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.

AN205208

FR Family MB91460 Series Standby Modes

This application notes describes the different standby modes (low power consumption modes) of the MB91460 Series.

Contents

1 Introduction

This chapter outlines the content of this document.

This document describes the different standby modes (low power consumption modes) of the MB91460 Series.

The document is separated into different parts.

- The chapter “Device States” outlines the different relevant device states of the MB91460 Series for the low power consumption modes.
- The chapter “Low Power Consumption Modes Features” lists the features of the different low power consumption modes.
- The chapter “Registers Related to Low Power Consumption Modes” describes the registers related to the low power consumption modes.
- The chapter “Programming” gives programming examples of the low power consumption modes. Also some settings for special applications are mentioned.
- The chapter “Transition Times” outlines the components of the transition times and shows measurement results for the transition time of the MB91F467DA.
- The chapter “Current Consumption” lists measurement results of the current consumption in low power consumption modes of the MB91F467DA.

Please see the hardware manual of the MB91460 Series, too.

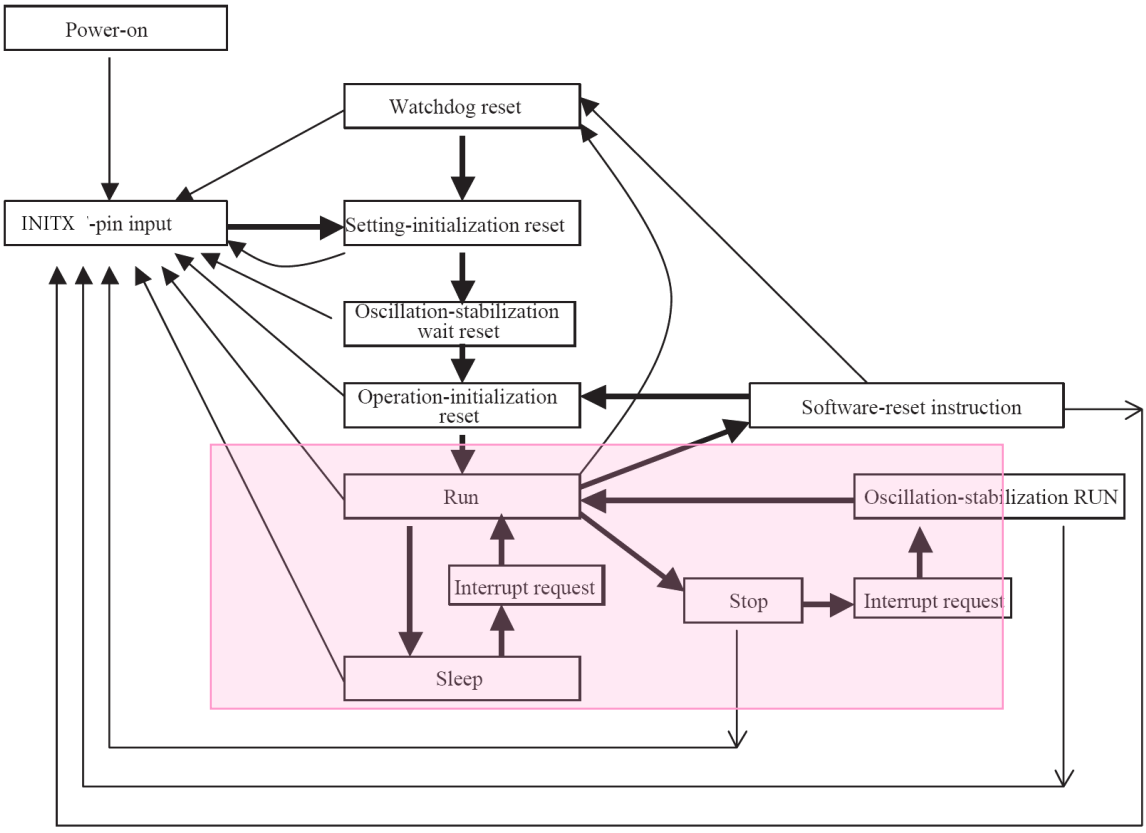
In the following “low power consumption modes” is used for “standby modes (low power consumption modes)”. The hardware manual of the MB91460 Series names the modes “standby modes”.

2 Device States

This chapter outlines the different relevant device states of the MB91460 Series for the low power consumption modes.

To understand the low power consumption modes it is important to know the different device states. The following diagram shows the basic device states of the MB91460 Series. A more detailed diagram is provided in the hardware manual.

Figure 1. Device States



The marked states and transitions are relevant for the low power consumption modes, with the following states.

- Run (Normal operation): State where the program is executed.
- Sleep: State where the program is stopped (peripheral circuits are operating and CPU core is stopped).
- Stop: State where the device is stopped (peripheral circuits and CPU core are stopped).
- Oscillation-stabilization-wait RUN: State to return from the STOP to the RUN state (waiting until clock oscillation is stabilized).

A detailed description of the stop and sleep modes is given in the following document.

3 Low Power Consumption Modes Features

This chapter lists the features of the different low power consumption modes.

There are two low power consumption modes available – the stop mode and the sleep mode. In the following the main features and differences are listed.

3.1 Stop Mode

3.1.1 Device State in Stop Mode

The device state in stop mode is characterised by the following

- **The overall device halts (internal circuits halt and the internal clock signals halt)**
- **Circuits that halt during stop mode**.
  - All internal circuits except those listed below.
- **Circuits that do not halt during stop mode**
  - Oscillation circuits that are not specified to be halted
    - Oscillation circuit for main clock (if not disabled)
    - Oscillation circuit for sub clock (if not disabled)
    - Main PLL circuit if oscillation circuit for main clock is enabled and PLL circuit is enabled and main regulator is kept enabled.
  - Peripheral functions that are driven directly by the oscillation and which have not been specified to be halted.
    - Real Time Clock (if not disabled) and main or sub clock oscillation is enabled and the RTC clock source is set to the enabled oscillation
    - LCDC (if LCD display enabled for sub-stop mode and sub clock selected as the clock source.)
- **Pin states (High impedance or maintain previous state)**
  - When pin outputs are set to go to high impedance during stop mode
    - High impedance output: Pins that are set as general purpose ports and pins that have been selected for use by peripheral functions.
  - When pin outputs are set to maintain their previous states during stop mode
    - Maintain previous state: Pins that are set as general purpose ports and pins that have been selected for use by peripheral functions.
  - When set as external interrupts
    - Input available state: Pins set as external interrupt inputs using level detection. (Whether the pin outputs during stop mode have been set to either high impedance or maintain previous state.)

3.1.2 Entering Stop Mode

The stop mode is invoked by the program. Writing “1” to the stop mode bit (STCR.STOP) changes to stop mode.

The device remains in this mode until an event occurs to wakeup the device from stop mode.

3.1.3 Options to Recover from Stop Mode

The following options cause a recovery from stop mode.

1. Any of the following interrupt requests cause the device to go to the oscillation stabilization wait RUN state and then to change back to RUN mode after the oscillation stabilization time elapses (return to normal operation).

- External interrupts set to level detection and that do not require a specific clock.
- Real Time Clock interrupt (if operating)
- Sub oscillation stabilisation timer when sub oscillation not halted.
- Oscillation stabilisation wait timer for the main clock when main oscillation not halted.

2. An INIT pin input or generation of a watchdog reset invokes an initialization reset (INIT) followed by an operation reset (RST) after the oscillation stabilization time.

3.2 Sleep Mode

3.2.1 Device State in Sleep Mode

The device state in sleep mode is characterised by the following

- **CPU program execution stops.**
- **Peripheral functions continue to operate.**
- **The internal memory and internal bus halt.**
- **Circuits that halt during sleep mode**
  - Bit search module
  - All internal memory (inclusive I-cache)
  - Internal/external bus
- **Circuits that do not halt during sleep mode**
  - Oscillation circuit, main PLL (if enabled)
  - Clock generation control circuit
  - Interrupt controller
  - External interrupts
  - DMA
  - Peripherals

3.2.2 Options to Entering Sleep Mode

The sleep mode is invoked by the program. Writing “1” to the sleep mode bit (STCR.SLEEP) changes to sleep mode.

The device remains in this mode until an event occurs to wakeup the device from sleep mode.

3.2.3 Options to Recover from Sleep Mode

The following options cause a recovery from sleep mode.

1. The generation of an interrupt request that is currently enabled changes the device back to RUN mode (Restores normal operation).
2. An INIT pin input or generation of a watchdog reset invokes an initialization reset (INIT) followed by an operation reset (RST).

4 Registers Related to Low Power Consumption Modes

This chapter describes the registers related to the low power consumption modes.

The following describes the related registers to the low power consumption modes. Relevant bits are marked.

4.1 Standby Control Register

This register is used to control transition to the stop and sleep modes, and to specify the pin states and whether to halt the oscillation during stop mode.

STCR: Address 0481h (Access: Byte)

7	6	5	4	3	2	1	0	bit
STOP	SLEEP	HIZ	SRST	0S1	0S0	0SCD2	0SCD1	
0	0	1	1	0	0	1	1	Initial value (INITX pin input)
0	0	1	1	X	X	1	1	Initial value (Watchdog reset)
0	0	X	1	X	X	X	X	Initial value (Software reset)
R/W	R/W	R/W	R1	R/W	R/W	R/W	R/W	Attribute

(See “A.1 Meaning of Bit Attribute Symbols” for details of the attributes.)

- Bit7: Stop mode

STOP	Operation
0	Does not change to stop mode.
1	Changes to stop mode.

  - Goes to “0” when a reset (INIT pin input or software reset) occurs or on recovery from stop mode.
  - Going directly from main PLL operation to stop mode is prohibited.

- Bit6: Sleep mode

SLEEP	Operation
0	Does not change to sleep mode.
1	Changes to sleep mode.

  - If this bit and the stop mode bit (STOP) bit are set to “1” at the same time, the device goes to stop mode.
  - Goes to “0” when a reset (INIT pin input or software reset) occurs or on recovery from sleep mode.

- Bit5: High impedance mode

HIZ	Operation
0	Maintain same pin states when changing to stop mode.
1	Set pin outputs to high impedance (Hi-z) during stop mode.

  - The default setting is high impedance.

- Bit4: Software reset (SRST)
  - Setting this bit to “0” invokes a software reset.

- Bit3-2: Oscillation stabilization time selection (OS[1:0])
  - • Setting these bits in the range “00”-“11” sets the oscillation stabilization time to use after recovering from stop mode.
    An INIT pin input reset or watchdog reset initialize this setting to its initial value.
    ($\Phi 2 \times 2^1$, main clock)
  - The count is performed by the timebase counter.

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

  - Φ2: Main clock divided by two or subclock

- Bit1: Sub clock oscillation halt

OSCD2	Operation
0	Continue oscillation
1	Halt oscillation

- Bit0: Main clock oscillation halt

OSCD1	Operation
0	Continue oscillation
1	Halt oscillation

4.2 Timebase Timer Control Register

This register controls the timebase timer interrupts and the options for resets and standby operation.

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 input)
0	0	X	X	X	X	0	0	Initial value (Watchdog reset)
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	Attribut

(See “A.1 Meaning of Bit Attribute Symbols” for details of the attributes.)

- Bit7: Interrupt request flag for timebase timer
  - This flag goes to “1” when a timebase timer interrupt request occurs

- •Bit6: Interrupt request enable for the timebase timer
  - Writing “1” to this bit enables timebase timer interrupt requests.
- Bit5-3: Interval time selection for timebase timer
  - Writing a value in the range “000”-“111” to these bits selects the interval time for the timebase timer. (F x 2^{11}, x 2^{12}, x 2^{13}, x 2^{22}, x 2^{23}, x 2^{24}, x 2^{25}, x 2^{26})
- Bit2: Reserved Writing does not affect the operation. The read value is undefined.
- Bit1: Enable synchronous reset operation
  - Selects a normal reset “0” or a synchronous reset “1”.
- Bit0: Enable synchronous standby operation

SYNCS	Operation
0	Normal reset operation (Not permitted on this model).
1	Enable synchronous standby operation (always set this before changing to a standby mode).

5 Programming

This chapter gives programming examples of the low power consumption modes. Also some settings for special applications are mentioned.

5.1 Low Power Consumption Mode Procedure

The following code examples show how a low power consumption mode procedure might be programmed.

5.1.1 Disable Interrupts

If interrupts are disabled in the interrupt control register (ICR=“00011111B”), the device will not recover from stop or sleep mode when an interrupt occurs.

To avoid that the wake-up sequence (e.g. for restoring the base clock settings) is interrupted, set the interrupt enable flag to 0 (disable all interrupts) and set the interrupt control register of the interrupt source, which should wake-up the microcontroller to an appropriate value.

This results in that the microcontroller is waked-up and continuous code execution. The interrupt service routine of the wake-up interrupt is not called, because the interrupts are disabled. Nevertheless the interrupt request(s) is (are) maintained and after setting the interrupt enable flag back to 1, the interrupt service routines are executed depending on their level.

The same effect can be achieved by setting the interrupt level mask.

```
__DI();                    /* Disable all interrupts */
```

5.1.2 Disable Low-Voltage Detection

If the low-voltage detection is not required in the stop / sleep mode, the low-voltage detection should be disabled to reduce the power consumption before entering a low power consumption mode.

```
LVDET_LVEPD = 1;           /* Power down external low voltage detection */
LVDET_LVIPD = 1;           /* Power down internal low voltage detection */
```

5.1.3 Reduce Sub-Regulator Voltage Level

The sub-regulator voltage level can be reduced. Please check the data sheet of the specific device for this change.

```
REGSEL_SUBSEL = 0x0      /* Set sub-regulator voltage level to 1.2 V */
```

5.1.4 Switch Base Clock

To save power in a low power consumption mode all clocks which are not required in the low power consumption mode can be disabled.

For the stop mode the halt of the oscillations can be set with the bits **OSCD2** and **OSCD1** of the standby control register (STCR). Before changing to stop mode, the main PLL must not be selected as base clock (operating clock source). Change the operation clock source selection to main clock divided by two before changing to stop mode. The restrictions that apply to the clock divide ratio setting are the same as for normal operation. Also, you do not necessarily have to halt the PLL oscillation.

For the sleep mode the clocks have to be halted by the appropriate clock control settings.

In the following code example the base clock (operating clock source) is switched from main PLL to main clock divided by two. The PLL is disabled afterwards.

```
PLLCTRL_GRDN = 0;         /* Clear gear-down flag */

CLKR &= 0xFC;             /* CLKR_CLKS = 0; -> clock source = main clock*/

while(!PLLCTRL_GRDN)      /* Wait until gear-down is finished */
   HWWD_CL = 0;
PLLCTRL_GRDN = 0;         /* Clear gear-down flag */

CLKR &= 0xF8;             /* CLKR_PLL1EN = 0; -> disable PLL*/
```

5.1.5 Enter Low Power Consumption Mode

When changing to a low power consumption mode, the synchronous standby operation enable bit has to be set (SYSNCS = 1). The setting (SYSNCS = 0) is prohibited.

In order to change to a low power consumption mode with synchronous standby operation enabled, the STCR register must be read after writing to the SLEEP or STOP bit. Always use the following sequence.

The meaning of the bits of the standby control register can be seen in chapter 4.1.

```
TBCR_SYNCS = 1;           /* Enable synchronous standby operation */

#pragma asm               /* NOP x 5 required for timing */
(LDI #lpc_value, R0)      /* lpc_value contains the write data for STCR */
(LDI #_STCR, R12)         /* _STCR is the address of STCR (481H) */
STB R0, @R12              /* Write to standby control register (STCR) */
LDUB @R12, R0             /* STCR read required for synchronous standby */
LDUB @R12, R0             /* Second dummy read to STCR */
NOP                       /* NOP x 5 required for timing */
NOP
NOP
NOP
NOP
#pragma endasm
```

5.1.6 Recover from Low Power Consumption Mode

The code execution continuous on the address it was stopped.

Recover from Stop Mode

The following options cause a recovery from stop mode.

1. Any of the following interrupt requests cause the device to go to the oscillation stabilization wait RUN state and then to change back to RUN mode after the oscillation stabilization time elapses (return to normal operation).

- External interrupts set to level detection and that do not require a specific clock.
- Real Time Clock interrupt (if operating)
- Sub oscillation stabilisation timer when oscillation not halted.
- Oscillation stabilisation wait timer for the main clock when oscillation not halted.

If using interrupt processing, remember the I flag (I), the interrupt level mask register (ILM), and the interrupt control register (ICR).

2. An INIT pin input or generation of a watchdog reset invokes an initialization reset (INIT) followed by an operation reset (RST) after the oscillation stabilization time.

Recover from Sleep Mode

The following options cause a recovery from sleep mode.

1. The generation of an interrupt request that is currently enabled changes the device back to RUN mode (Restores normal operation).
2. An INIT pin input or generation of a watchdog reset invokes an initialization reset (INIT) followed by an operation reset (RST).

5.1.7 Restore Previous Base Clock

The following shows how to restore the main PLL as base clock (operating clock source) again. If it is not necessary to restore the old settings immediately, this can be done later, too.

For example, if the microcontroller is waken-up only to update the clock and returns to a low power mode afterwards, it is not necessary to restore the previous base clock. With such a kind of application it is necessary to have criteria to decide when the base clock should be restored.

```
CLKR |= 0x04;            /* CLKR_PLL1EN = 1; -> enable PLL*/

TBCR = 0x00;             /* 0x00: 1 ms @ main clock*/
CTBR = 0xA5;             /* Clear timer*/
CTBR = 0x5A;
TBCR_TBIF = 0;           /* Clear interrupt flag */
while (TBCR_TBIF == 0)   /* Wait PLL stabilisation time */
   HWWD_CL = 0;

PLLCTRL_GRUP = 0;        /* Clear gear-up flag */
CLKR |= 0x02;            /* CLKR_CLKS = 2; -> clock source = PLL*/

While (!PLLCTRL_GRUP)    /* Wait until gear-up is finished */
   HWWD = 0x10;
PLLCTRL_GRUP = 0;        /* Clear gear-down flag */
```

5.1.8 Enable Low-Voltage Detection

If the low-voltage detection is not required in the stop / sleep mode, the low-voltage detection should be disabled to reduce the power consumption before entering a low power consumption mode.

```
LVDET_LVEPD = 0;          /* Power up external low voltage detection */
LVDET_LVIPD = 0;          /* Power up internal low voltage detection */
```

5.1.9 Increase Sub-Regulator Voltage Level

If it is necessary the sub-regulator voltage level can be increased again. Please check the data sheet of the specific device if this is necessary.

5.1.10 Enable Interrupts

After restoring all necessary settings, the interrupts should be enabled again.

```
__EI();                   /* Enable all interrupts*/
```

5.2 Settings for Special Application

In the following, hints for some special applications in connection with the low power consumption modes are given.

5.2.1 Wake-up by a CAN message.

Each CAN Rx is shared with an external interrupt. This feature allows to wake-up the microcontroller form stop mode by a CAN message without an external circuit.

To use this feature the appropriate interrupt control register of the external interrupt has to be set and an interrupt service routine has to be implemented.

The following table shows the correlation of external interrupt, CAN channel and I/O-Port. There are not all CAN channels available on all devices of the MB91460 Series. Please see the appropriate data sheet to identify which channel is available on a specific device.

Table 1. CAN Rx and External Interrupt

CAN Channel	Pin Name	I/O Port	External Interrupt
0	RX0	P23_0	INT8
1	RX1	P23_2	INT9
2	RX2	P23_4	INT10
3	RX3	P23_6	INT11
4	RX4	P22_0	INT12
5	RX5	P22_2	INT13

5.2.2 Fast wake-up from stop mode, to receive the first CAN message.

For a fast wake-up from stop mode to receive the first CAN message, the CAN clock prescaler has to be connected to the main oscillation and the main oscillation must not be stopped during stop mode.

Therefore set the PSCLOCKSOURCE to PSCLOCK_MAIN in the start-up file. Furthermore the bit STCR_OSCD0 has to be set to 0 before entering the stop mode.

5.2.3 Wake-up from stop mode by real time clock

It is possible to wake-up from stop mode by an interrupt from the real time clock. Relevant clock supplies for the real time clock in stop mode are the main clock divided by two (2 MHz), sub clock divided by to (32 kHz).

Therefore the source oscillation of the real time clock must not be stopped during stop mode, hence STCR_OSCD0 has to be set to 0, when the real time clock runs on the main oscillation divided by 2 or STCR_OSCD1 has to be set to 0, when the real time clock runs on the sub oscillation divided by 2.

6 Transition Times

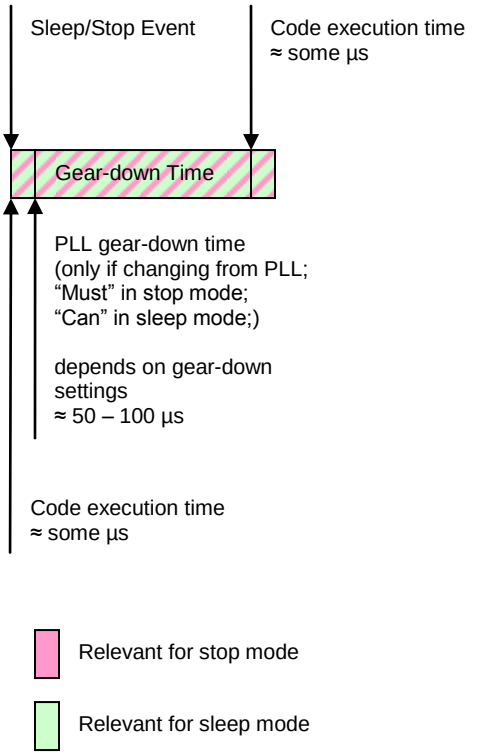
This chapter outlines the components of the transition times and shows measurement results for the transition time of the MB91F467DA.

6.1 Transition Time to Low Power Consumption Modes

6.1.1 Components of the Transition Time

The following figure shows the different components of the transition time to a low power consumption mode from run mode.

Figure 2. Components of the Transition Time to Low Power Consumption Modes



Two different times can be defined.

1. The time required to get into the low power mode (t_{trans}).
2. The time to switch to the desired base clock settings for the low power mode (t_{go_lpm}) consisting of t_{trans} and the time required to switch to the desired base clock settings.

6.1.2 Measurements

The measurements were done on a MB91F467DA and might differ slightly between other exemplars of this device. Before the transition to the low power mode, the base clock is set from main PLL to main clock divided by 2.

The following was included in the test program to visualise the different components of the wake-up time.

- INT1 is used to generate an interrupt request.
- P16_0 is used to indicate the state of the microcontroller.
  - Before the transition to the low power mode the PDR16_0 is set to ‘1’.

- To indicate the interrupt service routine, which set the microcontroller to the low power consumption mode, P16_0 is toggled.
- In stop mode the ports are set to high impedance (STCR_HIZ = 1). This state is indicated in the following graphs by a voltage of 1.8 V on P16_0.

The following is indicated by the figures.

1 Microcontroller is in run mode. Port 16_0 is set to high.

2 Generation of the interrupt request by INT1.

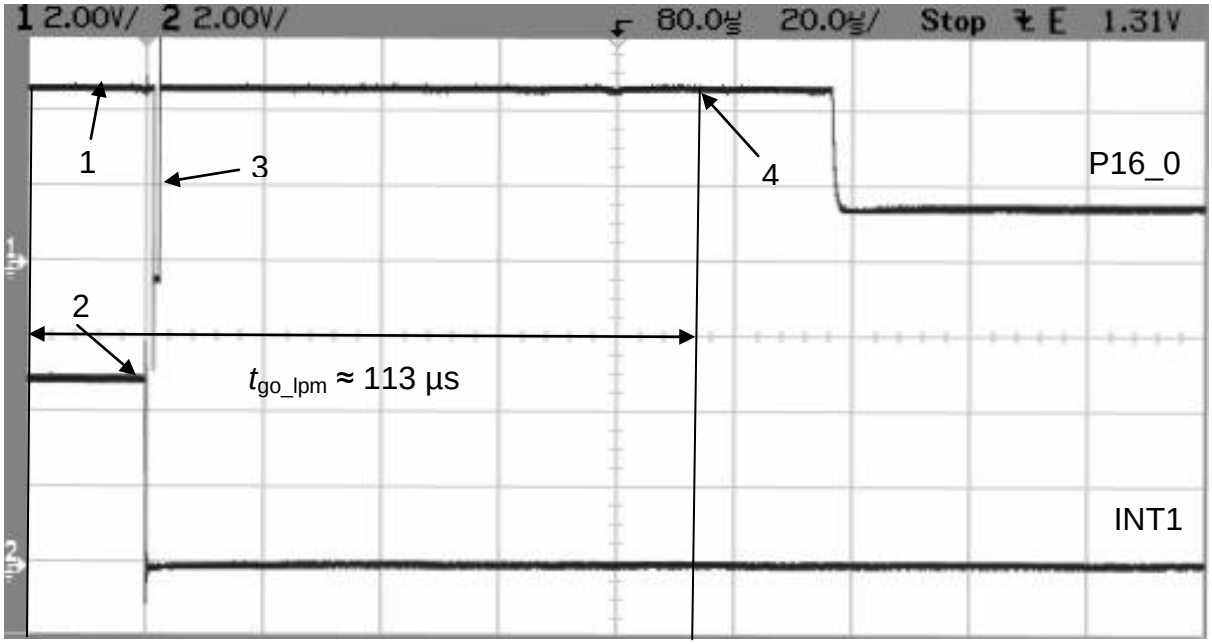
3 Start of interrupt service routine, setting microcontroller to low power consumption mode.

4 Microcontroller is in low power consumption mode.

3 – 4 t_{go_lpm}.

Figure 3 show the plot of an oscilloscope in the case of STCR = 0xBB, when the device is set to the low power mode.

Figure 3. STCR= 0xBB (Stop Mode, High-Imp.,
Main Osc. off, Sub Osc. Off, Osc. Stab. Time = 32 ms, PLL off, PLL Stab. Time = 1 ms)
t_{go_lpm} ≈ 113 µs



The numbers in the figure refer to the description in chapter 6.1.2.

Before setting the microcontroller to the stop mode, the PLL is disabled und the base clock source is set to main clock divided by two. The gear-down time of the PLL is about 75 µs.

t_{go_lpm} is about 113 µs. The time does not differ that much for other settings.

6.1.3 Summary

The following table summarizes the measured times. The gear-down time is about 75 µs. The PLL is always disabled in the low power consumption modes.

Table 2. Summary Transition Time to Low Power Consumption Modes

Transition to	STCR	t_{trans}	t_{go_lpm}
Stop-Mode	0xBB (Stop Mode, High-Imp., Main Osc. off, Sub Osc. Off, Osc. Stab. Time = 32 ms)	≈ some µs	≈ 0.113 ms

With:

t_{trans}: The time required to get into the low power mode (t_{trans})

t_{go_lpm}: The time to switch to the desired base clock settings for the low power mode (t_{go_lpm}) consisting of t_{trans} and the time required to switch to the desired base clock settings.

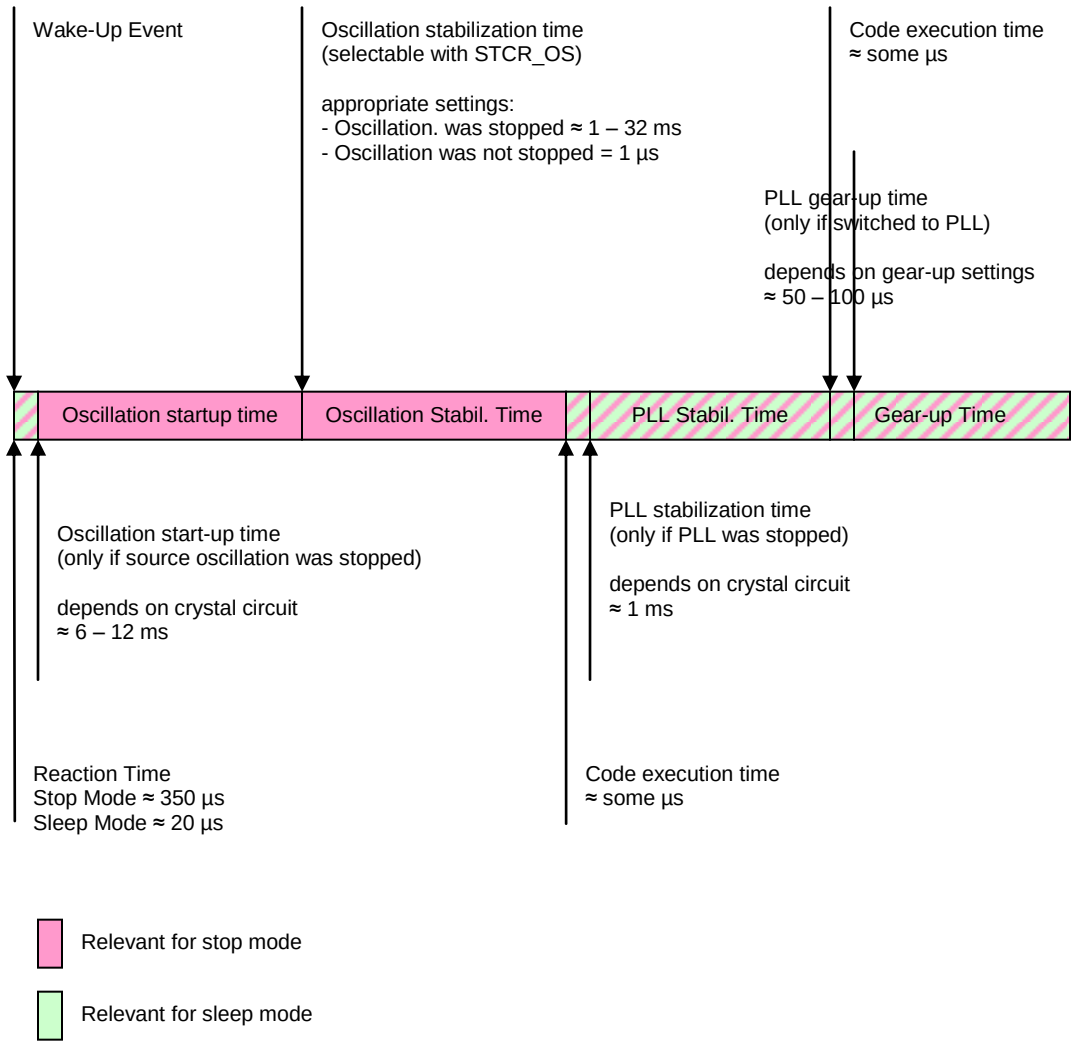
See Figure 1, too.

6.2 Transition Time from Low Power Consumption Modes

6.2.1 Components of the Transition Time

The following figure shows the different components of the transition time from a low power consumption mode to run mode.

Figure 4. Components of the Transition Time from Low Power Consumption Modes



Since the source oscillator is not stopped in sleep mode, the oscillation start-up time and the oscillation stabilisation time is not relevant for the sleep mode.

Two different times can be defined.

1. The wake-up time ($t_{wake\text{-}up}$) consisting of the reaction time, the oscillation start-up time and the oscillation stabilisation time.
2. The time to restore the previous base clock settings ($t_{restore}$) consisting of the wake-up time and the time required to switch to the desired base clock settings.

6.2.2 Measurements

The measurements were done on a MB91F467DA and might differ slightly between other exemplars of this device. Before the transition to the low power mode, the base clock was set to main clock divided by 2. After wake-up the base clock was set to Main PLL.

The following was included in the test program to visualise the different components of the wake-up time.

- INT0 is used to generate an interrupt request.
- P16_0 is used to indicate the state of the microcontroller.
  - Before the transition to the low power mode the PDR16_0 is set to '1'.
  - In stop mode the ports are set to high impedance (STCR_HIZ = 1). This state is indicated in the following graphs by a voltage of 1.8 V on P16_0.
  - The first code which is executed after the interrupt request is setting the PDR16_0 to '0'.
  - Finally, after enabling and switching to the PLL as base clock, the PDR16_0 is set back to '1'.

The following is indicated by the figures.

1 Microcontroller is in low power consumption mode. Port 16_0 has high impedance in case of stop mode and is set to high in case of sleep mode.

2 Generation of the interrupt request by INT0.

2 – 3 Reaction time.

3 – 4 Oscillation stabilisation + code execution time (wake-up time).

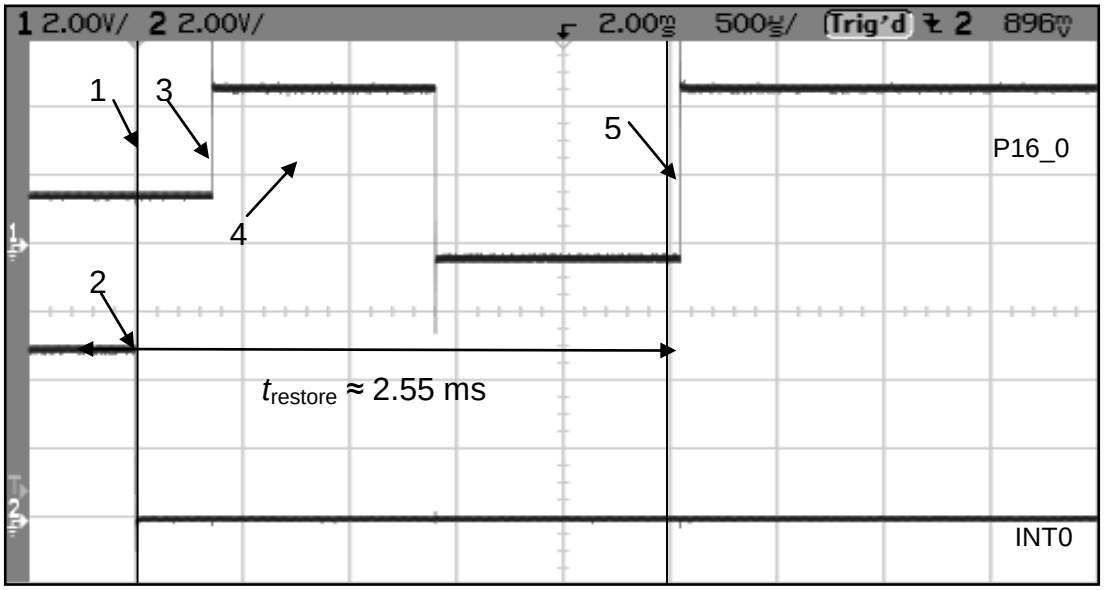
4 – 5 PLL stabilisation + PLL gear-up + code execution time (restore time).

5 Wake-up completed and base clock restored.

Stop Mode

Figure 5shows the plot of an oscilloscope in the case of STCR = 0xB6, when the device is waked-up from the low power mode.

Figure 5. STCR= 0xB6 (Stop Mode, High-Imp., Main Osc. on, Sub Osc. Off, Osc. Stab. Time ≈ 1 ms, PLL off, PLL Stab. Time ≈ 1 ms)
$t_{wake\text{-}up}$ ≈ 1.4 ms, $t_{restore}$ ≈ 2.55 ms



The numbers in the figure refer to the description in chapter 6.2.2.

Since the device is waked-up from stop mode the duration of the reaction time is about 350 µs (2-3). The main oscillation was not halted during stop mode; hence no oscillation start-up time applies.

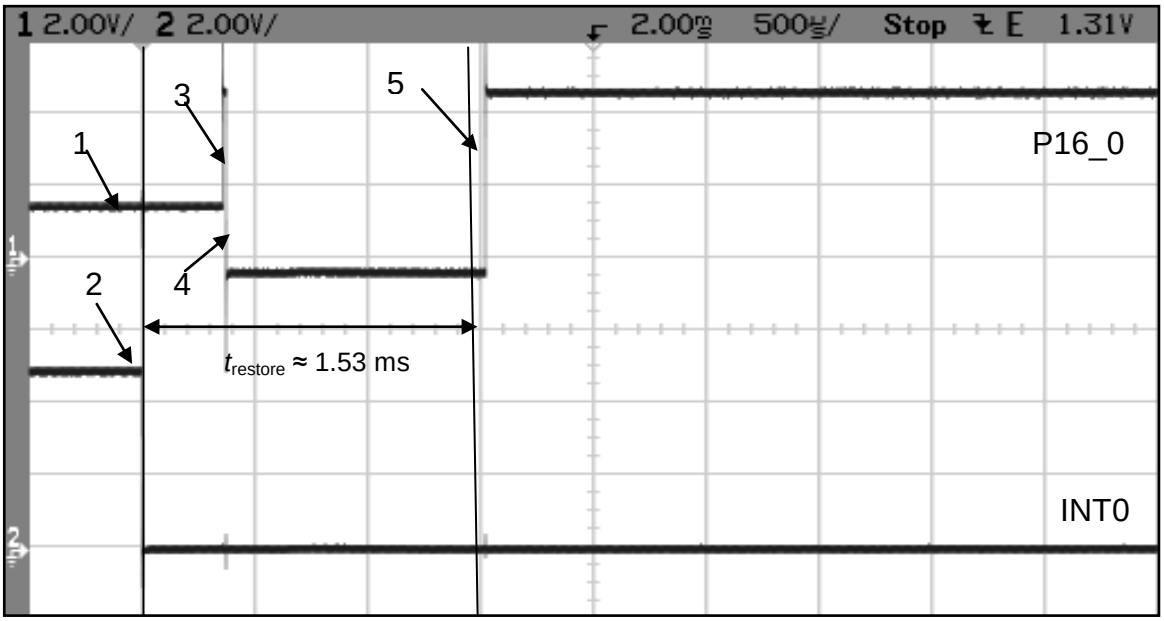
The oscillation stabilisation time is set to about 1 ms, this corresponds with the measurement (3-4). The measurement includes the time, required for setting the indication port from high to low.

The PLL stabilisation time is set to about 1 ms and the gear-up time is about 75 µs, this corresponds with the measurement (4-5). The measurement includes the time, required for setting the clock settings and for setting the indication port from low to high.

A wake-up time ($t_{wake\text{-}up}$) of about 1.4 ms and a time to restore the previous base clock settings ($t_{restore}$) of about 2.55 ms – under the above mentioned restrictions – can be measured.

In Figure 6 the oscillation stabilisation time was set to 1 µs (minimum value, STCR = 0xB2), because the main oscillation was not stopped during the low power mode; hence no oscillation stabilisation time is required.

Figure 6. STCR= 0xB2 (Stop Mode, High-Imp.,
Main Osc. on, Sub Osc. Off, Osc. Stab. Time = 1 µs, PLL off, PLL Stab. Time = 1 ms)
$t_{wake\text{-}up}$ ≈ 0.36 ms, $t_{restore}$ ≈1.53 ms



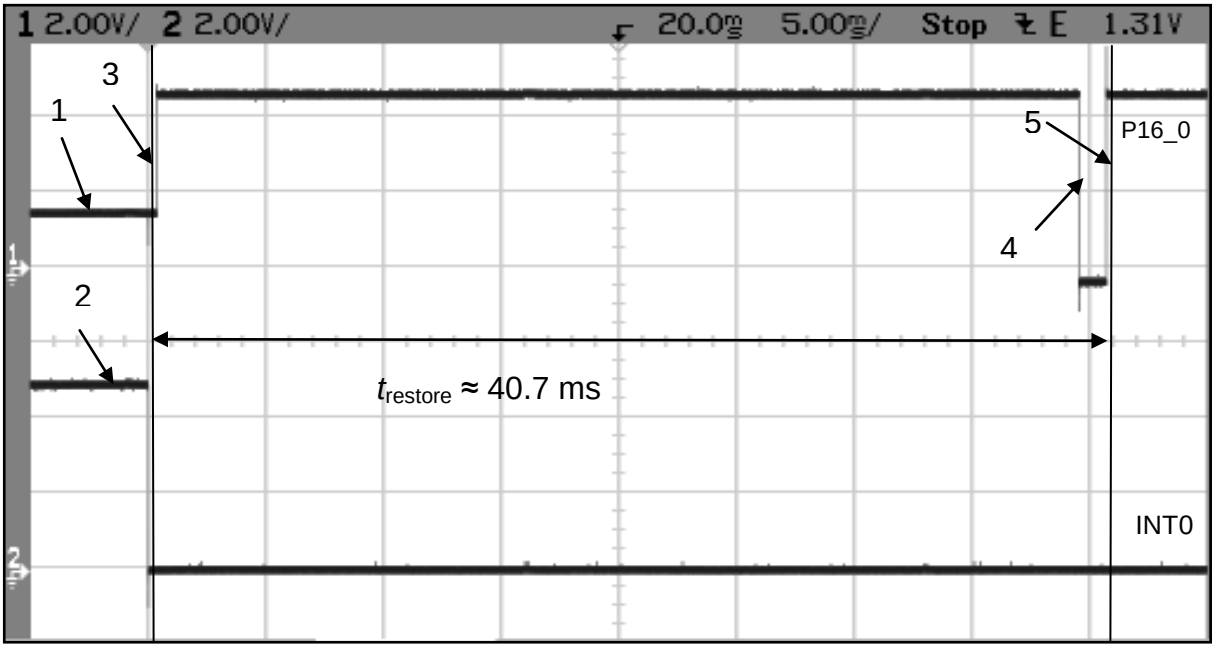
The numbers in the figure refer to the description in chapter 6.2.2. The explanation is analogously to the previous figure.

A wake-up time ($t_{wake\text{-}up}$) of about 0.36 ms and a time to restore the previous base clock settings ($t_{restore}$) of about 1.53 ms – under the above mentioned restrictions – can be measured.

In the following two measurements, the main clock was halted during low power mode, hence the oscillation stabilisation time applies and an appropriate oscillation stabilisation time is required.

In Figure 7oscillation stabilisation time is set to 32 ms (STCR = 0xBB). The appropriate oscillation stabilisation time depends on the crystal circuit.

Figure 7. STCR= 0xBB (Stop Mode, High-Imp.,
Main Osc. off, Sub Osc. Off, Osc. Stab. Time = 32 ms, PLL off, PLL Stab. Time = 1 ms)
$t_{wake-up}$ 39.6 ≈ ms, $t_{restore}$ ≈ 40.7 ms

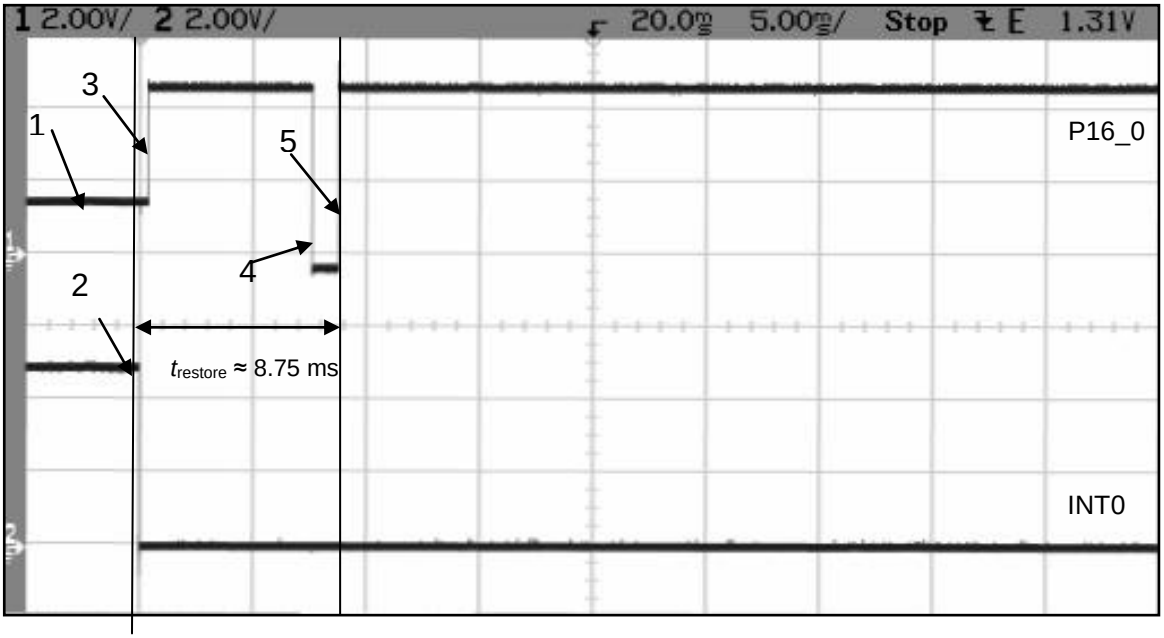


The numbers in the figure refer to the description in chapter 6.2.2. Since the main oscillation was stopped, the oscillation start-up time applies. The explanation is analogously to the previous figures.

A wake-up time ($t_{wake-up}$) of about 39.6 ms and a time to restore the previous base clock settings ($t_{restore}$) of about 40.7 ms – under the above mentioned restrictions – can be measured.

In Figure 8 the oscillation stabilisation time is set to 1 ms (STCR = 0xB7). The appropriate oscillation stabilisation time depends on the crystal circuit.

Figure 8. STCR= 0xB7 (Stop Mode, High-Imp.,
Main Osc. off, Sub Osc. Off, Osc. Stab. Time = 1 ms, PLL off, PLL Stab. Time = 1 ms)
$t_{wake\text{-}up}$ ≈ 7.6 ms, $t_{restore}$ ≈ 8.75 ms



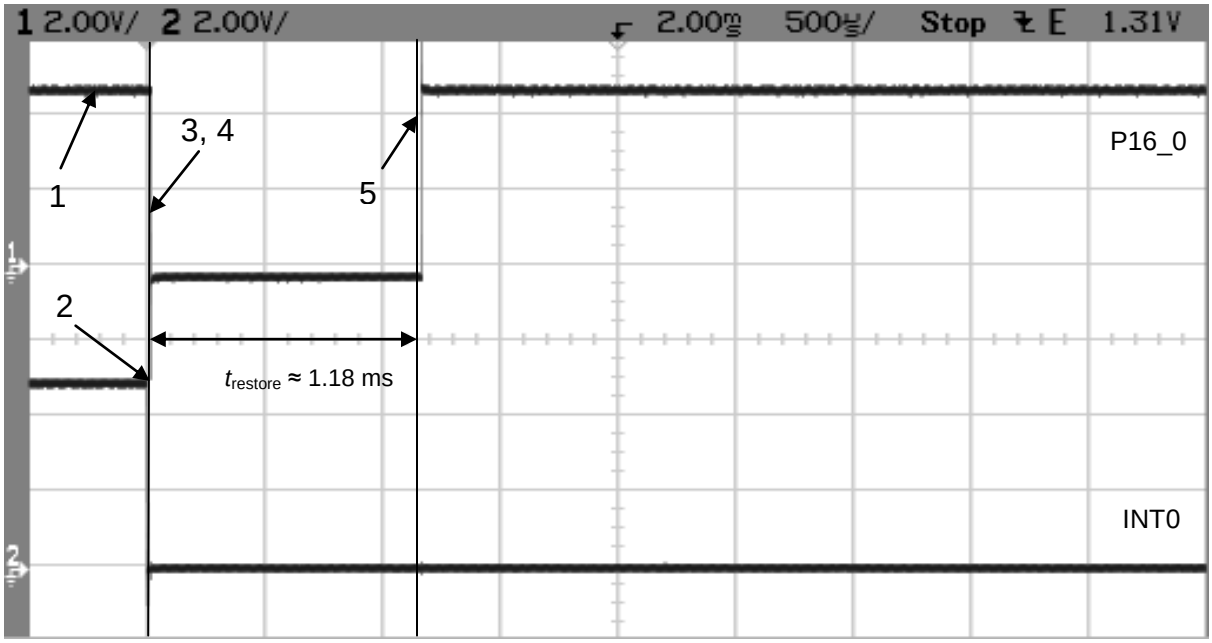
The numbers in the figure refer to the description in chapter 6.2.2. Since the main oscillation was stopped, the oscillation start-up time applies. The explanation is analogously to the previous figures.

A wake-up time ($t_{wake\text{-}up}$) of about 7.6 ms and a time to restore the previous base clock settings ($t_{restore}$) of about 8.75 ms – under the above mentioned restrictions – can be measured.

Sleep Mode

Figure 9 shows the plot of an oscilloscope in the case of STCR = 0x7B, when the device is waked-up from the low power mode.

Figure 9. STCR= 0x7B (Sleep Mode, High-Imp. (n.a.),
Main Osc. Off (n.a.), Sub Osc. Off (n.a.), Osc. Stab. Time = 32 ms (n.a.), PLL off, PLL Stab. Time = 1 ms)
$t_{wake\text{-}up} \approx 0.02$ ms, $t_{restore} \approx 1.18$ ms



The numbers in the figure refer to the description in chapter 6.2.2.

Since the device is waked-up from sleep mode the duration of the reaction time is about 20 µs (2-3). The main oscillation was not halted during sleep mode.

The PLL stabilisation time is set to about 1 ms and the gear-up time is about 75 µs, this corresponds with the measurement (4-5). The measurement includes the time, required for setting the clock settings and for setting the indication port from low to high.

A wake-up time ($t_{wake\text{-}up}$) of about 0.02 ms and a time to restore the previous base clock settings ($t_{restore}$) of about 1.18 ms – under the above mentioned restrictions – can be measured.

6.2.3 Summary

The following table summarizes the measured times. The PLL stabilisation time is set to about 1 ms and the gear-up time is about 75 µs. The PLL is always disabled in the low power consumption modes.

Table 3. Summary Transition Time from Low Power Consumption Modes

Transition from	STCR	$t_{wake-up}$	$t_{restore}$
Stop-Mode	0xB6 (**Stop Mode**, High-Imp., **Main Osc. on**, Sub Osc. Off, Osc. **Stab. Time ≈ 1 ms**)	≈ 1.4 ms	≈ 2.55 ms
Stop-Mode	0xB2 (**Stop Mode**, High-Imp., **Main Osc. on**, Sub Osc. Off, Osc. **Stab. Time = 1 µs**)	≈ 0.36 ms	≈ 1.53 ms
Stop-Mode	0xBB (**Stop Mode**, High-Imp., **Main Osc. off**, Sub Osc. Off, Osc. **Stab. Time = 32 ms**)	≈ 39.6 ms	≈ 40.7 ms
Stop-Mode	0xB7 (**Stop Mode**, High-Imp., **Main Osc. off**, Sub Osc. Off, Osc. **Stab. Time = 1 ms**)	≈ 7.6 ms	≈ 8.75 ms
Sleep-Mode	0x7B (**Sleep Mode**, High-Imp.[1], Main Osc. Off[1], Sub Osc. Off[1], Osc. Stab. Time = 32 ms[1])	≈ 0.02 ms	≈ 1.18 ms

With:

$t_{wake-up}$: The wake-up time ($t_{wake-up}$) consisting of the reaction time, the oscillation start-up time and the oscillation stabilisation time.

$t_{restore}$: The time to restore the previous base clock settings ($t_{restore}$) consisting of the wake-up time and the time required to switch to the desired base clock settings.

See Figure 2, too.

7 Current Consumption

This chapter list measurement results of the current consumption in low power consumption modes of the MB91F467DA.

The measurements were done on MB91F467DA and might differ slightly between other exemplars of this device.

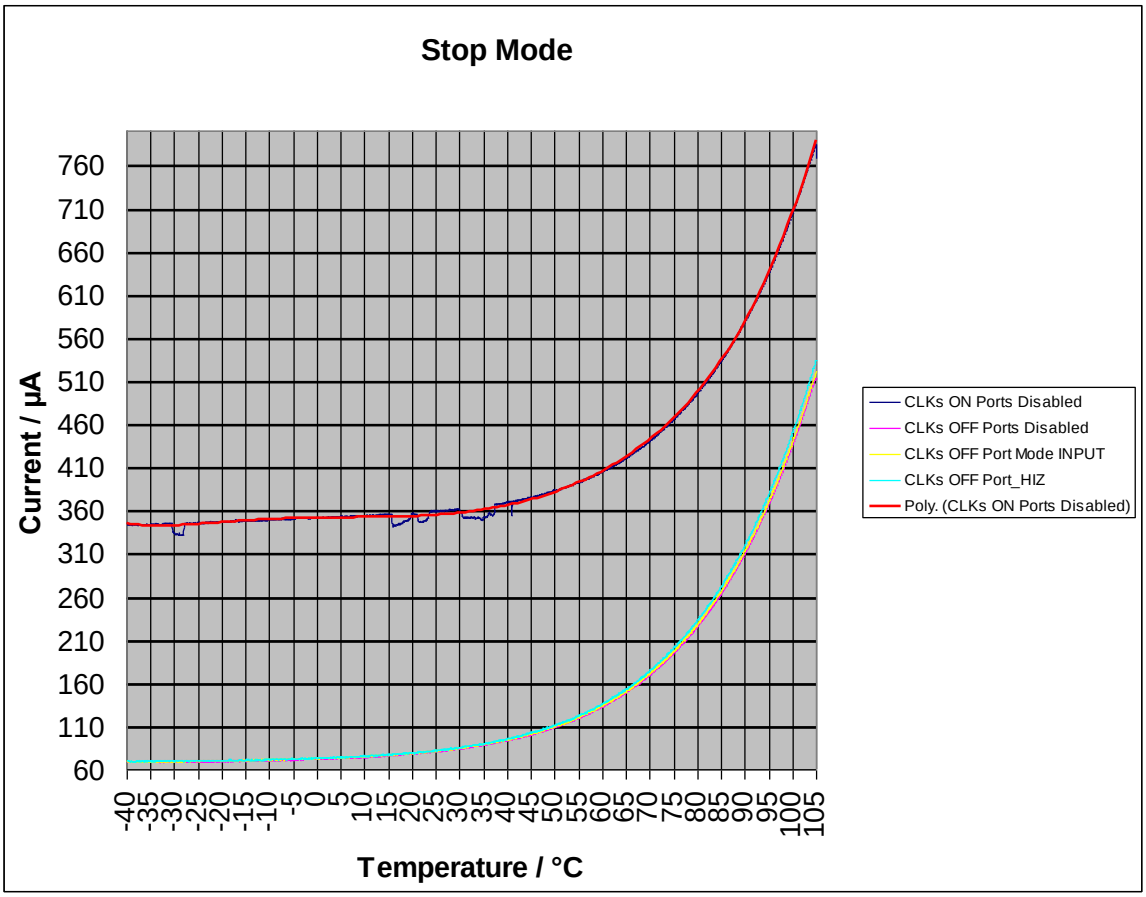
7.1 Current Consumption in Stop Mode

7.1.1 Enabled LV Detection and Default Sub-Regulator Voltage Level

Figure 10 is showing the dependency of the current *I* on the temperature *T* in the stop mode. The figure is showing different graphs for different port states during stop mode.

[1] Not applicable in sleep mode.

Figure 10. $I_{stop} = f(T)$; Stop Mode

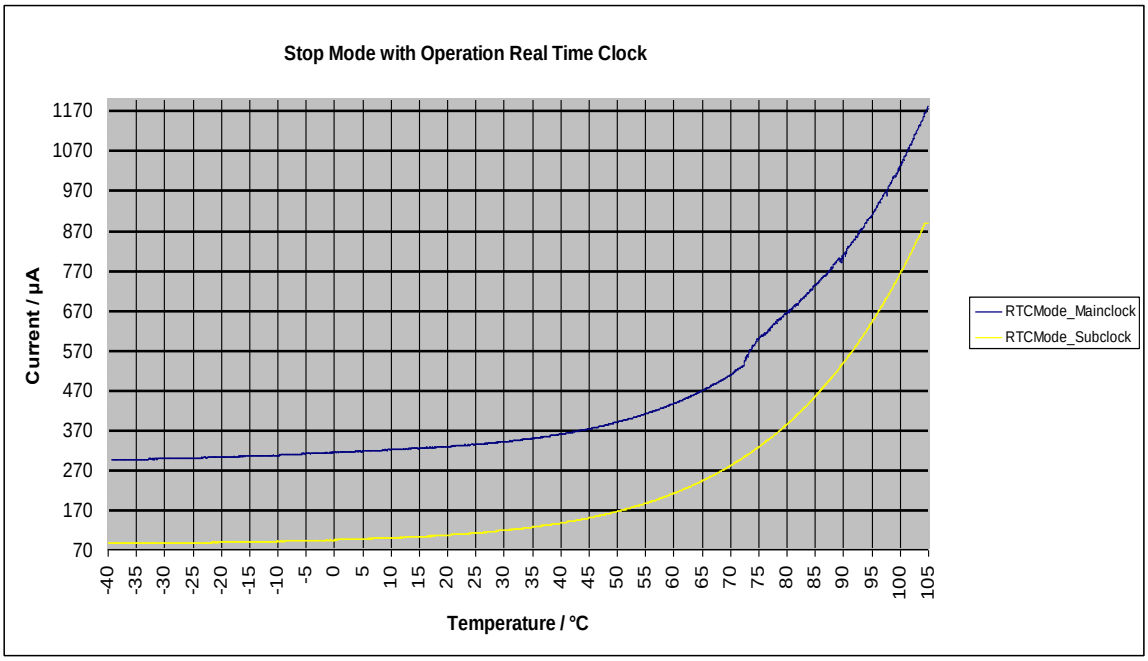


It can be seen, that the current consumption does not differ that much, when the clocks are disabled.

Enabled clocks during stop mode increase the current consumption significantly. The current consumption increases significantly with a higher temperature.

The next figure is showing the current consumption in the stop mode with operating real time clock over the temperature. The figure is showing two different graphs for the cases, that the real time clock is supplied by the main clock or by the subclock.

Figure 11. I_{stop} = *f(T);* Stop Mode with Operation Real Time Clock



It can be seen, that the current consumption main clock is significantly higher.

As seen in the previous figure the current consumption increases significantly with a higher temperature.

A further possibility to reduce the power consumption is to disable the low-voltage detection and to reduced the sub-regulator voltage level. The results can be seen in the next section.

7.1.2 Disabled LV Detection and Reduced Sub-Regulator Voltage Level

If the current is measured in stop mode as mentioned above with additionally disabled low-voltage detection and a reduced sub-regulator voltage level of 1.2 V the following value can be measure.

Table 4. Current in Stop Mode

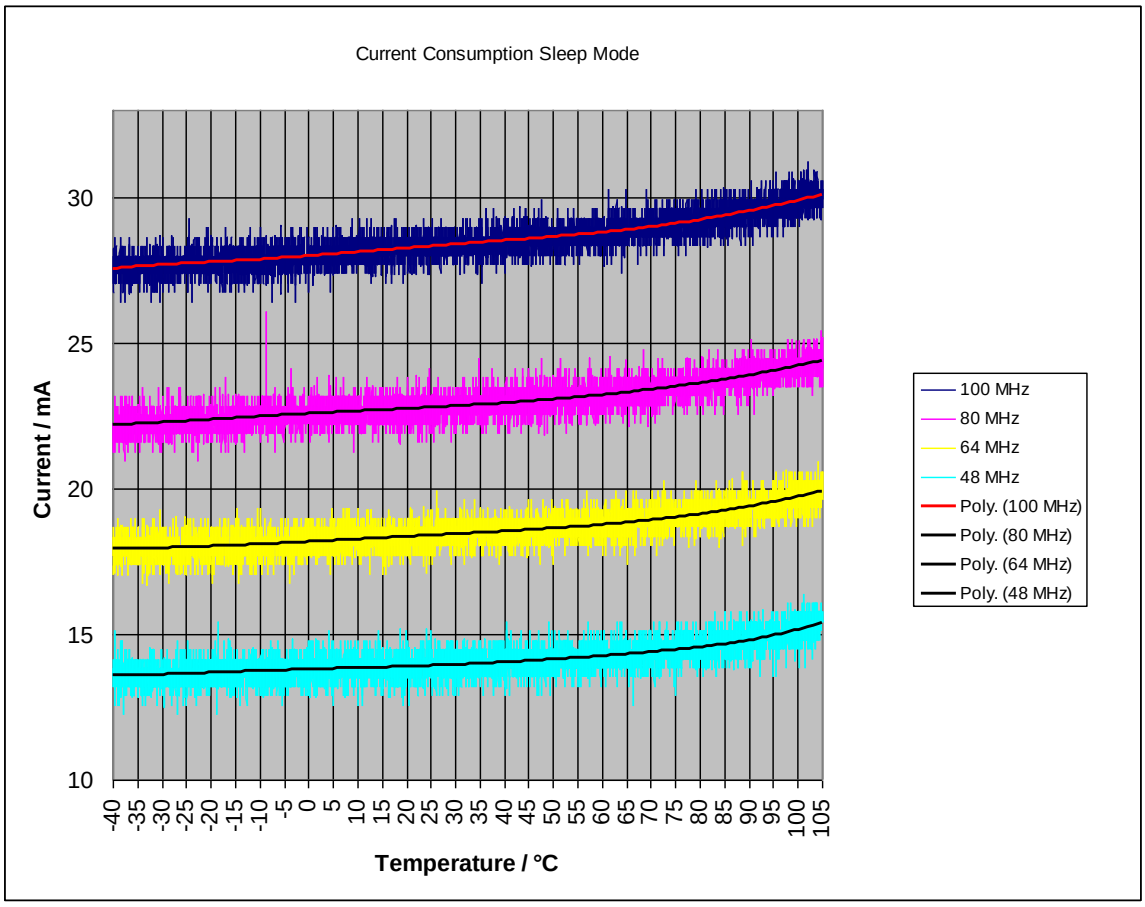
Mode	STCR	I_{stop}
Stop-Mode (Disabled LV detection, sub-reg voltage level = 1.2 V)	0xB7 (**Stop Mode**, High-Imp., **Main Osc. off**, Sub Osc. Off,)	≈ 40 µA

7.2 Current Consumption in Sleep Mode

7.2.1 Enabled LV Detection and Default Sub-Regulator Voltage Level

The following figure is showing the dependency on the current *I* on the temperature *T* in the sleep mode. The figure is showing different graphs for different clock speeds during sleep mode.

Figure 12. I_{sleep} = f(T); Sleep Mode



The current consumption rises with the clock frequency. Therefore it is best if the clock speed can be reduced or disabled during sleep mode.

As seen in the previous figure the current consumption increases with a higher temperature.

A further possibility to reduce the power consumption is to disable the low-voltage detection and to reduced the sub-regulator voltage level.

A Appendix

A.1 Meaning of Bit Attribute Symbols

R : Readable

W : Writable

RM : Reading operation during read/modify/write operation.

"/" (Slash) R/W: Readable and writable. (The read value is the value written.)

"," (comma) R,W: Values are different between read and write. (The read value is different from the value written.)

R0 : The read value is "0".

R1 : The read value is "1".

W0 : Always write "0".

W1 : Always write "1".

(RM0) : read/modify/write operation reads "0".

(RM1) : read/modify/write operation reads "1".

RX : The read value is indeterminate. (Reserved bit or undefined bit)

WX : Writing does not affect the operation. (Undefined bit)

Example of how R/W is used

R/W : Readable and writable. (The read value is the value written.)

R,W : Readable and writable. (The read value and written value are different.)

R,RM/W : Readable and writable. (The read value and written value are different. Read/modify/write command reads the value written.) Example: port data register

R(RM1),W : Readable and writable. (The read value and written value are different. Read/modify/write command reads 1.) Example: interrupt request flag

R/WX : Read-only (Read-only. Writing does not affect the operation.)

R1,W : Write-only (Write-only. The read value is 1.)

R0,W : Write-only (Write-only. The read value is 0.)

RX,W : Write-only (Write-only. The read value is indeterminate.)

R/W0 : Reserved bit (The written value is 0. The read value is the value written.)

R0/W0 : Reserved bit (The written value is 0. The read value is 0.)

R1,W0 : Reserved bit (The written value is 0. The read value is 1.)

RX,W0 : Reserved bit (The written value is 0. The read value is indeterminate.)

R/W1 : Reserved bit (The written value is 1. The read value is the value written.)

R1/W1 : Reserved bit (The written value is 1. The read value is 1.)

R0,W1 : Reserved bit (The written value is 1. The read value is 0.)

RX,W1 : Reserved bit (The written value is 1. The read value is indeterminate.)

RX/WX : Undefined bit (The read value is indeterminate. Writing does not affect the operation.)

R0/WX : Undefined bit (The read value is 0. Writing does not affect the operation.)

8 Document History

Document Title: AN205208 - FR Family MB91460 Series Standby Modes

Document Number: 002-05208

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	09/22/2006	Initial release
			01/19/2007	Added: Disable LV detection and reduce sub regulator voltage
			04/04/2008	“3.1 Stop Mode”, typo corrected: “stop mode” instead of “sleep mode”
*A	5123678	NOFL	22/03/2016	Migrated Spansion Application Note from MCU-AN-300032-E-V12 to Cypress format
*B	5872327	AESATMP9	09/04/2017	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

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