# ATA6020N

## Low-current Microcontroller for Watchdog Function

**DATASHEET**

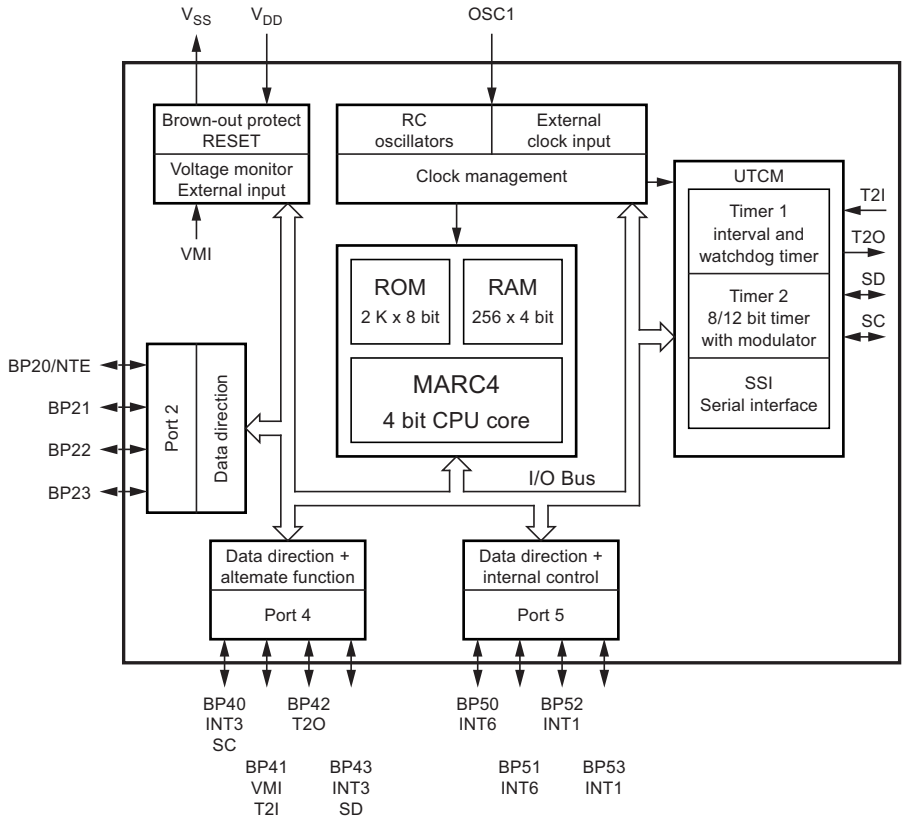
## Features/Benefits

- Programmable system clock with prescaler and three different clock sources
- Very low sleep current (< 1µA)
- Very low power consumption in active, power-down and sleep mode
- 2-Kbyte ROM, 256 × 4-bit RAM
- 12 bi-directional I/Os
- Up to 6 external/internal interrupt sources
- Synchronous serial interface (2-wire, 3-wire)
- Multifunction timer/counter with
  - Watchdog, POR and brown-out function
  - Voltage monitoring inclusive Lo_BAT detection
  - Flash controller ATAM893 available (SSO20)
  - Code-efficient instruction set
  - High-level language programming with qFORTH compiler

## Description

The Atmel® ATA6020N is a member of Atmel 4-bit single-chip microcontroller family. It contains ROM, RAM, parallel I/O ports, one 8-bit programmable multifunction timer/counter with modulator function, voltage supervisor, interval timer with watchdog function and a sophisticated on-chip clock generation with external clock input and integrated RC-oscillators.

**Figure 1. Block Diagram**



# 1. Pin Configuration

**Figure 1-1. Pinning SSO20 Package**

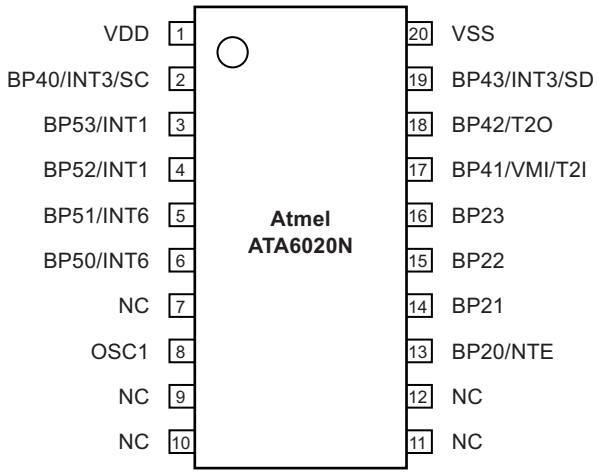



**Table 1-1. Pin Description**

| Name | Type | Function | Alternate Function | Pin Number SS020 | Reset State |
|---|---|---|---|---|---|
| VDD | – | Supply voltage | – | 1 | NA |
| VSS | – | Circuit ground | – | 20 | NA |
| NC | – | Not connected | – | 10 | – |
| NC | – | Not connected | – | 11 | – |
| BP20 | I/O | Bi-directional I/O line of port 2.0 | NTE test mode enable, see also Section 3.2 “Master Reset” on page 11' | 13 | Input |
| BP21 | I/O | Bi-directional I/O line of port 2.1 | – | 14 | Input |
| BP22 | I/O | Bi-directional I/O line of port 2.2 | – | 15 | Input |
| BP23 | I/O | Bi-directional I/O line of port 2.3 | – | 16 | Input |
| BP40 | I/O | Bi-directional I/O line of port 4.0 | SC serial clock or INT3 external interrupt input | 2 | Input |
| BP41 | I/O | Bi-directional I/O line of port 4.1 | VMI voltage monitor input or T2I external clock input Timer 2 | 17 | Input |
| BP42 | I/O | Bi-directional I/O line of port 4.2 | T2O Timer 2 output | 18 | Input |
| BP43 | I/O | Bi-directional I/O line of port 4.3 | SD serial data I/O or INT3 external interrupt input | 19 | Input |
| BP50 | I/O | Bi-directional I/O line of port 5.0 | INT6 external interrupt input | 6 | Input |
| BP51 | I/O | Bi-directional I/O line of port 5.1 | INT6 external interrupt input | 5 | Input |
| BP52 | I/O | Bi-directional I/O line of port 5.2 | INT1 external interrupt input | 4 | Input |
| BP53 | I/O | Bi-directional I/O line of port 5.3 | INT1 external interrupt input | 3 | Input |
| NC | – | Not connected | – | 9 | – |
| NC | – | Not connected | – | 12 | – |
| NC | – | Not connected | – | 7 | – |
| OSC1 | I | Oscillator input | External clock input or external trimming resistor input | 8 | Input |

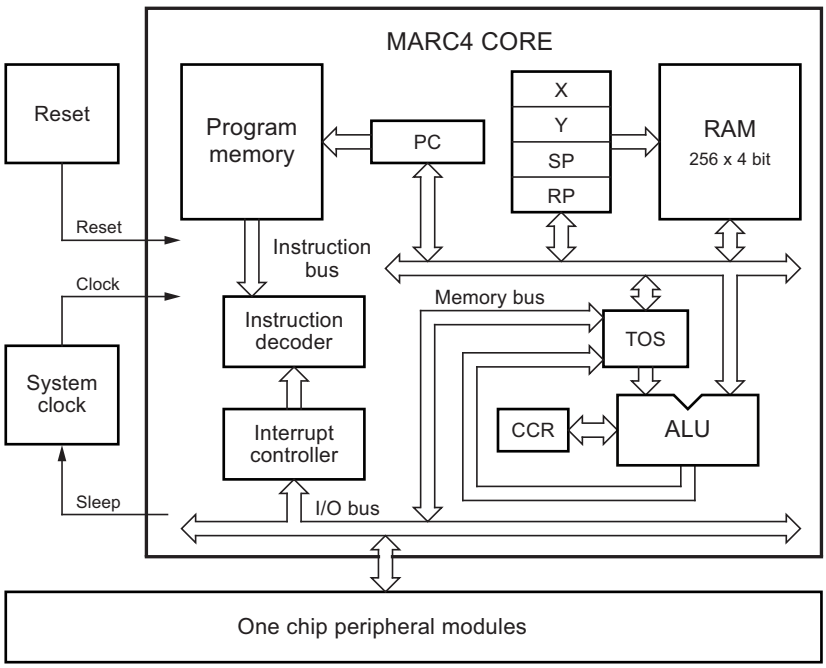
# 2. Introduction

The Atmel® ATA6020N is a member of Atmel 4-bit single-chip microcontroller family. It contains ROM, RAM, parallel I/O ports, one 8-bit programmable multifunction timer/counter, voltage supervisor, interval timer with watchdog function and a sophisticated on-chip clock generation with integrated RC-oscillators.

# 3. MARC4 Architecture General Description

The MARC4 microcontroller consists of an advanced stack-based, 4-bit CPU core and on-chip peripherals. The CPU is based on the HARVARD architecture with physically separated program memory (ROM) and data memory (RAM). Three independent buses, the instruction bus, the memory bus and the I/O bus, are used for parallel communication between ROM, RAM and peripherals. This enhances program execution speed by allowing both instruction prefetching, and a simultaneous communication to the on-chip peripheral circuitry. The extremely powerful integrated interrupt controller with associated eight prioritized interrupt levels supports fast and efficient processing of hardware events. The MARC4 is designed for the high-level programming language qFORTH. The core includes both, an expression and a return stack. This architecture enables high-level language programming without any loss of efficiency or code density.

**Figure 3-1. MARC4 Core**



## 3.1 Components of MARC4 Core

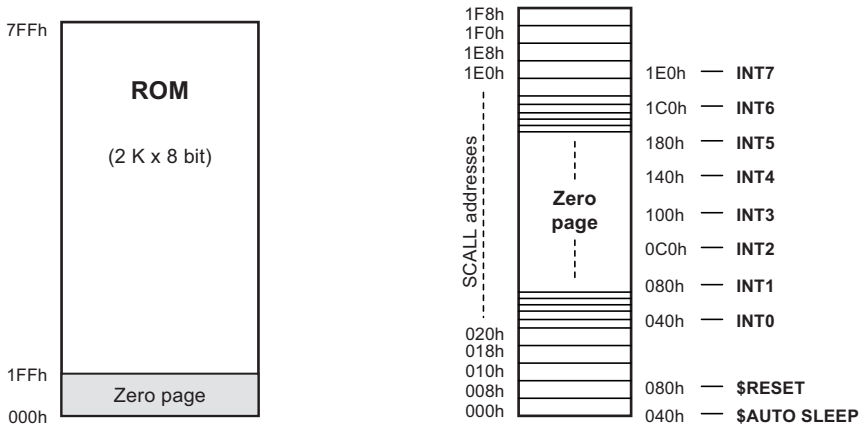
The core contains ROM, RAM, ALU, a program counter, RAM address registers, an instruction decoder and interrupt controller. The following sections describe each functional block in more detail.

### 3.1.1 ROM

The program memory (ROM) is mask programmed with the customer application program during the fabrication of the microcontroller. The ROM is addressed by a 12-bit wide program counter, thus predefining a maximum program bank size of 2-Kbytes. An additional 1-Kbyte of ROM exists, which is reserved for quality control self-test software The lowest user ROM address segment is taken up by a 512-byte Zero page which contains predefined start addresses for interrupt service routines and special subroutines accessible with single byte instructions (SCALL).

The corresponding memory map is shown in Figure 3-2. Look-up tables of constants can also be held in ROM and are accessed via the MARC4's built-in TABLE instruction.

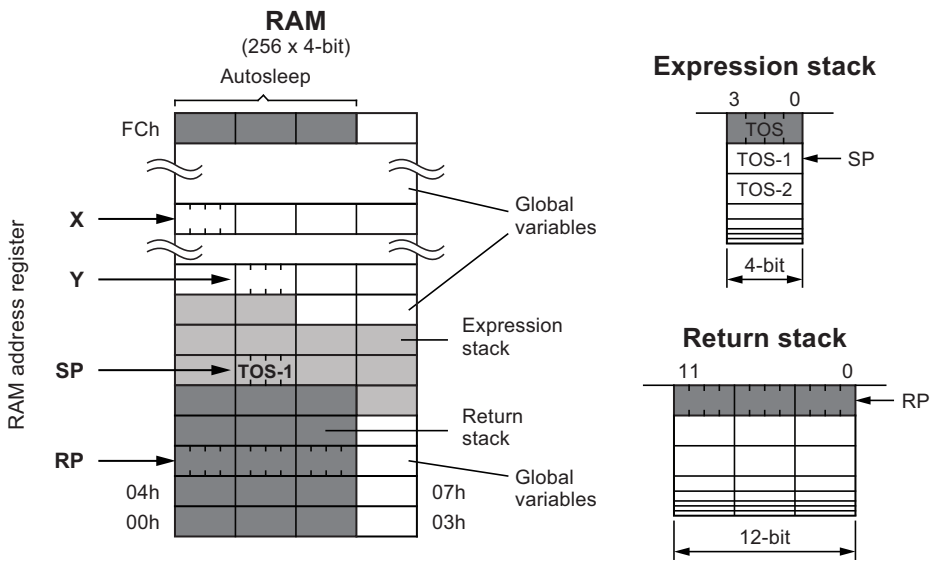
**Figure 3-2. ROM Map of ATA6020N**




### 3.1.2 RAM

The Atmel ATA6020N contains 256 x 4-bit wide static random access memory (RAM), which is used for the expression stack. The return stack and data memory are used for variables and arrays. The RAM is addressed by any of the four 8-bit wide RAM address registers SP, RP, X and Y.

**Figure 3-3. RAM Map**



### 3.1.2.1 Expression Stack

The 4-bit wide expression stack is addressed with the expression stack pointer (SP). All arithmetic, I/O and memory reference operations take their operands, and return their results to the expression stack. The MARC4 performs the operations with the top of stack items (TOS and TOS-1). The TOS register contains the top element of the expression stack and works in the same way as an accumulator. This stack is also used for passing parameters between subroutines and as a scratch pad area for temporary storage of data.

### 3.1.2.2 Return Stack

The 12-bit wide return stack is addressed by the return stack pointer (RP). It is used for storing return addresses of subroutines, interrupt routines and for keeping loop index counts. The return stack can also be used as a temporary storage area.

The MARC4 instruction set supports the exchange of data between the top elements of the expression stack and the return stack. The two stacks within the RAM have a user definable location and maximum depth.

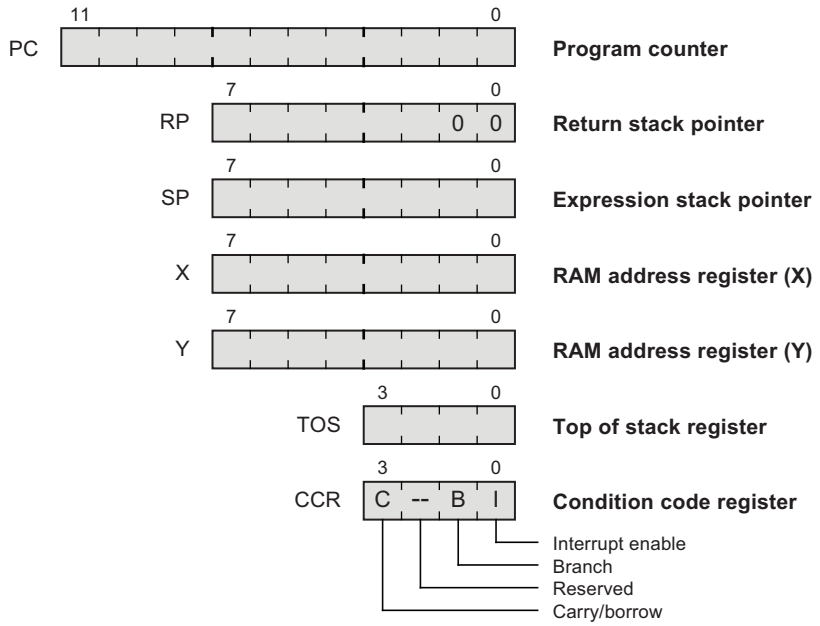
## 3.1.3 Registers

The MARC4 controller has seven programmable registers and one condition code register. They are shown in the following programming model.

### 3.1.3.1 Program Counter (PC)

The program counter is a 12-bit register which contains the address of the next instruction to be fetched from ROM. Instructions currently being executed are decoded in the instruction decoder to determine the internal micro-operations. For linear code (no calls or branches) the program counter is incremented with every instruction cycle. If a branch-, call-, return-instruction or an interrupt is executed, the program counter is loaded with a new address. The program counter is also used with the TABLE instruction to fetch 8-bit wide ROM constants.

**Figure 3-4. Programming Model**



#### 3.1.3.2 RAM Address Registers

The RAM is addressed with the four 8-bit wide RAM address registers: SP, RP, X and Y. These registers allow access to any of the 256 RAM nibbles.

#### 3.1.3.3 Expression Stack Pointer (SP)

The stack pointer contains the address of the next-to-top 4-bit item (TOS-1) of the expression stack. The pointer is automatically pre-incremented if a nibble is moved onto the stack or post-decremented if a nibble is removed from the stack. Every post-decrement operation moves the item (TOS-1) to the TOS register before the SP is decremented. After a reset, the stack pointer has to be initialized with >SP S0 to allocate the start address of the expression stack area.

#### 3.1.3.4 Return Stack Pointer (RP)

The return stack pointer points to the top element of the 12-bit wide return stack. The pointer automatically pre-increments if an element is moved onto the stack, or it post-decrements if an element is removed from the stack. The return stack pointer increments and decrements in steps of 4. This means that every time a 12-bit element is stacked, a 4-bit RAM location is left unwritten. This location is used by the qFORTH compiler to allocate 4-bit variables. After a reset the return stack pointer has to be initialized via >RP FCh.

#### 3.1.3.5 RAM Address Registers (X and Y)

The X and Y registers are used to address any 4-bit item in RAM. A fetch operation moves the addressed nibble onto the TOS. A store operation moves the TOS to the addressed RAM location. By using either the pre-increment or post-decrement addressing mode arrays in RAM can be compared, filled or moved.

#### 3.1.3.6 Top of Stack (TOS)

The top of stack register is the accumulator of the MARC4. All arithmetic/logic, memory reference and I/O operations use this register. The TOS register receives data from the ALU, ROM, RAM or I/O bus.

#### 3.1.3.7 Condition Code Register (CCR)

The 4-bit wide condition code register contains the branch, the carry and the interrupt enable flag. These bits indicate the current state of the CPU. The CCR flags are set or reset by ALU operations. The instructions SET_BCF, TOG_BF, CCR! and DI allow direct manipulation of the condition code register.

#### 3.1.3.8 Carry/Borrow (C)

The carry/borrow flag indicates that the borrowing or carrying out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. During shift and rotate operations, this bit is used as a fifth bit. Boolean operations have no effect on the C-flag.

#### 3.1.3.9 Branch (B)

The branch flag controls the conditional program branching. Should the branch flag has been set by a previous instruction a conditional branch will cause a jump. This flag is affected by arithmetic, logic, shift, and rotate operations.

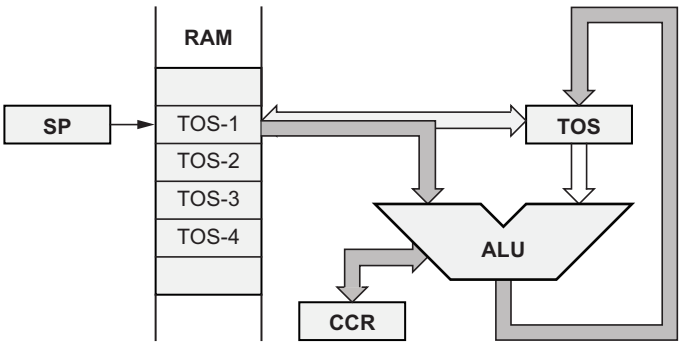
#### 3.1.3.10 Interrupt Enable (I)

The interrupt enable flag globally enables or disables the triggering of all interrupt routines with the exception of the non-maskable reset. After a reset or on executing the DI instruction, the interrupt enable flag is reset thus disabling all interrupts. The core will not accept any further interrupt requests until the interrupt enable flag has been set again by either executing an EI or SLEEP instruction.

### 3.1.4 ALU

The 4-bit ALU performs all the arithmetic, logical, shift and rotate operations with the top two elements of the expression stack (TOS and TOS-1) and returns the result to the TOS. The ALU operations affects the carry/borrow and branch flag in the condition code register (CCR).

**Figure 3-5. ALU Zero-address Operations**




### 3.1.5 I/O Bus

The I/O ports and the registers of the peripheral modules are I/O mapped. All communication between the core and the on-chip peripherals take place via the I/O bus and the associated I/O control. With the MARC4 IN and OUT instructions, the I/O bus allows a direct read or write access to one of the 16 primary I/O addresses. More about the I/O access to the on-chip peripherals is described in the section Section 4. “Peripheral Modules” on page 19. The I/O bus is internal and is not accessible by the customer on the final microcontroller device, but it is used as the interface for the MARC4 emulation (see Section 7.2 “Emulation” on page 64).

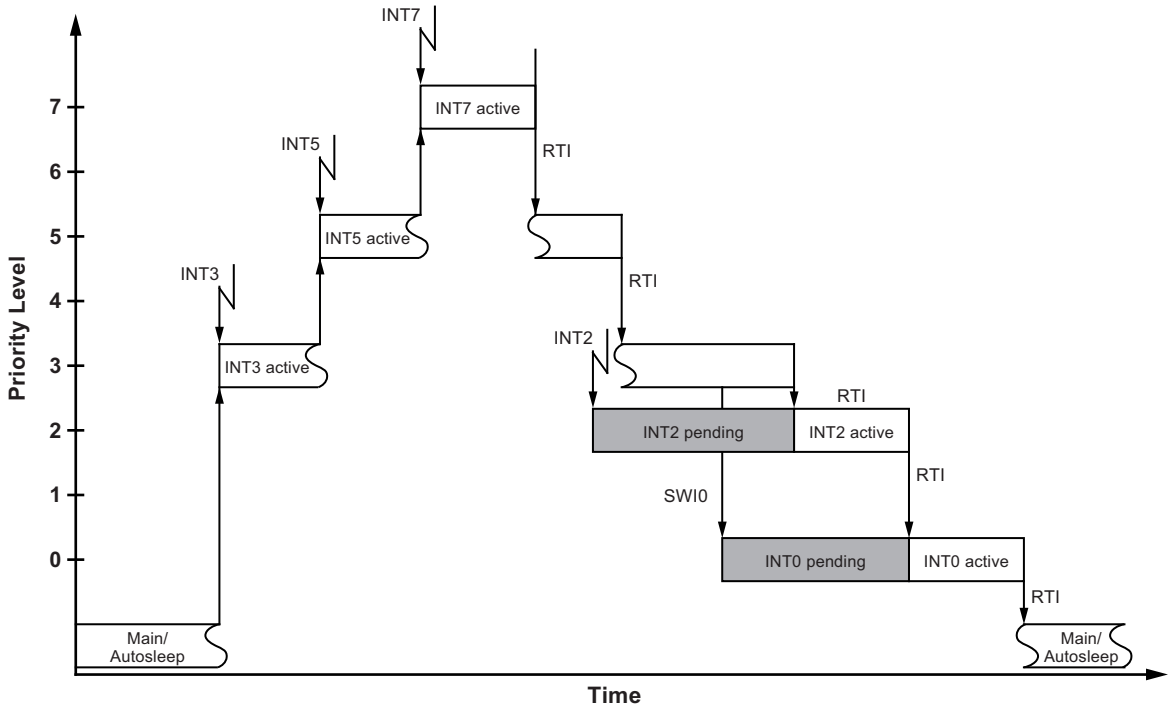
### 3.1.6 Instruction Set

The MARC4 instruction set is optimized for the high level programming language qFORTH. Many MARC4 instructions are qFORTH words. This enables the compiler to generate a fast and compact program code. The CPU has an instruction pipeline allowing the controller to prefetch an instruction from ROM at the same time as the present instruction is being executed. The MARC4 is a zero address machine, the instructions containing only the operation to be performed and no source or destination address fields. The operations are implicitly performed on the data placed on the stack. There are one- and two-byte instructions which are executed within 1 to 4 machine cycles. A MARC4 machine cycle is made up of two system clock cycles (SYSCL). Most of the instructions are only one byte long and are executed in a single machine cycle. For more information refer to the “MARC4 Programmer’s Guide”.

### 3.1.7 Interrupt Structure

The MARC4 can handle interrupts with eight different priority levels. They can be generated from the internal and external interrupt sources or by a software interrupt from the CPU itself. Each interrupt level has a hard-wired priority and an associated vector for the service routine in ROM (see Table 3-1 on page 10). The programmer can postpone the processing of interrupts by resetting the interrupt enable flag (I) in the CCR. An interrupt occurrence will still be registered, but the interrupt routine only starts after the I-flag is set. All interrupts can be masked, and the priority individually software configured by programming the appropriate control register of the interrupting module (see Section 4. “Peripheral Modules” on page 19).

**Figure 3-6. Interrupt Handling**




### 3.1.7.1 Interrupt Processing

In order to process the eight interrupt levels, the MARC4 includes an interrupt controller with two 8-bit wide interrupt pending and interrupt active registers. The interrupt controller samples all interrupt requests during every non-I/O instruction cycle and latches these in the interrupt pending register. If no higher priority interrupt is present in the interrupt active register, it signals the CPU to interrupt the current program execution. If the interrupt enable bit is set, the processor enters an interrupt acknowledge cycle. During this cycle a short call (SCALL) instruction to the service routine is executed and the current PC is saved on the return stack. An interrupt service routine is completed with the RTI instruction. This instruction resets the corresponding bits in the interrupt pending/active register and fetches the return address from the return stack to the program counter. When the interrupt enable flag is reset (triggering of interrupt routines are disabled), the execution of new interrupt service routines is inhibited but not the logging of the interrupt requests in the interrupt pending register. The execution of the interrupt is delayed until the interrupt enable flag is set again. Note that interrupts are only lost if an interrupt request occurs while the corresponding bit in the pending register is still set (i.e., the interrupt service routine is not yet finished).

It should be noted that automatic stacking of the RBR is not carried out by the hardware and so if ROM banking is used, the RBR must be stacked on the expression stack by the application program and restored before the RTI. After a master reset (power-on, brown-out or watchdog reset), the interrupt enable flag and the interrupt pending and interrupt active register are all reset.

### 3.1.7.2 Interrupt Latency

The interrupt latency is the time from the occurrence of the interrupt to the interrupt service routine being activated. This is extremely short (taking between 3 to 5 machine cycles depending on the state of the core).

**Table 3-1. Interrupt Priority Table**

| Interrupt | Priority | ROM Address | Interrupt Opcode | Function |
|---|---|---|---|---|
| INT0 | lowest | 040h | C8h (SCALL 040h) | Software interrupt (SWI0) |
| INT1 | \| | 080h | D0h (SCALL 080h) | External hardware interrupt, any edge at BP52 or BP53 |
| INT2 | \| | 0C0h | D8h (SCALL 0C0h) | Timer 1 interrupt |
| INT3 | \| | 100h | E8h (SCALL 100h) | SSI interrupt or external hardware interrupt at BP40 or BP43 |
| INT4 | \| | 140h | E8h (SCALL 140h) | Timer 2 interrupt |
| INT5 | \| | 180h | F0h (SCALL 180h) | Software interrupt (SW15) |
| INT6 | ↓ | 1C0h | F8h (SCALL 1C0h) | External hardware interrupt, at any edge at BP50 or BP51 |
| INT7 | highest | 1E0h | FCh (SCALL 1E0h) | Voltage monitor (VM) interrupt |

**Table 3-2. Hardware Interrupts**

| Interrupt | Interrupt Mask | | Interrupt Source |
|---|---|---|---|
| | Register | Bit | |
| INT1 | P5CR | P52M1, P52M2<br>P53M1, P53M2 | Any edge at BP52<br>Any edge at BP53 |
| INT2 | T1M | T1IM | Timer 1 |
| INT3 | SISC | SIM | SSI buffer full/empty or BP40/BP43 interrupt |
| INT4 | T2CM | T2IM | Timer 2 compare match/overflow |
| INT6 | P5CR | P50M1, P50M2<br>P51M1, P51M2 | Any edge at BP50<br>Any edge at BP51 |
| INT7 | VCM | VIM | External/internal voltage monitoring |

### 3.1.7.3 Software Interrupts

The programmer can generate interrupts by using the software interrupt instruction (SWI), which is supported in qFORTH by predefined macros named SWI0...SWI7. The software triggered interrupt operates exactly like any hardware triggered interrupt. The SWI instruction takes the top two elements from the expression stack and writes the corresponding bits via the I/O bus to the interrupt pending register. Therefore, by using the SWI instruction, interrupts can be re-prioritized or lower priority processes scheduled for later execution.

### 3.1.7.4 Hardware Interrupts

In the ATA6020N, there are eleven hardware interrupt sources with seven different levels. Each source can be masked individually by mask bits in the corresponding control registers. An overview of the possible hardware configurations is shown in Table 3-2.

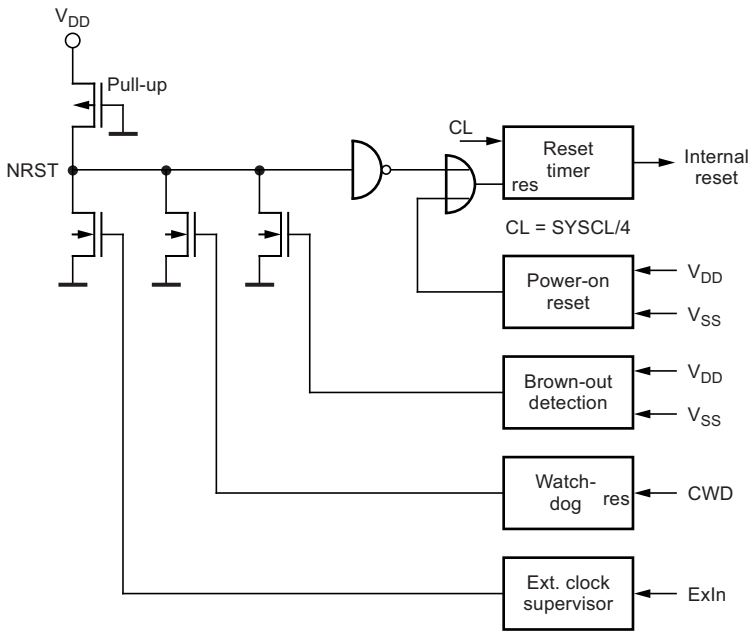
## 3.2 Master Reset

The master reset forces the CPU into a well-defined condition. It is unmaskable and is activated independent of the current program state. It can be triggered by either initial supply power-up, a short collapse of the power supply, the brown-out detection circuitry, a watchdog time-out, or an external input clock supervisor stage (see Figure 3-7). A master reset activation will reset the interrupt enable flag, the interrupt pending register and the interrupt active register. During the power-on reset phase, the I/O bus control signals are set to reset mode, thereby, initializing all on-chip peripherals. All bi-directional ports are set to input mode.

**Attention:** During any reset phase, the BP20/NTE input is driven towards $V_{DD}$ by an additional internal strong pull-up transistor. This pin must not be pulled down to $V_{SS}$ during reset by any external circuitry representing a resistor of less than 150kΩ.

Releasing the reset results in a short call instruction (opcode C1h) to the ROM address 008h. This activates the initialization routine $RESET which in turn has to initialize all necessary RAM variables, stack pointers and peripheral configuration registers.

**Figure 3-7. Reset Configuration**




### 3.2.1 Power-on Reset and Brown-out Detection

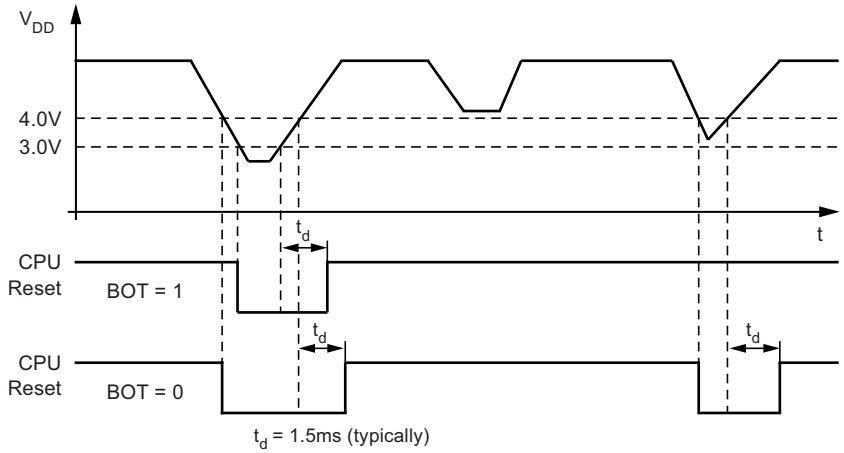
The Atmel® ATA6020N has a fully integrated power-on reset and brown-out detection circuitry. For reset generation no external components are needed.

These circuits ensure that the core is held in the reset state until the minimum operating supply voltage has been reached. A reset condition will also be generated should the supply voltage drop momentarily below the minimum operating level except when a power-down mode is activated (the core is in SLEEP mode and the peripheral clock is stopped). In this power-down mode the brown-out detection is disabled.

Two values for the brown-out voltage threshold are programmable via the BOT bit in the SC-register.

A power-on reset pulse is generated by a $V_{DD}$ rise across the default BOT voltage level (3.0V). A brown-out reset pulse is generated when $V_{DD}$ falls below the brown-out voltage threshold. Two values for the brown-out voltage threshold are programmable via the BOT-bit in the SC-register. When the controller runs in the upper supply voltage range with a high system clock frequency, the high threshold must be used. When it runs with a lower system clock frequency, the low threshold and a wider supply voltage range may be chosen. For further details, see the electrical specification and the SC-register description for BOT programming.

**Figure 3-8. Brown-out Detection**




BOT = 1, low brown-out voltage threshold. (3.0V is the reset value).
BOT = 0, high brown-out voltage threshold (4.0V).

### 3.2.2 Watchdog Reset

The watchdog's function can be enabled at the WDC-register and triggers a reset with every watchdog counter overflow. To suppress the watchdog reset, the watchdog counter must be regularly reset by reading the watchdog register address (CWD). The CPU reacts in exactly the same manner as a reset stimulus from any of the above sources.

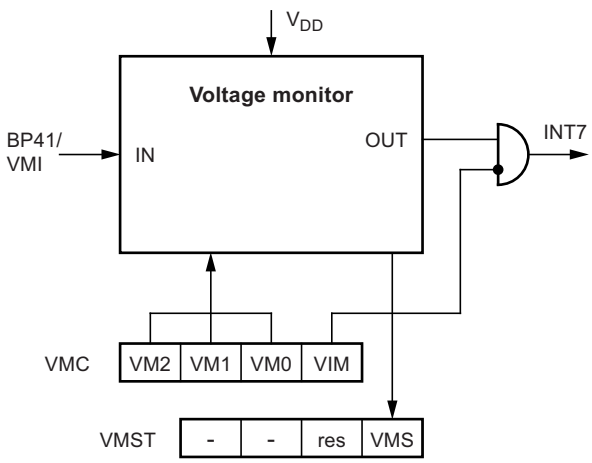
### 3.2.3 External Clock Supervisor

The external input clock supervisor function can be enabled if the external input clock is selected within the CM- and SC-registers of the clock module. The CPU reacts in exactly the same manner as a reset stimulus from any of the above sources.

## 3.3 Voltage Monitor

The voltage monitor consists of a comparator with internal voltage reference. It is used to supervise the supply voltage or an external voltage at the VMI pin. The comparator for the supply voltage has two internal programmable thresholds: one lower threshold (4.0V) and one higher threshold (5.0V). For external voltages at the VMI pin, the comparator threshold is set to $V_{BG}$ = 1.25V. The VMS-bit indicates if the supervised voltage is below (VMS = 0) or above (VMS = 1) this threshold. An interrupt can be generated when the VMS-bit is set or reset to detect a rising or falling slope. A voltage monitor interrupt (INT7) is enabled when the interrupt mask bit (VIM) is reset in the VMC-register.

**Figure 3-9. Voltage Monitor**



### 3.3.1 Voltage Monitor Control/Status Register

Primary register address: 'F'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **VMC: Write** | **VM2** | **VM1** | **VM0** | **VIM** | **Reset value: 1111b** |
| **VMST: Read** | **—** | **—** | **reserved** | **VMS** | **Reset value: xx11b** |

**VM2**: **V**oltage monitor **M**ode bit **2**
**VM1**: **V**oltage monitor **M**ode bit **1**
**VM0**: **V**oltage monitor **M**ode bit **0**

**Table 3-3. Voltage Monitor Modes**

| VM2 | VM1 | VM0 | Function |
|---|---|---|---|
| 1 | 1 | 1 | Disable voltage monitor |
| 1 | 1 | 0 | External (VIM input), internal reference threshold (1.25V), interrupt with negative slope |
| 1 | 0 | 1 | Not allowed |
| 1 | 0 | 0 | External (VMI input), internal reference threshold (1.25V), interrupt with positive slope |
| 0 | 1 | 1 | Internal (supply voltage), high threshold (5.0V), interrupt with negative slope |
| 0 | 1 | 0 | Not allowed |
| 0 | 0 | 1 | Internal (supply voltage), low threshold (4.0V), interrupt with negative slope |
| 0 | 0 | 0 | Not allowed |

**VIM** **V**oltage **I**nterrupt **M**ask bit

- VIM = 0, voltage monitor interrupt is enabled
- VIM = 1, voltage monitor interrupt is disabled

**VMS** **V**oltage **M**onitor **S**tatus bit

- VMS = 0, the voltage at the comparator input is below $V_{Ref}$
- VMS = 1, the voltage at the comparator input is above $V_{Ref}$

**Figure 3-10. Internal Supply Voltage Supervisor**

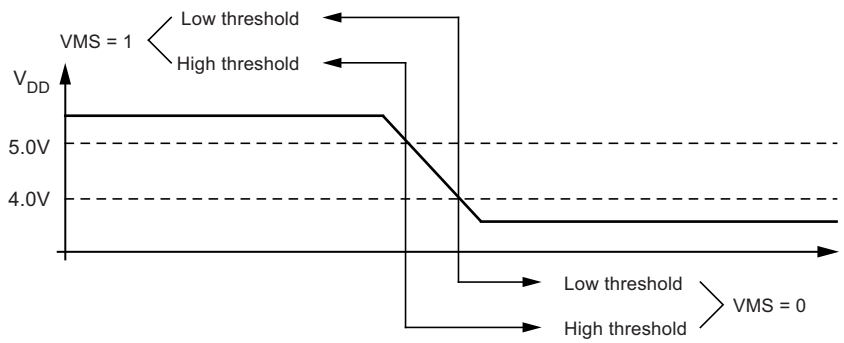
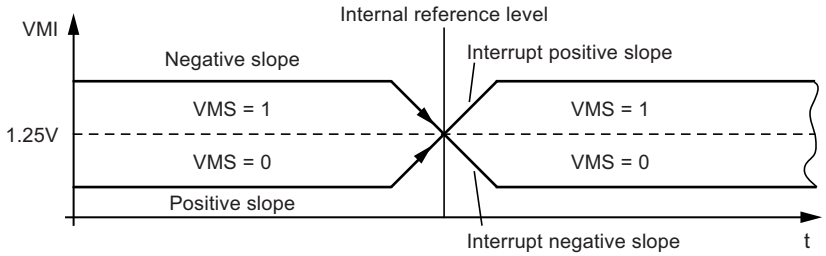


**Figure 3-11. External Input Voltage Supervisor**




## 3.4 Clock Generation

### 3.4.1 Clock Module

The Atmel® ATA6020N contains a clock module with two different internal RC-oscillator types. OSC1 can be used as input for external clocks or to connect an external trimming resistor for RC-oscillator 2. All necessary circuitry, except the trimming resistor, is integrated on-chip. One of these oscillator types or an external input clock can be selected to generate the system clock (SYSCL).

In applications that do not require exact timing, it is possible to use the fully integrated RC-oscillator 1 without any external components. The RC-oscillator 1 center frequency tolerance is better than ±50%. RC-oscillator 2 is a trimmable oscillator whereby the oscillator frequency can be trimmed with an external resistor attached between OSC1 and GND. In this configuration, RC-oscillator 2 frequency can be maintained stable to within a tolerance of ±15% over the full operating temperature and voltage range.

The clock module is programmable via software with the clock management register (CM) and the system configuration register (SC). The required oscillator configuration can be selected with the OS1-bit and the OS0-bit in the SC-register. A programmable 4-bit divider stage allows the adjustment of the system clock speed. A special feature of the clock management is that an external oscillator may be used and switched on and off via a port pin for the power-down mode. Before the external clock is switched off, the internal RC-oscillator 1 must be selected with the CCS-bit and then the SLEEP mode may be activated. In this state an interrupt can wake up the controller with the RC-oscillator, and the external oscillator can be activated and selected by software. A synchronization stage avoids clock periods that are too short if the clock source or the clock speed is changed. If an external input clock is selected, a supervisor circuit monitors the external input and generates a hardware reset if the external clock source fails or drops below 500kHz for more than 1ms.

**Figure 3-12. Clock Module**

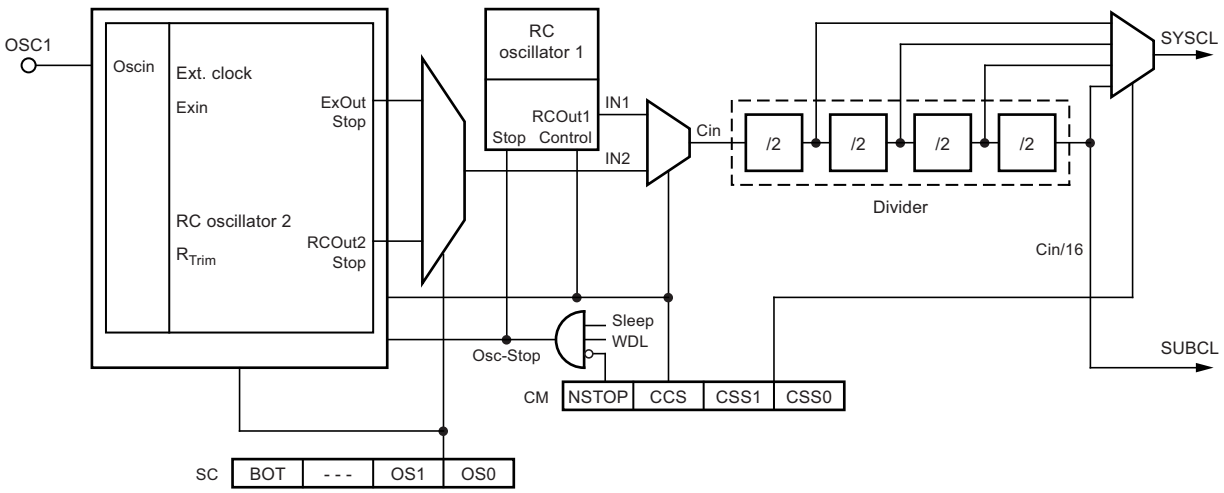


**Table 3-4. Clock Modes**

| Mode | OS1 | OS0 | Clock Source for SYSCL | | Clock Source for SUBCL |
|---|---|---|---|---|---|
| | | | CCS = 1 | CCS = 0 | |
| 1 | 1 | 1 | RC-oscillator 1 (internal) | External input clock | $C_{in}/16$ |
| 2 | 0 | 1 | RC-oscillator 1 (internal) | RC-oscillator 2 with external trimming resistor | $C_{in}/16$ |

The clock module generates two output clocks. One is the system clock (SYSCL) and the other the periphery (SUBCL). The SYSCL can supply the core and the peripherals and the SUBCL can supply only the peripherals with clocks. The modes for clock sources are programmable with the OS1-bit and OS0-bit in the SC-register and the CCS-bit in the CM-register.

### 3.4.2 Oscillator Circuits and External Clock Input Stage

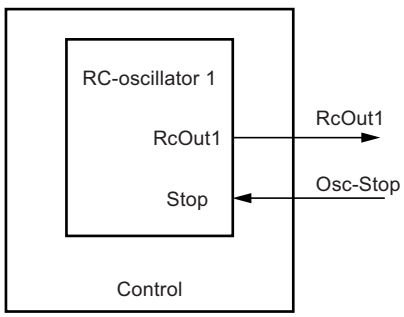
The Atmel® ATA6020N consists of two different internal RC-oscillators and one external clock input stage.

#### 3.4.2.1 RC-oscillator 1 Fully Integrated

For timing insensitive applications, it is possible to use the fully integrated RC-oscillator 1. It operates without any external components and saves additional costs. The RC-oscillator 1 center frequency tolerance is better than ±50% over the full temperature and voltage range.

The basic center frequency of the RC-oscillator 1 is $f_O \approx 4.0$MHz The RC-oscillator 1 is selected by default after power-on reset.

**Figure 3-13. RC-oscillator 1**



### 3.4.2.2 External Input Clock

The OSC1 can be driven by an external clock source provided it meets the specified duty cycle, rise and fall times and input levels. Additionally, the external clock stage contains a supervisory circuit for the input clock. The supervisor function is controlled via the OS1, OS0-bit in the SC-register and the CCS-bit in the CM-register. If the external input clock fails and CCS = 0 is set in the CM-register, the supervisory circuit generates a hardware reset. The input clock has failed if the frequency is less than 500kHz for more than 1ms.

**Figure 3-14. External Input Clock**

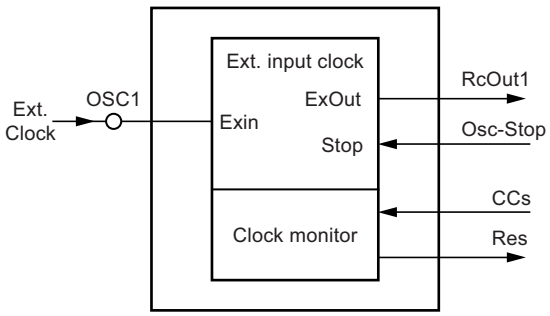



**Table 3-5. Supervisor Function Control Bits**

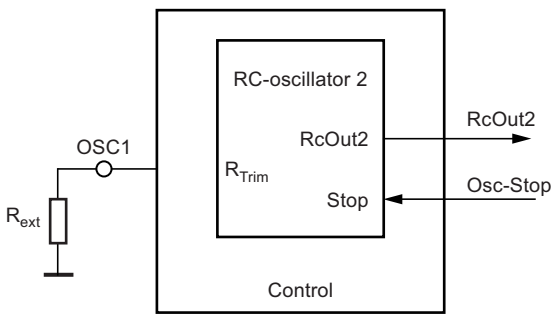
| OS1 | OS0 | CCS | Supervisor Reset Output (Res) |
|---|---|---|---|
| 1 | 1 | 0 | Enable |
| 1 | 1 | 1 | Disable |
| x | 0 | x | Disable |

### 3.4.2.3 RC-oscillator 2 with External Trimming Resistor

The RC-oscillator 2 is a high resolution trimmable oscillator whereby the oscillator frequency can be trimmed with an external resistor between OSC1 and $V_{DD}$. In this configuration, the RC-oscillator 2 frequency can be maintained stable to within a tolerance of ±10% over the full operating temperature and voltage range from $V_{DD}$ = 3.5V to 5.5V.

For example: An output frequency at the RC-oscillator 2 of 1.6MHz, can be obtained by connecting a resistor $R_{ext}$ = 47kΩ (see Figure 3-15).

**Figure 3-15. RC-oscillator 2**



### 3.4.3 Clock Management

The clock management register controls the system clock divider and synchronization stage. Writing to this register triggers the synchronization cycle.

#### 3.4.3.1 Clock Management Register (CM)

Auxiliary register address:'3'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **CM** | **NSTOP** | **CCS** | **CSS1** | **CSS0** | **Reset value: 1111b** |

**NSTOP** — **N**ot **STOP** peripheral clock
NSTOP = 0, stops the peripheral clock while the core is in SLEEP mode
NSTOP = 1, enables the peripheral clock while the core is in SLEEP mode

**CCS** — **C**ore **C**lock **S**elect
CCS = 1, the internal RC-oscillator 1 generates SYSCL
CCS = 0, an external clock source or the RC-oscillator 2 with the external resistor at OSC1 generates SYSCL dependent on the setting of OS0 and OS1 in the system configuration register

**CSS1** — **C**ore **S**peed **S**elect **1**

**CSS0** — **C**ore **S**peed **S**elect **0**

**Table 3-6. Core Speed Select**

| CSS1 | CSS0 | Divider | Note |
|---|---|---|---|
| 0 | 0 | 16 | – |
| 1 | 1 | 8 | Reset value |
| 1 | 0 | 4 | – |
| 0 | 1 | 2 | – |

#### 3.4.3.2 System Configuration Register (SC)

Primary register address:'3'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **SC: write** | **BOT** | **–** | **OS1** | **OS0** | **Reset value: 1x11b** |

**BOT** — **B**rown-**O**ut **T**hreshold
BOT = 1, low brown-out voltage threshold (3.0V)
BOT = 0, high brown-out voltage threshold (4.0V)

**OS1** — **O**scillator **S**elect **1**

**OS0** — **O**scillator **S**elect **0**

**Table 3-7. Oscillator Select**

| Mode | OS1 | OS0 | Input for SUBCL | Selected Oscillators |
|---|---|---|---|---|
| 1 | 1 | 1 | $C_{in}$/16 | RC-oscillator 1 and external input clock |
| 2 | 0 | 1 | $C_{in}$/16 | RC-oscillator 1 and RC-oscillator 2 |

Note: f bit CCS = 0 in the CM-register, the RC-oscillator 1 always stops.

## 3.5 Power-down Modes

The sleep mode is a shut-down condition which is used to reduce the average system power consumption in applications where the microcontroller is not fully utilized. In this mode, the system clock is stopped. The sleep mode is entered via the SLEEP instruction. This instruction sets the interrupt enable bit (I) in the condition code register to enable all interrupts and stops the core. During the sleep mode the peripheral modules remain active and are able to generate interrupts. The microcontroller exits the sleep mode by carrying out any interrupt or a reset.

The sleep mode can only be kept when none of the interrupt pending or active register bits are set. The application of the $AUTOSLEEP routine ensures the correct function of the sleep mode.

The total power consumption is directly proportional to the active time of the microcontroller. For a rough estimation of the expected average system current consumption, the following formula should be used:

$I_{total}\ (V_{DD}, f_{syscl}) = I_{Sleep} + (I_{DD} \times t_{active}/t_{total})$

$I_{DD}$ depends on $V_{DD}$ and $f_{syscl}$

The Atmel® ATA6020N has various power-down modes. During the sleep mode the clock for the MARC4 core is stopped. With the NSTOP-bit in the clock management register (CM), it is programmable if the clock for the on-chip peripherals is active or stopped during the sleep mode. If the clock for the core and the peripherals is stopped the selected oscillator is switched off.

**Table 3-8. Power-down Modes**

| Mode | CPU Core | Osc-Stop[1] | Brown-out Function | RC-oscillator 1 RC-oscillator 2 | External Input Clock |
|---|---|---|---|---|---|
| Active | RUN | NO | Active | RUN | YES |
| Power-down | SLEEP | NO | Active | RUN | YES |
| SLEEP | SLEEP | YES | STOP | STOP | STOP |

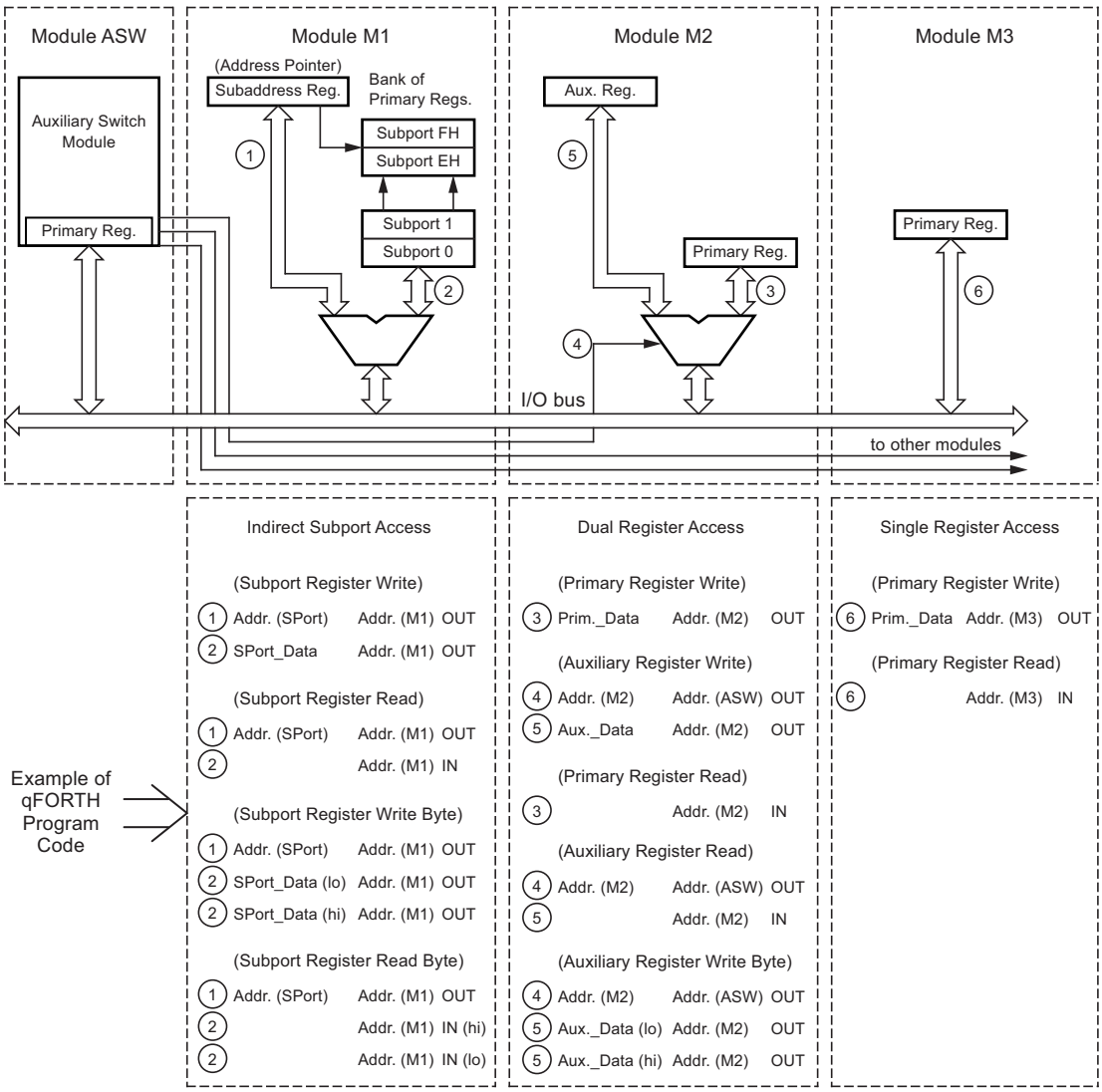
Note: 1. Osc-Stop = SLEEP and NSTOP and WDL

# 4. Peripheral Modules

## 4.1 Addressing Peripherals

Accessing the peripheral modules takes place via the I/O bus (see Figure 4-1). The IN or OUT instructions allow direct addressing of up to 16 I/O modules. A dual register addressing scheme has been adopted to enable direct addressing of the primary register. To address the auxiliary register, the access must be switched with an auxiliary switching module. Thus, a single IN (or OUT) to the module address will read (or write into) the modules primary register. Accessing the auxiliary register is performed with the same instruction preceded by writing the module address into the auxiliary switching module. Byte wide registers are accessed by multiple IN (or OUT) instructions. For more complex peripheral modules, with a larger number of registers, extended addressing is used. In this case, a bank of up to 16 subport registers are indirectly addressed with the subport address. The first OUT-instruction writes the subport address to the sub-address register, the second IN or OUT instruction reads data from or writes data to the addressed subport.

**Figure 4-1. Example of I/O Addressing**




Addr. (ASW) = Auxiliary Switch Module Address
Addr. (MX) = Module Mx Address
Addr. (SPort) = Subport Address
Prim._Data = Data to be written into Primary Register
Aux._Data = Data to be written into Auxiliary Register
Aux._Data (lo) = Data to be written into Auxiliary Register (low nibble)
Aux._Data (hi) = Data to be written into Auxiliary Register (high nibble)
SPort_Data (lo) = Data to be written into Subport (low nibble)
SPort_Data (hi) = Data to be written into Subport (high nibble)
(lo) = SPort_Data (low nibble)
(hi) = SPort_Data (high nibble)

**Table 4-1. Peripheral Addresses**

| Port Address | | Name | Write/Read | Reset Value | Register Function | Module Type | See Page |
|---|---|---|---|---|---|---|---|
| 2 | | P2DAT | W/R | 1111b | Port 2 - data register/pin data | M2 | 22 |
| | Aux. | P2CR | W | 1111b | Port 2 - control register | | 22 |
| 3 | | SC | W | 1x11b | Port 3 - system configuration register | M3 | 17 |
| | | CWD | R | xxxxb | Watchdog reset | M3 | 12 |
| | Aux. | CM | W | 1111b | Port 3 - clock management register | M2 | 17 |
| 4 | | P4DAT | W/R | 1111b | Port 4 - data register/pin data | M2 | 25 |
| | Aux. | P4CR | W | 1111 1111b | Port 4 - control register (byte) | | 25 |
| 5 | | P5DAT | W/R | 1111b | Port 5 - data register/pin data | M2 | 24 |
| | Aux. | P5CR | W | 1111 1111b | Port 5 - control register (byte) | | 24 |
| 6 | | — | | | Reserved | | |
| 7 | | T12SUB | W | — | Data to Timer 1/2 subport | M1 | 19 |
| | Support address | | | | | | |
| | 0 | T2C | W | 0000b | Timer 2 control register | M1 | 37 |
| | 1 | T2M1 | W | 1111b | Timer 2 mode register 1 | M1 | 37 |
| | 2 | T2M2 | W | 1111b | Timer 2 mode register 2 | M1 | 39 |
| | 3 | T2CM | W | 0000b | Timer 2 compare mode register | M1 | 40 |
| | 4 | T2CO1 | W | 1111b | Timer 2 compare register 1 | M1 | 40 |
| | 5 | T2CO2 | W | 1111 1111b | Timer 2 compare register 2 (byte) | M1 | 40 |
| | 6 | — | — | — | Reserved | | |
| | 7 | — | — | — | Reserved | | |
| | 8 | T1C1 | W | 1111b | Timer 1 control register 1 | M1 | 28 |
| | 9 | T1C2 | W | x111b | Timer 1 control register 2 | M1 | 29 |
| | A | WDC | W | 1111b | Watchdog control register | M1 | 29 |
| | B-F | | | | Reserved | | |
| 8 | | ASW | W | 1111b | Auxiliary/switch register | ASW | 19 |
| 9 | | STB | W | xxxx xxxxb | Serial transmit buffer (byte) | M2 | 50 |
| | | SRB | R | xxxx xxxxb | Serial receive buffer (byte) | | 50 |
| | Aux. | SIC1 | W | 1111b | Serial interface control register 1 | | 48 |
| A | | SISC | W/R | 1x11b | Serial interface status/control register | M2 | 50 |
| | Aux. | SIC2 | W | 1111b | Serial interface control register 2 | | 49 |
| B | | — | | | Reserved | | |
| C | | — | | | Reserved | | |
| D | | RBR | W | 0000b | ROM bank switch register | M3 | 7 |
| E | | — | | — | Reserved | | |
| F | | VMC | W | 1111b | Voltage monitor control register | M3 | 13 |
| | | VMST | R | xx11b | Voltage monitor status register | M3 | 13 |

## 4.2 Bi-directional Ports

Ports 2, 4 and 5 are 4 bits wide. All ports may be used for data input or output. All ports are equipped with Schmitt trigger inputs and a variety of mask options for open-drain, open-source, full-complementary outputs, pull-up and pull-down transistors. All Port Data Registers (PxDAT) are I/O mapped to the primary address register of the respective port address and the Port Control Register (PxCR), to the corresponding auxiliary register.

There are three different directional ports available:

Port 2 — 4-bit wide bitwise-programmable I/O port.

Port 5 — 4-bit wide bitwise-programmable bi-directional port with optional static pull-ups and programmable interrupt logic.

Port 4 — 4-bit wide bitwise-programmable bi-directional port also provides the I/O interface to Timer 2, SSI, voltage monitor input and external interrupt input.

### 4.2.1 Bi-directional Port 2

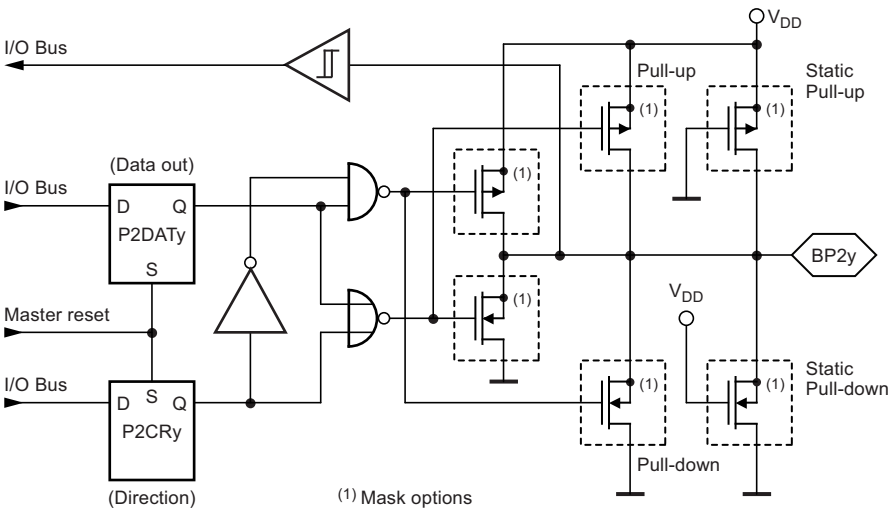
This, and all other bi-directional ports include a bitwise-programmable Control Register (P2CR), which enables the individual programming of each port bit as input or output. It also opens up the possibility of reading the pin condition when in output mode. This is a useful feature for self-testing and for serial bus applications.

Port 2, however, has an increased drive capability and an additional low resistance pull-up/-down transistor mask option.

Care should be taken connecting external components to BP20/NTE. During any reset phase, the BP20/NTE input is driven towards $V_{DD}$ by an additional internal strong pull-up transistor. This pin must not be pulled down (active or passive) to $V_{SS}$ during reset by any external circuitry representing a resistor of less than 150 kΩ. This prevents the circuit from unintended switching to test mode enable through the application circuitry at pin BP20/NTE. Resistors less than 150kΩ might lead to an undefined state of the internal test logic thus disabling the application firmware.

To avoid any conflict with the optional internal pull-down transistors, BP20 handles the pull-down options in a different way than all other ports. BP20 is the only port that switches off the pull-down transistors during reset.

**Figure 4-2. Bi-directional Port 2**



#### 4.2.1.1 Port 2 Data Register (P2DAT)

Primary register address:'2'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **P2DAT** | **P2DAT3** | **P2DAT2** | **P2DAT1** | **P2DAT0** | **Reset value: 1111b** |

Bit 3 = MSB, Bit 0 = LSB

#### 4.2.1.2 Port 2 Control Register (P2CR)

Auxiliary register address:'2'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **P2CR** | **P2CR3** | **P2CR2** | **P2CR1** | **P2CR0** | **Reset value: 1111b** |

Value: 1111b means all pins in input mode

**Table 4-2. Port 2 Control Register**

| Code<br>3 2 1 0 | Function |
|---|---|
| x x x 1 | BP20 in input mode |
| x x x 0 | BP20 in output mode |
| x x 1 x | BP21 in input mode |
| x x 0 x | BP21 in output mode |
| x 1 x x | BP22 in input mode |
| x 0 x x | BP22 in output mode |
| 1 x x x | BP23 in input mode |
| 0 x x x | BP23 in output mode |

### 4.2.2 Bi-directional Port 5

This, and all other bi-directional ports include a bitwise-programmable control register (P5CR), which allows individual programming of each port bit as input or output. It also opens up the possibility of reading the pin condition when in output mode. This is a useful feature for self testing and for serial bus applications.

The port pins can also be used as external interrupt inputs (see Figure 4-3 on page 23 and Figure 4-4 on page 23). The interrupts (INT1 and INT6) can be masked or independently configured to trigger on either edge. The interrupt configuration and port direction is controlled by the Port 5 control register (P5CR). An additional low resistance pull-up/-down transistor mask option provides an internal bus pull-up for serial bus applications.

The port 5 data register (P5DAT) is I/O mapped to the primary address register of address '5'h and the port 5 control register (P5CR) to the corresponding auxiliary register. The P5CR is a byte-wide register and is configured by writing first the low nibble then the high nibble (see Section 4.1 "Addressing Peripherals" on page 19).

**Figure 4-3. Bi-directional Port 5**

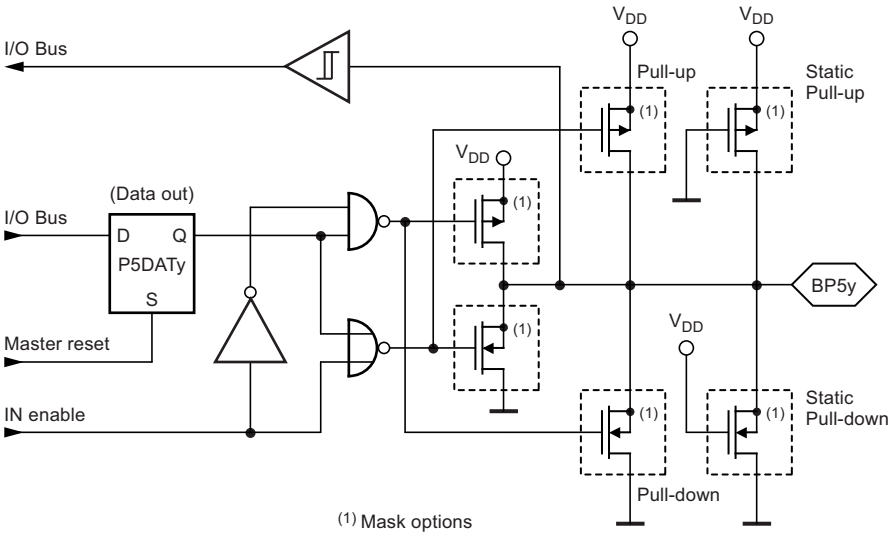
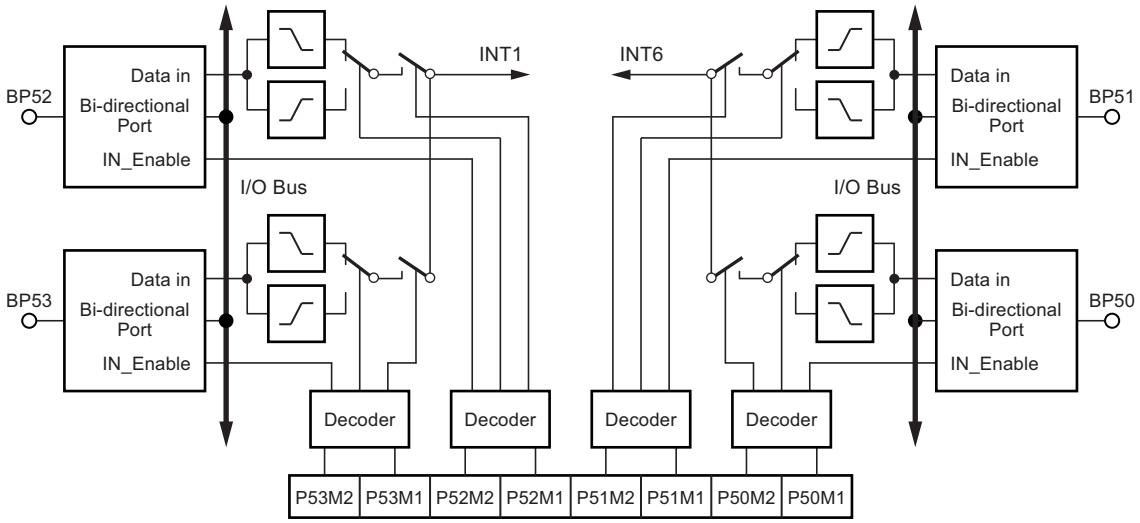



**Figure 4-4. Port 5 External Interrupts**



#### 4.2.2.1 Port 5 Data Register (P5DAT)

Primary register address:'5'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **P5DAT** | **P5DAT3** | **P5DAT2** | **P5DAT1** | **P5DAT0** | **Reset value: 1111b** |

#### 4.2.2.2 Port 5 Control Register (P5CR) Byte Write

Auxiliary register address:'5'hex

| | | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|---|
| **P5CR** | **First write cycle** | **P51M2** | **P51M1** | **P50M2** | **P50M1** | **Reset value: 1111b** |
| | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | |
| | **Second write cycle** | **P53M2** | **P53M1** | **P52M2** | **P52M1** | **Reset value: 1111b** |

P5xM2, P5xM1 – Port 5x Interrupt Mode/Direction Code

**Table 4-3. Port 5 Control Register**

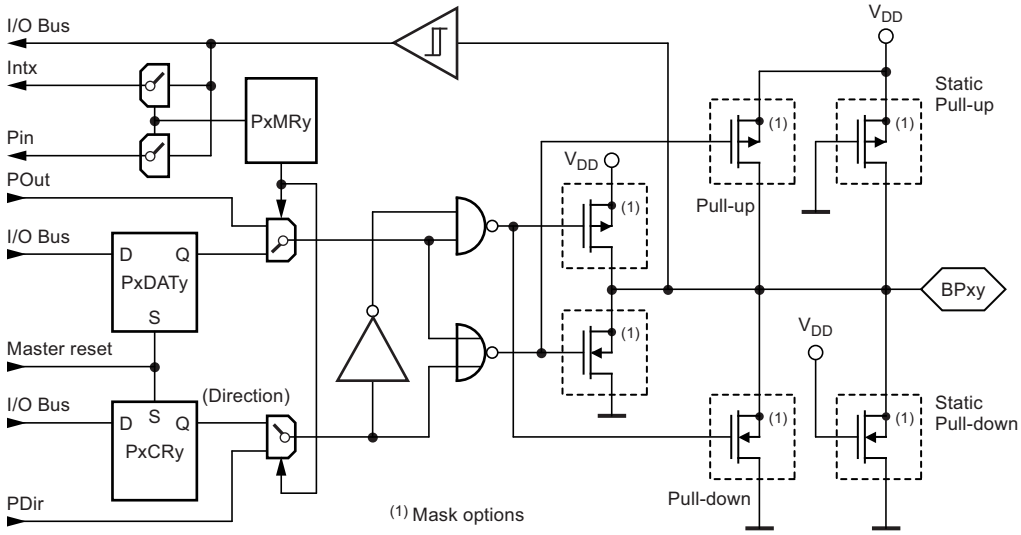
| Auxiliary Address:'5'hex First Write Cycle | | Second Write Cycle | |
|---|---|---|---|
| **Code 3 2 1 0** | **Function** | **Code 3 2 1 0** | **Function** |
| x x 1 1 | BP50 in input mode - interrupt disabled | x x 1 1 | BP52 in input mode – interrupt disabled |
| x x 0 1 | BP50 in input mode - rising edge interrupt | x x 0 1 | BP52 in input mode – rising edge interrupt |
| x x 1 0 | BP50 in input mode - falling edge interrupt | x x 1 0 | BP52 in input mode – falling edge interrupt |
| x x 0 0 | BP50 in output mode - interrupt disabled | x x 0 0 | BP52 in output mode – interrupt disabled |
| 1 1 x x | BP51 in input mode - interrupt disabled | 1 1 x x | BP53 in input mode – interrupt disabled |
| 0 1 x x | BP51 in input mode - rising edge interrupt | 0 1 x x | BP53 in input mode – rising edge interrupt |
| 1 0 x x | BP51 in input mode - falling edge interrupt | 1 0 x x | BP53 in input mode – falling edge interrupt |
| 0 0 x x | BP51 in output mode - interrupt disabled | 0 0 x x | BP53 in output mode – interrupt disabled |

### 4.2.3 Bi-directional Port 4

The bi-directional port 4 is both a bitwise configurable I/O port and provides the external pins for the Timer 2, SSI and the voltage monitor input (VMI). As a normal port, it performs in exactly the same way as bi-directional port 2 (see Figure 4-2 on page 21). Two additional multiplexes allow data and port direction control to be passed over to other internal modules (timer 2, VM or SSI). The I/O-pins for the SC and SD lines have an additional mode to generate an SSI-interrupt.

All four port 4 pins can be individually switched by the P4CR-register. Figure 4-5 on page 25 shows the internal interfaces to bi-directional port 4.

**Figure 4-5. Bi-directional Port 4**




### 4.2.3.1 Port 4 Data Register (P4DAT)

Primary register address: ’4’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **P4DAT** | **P4DAT3** | **P4DAT2** | **P4DAT1** | **P4DAT0** | Reset value: 1111b |

### 4.2.3.2 Port 4 Control Register (P4CR) Byte Write

Auxiliary register address: ’4’hex

| | | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|---|
| **P4CR** | **First write cycle** | **P41M2** | **P41M1** | **P40M2** | **P40M1** | **Reset value: 1111b** |
| | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | |
| | **Second write cycle** | **P43M2** | **P43M1** | **P42M2** | **P42M1** | **Reset value: 1111b** |

P4xM2, P4xM1 – Port 4x Interrupt Mode/Direction Code

**Table 4-4. Port 4 Control Register**

| Auxiliary Address: ’4’hex First Write Cycle | | Second Write Cycle | |
|---|---|---|---|
| **Code 3 2 1 0** | **Function** | **Code 3 2 1 0** | **Function** |
| x x 1 1 | BP40 in input mode | x x 1 1 | BP42 in input mode |
| x x 1 0 | BP40 in output mode | x x 1 0 | BP42 in output mode |
| x x 0 1 | BP40 enable alternate function (SC for SSI) | x x 0 x | BP42 enable alternate function (T2O for Timer 2) |
| x x 0 0 | BP40 enable alternate function (falling edge interrupt input for INT3) | 1 1 x x | BP43 in input mode |
| 1 1 x x | BP41 in input mode | 1 0 x x | BP43 in output mode |
| 1 0 x x | BP41 in output mode | 0 1 x x | BP43 enable alternate function (SD for SSI) |
| 0 1 x x | BP41 enable alternate function (VMI for voltage monitor input) | 0 0 x x | BP43 enable alternate function (falling edge interrupt input for INT3) |
| 0 0 x x | BP41 enable alternate function (T2I external clock input for Timer 2) | – | – |

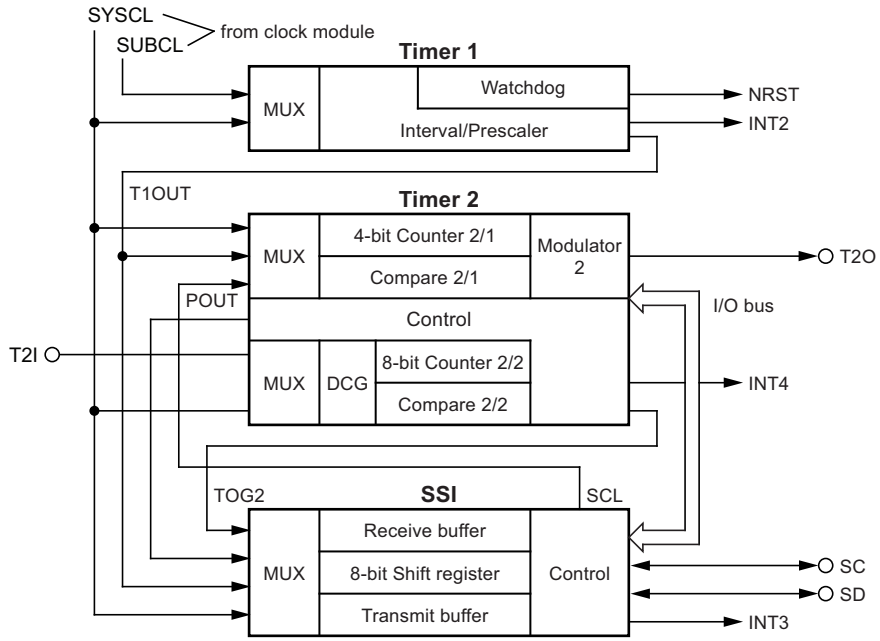
### 4.2.4 Universal Timer/Counter/ Communication Module (UTCM)

The universal timer/counter/communication module (UTCM) consists of timer 1, timer 2 and a synchronous serial interface (SSI).

- Timer 1 is an interval timer that can be used to generate periodical interrupts and as prescaler for timer 2, the serial interface and the watchdog function.
- Timer 2 is an 8/12-bit timer with an external clock input (T2I) and an output (T2O).
- The SSI operates as a two-wire serial interface or as a shift register for modulation. The modulator units work together with the timers and shift the data bits out of the shift register.

There is a multitude of modes in which the timers and the serial interface can work together.

**Figure 4-6. UTCM Block Diagram**




### 4.2.5 Timer 1

Timer 1 is an interval timer which can be used to generate periodic interrupts and as a prescaler for timer 2, timer 3, the serial interface and the watchdog function.

Timer 1 consists of a programmable 14-stage divider that is driven by either SUBCL or SYSCL. The timer output signal can be used as a prescaler clock or as SUBCL and as source for the timer 1 interrupt. Because of other system requirements timer 1 output T1OUT is synchronized with SYSCL. Therefore, in the power-down mode SLEEP (CPU core -> sleep and OSC-Stop -> yes) the output T1OUT is stopped (T1OUT = 0). Nevertheless, timer 1 can be active in SLEEP and generate timer 1 interrupts. The interrupt is maskable via the T1IM bit and the SUBCL can be bypassed via the T1BP bit of the T1C2 register. The time interval for the timer output can be programmed via the timer 1 control register T1C1.

This timer starts running automatically after any power-on reset! If the watchdog function is not activated, the timer can be restarted by writing into the T1C1 register with T1RM = 1.

Timer 1 can also be used as a watchdog timer to prevent a system from stalling. The watchdog timer is a 3-bit counter that is supplied by a separate output of timer 1. It generates a system reset when the 3-bit counter overflows. To avoid this, the 3-bit counter must be reset before it overflows. The application software has to accomplish this by reading the CWD register.

After power-on reset the watchdog must be activated by software in the $RESET initialization routine. There are two watchdog modes, in one mode the watchdog can be switched on and off by software, in the other mode the watchdog is active and locked. This mode can only be stopped by carrying out a system reset.

The watchdog timer operation mode and the time interval for the watchdog reset can be programmed via the watchdog control register (WDC).

**Figure 4-7. Timer 1 Module**

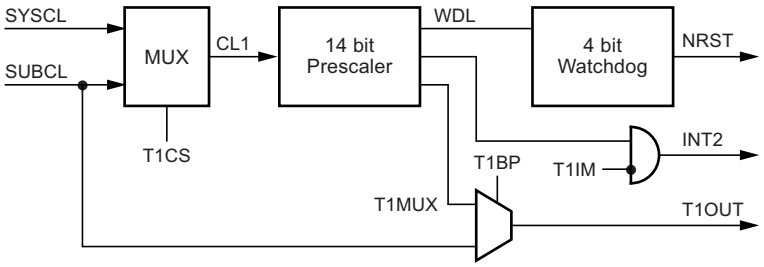
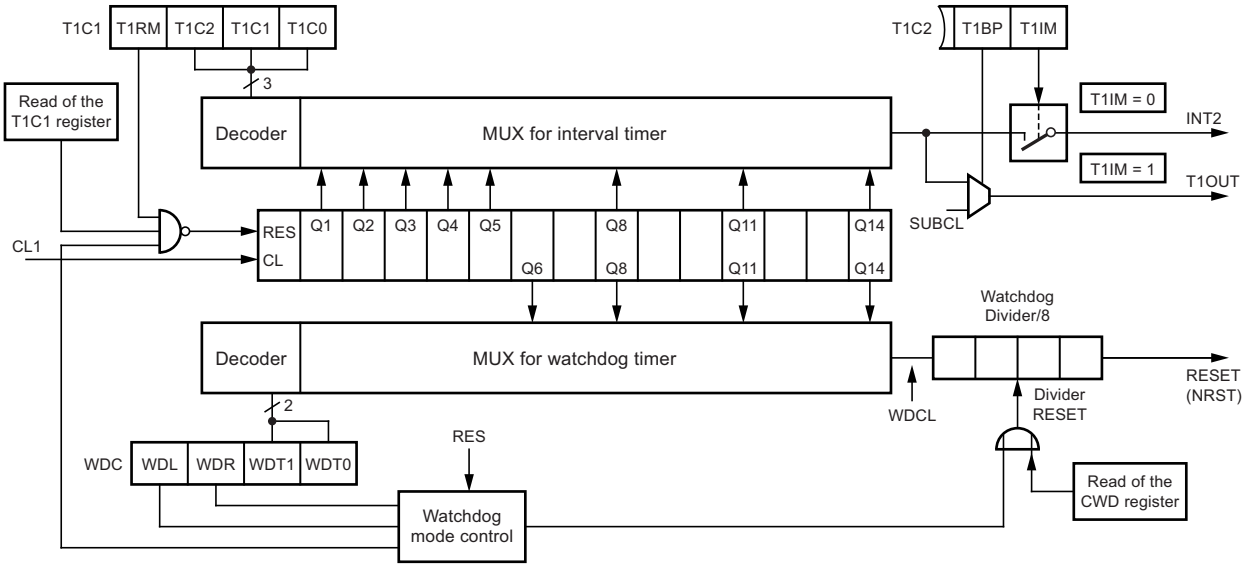



**Figure 4-8. Timer 1 and Watchdog**



### 4.2.5.1 Timer 1 Control Register 1 (T1C1)

Address: '7'hex – Subaddress: '8'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **T1C1** | **T1RM** | **T1C2** | **T1C1** | **T1C0** | **Reset value: 1111b** |

Bit 3 = MSB, Bit 0 = LSB

| | | |
|---|---|---|
| **T1RM** | **T**imer **1** **R**estart **M**ode<br>**Note:** If WDL = 0, Timer 1 restart is impossible | T1RM = 0, write access without timer 1 restart<br>T1RM = 1, write access with timer 1 restart |
| **T1C2** | **T**imer **1** **C**ontrol bit **2** | |
| **T1C1** | **T**imer **1** **C**ontrol bit **1** | |
| **T1C0** | **T**imer **1** **C**ontrol bit **0** | |

The three bits T1C[2:0] select the divider for timer 1. The resulting time interval depends on this divider and the timer 1 input clock source. The timer input can be supplied by the system clock or via clock management. If the clock management generates the SUBCL, the selected input clock from the RC oscillator or an external clock is divided by 16.

**Table 4-5. Timer 1 Control Bits**

| T1C2 | T1C1 | T1C0 | Divider | Time Interval with SUBCL from Clock Management | Time Interval with SYSCL = 2/1 MHz |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 2 | Tin × 32 | 1µs/2µs |
| 0 | 0 | 1 | 4 | Tin × 64 | 2µs/4µs |
| 0 | 1 | 0 | 8 | Tin × 128 | 4µs/8µs |
| 0 | 1 | 1 | 16 | Tin × 256 | 8µs/16µs |
| 1 | 0 | 0 | 32 | Tin × 512 | 16µs/32µs |
| 1 | 0 | 1 | 256 | Tin × 4096 | 128µs/256µs |
| 1 | 1 | 0 | 2048 | Tin × 32768 | 1024µs/2048µs |
| 1 | 1 | 1 | 16384 | Tin × 262144 | 8192µs/16384µs |

Note: Tin: input clock period = 1/Cin (see Figure 3-12 on page 14)

### 4.2.5.2 Timer 1 Control Register 2 (T1C2)

Address: '7'hex – Subaddress: '9'hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **T1C2** | **–** | **T1BP** | **T1CS** | **T1IM** | **Reset value: x111b** |

Bit 3 = MSB, Bit 0 = LSB

| | |
|---|---|
| **T1BP** | **T**imer **1** SUBCL **B**y**P**assed<br>T1BP = 1, TIOUT = T1MUX<br>T1BP = 0, T1OUT = SUBCL |
| **T1CS** | **T**imer **1** input **C**lock **S**elect<br>T1CS = 1, CL1 = SUBCL (see Figure 4-11 on page 32)<br>T1CS = 0, CL1 = SYSCL (see Figure 4-11 on page 32) |
| **T1IM** | **T**imer **1** **I**nterrupt **M**ask<br>T1IM = 1, disables timer 1 interrupt<br>T1IM = 0, enables timer 1 interrupt |

### 4.2.5.3 Watchdog Control Register (WDC)

Address: ’7’hex – Subaddress: ’A’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **WDC** | **WDL** | **WDR** | **WDT1** | **WDT0** | **Reset value: 1111b** |

Bit 3 = MSB, Bit 0 = LSB

| | |
|---|---|
| **WDL** | **W**atch**D**og **L**ock mode<br>WDL = 1, the watchdog can be enabled and disabled by using the WDR-bit<br>WDL = 0, the watchdog is enabled and locked. In this mode the WDR-bit has no effect. After the WDL-bit is cleared, the watchdog is active until a system reset or power-on reset occurs. |
| **WDR** | **W**atch**D**og **R**un and stop mode<br>WDR = 1, the watchdog is stopped/disabled<br>WDR = 0, the watchdog is active/enabled |
| **WDT1** | **W**atch**D**og **T**ime **1** |
| **WDT0** | **W**atch**D**og **T**ime **0** |

Both these bits control the time interval for the watchdog reset

**Table 4-6. Watchdog Time Control Bits**

| WDT1 | WDT0 | Divider | Delay Time to Reset with $t_{in}$ = 1/(2/1MHz) |
|---|---|---|---|
| 0 | 0 | 512 | 0.256ms/0.512ms |
| 0 | 1 | 2048 | 1.024ms/2.048ms |
| 1 | 0 | 16384 | 8.2ms/16.4ms |
| 1 | 1 | 131072 | 65.5ms/131ms |

Note: $t_{in}$: input clock period = $1/C_{in}$ (see Figure 3-12 on page 14)

### 4.2.6 Timer 2

8-/12 Bit Timer for:

- Interrupt, square-wave, pulse and duty cycle generation
- Baud rate generation for the internal shift register
- Manchester and bi-phase modulation together with the SSI
- Carrier frequency generation and modulation together with the SSI

Timer 2 can be used as an interval timer for interrupt generation, as signal generator or as baud rate generator and modulator for the serial interface. It consists of a 4-bit and an 8-bit up counter stage which both have compare registers. The 4-bit counter stages of timer 2 are cascadable as a 12-bit timer or as an 8-bit timer with a 4-bit prescaler. The timer can also be configured as an 8-bit timer and a separate 4-bit prescaler.

The timer 2 input can be supplied via the system clock, the external input clock (T2I), the Timer 1 output clock or the shift clock of the serial interface. The external input clock T2I is not synchronized with SYSCL. Therefore, it is possible to use timer 2 with a higher clock speed than SYSCL. Furthermore; with that input clock timer 2 operates in the power-down mode SLEEP (CPU core -> sleep and OSC-Stop -> yes) as well as in the POWER-DOWN (CPU core -> sleep and OSC-stop -> no). All other clock sources supplied no clock signal in SLEEP. The 4-bit counter stages of timer 2 have an additional clock output (POUT).

Its output has a modulator stage that allows the generation of pulses as well as the generation and modulation of carrier frequencies. Timer 2 output can modulate with the shift register internal data output to generate bi-phase- or manchester-code.

If the serial interface is used to modulate a bit-stream, the 4-bit stage of timer 2 has a special task. The shift register can only handle bit-stream lengths divisible by 8. For other lengths, the 4-bit counter stage can be used to stop the modulator after the right bit-count is shifted out.

If the timer is used for carrier frequency modulation, the 4-bit stage works together with an additional 2-bit duty cycle generator like a 6-bit prescaler to generate carrier frequency and duty cycle. The 8-bit counter is used to enable and disable the modulator output for a programmable count of pulses.

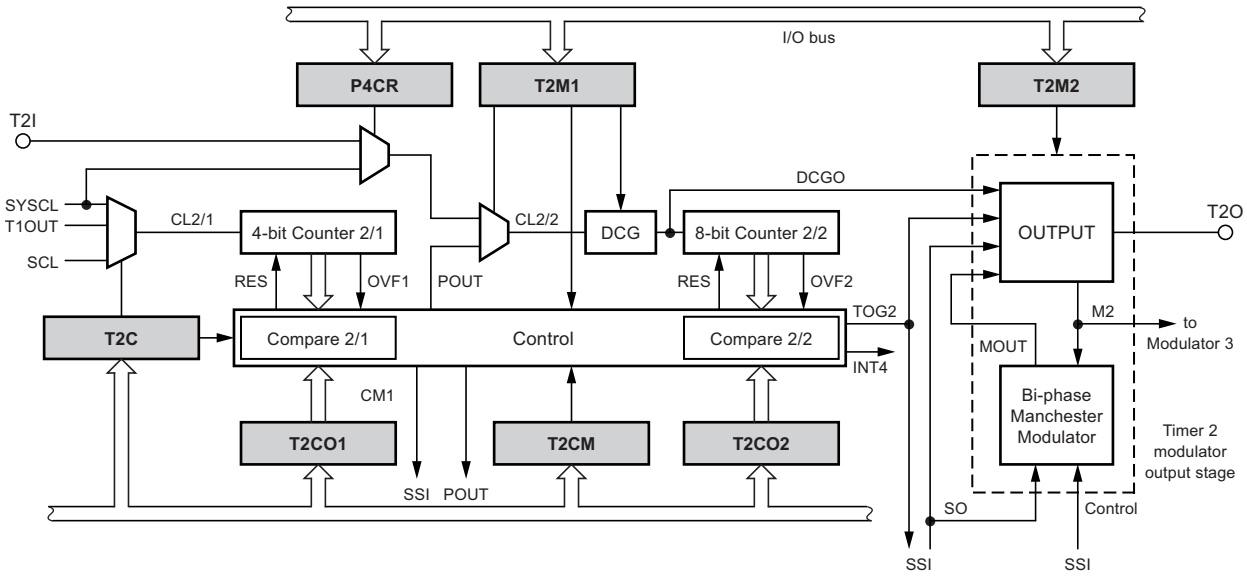
The timer has a 4-bit and an 8-bit compare register for programming the time interval, t. For programming the timer function, it has four mode and control registers. The comparator output of stage 2 is controlled by a special compare mode register (T2CM). This register contains mask bits for the actions (counter reset, output toggle, timer interrupt) which can be triggered by a compare match event or the counter overflow. This architecture enables the timer function for various modes.

Timer 2 compare data values.

Timer 2 has a 4-bit compare register (T2CO1) and an 8-bit compare register (T2CO2). Both these compare registers are cascadable as a 12-bit compare register, or 8-bit compare register and 4-bit compare register.

| | | |
|---|---|---|
| For 12-bit compare data value: | m = x +1 | $0 \le x \le 4095$ |
| For 8-bit compare data value: | n = y +1 | $0 \le y \le 255$ |
| For 4-bit compare data value: | I = z +1 | $0 \le z \le 15$ |

**Figure 4-9. Timer 2**

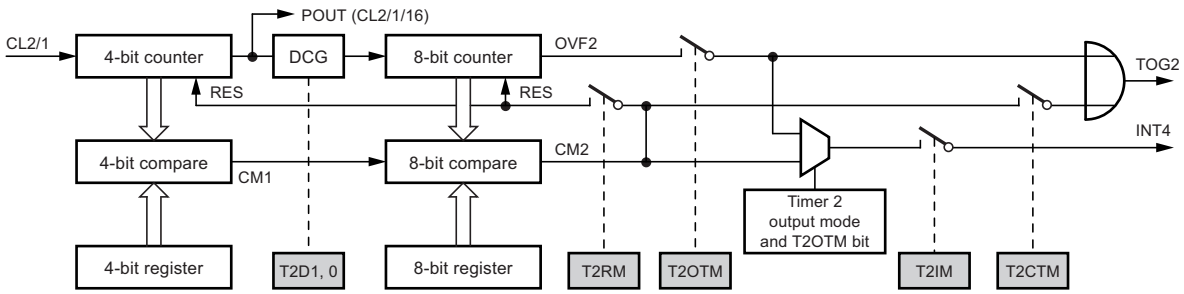



### 4.2.6.1 Timer 2 Modes

#### Mode 1: 12-bit Compare Counter

The 4-bit stage and the 8-bit stage work together as a 12-bit compare counter. A compare match signal of the 4-bit and the 8-bit stage generates the signal for the counter reset, toggle flip-flop or interrupt. The compare action is programmable via the compare mode register (T2CM). The 4-bit counter overflow (OVF1) supplies the clock output (POUT) with clocks. The duty cycle generator (DCG) has to be bypassed in this mode.

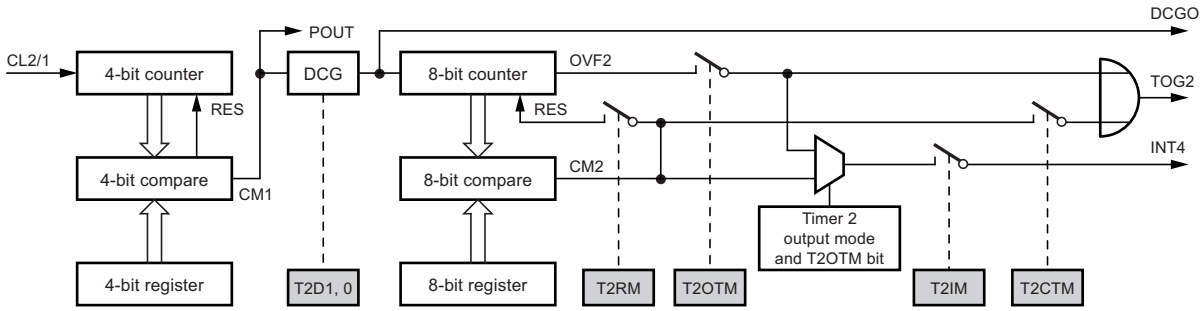
**Figure 4-10. 12-bit Compare Counter**



### Mode 2: 8-bit Compare Counter with 4-bit Programmable Prescaler

The 4-bit stage is used as a programmable prescaler for the 8-bit counter stage. In this mode, a duty cycle stage is also available. This stage can be used as an additional 2-bit prescaler or for generating duty cycles of 25%, 33% and 50%. The 4-bit compare output (CM1) supplies the clock output (POUT) with clocks.

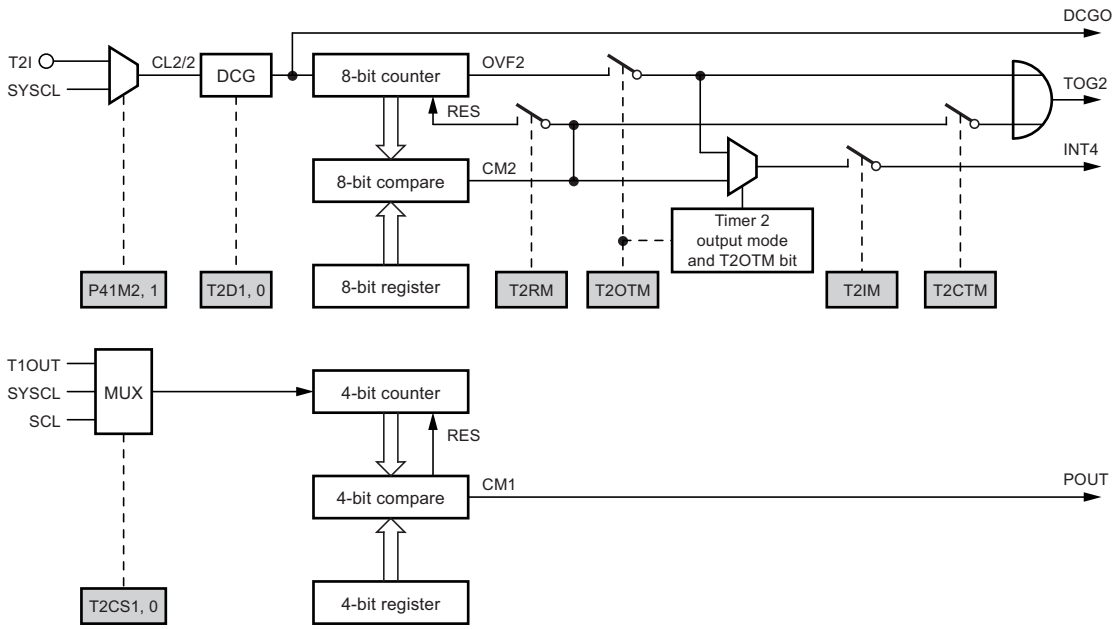
**Figure 4-11. 8-bit Compare Counter**




### Mode 3/4: 8-bit Compare Counter and 4-bit Programmable Prescaler

In these modes the 4-bit and the 8-bit counter stages work independently as a 4-bit prescaler and an 8-bit timer with a 2-bit prescaler or as a duty cycle generator. Only in mode 3 and mode 4 can the 8-bit counter be supplied via the external clock input (T2I) which is selected via the P4CR register. The 4-bit prescaler is started by activating mode 3 and stopped and reset in mode 4. Changing mode 3 and 4 has no effect for the 8-bit timer stage. The 4-bit stage can be used as a prescaler for the SSI or to generate the stop signal for modulator 2.

**Figure 4-12. 4-/8-bit Compare Counter**

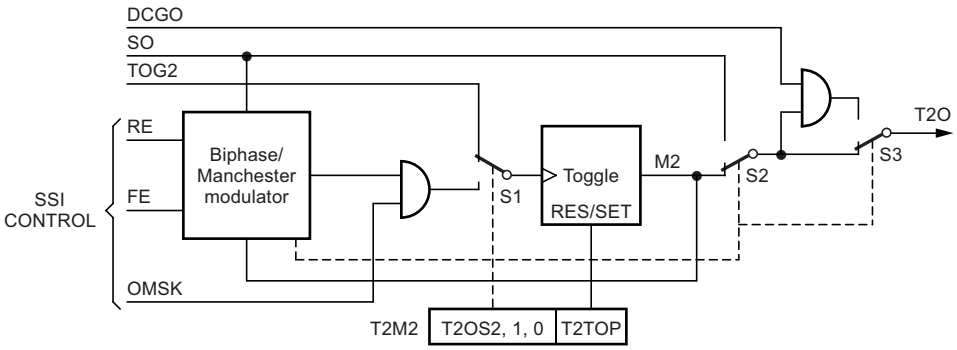


### 4.2.6.2 Timer 2 Output Modes

The signal at the timer output is generated via modulator 2. In the toggle mode, the compare match event toggles the output T2O. For high resolution duty cycle modulation 8 bits or 12 bits can be used to toggle the output. In the duty cycle burst modulator modes the DCG output is connected to T2O and switched on and off either by the toggle flipflop output or the serial data line of the SSI. Modulator 2 also has 2 modes to output the content of the serial interface as bi-phase or manchester code.

The modulator output stage can be configured by the output control bits in the T2M2 register. The modulator is started with the start of the shift register (SIR = 0) and stopped either by carrying out a shift register stop (SIR = 1) or compare match event of stage 1 (CM1) of timer 2. For this task, timer 2 mode 3 must be used and the prescaler has to be supplied with the internal shift clock (SCL).

**Figure 4-13. Timer 2 Modulator Output Stage**

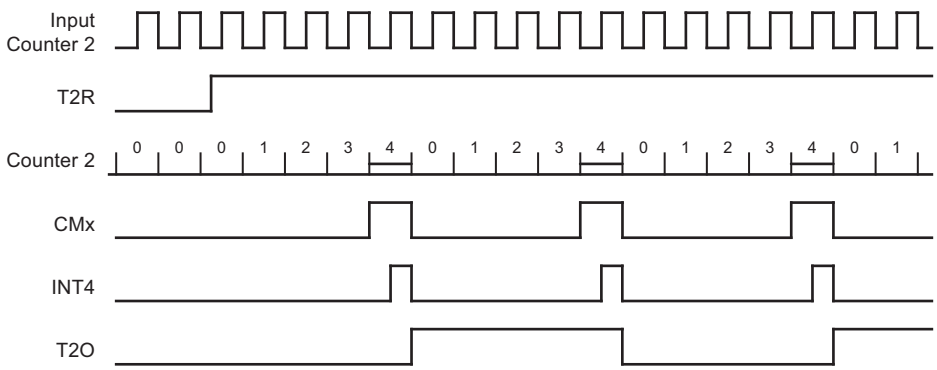



### 4.2.6.3 Timer 2 Output Signals

#### Timer 2 Output Mode 1

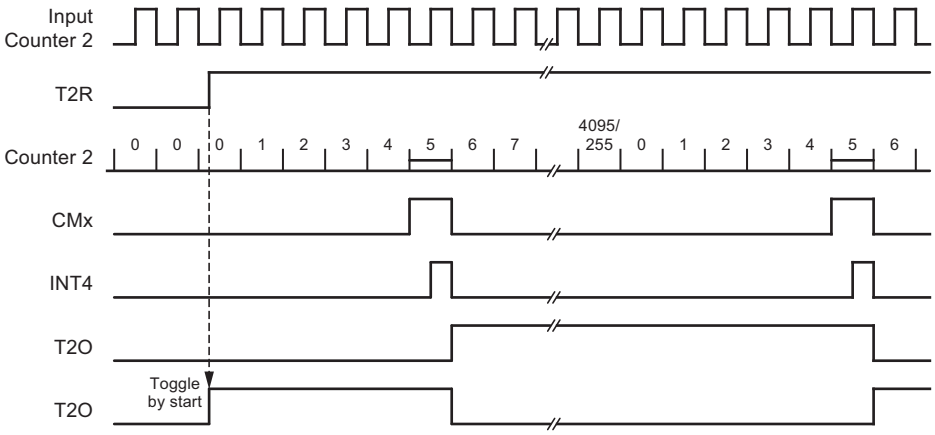
**Toggle Mode A**: A timer 2 compare match toggles the output flip-flop (M2) -> T2O

**Figure 4-14. Interrupt Timer/Square Wave Generator — Output Toggles with Each Edge Compare Match Event**



**Toggle Mode B**: A timer 2 compare match toggles the output flip-flop (M2) -> T2O

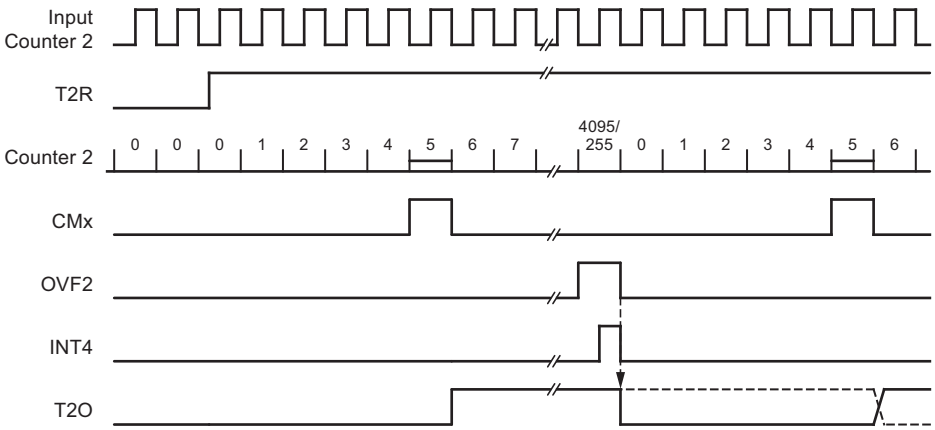
**Figure 4-15. Pulse Generator — Timer Output Toggles with Timer Start If T2TS-Bit is Set**




## Timer 2 Output Mode 1

**Toggle Mode C**: A timer 2 compare match toggles the output flip-flop (M2) -> T2O

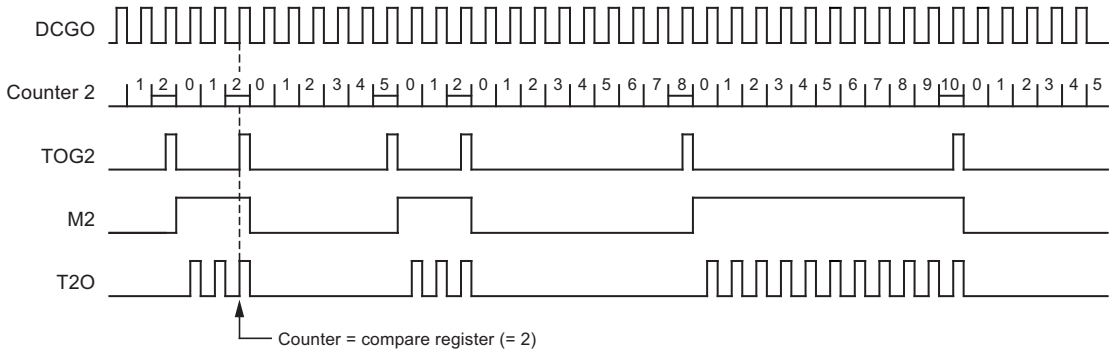
**Figure 4-16. Pulse Generator — Timer Toggles with Timer Overflow and Compare Match**



### Timer 2 Output Mode 2

Duty cycle burst generator 1: The DCG output signal (DCGO) is given to the output, and gated by the output flip-flop (M2)

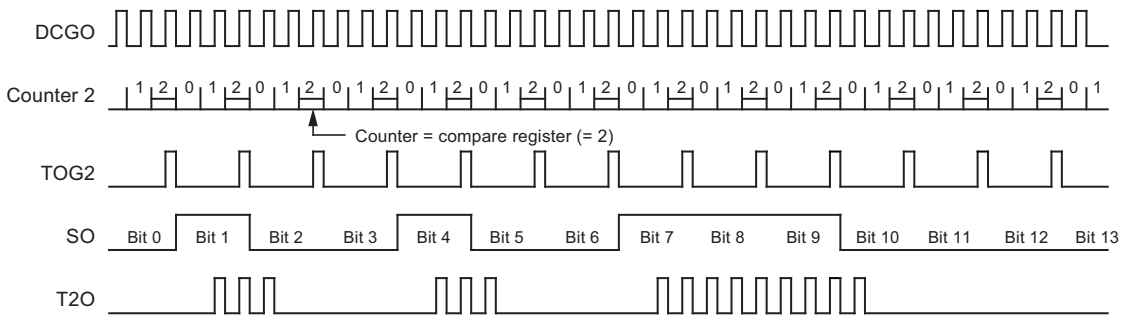
**Figure 4-17. Carrier Frequency Burst Modulation with Timer 2 Toggle Flip-Flop Output**




### Timer 2 Output Mode 3

Duty cycle burst generator 2: The DCG output signal (DCGO) is given to the output, and gated by the SSI internal data output (SO)

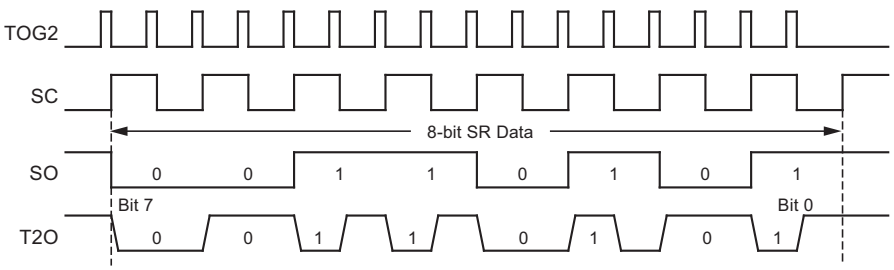
**Figure 4-18. Carrier Frequency Burst Modulation with SSI Data Output**




### Timer 2 Output Mode 4

**Bi-phase Modulator:** Timer 2 modulates the SSI internal data output (SO) to bi-phase code.

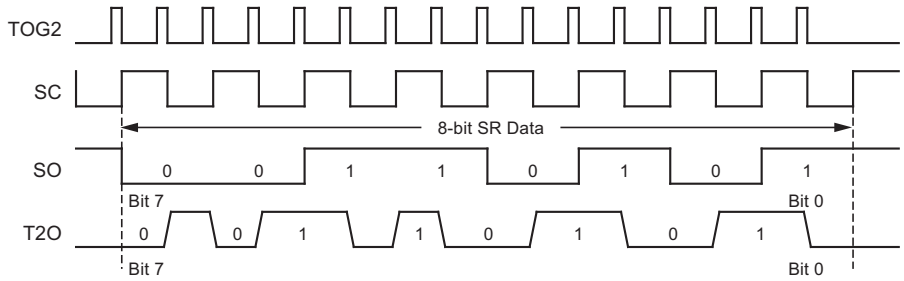
**Figure 4-19. Bi-phase Modulation**



### Timer 2 Output Mode 5

Manchester modulator: Timer 2 modulates the SSI internal data output (SO) to manchester code.

**Figure 4-20. Manchester Modulation**

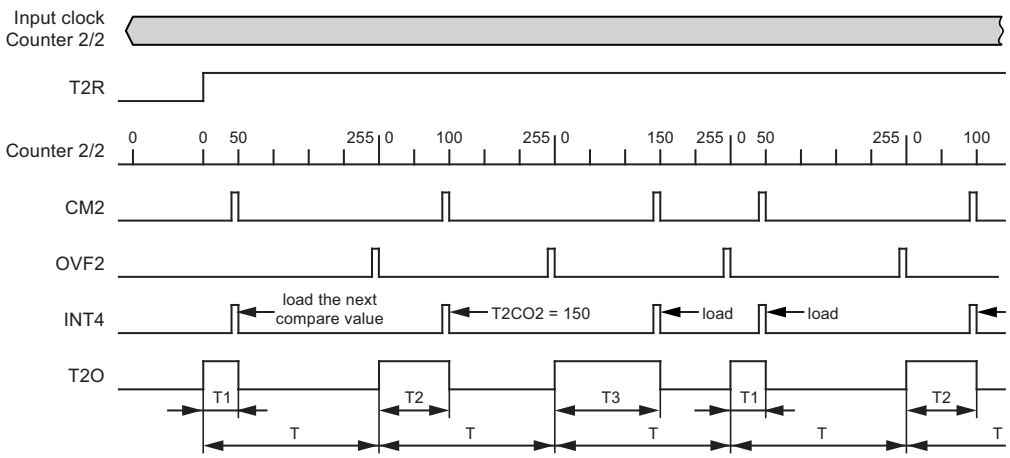



### Timer 2 Output Mode 7

**PWM Mode:** Pulse-width modulation output on timer 2 output pin (T2O)

In this mode the timer overflow defines the period and the compare register defines the duty cycle. During one period only the first compare match occurrence is used to toggle the timer output flip-flop, until overflow occur all further compare match are ignored. This avoids the situation that changing the compare register causes the occurrence of several compare match during one period. The resolution at the pulse-width modulation timer 2 mode 1 is 12-bit and all other timer 2 modes are 8-bit.

**Figure 4-21. PWM Modulation**




## 4.2.6.4 Timer 2 Registers

Timer 2 has 6 control registers to configure the timer mode, the time interval, the input clock and its output function. All registers are indirectly addressed using extended addressing as described in Section 4.1 “Addressing Peripherals” on page 19. The alternate functions of the ports BP41 or BP42 must be selected with the port 4 control register P4CR, if one of the timer 2 modes require an input at T2I/BP41 or an output at T2O/BP42.

### 4.2.6.5 Timer 2 Control Register (T2C)

Address: ’7’hex – Subaddress: ’0’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **T2C** | **T2CS1** | **T2CS0** | **T2TS** | **T2R** | **Reset value: 0000b** |

T2CS1 **T**imer **2** **C**lock **S**elect bit **1**

T2CS0 **T**imer **2** **C**lock **S**elect bit **0**

**Table 4-7. Timer 2 Clock Select Bits**

| T2CS1 | T2CS0 | Input Clock (CL 2/1) of Counter Stage 2/1 |
|---|---|---|
| 0 | 0 | System clock (SYSCL) |
| 0 | 1 | Output signal of timer 1 (T1OUT) |
| 1 | 0 | Internal shift clock of SSI (SCL) |
| 1 | 1 | Reserved |

**T2TS** **T**imer **2** **T**oggle with **S**tart
T2TS = 0, the output flip–flop of timer 2 is not toggled with the timer start
T2TS = 1, the output flip–flop of timer 2 is toggled when the timer is started with T2R

**T2R** **T**imer **2** **R**un
T2R = 0, timer 2 stop and reset
T2R = 1, timer 2 run

### 4.2.6.6 Timer 2 Mode Register 1 (T2M1)

Address: ’7’hex – Subaddress: ’1’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **T2M1** | **T2D1** | **T2D0** | **T2MS1** | **T2MS0** | **Reset value: 1111b** |

**T2D1** **T**imer **2** **D**uty cycle bit **1**

**T2D0** **T**imer **2** **D**uty cycle bit **0**

**Table 4-8. Timer 2 Duty Cycle Bits**

| T2D1 | T2D0 | Function of Duty Cycle Generator (DCG) | Additional Divider Effect |
|---|---|---|---|
| 1 | 1 | Bypassed (DCGO0) | /1 |
| 1 | 0 | Duty cycle 1/1 (DCGO1) | /2 |
| 0 | 1 | Duty cycle 1/2 (DCGO2) | /3 |

T2MS1 **T**imer **2** **M**ode **S**elect bit **1**

T2MS0 **T**imer **2** **M**ode **S**elect bit **0**

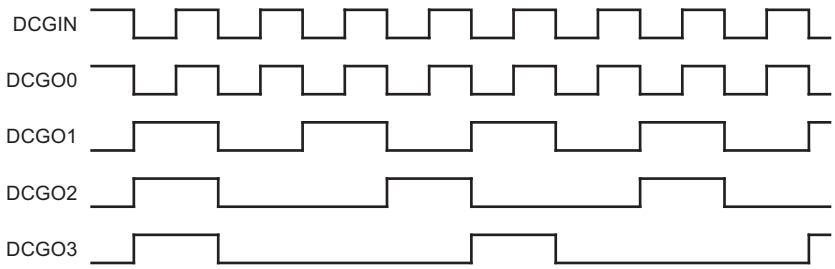
**Table 4-9. Timer 2 Mode Select Bits**

| Mode | T2MS1 | T2MS0 | Clock Output (POUT) | Timer 2 Modes |
|---|---|---|---|---|
| 1 | 1 | 1 | 4-bit counter overflow (OVF1) | 12-bit compare counter, the DCG have to be bypassed in this mode |
| 2 | 1 | 0 | 4-bit compare output (CM1) | 8-bit compare counter with 4-bit programmable prescaler and duty cycle generator |
| 3 | 0 | 1 | 4-bit compare output (CM1) | 8-bit compare counter clocked by SYSCL or the external clock input T2I, 4-bit prescaler run, the counter 2/1 starts after writing mode 3 |
| 4 | 0 | 0 | 4-bit compare output (CM1) | 8-bit compare counter clocked by SYSCL or the external clock input T2I, 4-bit prescaler stop and resets |

### 4.2.6.7 Duty Cycle Generator

The duty cycle generator generates duty cycles from 25%, 33% or 50%. The frequency at the duty cycle generator output depends on the duty cycle and the Timer 2 prescaler setting. The DCG-stage can also be used as an additional programmable prescaler for timer 2.

**Figure 4-22. DCG Output Signals**



### 4.2.6.8 Timer 2 Mode Register 2 (T2M2)

Address: ’7’hex – Subaddress: ’2’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **T2M2** | **T2TOP** | **T2OS2** | **T2OS1** | **T2OS0** | **Reset value: 1111b** |

| | |
|---|---|
| **T2TOP** | **T**imer **2** **T**oggle **O**utput **P**reset<br>This bit allows the programmer to preset the timer 2 output T2O.<br>T2TOP = 0, resets the toggle outputs with the write cycle (M2 = 0)<br>T2TOP = 1, sets toggle outputs with the write cycle (M2 = 1)<br>**Note: If T2R = 1, no output preset is possible** |
| **T2OS2** | **T**imer **2** **O**utput **S**elect bit **2** |
| **T2OS1** | **T**imer **2** **O**utput **S**elect bit **1** |
| **T2OS0** | **T**imer **2** **O**utput **S**elect bit **0** |

**Table 4-10. Timer 2 Output Select Bits**

| Output Mode | T2OS2 | T2MS1 | T2MS0 | Clock Output (POUT) |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | Toggle mode: A timer 2 compare match toggles the output flip-flop (M2) -> T2O |
| 2 | 1 | 1 | 0 | Duty cycle burst generator 1: the DCG output signal (DCG0) is given to the output and gated by the output flip-flop (M2) |
| 3 | 1 | 0 | 1 | Duty cycle burst generator 2: the DCG output signal (DCGO) is given to the output and gated by the SSI internal data output (SO) |
| 4 | 1 | 0 | 0 | Bi-phase modulator: Timer 2 modulates the SSI internal data output (SO) to Bi-phase code |
| 5 | 0 | 1 | 1 | Manchester modulator: Timer 2 modulates the SSI internal data output (SO) to manchester code |
| 6 | 0 | 1 | 0 | SSI output: T2O is used directly as SSI internal data output (SO) |
| 7 | 0 | 0 | 1 | PWM mode: an 8/12-bit PWM mode |
| 8 | 0 | 0 | 0 | Not allowed |

Note: If one of these output modes is used the T2O alternate function of Port 4 must also be activated.

### 4.2.6.9 Timer 2 Compare and Compare Mode Registers

Timer 2 has two separate compare registers, T2CO1 for the 4-bit stage and T2CO2 for the 8-bit stage of timer 2. The timer compares the contents of the compare register current counter value and if it matches it generates an output signal. Depending on the timer mode, this signal is used to generate a timer interrupt, to toggle the output flip-flop as SSI clock or as a clock for the next counter stage.

In the 12-bit timer mode, T2CO1 contains bits 0 to 3 and T2CO2 bits 4 to 11 of the 12-bit compare value. In all other modes, the two compare registers work independently as a 4-bit and 8-bit compare register. When assigned to the compare register a compare event will be suppressed.

### 4.2.6.10 Timer 2 Compare Mode Register (T2CM)

Address: ’7’hex – Subaddress: ’3’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **T2CM** | **T2OTM** | **T2CTM** | **T2RM** | **T2IM** | **Reset value: 0000b** |

**T2OTM** — **T**imer **2** **O**verflow **T**oggle **M**ask bit
T2OTM = 0, disable overflow toggle
T2OTM = 1, enable overflow toggle, a counter overflow (OVF2) toggles the output flip-flop (TOG2). If the T2OTM-bit is set, only a counter overflow can generate an interrupt except on the timer 2 output mode 7.

**T2CTM** — **T**imer **2** **C**ompare **T**oggle **M**ask bit
T2CTM = 0, disable compare toggle
T2CTM = 1, enable compare toggle, a match of the counter with the compare register toggles output flip-flop (TOG2). In timer 2 output mode 7 and when the T2CTM-bit is set, only a match of the counter with the compare register can generate an interrupt.

**T2RM** — **T**imer **2** **R**eset **M**ask bit
T2RM = 0, disable counter reset
T2RM = 1, enable counter reset, a match of the counter with the compare register resets the counter

**T2IM** — **T**imer **2** **I**nterrupt **M**ask bit
T2IM = 0, disable Timer 2 interrupt
T2IM = 1, enable Timer 2 interrupt

**Table 4-11. Timer 2 Toggle Mask Bits**

| Timer 2 Output Mode | T2OTM | T2CTM | Timer 2 Interrupt Source |
|---|---|---|---|
| 1, 2, 3, 4, 5 and 6 | 0 | x | Compare match (CM2) |
| 1, 2, 3, 4, 5 and 6 | 1 | x | Overflow (OVF2) |
| 7 | x | 1 | Compare match (CM2) |

### 4.2.6.11 Timer 2 COmpare Register 1 (T2CO1)

Address: ’7’hex – Subaddress: ’4’hex

| | | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|---|
| **T2CO1** | **Write cycle** | | | | | **Reset value: 1111b** |

In prescaler mode the clock is bypassed if the compare register T2CO1 contains 0.

### 4.2.6.12 Timer 2 COmpare Register 2 (T2CO2) Byte Write

Address: ’7’hex – Subaddress: ’5’hex

| | | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|---|
| **T2CO2** | **First write cycle** | | | | | **Reset value: 1111b** |
| | **Second write cycle** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Reset value: 1111b** |

### 4.2.7 Synchronous Serial Interface (SSI)

#### 4.2.7.1 SSI Features

- 2- and 3-wire NRZ
- 2-wire mode

With timer 2:

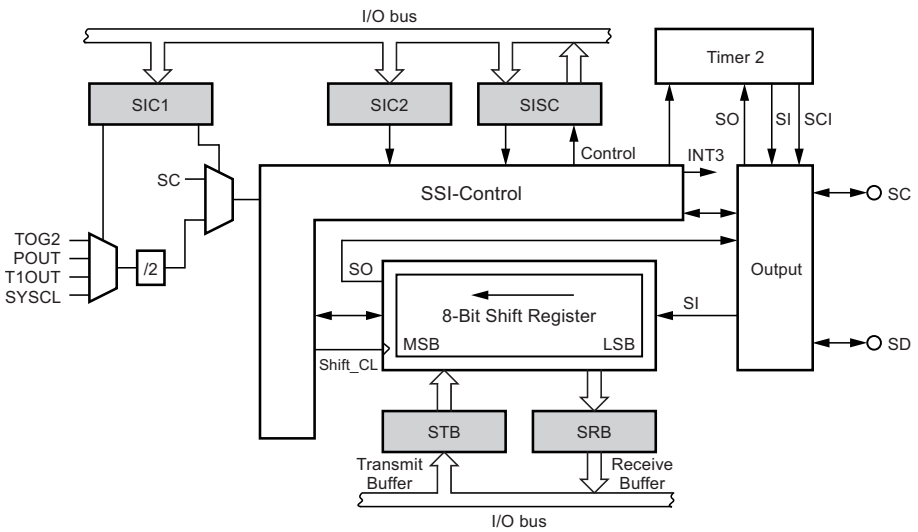
- Bi-phase modulation
- Manchester modulation
- Pulse-width demodulation
- Burst modulation

#### 4.2.7.2 SSI Peripheral Configuration

The synchronous serial interface (SSI) can be used either for serial communication with external devices such as EEPROMs, shift registers, display drivers, other microcontrollers, or as a means for generating and capturing on-chip serial streams of data. External data communication takes place via port 4's (BP4) multi-functional port which can be software configured by writing the appropriate control word into the P4CR register. The SSI can be configured in any one of the following ways:

1. 2-wire external interface for bi-directional data communication with one data terminal and one shift clock. The SSI uses Port BP43 as a bi-directional serial data line (SD) and BP40 as a shift clock line (SC).
2. 3-wire external interface for simultaneous input and output of serial data, with a serial input data terminal (SI), a serial output data terminal (SO) and a shift clock (SC). The SSI uses BP40 as a shift clock (SC), while the serial data input (SI) is applied to BP43 (configured in P4CR as input). Serial output data (SO) in this case is passed through to BP42 (configured in P4CR to T2O) via timer 2 output stage (T2M2 configured in mode 6).
3. Timer/SSI combined modes – the SSI used together with timer 2 is capable of performing a variety of data modulation and functions (see Section 4.2.5 "Timer 1" on page 26). The modulating data is converted by the SSI into a continuous serial stream of data which is in turn modulated in one of the timer functional blocks.

**Figure 4-23. Block Diagram of the Synchronous Serial Interface**



#### 4.2.7.3 General SSI Operation

The SSI is comprised essentially of an 8–bit shift register with two associated 8-bit buffers - the receive buffer (SRB) for capturing the incoming serial data and a transmit buffer (STB) for intermediate storage of data to be serially output. Both buffers are directly accessible by software. Transferring the parallel buffer data into and out of the shift register is controlled automatically by the SSI control, so that both single byte transfers or continuous bit streams can be supported.

The SSI can generate the shift clock (SC) either from one of several on-chip clock sources or accept an external clock. The external shift clock is output on, or applied to the port BP40. Selection of an external clock source is performed by the serial clock direction control bit (SCD). In the combinational modes, the required clock is selected by the corresponding timer mode.

The SSI can operate in three data transfer modes – synchronous 8-bit shift mode, a 9-bit multi-chip link mode (MCL), containing 8-bit data and 1-bit acknowledge, and a corresponding 8-bit MCL mode without acknowledge. In both MCL modes the data transmission begins after a valid start condition and ends with a valid stop condition.

External SSI clocking is not supported in these modes. The SSI should thus generate and have full control over the shift clock so that it can always be regarded as an MCL-bus master device.

All directional control of the external data port used by the SSI is handled automatically and is dependent on the transmission direction set by the serial data direction (SDD) control bit. This control bit defines whether the SSI is currently operating in transmit (TX) mode or receive (RX) mode.

Serial data is organized in 8-bit telegrams which are shifted with the most significant bit first. In the 9-bit MCL mode, an additional acknowledge bit is appended to the end of the telegram for handshaking purposes (see “MCL protocol”).

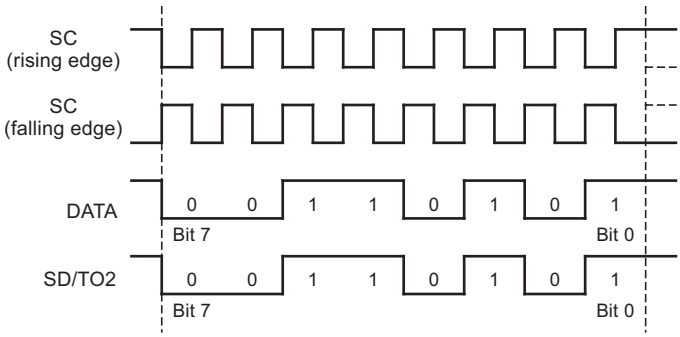
At the beginning of every telegram, the SSI control loads the transmit buffer into the shift register and proceeds immediately to shift data serially out. At the same time, incoming data is shifted into the shift register input. This incoming data is automatically loaded into the receive buffer when the complete telegram has been received. Data can, if required thus be simultaneously received and transmitted.

Before data can be transferred, the SSI must first be activated. This is performed by means of the SSI reset control (SIR) bit. All further operation then depends on the data directional mode (TX/RX) and the present status of the SSI buffer registers shown by the serial interface ready status flag (SRDY). This SRDY flag indicates the (empty/full) status of either the transmit buffer (in TX mode), or the receive buffer (in RX mode). The control logic ensures that data shifting is temporarily halted at any time, if the appropriate receive/transmit buffer is not ready (SRDY = 0). The SRDY status will then automatically be set back to “1” and data shifting resumed as soon as the application software loads the new data into the transmit register (in TX mode) or frees the shift register by reading it into the receive buffer (in RX mode).

A further activity status (ACT) bit indicates the present status of serial communication. The ACT bit remains high for the duration of the serial telegram or if MCL stop or start conditions are currently being generated. Both the current SRDY and ACT status can be read in the SSI status register. To deactivate the SSI, the SIR bit must be set high.

### 4.2.7.4 8-bit Synchronous Mode

**Figure 4-24. 8-bit Synchronous Mode**




In the 8-bit synchronous mode, the SSI can operate as either a 2- or 3-wire interface (see Section 4.2.7.2 “SSI Peripheral Configuration” on page 41). The serial data (SD) is received or transmitted in NRZ format, synchronized to either the rising or falling edge of the shift clock (SC). The choice of clock edge is defined by the serial mode control bits (SM0, SM1). It should be noted that the transmission edge refers to the SC clock edge with which the SD changes. To avoid clock skew problems, the incoming serial input data is shifted in with the opposite edge.

When used together with one of the timer modulator or demodulator stages, the SSI must be set in the 8-bit synchronous mode 1.

In RX mode, as soon as the SSI is activated (SIR = 0), 8 shift clocks are generated and the incoming serial data is shifted into the shift register. This first telegram is automatically transferred into the receive buffer and the SRDY flag is set to 0 indicating that the receive buffer contains valid data. At the same time an interrupt (if enabled) is generated. The SSI then continues shifting in the following 8-bit telegram. If, during this time the first telegram has been read by the controller, the second telegram will also be transferred in the same way into the receive buffer and the SSI will continue clocking in the next telegram. Should, however, the first telegram not have been read (SRDY = 1), then the SSI will stop, temporarily holding the second telegram in the shift register until a certain point in time when the controller is able to service the receive buffer. In this way no data is lost or overwritten.

Deactivating the SSI (SIR = 1) in mid-telegram will immediately stop the shift clock and latch the present contents of the shift register into the receive buffer. This can be used for clocking in a data telegram of less than 8 bits in length. Care should be taken to read out the final complete 8-bit data telegram of a multiple word message before deactivating the SSI (SIR = 1) and terminating the reception. After termination, the shift register contents will overwrite the receive buffer.

**Figure 4-25. Example of 8-bit Synchronous Transmit Operation**

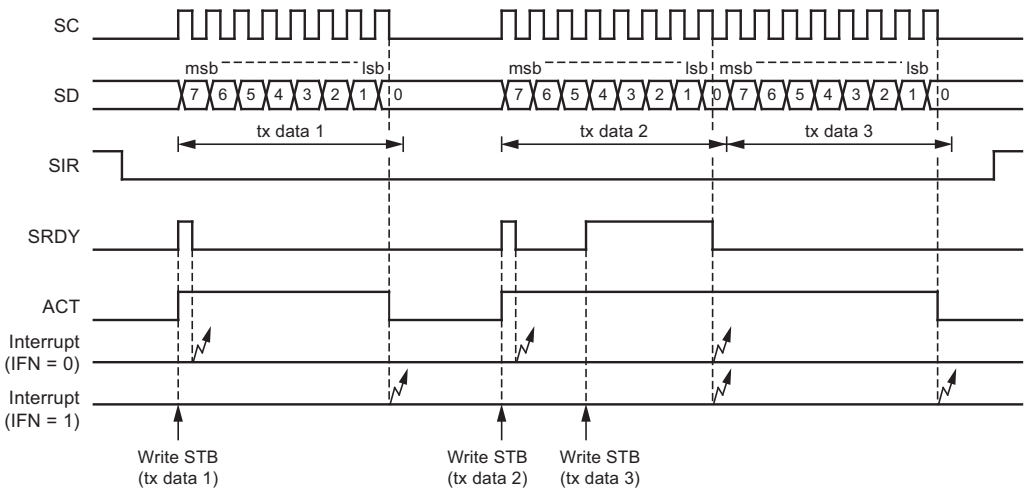
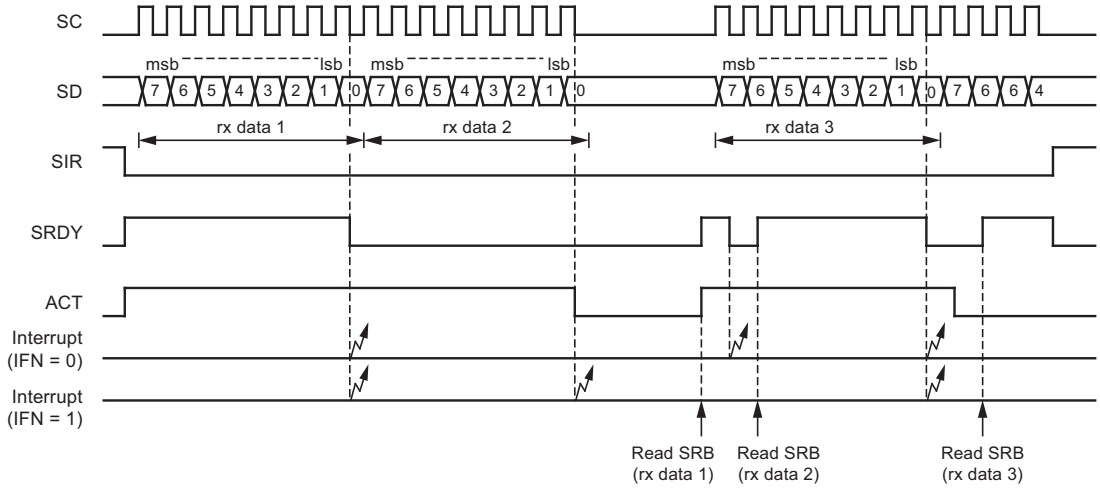


**Figure 4-26. Example of 8-bit Synchronous Receive Operation**




### 4.2.7.5 9-bit Shift Mode

In the 9-bit shift mode, the SSI is able to handle the MCL protocol (described below). It always operates as an MCL master device, i.e., SC is always generated and output by the SSI. Both the MCL start and stop conditions are automatically generated whenever the SSI is activated or deactivated by the SIR-bit. In accordance with the MCL protocol, the output data is always changed in the clock low phase and shifted in on the high phase.

Before activating the SSI (SIR = 0) and commencing an MCL dialog, the appropriate data direction for the first word must be set using the SDD control bit. The state of this bit controls the direction of the data port (BP43 or MCL_SD). Once started, the 8 data bits are, depending on the selected direction, either clocked into or out of the shift register. During the 9th clock period, the port direction is automatically switched over so that the corresponding acknowledge bit can be shifted out or read in. In transmit mode, the acknowledge bit received from the device is captured in the SSI status register (TACK) where it can be read by the controller. A receive mode, the state of the acknowledge bit to be returned to the device is predetermined by the SSI status register (RACK).

Changing the directional mode (TX/RX) should not be performed during the transfer of an MCL telegram. One should wait until the end of the telegram which can be detected using the SSI interrupt (IFN = 1) or by interrogating the ACT status.

A 9-bit telegram, once started will always run to completion and will not be prematurely terminated by the SIR bit. So, if the SIR-bit is set to “1” in with telegram, the SSI will complete the current transfer and terminate the dialog with an MCL stop condition.

**Figure 4-27. Example of MCL Transmit Dialog**

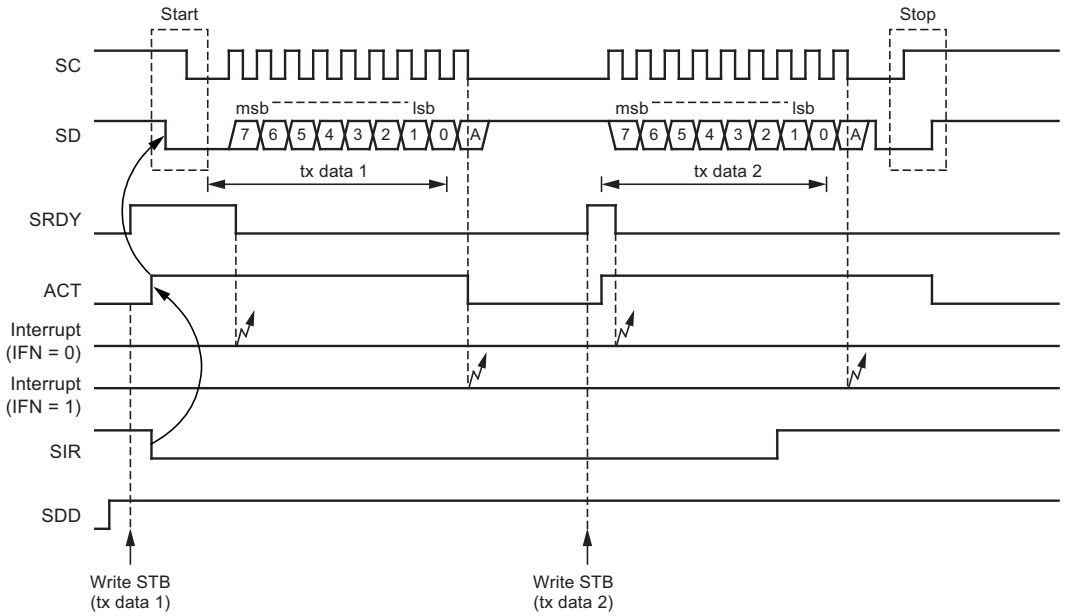
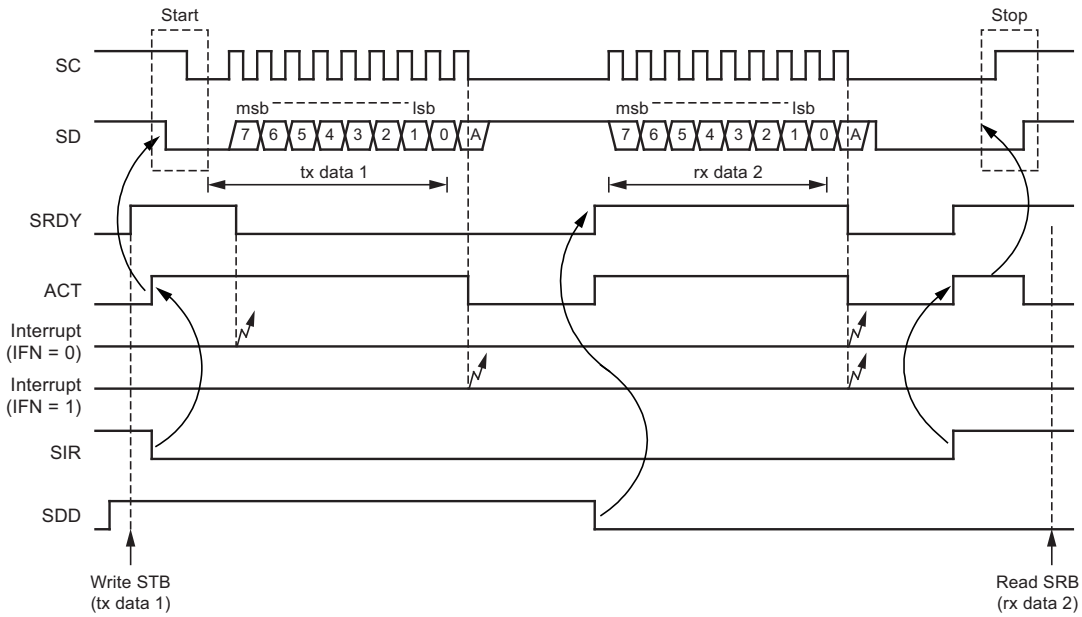



**Figure 4-28. Example of MCL Receive Dialog**



### 4.2.7.6 8-bit Pseudo MCL Mode

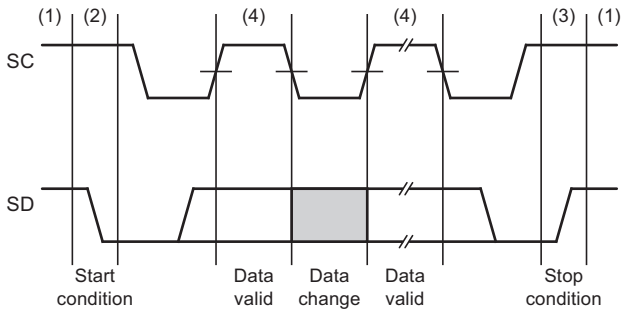
In this mode, the SSI exhibits all the typical MCL operational features except for the acknowledge-bit which is never expected or transmitted.

### 4.2.7.7 MCL Bus Protocol

The MCL protocol constitutes a simple 2-wire bi-directional communication highway via which devices can communicate control and data information. Although the MCL protocol can support multi-master bus configurations, the SSI, in MCL mode is intended for use purely as a master controller on a single master bus system. So all reference to multiple bus control and bus contention will be omitted at this point.

All data is packaged into 8-bit telegrams plus a trailing handshaking or acknowledge-bit. Normally the communication channel is opened with a so-called start condition, which initializes all devices connected to the bus. This is then followed by a data telegram, transmitted by the master controller device. This telegram usually contains an 8-bit address code to activate a single slave device connected onto the MCL bus. Each slave receives this address and compares it with its own unique address. The addressed slave device, if ready to receive data will respond by pulling the SD line low during the 9th clock pulse. This represents a so-called MCL acknowledge. The controller on detecting this affirmative acknowledge then opens a connection to the required slave. Data can then be passed back and forth by the master controller, each 8-bit telegram being acknowledged by the respective recipient. The communication is finally closed by the master device and the slave device put back into standby by applying a stop condition onto the bus.

**Figure 4-29. MCL Bus Protocol 1**




**Bus not busy (1)** Both data and clock lines remain HIGH.

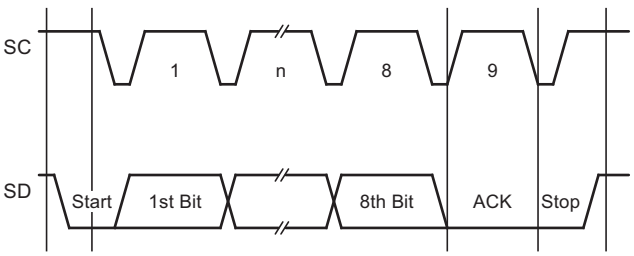
**Start data transfer (2)** A HIGH to LOW transition of the SD line while the clock (SC) is HIGH defines a START condition.

**Stop data transfer (3)** A LOW to HIGH transition of the SD line while the clock (SC) is HIGH defines a STOP condition.

**Data valid (4)** The state of the data line represents valid data when, after START condition, the data line is stable for the duration of the HIGH period of the clock signal.

**Acknowledge** All address and data words are serially transmitted to and from the device in eight-bit words. The receiving device returns a zero on the data line during the ninth clock cycle to acknowledge word receipt.

**Figure 4-30. MCL Bus Protocol 2**



### 4.2.7.8 SSI Interrupt

The SSI interrupt INT3 can be generated either by an SSI buffer register status (i.e., transmit buffer empty or receive buffer full) at the end of an SSI data telegram or on the falling edge of the SC/SD pins on Port 4 (see P4CR). SSI interrupt selection is performed by the interrupt function control bit (IFN). The SSI interrupt is usually used to synchronize the software control of the SSI and inform the controller of the present SSI status. Port 4 interrupts can be used together with the SSI or, if the SSI itself is not required, as additional external interrupt sources. In either case this interrupt is capable of waking the controller out of sleep mode.

To enable and select the SSI relevant interrupts use the SSI interrupt mask (SIM) and the interrupt function (IFN) while port 4 interrupts are enabled by setting appropriate control bits in P4CR register.

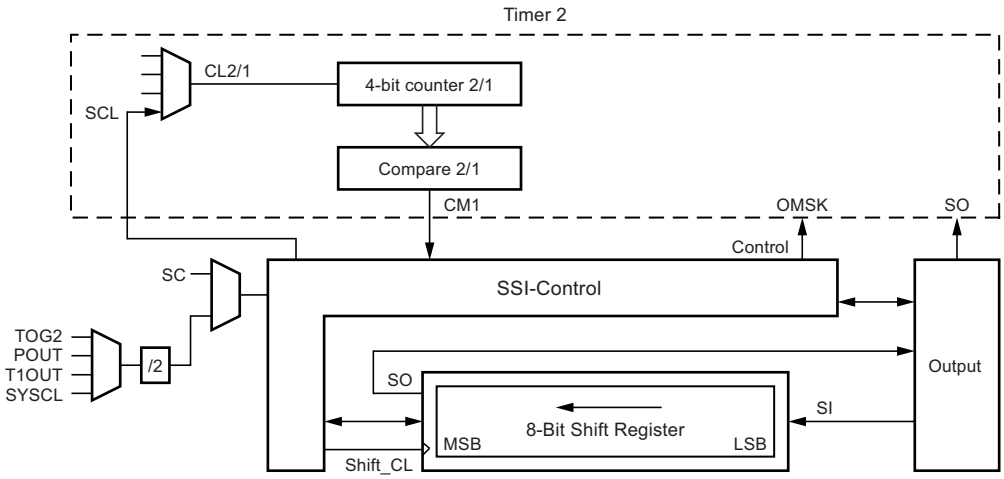
### 4.2.7.9 Modulation

If the shift register is used together with timer 2 for modulation purposes, the 8-bit synchronous mode must be used. In this case, the unused port 4 pins can be used as conventional bi-directional ports.

The modulation stage, if enabled, operates as soon as the SSI is activated (SIR = 0) and ceases when deactivated (SIR = 1).

Due to the byte-orientated data control, the SSI (when running normally) generates serial bit-streams which are submultiples of 8 bits. However, an SSI output masking (OMSK) function permits, however, the generation of bit-streams of any length. The OMSK signal is derived indirectly from the 4-bit prescaler of the timer 2 and masks out a programmable number of unrequired trailing data bits during the shifting out of the final data word in the bit stream. The number of non-masked data bits is defined by the value pre-programmed in the prescaler compare register. To use output masking, the modulator stop mode bit (MSM) must be set to “0” before programming the final data word into the SSI transmit buffer. This in turn, enables shift clocks to the prescaler when this final word is shifted out. On reaching the compare value, the prescaler triggers the OMSK signal and all following data bits are blanked.

**Figure 4-31. SSI Output Masking Function**



### 4.2.7.10 Serial Interface Registers

### 4.2.7.11 Serial Interface Control Register 1 (SIC1)

Auxiliary register address: ’9’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **SIC1** | **SIR** | **SCD** | **SCS1** | **SCS0** | **Reset value: 1111b** |

**SIR** **S**erial **I**nterface **R**eset
SIR = 1, SSI inactive
SIR = 0, SSI active

**SCD** **S**erial **C**lock **D**irection
SCD = 1, SC line used as output
SCD = 0, SC line used as input
**Note:** This bit has to be set to '1' during the MCL mode

**SCS1** **S**erial **C**lock source **S**elect bit **1**

**SCS0** **S**erial **C**lock source **S**elect bit **0**

Note: with SCD = '0' the bits SCS1 and SCS0 are insignificant

**Table 4-12. Serial Clock Source Select Bits**

| SCS1 | SCS0 | Internal Clock for SSI |
|---|---|---|
| 1 | 1 | SYSCL/2 |
| 1 | 0 | T1OUT/2 |
| 0 | 1 | POUT/2 |
| 0 | 0 | TOG2/2 |

- In transmit mode (SDD = 1) shifting starts only if the transmit buffer has been loaded (SRDY = 1).
- Setting SIR-bit loads the contents of the shift register into the receive buffer (synchronous 8-bit mode only).
- In MCL modes, writing a 0 to SIR generates a start condition and writing a 1 generates a stop condition.

### 4.2.7.12 Serial Interface Control Register 2 (SIC2)

Auxiliary register address: ’A’hex

| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|
| **SIC2** | **MSM** | **SM1** | **SM0** | **SDD** | **Reset value: 1111b** |

**MSM** — **M**odular **S**top **M**ode
MSM = 1, modulator stop mode disabled (output masking off)
MSM = 0, modulator stop mode enabled (output masking on) - used in modulation modes for generating bit streams which are not sub–multiples of 8 bit.

**SM1** — **S**erial **M**ode control bit **1**

**SM0** — **S**erial **M**ode control bit **0**

**Table 4-13. Serial Mode Control Bits**

| Mode | SM1 | SM0 | SSI Mode |
|---|---|---|---|
| 1 | 1 | 1 | 8-bit NRZ-Data changes with the rising edge of SC |
| 2 | 1 | 0 | 8-bit NRZ-Data changes with the falling edge of SC |
| 3 | 0 | 1 | 9-bit two-wire MCL compatible |
| 4 | 0 | 0 | 8-bit two-wire pseudo MCL compatible (no acknowledge) |

**SDD** — **S**erial **D**ata **D**irection
SDD = 1, transmit mode - SD line used as output (transmit data). SRDY is set by a transmit buffer write access
SDD = 0, receive mode - SD line used as input (receive data). SRDY is set by a receive buffer read access

SDD controls port directional control and defines the reset function for the SRDY-flag.

### 4.2.7.13 Serial Interface Status and Control Register (SISC)

Primary register address: ’A’hex

| | | Bit 3 | Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|---|---|---|
| **SISC** | **write** | | **RACK** | **SIM** | **IFN** | **Reset value: 1111b** |
| **SISC** | **read** | **–** | **TACK** | **ACT** | **SRDY** | **Reset value: xxxxb** |

| | |
|---|---|
| **RACK** | **R**eceive **ACK**nowledge status/control bit for MCL mode<br>RACK = 0, transmit acknowledge in next receive telegram<br>RACK = 1, transmit no acknowledge in last receive telegram |
| **TACK** | **T**ransmit **ACK**nowledge status/control bit for MCL mode<br>TACK = 0, acknowledge received in last transmit telegram<br>TACK = 1, no acknowledge received in last transmit telegram |
| **SIM** | **S**erial **I**nterrupt **M**ask<br>SIM = 1, disable interrupts<br>SIM = 0, enable serial interrupt. An interrupt is generated. |
| **IFN** | **I**nterrupt **F**u**N**ction<br>IFN = 1, the serial interrupt is generated at the end of the telegram<br>IFN = 0, the serial interrupt is generated when the SRDY goes low<br>(i.e., buffer becomes empty/full in transmit/receive mode) |
| **SRDY** | **S**erial interface buffer **R**ea**DY** status flag<br>SRDY = 1, in receive mode: receive buffer empty in transmit mode: transmit buffer full<br>SRDY = 0, in receive mode: receive buffer full in transmit mode: transmit buffer empty |
| **ACT** | Transmission **ACT**ive status flag<br>ACT = 1, transmission is active, i.e., serial data transfer. Stop or start conditions are currently in progress.<br>ACT = 0, transmission is inactive |

### 4.2.7.14 Serial Transmit Buffer (STB) – Byte Write

Primary register address: ’9’hex

| | | | | | | |
|---|---|---|---|---|---|---|
| **STB** | **First write cycle** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** | **Reset value: xxxxb** |
| | **Second write cycle** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Reset value: xxxxb** |

The STB is the transmit buffer of the SSI. The SSI transfers the transmit buffer into the shift register and starts shifting with the most significant bit.

### 4.2.7.15 Serial Receive Buffer (SRB) – Byte Read

Primary register address: ’9’hex

| | | | | | | |
|---|---|---|---|---|---|---|
| **SRB** | **First read cycle** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Reset value: xxxxb** |
| | **Second read cycle** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** | **Reset value: xxxxb** |

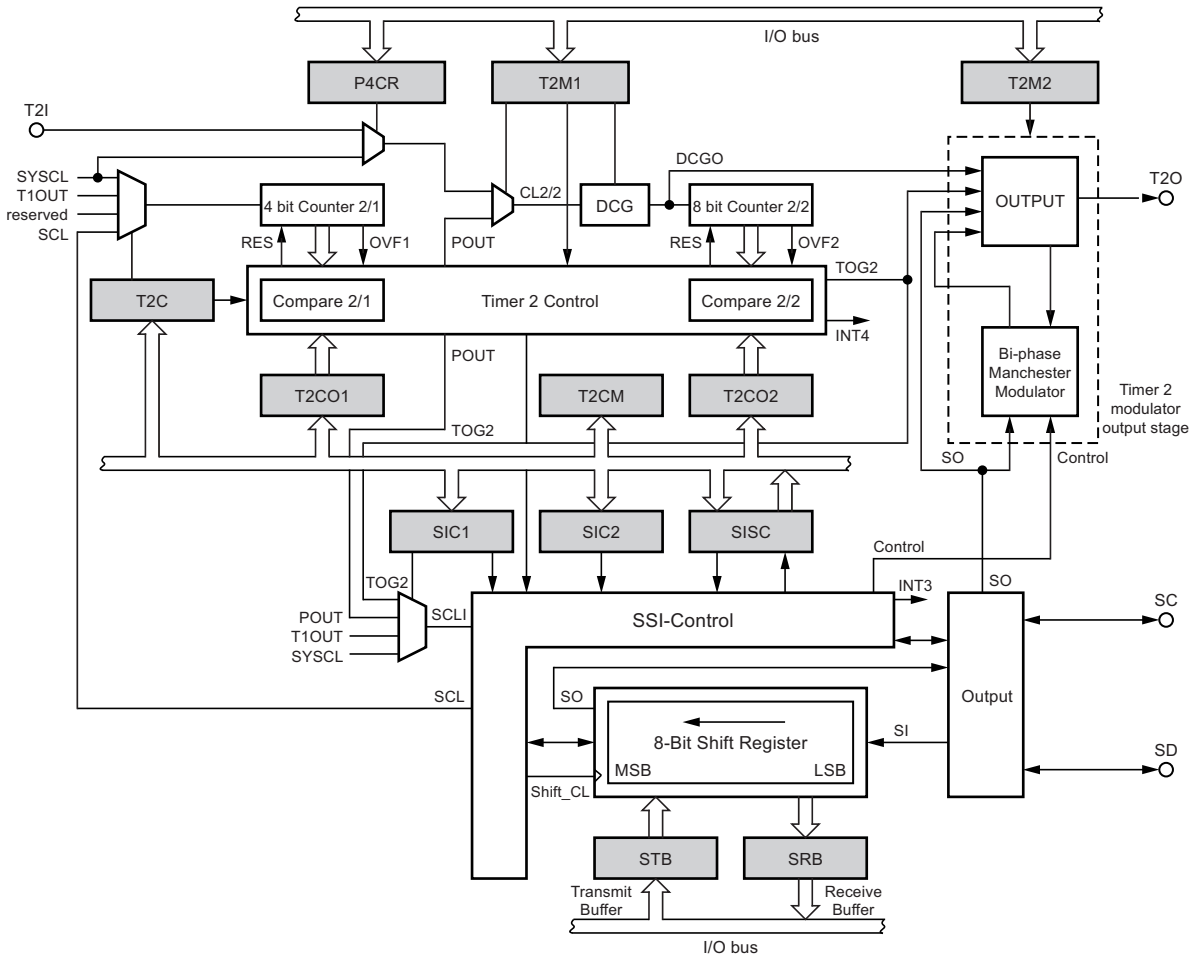
The SRB is the receive buffer of the SSI. The shift register clocks serial data in (most significant bit first) and loads content into the receive buffer when complete telegram has been received.

### 4.2.8 Combination Modes

The UTCM consists of one timer (Timer 2) and a serial interface. There is a multitude of modes in which the timers and serial interface can work together. The 8-bit wide serial interface operates as shift register for modulation. The modulator units work together with the timers and shift the data bits into or out of the shift register.

#### 4.2.8.1 Combination Mode Timer 2 and SSI

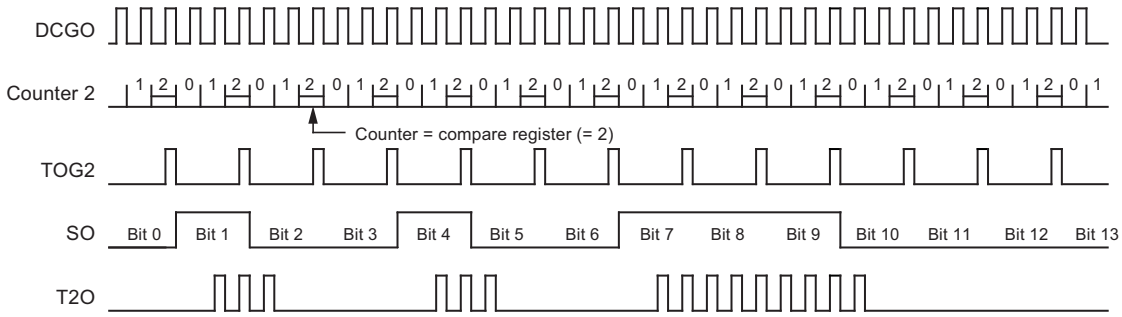
**Figure 4-32. Combination Timer 2 and SSI**



### Combination Mode 1 Burst Modulation

| | |
|---|---|
| SSI mode 1: | 8-bit NRZ and internal data SO output to the timer 2 modulator stage |
| Timer 2 mode 1, 2, 3 or 4: | 8-bit compare counter with 4-bit programmable prescaler and DCG |
| Timer 2 output mode 3: | Duty cycle burst generator |

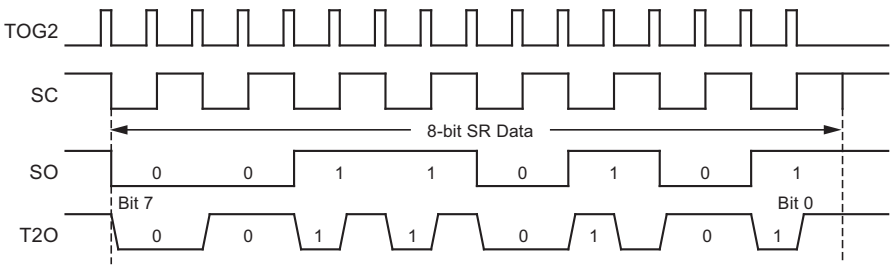
**Figure 4-33. Carrier Frequency Burst Modulation with the SSI Internal Data Output**




### Combination Mode 2: Bi-phase Modulation 1

| | |
|---|---|
| SSI mode 1: | 8-bit shift register internal data output (SO) to the timer 2 modulator stage |
| Timer 2 mode 1, 2, 3 or 4: | 8-bit compare counter with 4-bit programmable prescaler |
| Timer 2 output mode 4: | The Modulator 2 of Timer 2 modulates the SSI internal data out put to bi-phase code |

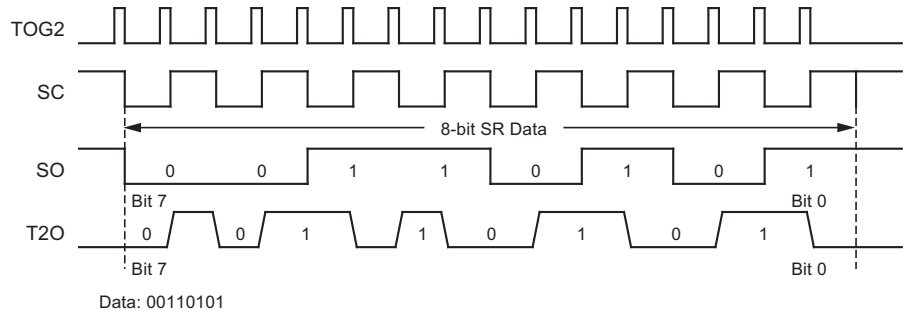
**Figure 4-34. Bi-phase Modulation 1**



### Combination Mode 3: Manchester Modulation 1

| | |
|---|---|
| SSI mode 1: | 8-bit shift register internal data output (SO) to timer 2 modulator stage |
| Timer 2 mode 1, 2, 3 or 4: | 8-bit compare counter with 4-bit programmable prescaler |
| Timer 2 output mode 5: | The Modulator 2 of Timer 2 modulates the SSI internal data out put to Manchester code |

**Figure 4-35. Manchester Modulation 1**

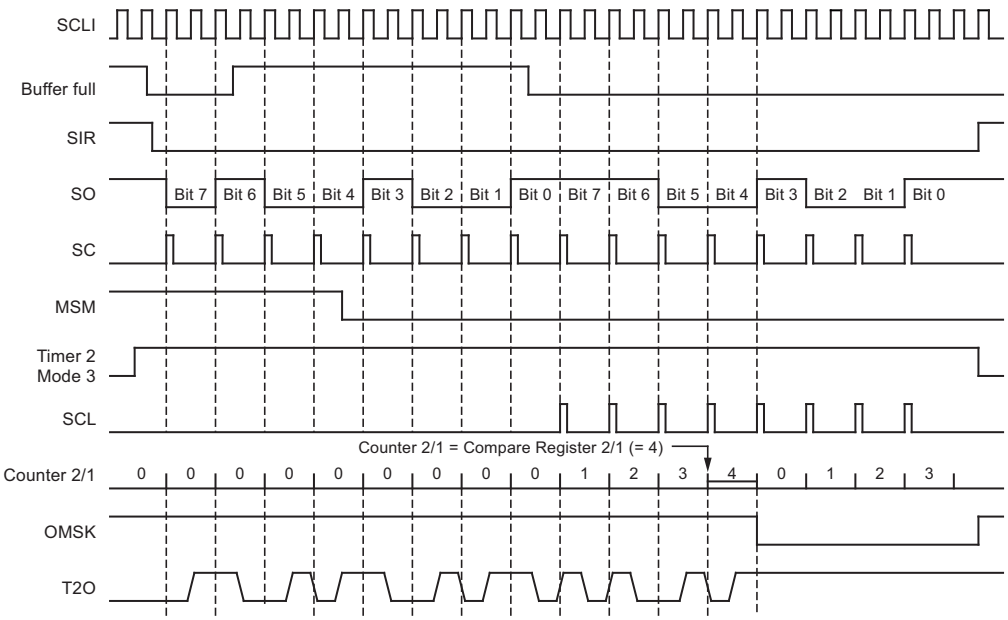



### Combination Mode 4: Manchester Modulation 2

| | |
|---|---|
| SSI mode 1: | 8-bit shift register internal data output (SO) to the timer 2 modulator stage |
| Timer 2 mode 3: | 8-bit compare counter and 4-bit prescaler |
| Timer 2 output mode 5: | The Modulator 2 of Timer 2 modulates the SSI data output to manchester code |

The 4-bit stage can be used as prescaler for the SSI to generate the stop signal for modulator 2. The SSI has a special mode to supply the prescaler with the shift-clock. The control output signal (OMSK) of the SSI is used as stop signal for the modulator. Figure 4-36 is an example for a 12-bit manchester telegram.

**Figure 4-36. Manchester Modulation 2**

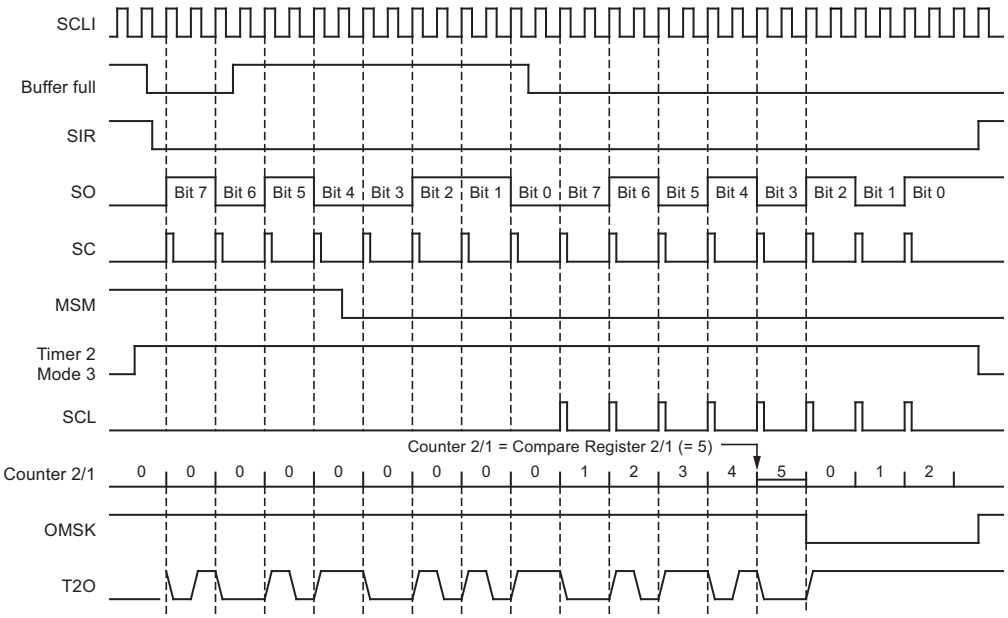


### Combination Mode 5: Bi-phase Modulation 2

| | |
|---|---|
| SSI mode 1: | 8-bit shift register internal data output (SO) to timer 2 modulator stage |
| Timer 2 mode 3: | 8-bit compare counter and 4-bit prescaler |
| Timer 2 output mode 4: | The modulator 2 of Timer 2 modulates the SSI data output to bi-phase code |

The 4-bit stage can be used as prescaler for the SSI to generate the stop signal for modulator 2. The SSI has a special mode to supply the prescaler via the shift-clock. The control output signal (OMSK) of the SSI is used as a stop signal for the modulator. Figure 4-37 is an example for a 13-bit Bi-phase telegram.

**Figure 4-37. Bi-phase Modulation 2**



# 5. Absolute Maximum Ratings

Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

All inputs and outputs are protected against high electrostatic voltages or electric fields. However, precautions to minimize the build-up of electrostatic charges during handling are recommended. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., $V_{DD}$).

Voltages are given relative to $V_{SS}$

| Parameters | Symbol | Value | Unit |
|---|---|---|---|
| Supply voltage | $V_{DD}$ | –0.3 to +6.5 | V |
| Input voltage (on any pin) | $V_{IN}$ | $V_{SS}$ –0.3 ≤ $V_{IN}$ ≤ $V_{DD}$ +0.3 | V |
| Output short circuit duration | $t_{short}$ | Indefinite | s |
| Operating temperature range | $T_{amb}$ | –40 to +85 | °C |
| Storage temperature range | $T_{stg}$ | –40 to +130 | °C |
| Thermal resistance (SSO20) | $R_{thJA}$ | 140 | K/W |
| Soldering temperature (t ≤ 10s) | $T_{sld}$ | 260 | °C |

# 6. Operating Characteristics

$V_{DD}$ = 5V, $V_{SS}$ = 0V, $T_{amb}$ = –40°C to +85°C unless otherwise specified

| Parameters | Test Conditions | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Power Supply | | | | | | |
| Active current<br>CPU active | $R_{ext}$ = 47kΩ<br>$f_{SYSCL}$ = $f_{RCext}$/2<br>$f_{SYSCL}$ = $f_{RCext}$/4 | $I_{DD}$ | | <br>330<br>170 | <br>370<br>190 | <br>µA<br>µA |
| Power down current<br>(CPU sleep, RC-oscillator active) | $R_{ext}$ = 47kΩ<br>$f_{SYSCL}$ = $f_{RCext}$/2<br>$f_{SYSCL}$ = $f_{RCext}$/4<br>$f_{SYSCL}$ = $f_{RCext}$/16 | $I_{PD}$ | | <br>40<br>35<br>30 | <br>45<br>40<br>35 | <br>µA<br>µA<br>µA |
| Sleep current<br>(CPU sleep, RC-oscillator inactive) | $V_{DD}$ = 6.5V | $I_{Sleep}$ | | 0.5 | 0.8 | µA |

$V_{DD}$ = 5.5V, $V_{SS}$ = 0V, $T_{amb}$ = –40°C to +85°C unless otherwise specified.

| Parameters | Test Conditions | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Active current<br>CPU active | $R_{ext}$ = 47kΩ<br>$f_{SYSCL}$ = $f_{RCext}$/2<br>$f_{SYSCL}$ = $f_{RCext}$/4 | IDD | | <br>370<br>190 | <br>410<br>210 | <br>µA<br>µA |
| Power down current<br>(CPU sleep, RC oscillator active) | $R_{ext}$ = 47kΩ<br>$f_{SYSCL}$ = $f_{RCext}$/2<br>$f_{SYSCL}$ = $f_{RCext}$/4<br>$f_{SYSCL}$ = $f_{RCext}$/16 | IPD | | <br>45<br>40<br>35 | <br>50<br>45<br>40 | <br>µA<br>µA<br>µA |

$V_{SS}$ = 0V, $T_{amb}$ = 25°C unless otherwise specified.

| Parameters | Test Conditions | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Power-on Reset Threshold Voltage | | | | | | |
| POR threshold voltage | BOT = 1 | $V_{POR}$ | 2.5 | 3.0 | 3.5 | V |
| POR threshold voltage | BOT = 0 | $V_{POR}$ | 3.5 | 4.0 | 4.5 | V |
| POR hysteresis | | $V_{POR}$ | | 50 | | mV |
| Voltage Monitor Threshold Voltage | | | | | | |
| VM high threshold voltage | $V_{DD}$ > VM, VMS = 1 | $V_{MThh}$ | | 5.0 | 5.5 | V |
| VM high threshold voltage | $V_{DD}$ < VM, VMS = 0 | $V_{MThh}$ | 4.5 | 5.0 | | V |
| VM low threshold voltage | $V_{DD}$ > VM, VMS = 1 | $V_{MThl}$ | | 4.0 | 4.5 | V |
| VM low threshold voltage | $V_{DD}$ < VM, VMS = 0 | $V_{MThl}$ | 3.5 | 4.0 | | V |
| External Input Voltage | | | | | | |
| VMI | VMI > VBG, VMS = 1 | $V_{VMI}$ | | 1.25 | 1.4 | V |
| VMI | VMI < VBG, VMS = 0 | $V_{VMI}$ | 1.1 | 1.25 | | V |

## 6.1 All Bi-directional Ports

$V_{SS}$ = 0V, $T_{amb}$ = –40°C to +85°C unless otherwise specified.

| Parameters | Test Conditions | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Input voltage LOW | $V_{DD}$ = 3.5V to 6.5V | $V_{IL}$ | $V_{SS}$ | | $0.2 \times V_{DD}$ | V |
| Input voltage HIGH | $V_{DD}$ = 3.5V to 6.5V | $V_{IH}$ | $0.8 \times V_{DD}$ | | $V_{DD}$ | V |
| Input LOW current (dynamic pull-up) | $V_{DD}$ = 3.5V, $V_{IL}$ = $V_{SS}$<br>$V_{DD}$ = 6.5V | $I_{IL}$ | –15<br>–50 | –30<br>–100 | –50<br>–200 | µA<br>µA |
| Input HIGH current (dynamic pull-down) | $V_{DD}$ = 3.5V, $V_{IH}$ = $V_{DD}$<br>$V_{DD}$ = 6.5V | $I_{IH}$ | 15<br>50 | 30<br>100 | 50<br>200 | µA<br>µA |
| Input LOW current (static pull-up) | $V_{DD}$ = 3.5V, $V_{IL}$ = $V_{SS}$<br>$V_{DD}$ = 6.5V | $I_{IL}$ | –120<br>–300 | –250<br>–600 | –500<br>–1200 | µA<br>µA |
| Input LOW current (static pull-down) | $V_{DD}$ = 3.5V, $V_{IH}$ = $V_{DD}$<br>$V_{DD}$ = 6.5V | $I_{IH}$ | 120<br>300 | 250<br>600 | 500<br>1200 | µA<br>µA |
| Output LOW current | $V_{OL}$ = 0.2$V_{DD}$<br>$V_{DD}$ = 3.5V,<br>$V_{DD}$ = 6.5V | $I_{OL}$ | 3<br>8 | 5<br>15 | 8<br>22 | mA<br>mA |
| Output HIGH current | $V_{OH}$ = 0.8$V_{DD}$<br>$V_{DD}$ = 3.5V,<br>$V_{DD}$ = 6.5V | $I_{OH}$ | –3<br>–8 | –5<br>–16 | –8<br>–24 | mA<br>mA |

Note: The pin BP20/NTE has a static pull-up resistor during the reset-phase of the microcontroller:

# 7. AC Characteristics

## 7.1 Operation Cycle Time

$V_{SS}$ = 0V

| Parameters | Test Conditions | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| System clock cycle | $V_{DD}$ = 2.5V to 6.5V<br>$T_{amb}$ = –40°C to +85°C | $t_{SYSCL}$ | 0.25 | | 100 | µs |

Supply voltage $V_{DD}$ = 2.5V to 6.5V, $V_{SS}$ = 0V, $T_{amb}$ = 25°C unless otherwise specified.

| Parameters | Test Conditions | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Timer 2 Input Timing Pin T2I | | | | | | |
| Timer 2 input clock | | $f_{T2I}$ | | | 5 | MHz |
| Timer 2 input LOW time | Rise/fall time < 10ns | $t_{T2IL}$ | 100 | | | ns |
| Timer 2 input HIGH time | Rise/fall time < 10ns | $t_{T2IH}$ | 100 | | | ns |
| Interrupt Request Input Timing | | | | | | |
| Interrupt request LOW time | Rise/fall time < 10ns | $t_{IRL}$ | 100 | | | ns |
| Interrupt request HIGH time | Rise/fall time < 10ns | $t_{IRH}$ | 100 | | | ns |
| External System Clock | | | | | | |
| EXSCL at OSC1 input | ECM = EN<br>Rise/fall time < 10ns | $f_{EXSCL}$ | 0.5 | | 8 | MHz |
| EXSCL at OSC1 input | ECM = DI<br>Rise/fall time < 10ns | $f_{EXSCL}$ | 0.02 | | 8 | MHz |
| Input HIGH time | Rise/fall time < 10ns | $t_{IH}$ | 0.1 | | | µs |
| Reset Timing | | | | | | |
| Power-on reset time | $V_{DD}$ >$V_{POR}$ | $t_{POR}$ | | 1.5 | 5 | ms |
| RC-oscillator 1 | | | | | | |
| Frequency | | $f_{RcOut1}$ | | 4 | | MHz |
| Stability | $V_{DD}$ = 3.5V to 5.5V<br>$T_{amb}$ = –40°C to +85°C | Δf/f | | | ±50 | % |
| Stabilization time | $V_{DD}$ = 3.5V to 5.5V | $t_S$ | | | 1 | ms |
| RC-oscillator 2 – External Resistor | | | | | | |
| Frequency | $R_{ext}$ = 47kΩ | $f_{RcOut2}$ | | 1.6 | | MHz |
| Stability | $V_{DD}$ = 3.5V to 5.5V<br>$T_{amb}$ = –40°C to +85°C | Δf/f | | | 10 | % |
| Stabilization time | $V_{DD}$ = 3.5V to 5.5V | $t_S$ | | | 1 | ms |
| External resistor | | $R_{ext}$ | 12 | 47 | 100 | kΩ |

**Figure 7-1. Active Supply Current versus Frequency**

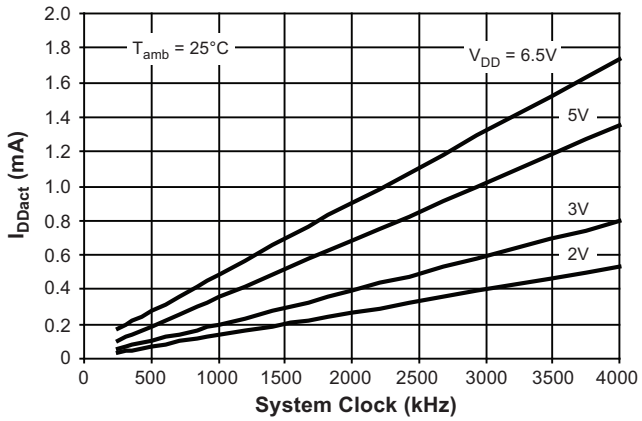



**Figure 7-2. Power-down Supply Current versus Frequency**

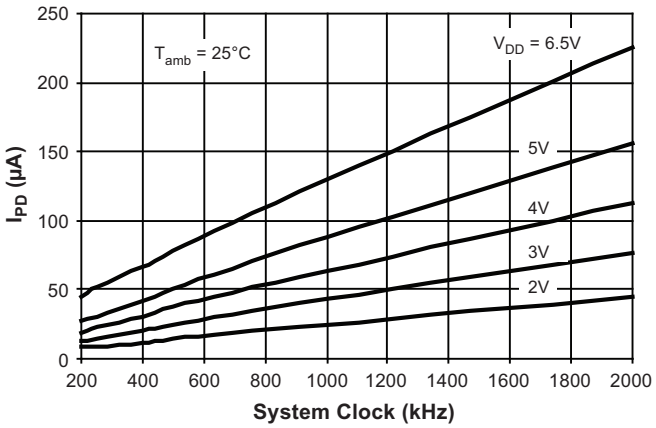



**Figure 7-3. Active Supply Current versus $V_{DD}$**

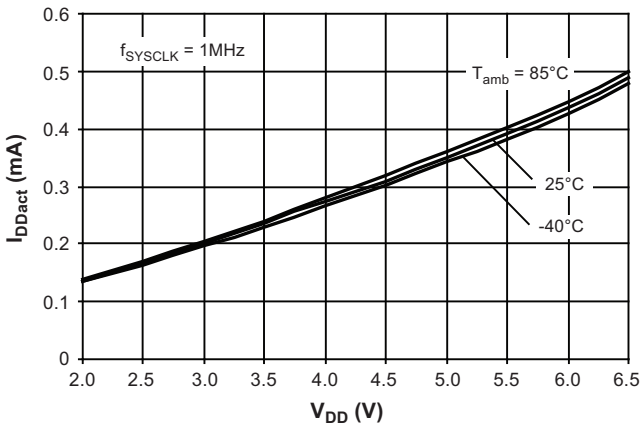


**Figure 7-4. Power-down Supply Current versus $V_{DD}$**

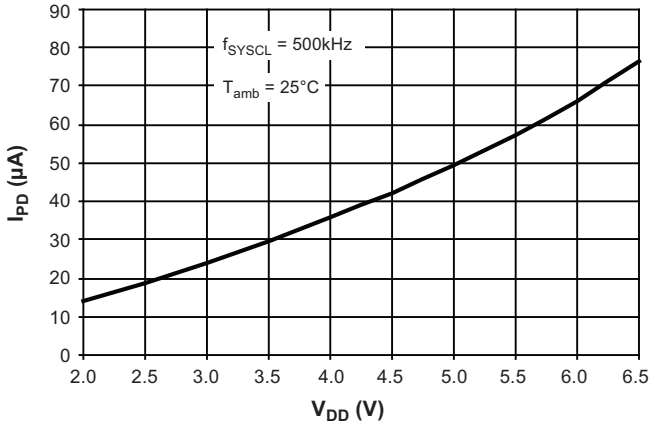



**Figure 7-5. Internal RC Frequency versus $V_{DD}$**

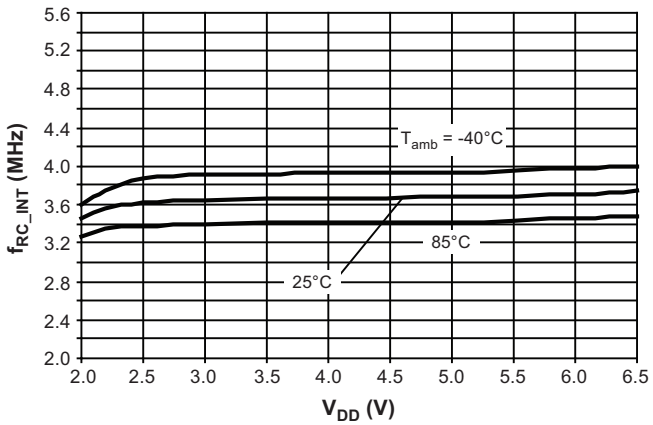



**Figure 7-6. External RC Frequency versus $V_{DD}$**

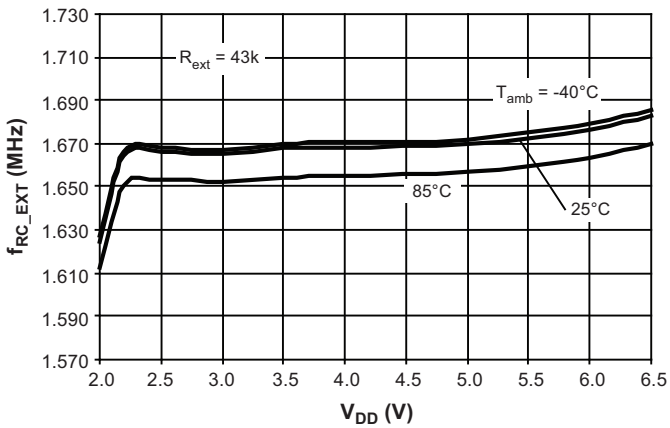


**Figure 7-7. Maximum System Clock versus $V_{DD}$**

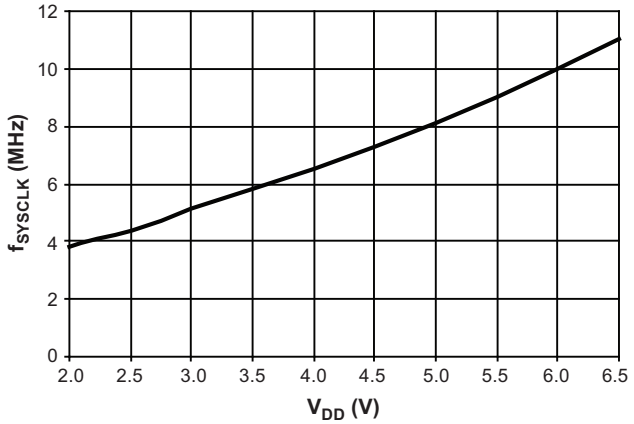



**Figure 7-8. Internal RC Frequency versus $T_{amb}$**

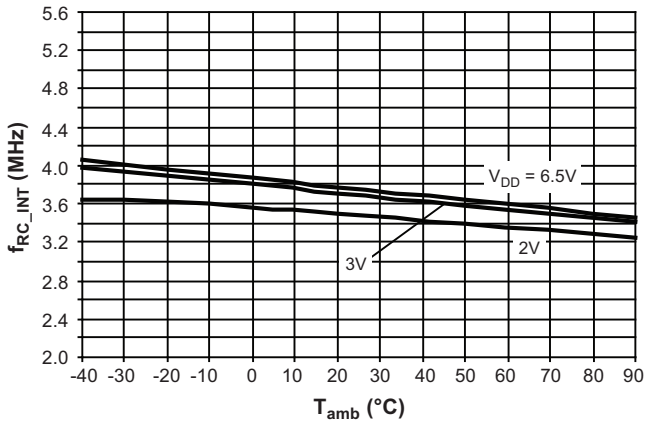



**Figure 7-9. External RC Frequency versus $T_{amb}$**

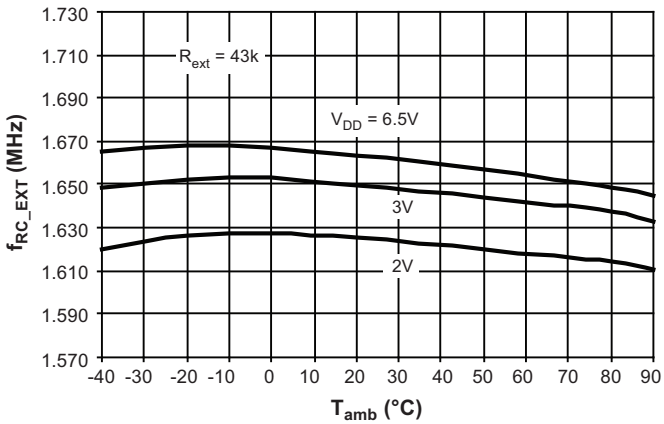


**Figure 7-10. External RC Frequency versus $R_{ext}$**

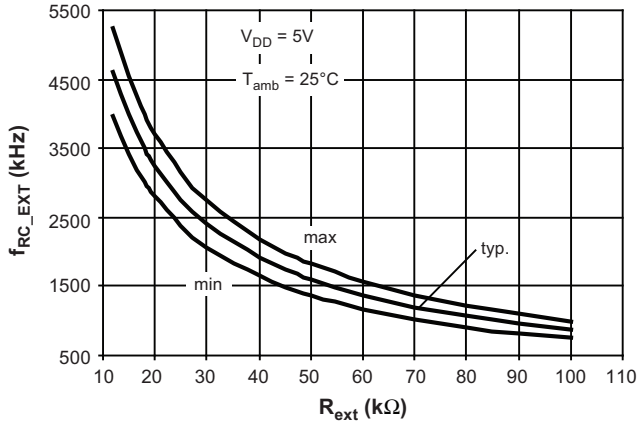



**Figure 7-11. Pull-up Resistor versus $V_{DD}$**

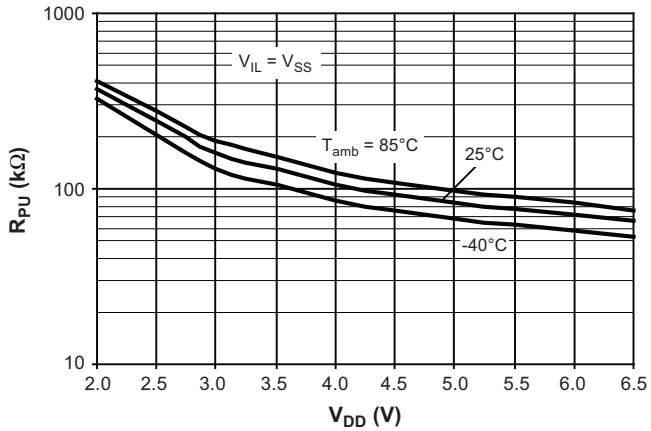



**Figure 7-12. Strong Pull-up Resistor versus $V_{DD}$**

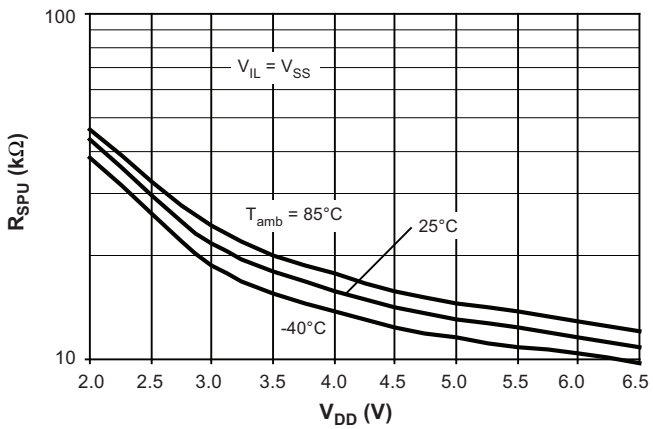


**Figure 7-13. Output High Current versus $V_{DD}$ - Output High Voltage**

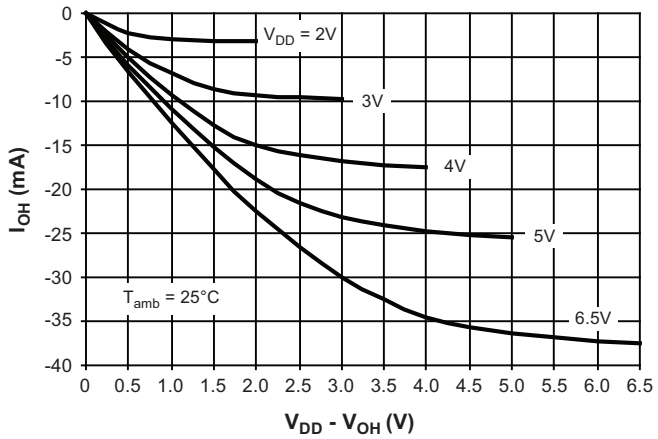



**Figure 7-14. Pull-down Resistor versus $V_{DD}$**

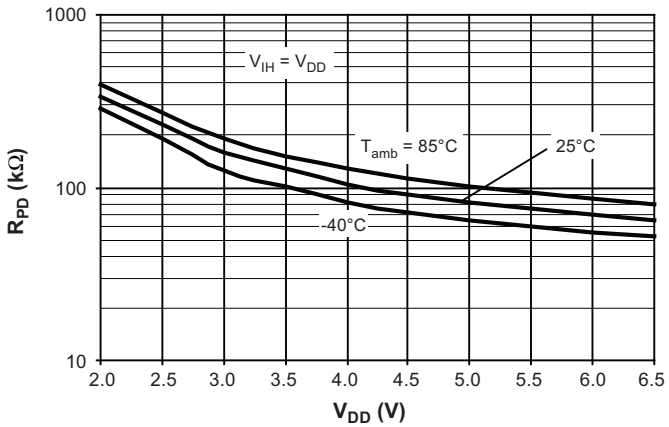



**Figure 7-15. Strong Pull-down Resistor versus $V_{DD}$**

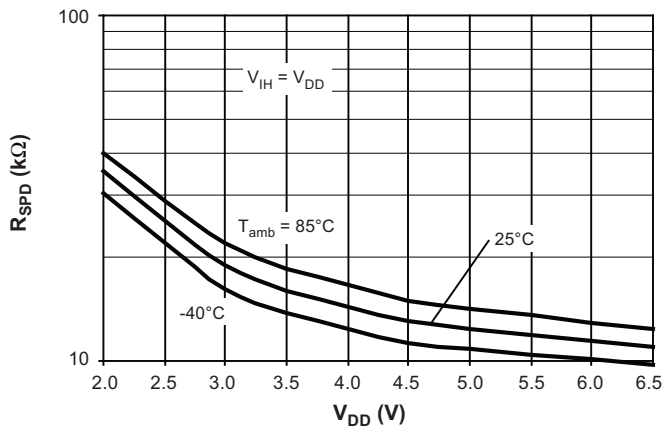


**Figure 7-16. Output Low Current versus Output Low Voltage**

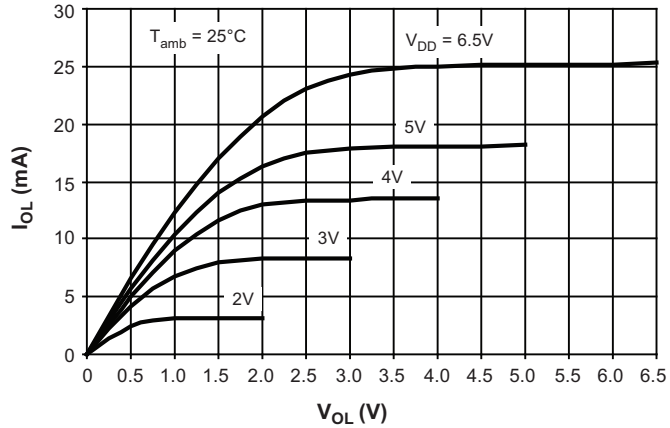



**Figure 7-17. Output High Current versus $T_{amb}$ = 25°C, $V_{DD}$ = 6.5V, $V_{OH}$ = 0.8 × $V_{DD}$**

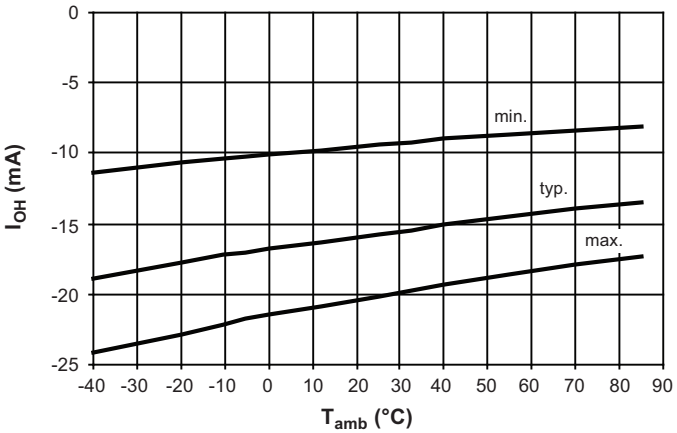
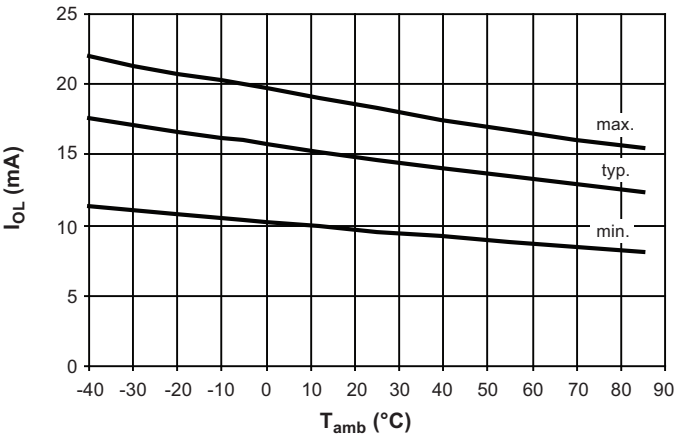



**Figure 7-18. Output Low Current versus $T_{amb}$, $V_{DD}$ = 6.5V, $V_{OL}$ = 0.2 × $V_{DD}$**

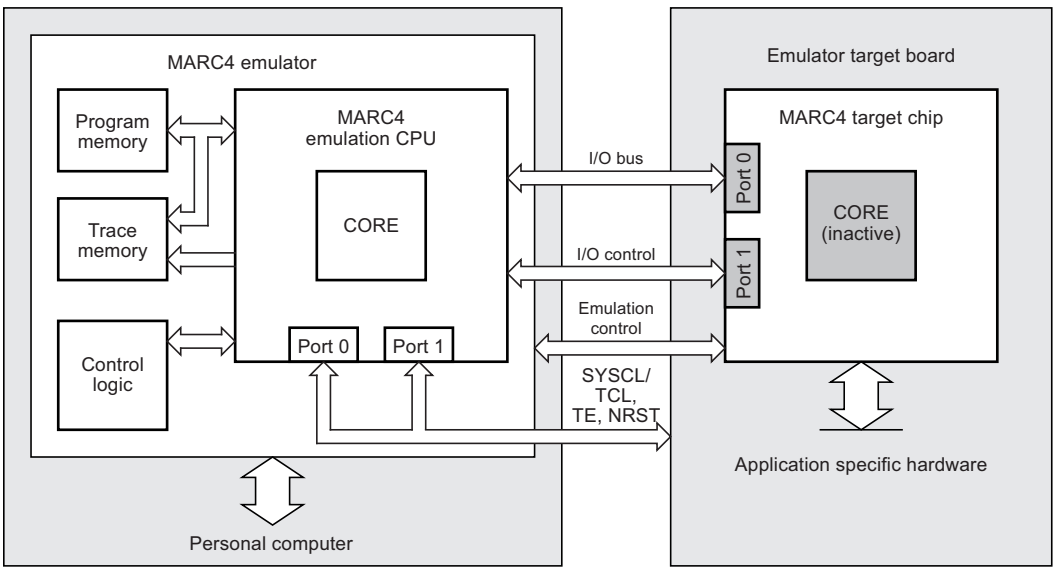


## 7.2 Emulation

The basic function of emulation is to test and evaluate the customer's program and hardware in real time. This therefore enables the analysis of any timing, hardware or software problem. For emulation purposes, all MARC4 controllers include a special emulation mode. In this mode, the internal CPU core is inactive and the I/O buses are available via port 0 and port 1 to allow an external access to the on-chip peripherals. The MARC4 emulator uses this mode to control the peripherals of any MARC4 controller (target chip) and emulates the lost ports for the application.

The MARC4 emulator can stop and restart a program at specified points during execution, making it possible for the applications engineer to view the memory contents and those of various registers during program execution. The designer also gains the ability to analyze the executed instruction sequences and all the I/O activities.

**Figure 7-19. MARC4 Emulation**



# 8. Option Settings for Ordering

Please select the option settings from the list below and insert ROM CRC.

| | Output | Input |
|---|---|---|
| **Port 2** | | |
| BP20 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>Pull-down static |
| BP21 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| **Port 4** | | |
| BP40 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| BP41 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| BP42 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| BP43 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |

| | Output | Input |
|---|---|---|
| **Port 5** | | |
| BP50 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| BP51 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| BP52 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| BP53 | [ ] CMOS<br>[ ] Open drain [N]<br>[ ] Open drain [P] | [ ] Pull-up<br>[ ] Pull-down<br>[ ] Pull-up static<br>[ ] Pull-down static |
| **ECM (External Clock Monitor)** | | |
| | [ ] Enable<br>[ ] Disable | |
| **Watchdog** | | |
| | [ ] Softlock<br>[ ] Hardlock | |
| **Used oscillator** | | |
| | [ ] Ext. RC<br>[ ] Ext. clock | |

**Please attach this page to the approval form**.

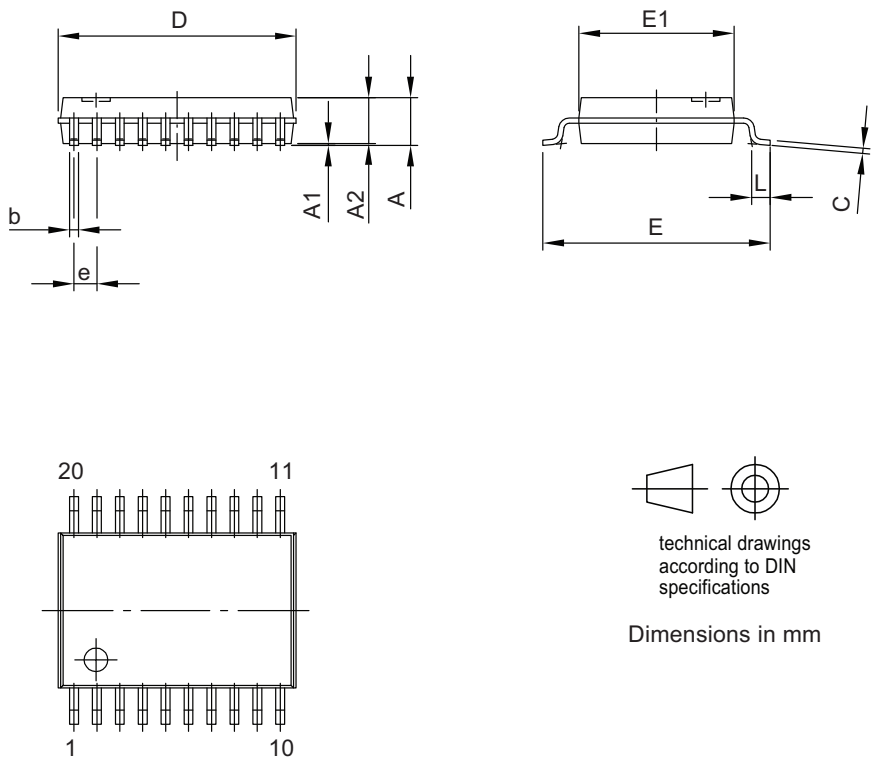
Date: _____ Signature: _____ Company: _____

## 9. Ordering Information

| Extended Type Number[1] | Program Memory | Data-EEPROM | Package | Delivery | MOQ |
|---|---|---|---|---|---|
| ATA6020x-yyy-TKQW | 2kB ROM | No | SSO20, Pb-free | Taped and reeled | 4,000 |

Notes: 1. x = Hardware revision
yyy = Customer specific ROM-version

## 10. Package Information




technical drawings
according to DIN
specifications

Dimensions in mm

| COMMON DIMENSIONS (Unit of Measure = mm) | | | | |
|---|---|---|---|---|
| Symbol | MIN | NOM | MAX | NOTE |
| A | 0.9 | 1.0 | 1.1 | |
| A1 | 0.05 | 0.1 | 0.15 | |
| A2 | 0.85 | 0.9 | 0.95 | |
| D | 6.4 | 6.5 | 6.6 | |
| E | 6.3 | 6.4 | 6.5 | |
| E1 | 4.3 | 4.4 | 4.5 | |
| L | 0.5 | 0.6 | 0.7 | |
| C | 0.1 | 0.15 | 0.2 | |
| b | 0.2 | 0.25 | 0.3 | |
| e | | 0.65 BSC | | |

04/16/14

| Package Drawing Contact: packagedrawings@atmel.com | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| | Package: SSO20<br>4.4mm | | 6.543-5182.01-4 | 1 |

# 11. Revision History

Please note that the following page numbers referred to in this section refer to the specific revision mentioned, not to this document.

| Revision No. | History |
|---|---|
| 4708F-4BMCU-10/14 | • Section 9 “Ordering Information” on page 66 updated<br>• Section 10 “Package Information” on page 66 updated |
| 4708E-4BMCU-03/14 | • Put datasheet in the latest template |
| 4708D-4BMCU-09/05 | • Put datasheet in the latest template<br>• Pb-free Logo on page 1 added<br>• Ordering Information on page 67 updated |
| 4708C-4BMCU-02/04 | • Figure 4 “ROM MAP” on page 4 updated<br>• Figure 55 to Figure 72 on page 56 to page 61 added |
| 4708B-4BMCU-12/03 | • Put datasheet in the latest template<br>• Figure 5 “RAM MAP” on page 4 updated<br>• Table 9 “Peripheral Addresses” on page 19 updated<br>• New heading rows at Table “Absolute Maximum Ratings” on page 53 added<br>• Section “Emulation” on page 56 added<br>• Table “Ordering Information” on page 58 added<br>• Table name on page 57 updated |






**Atmel Corporation** 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

© 2014 Atmel Corporation. / Rev.: Rev.: 4708F–4BMCU–10/14

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.