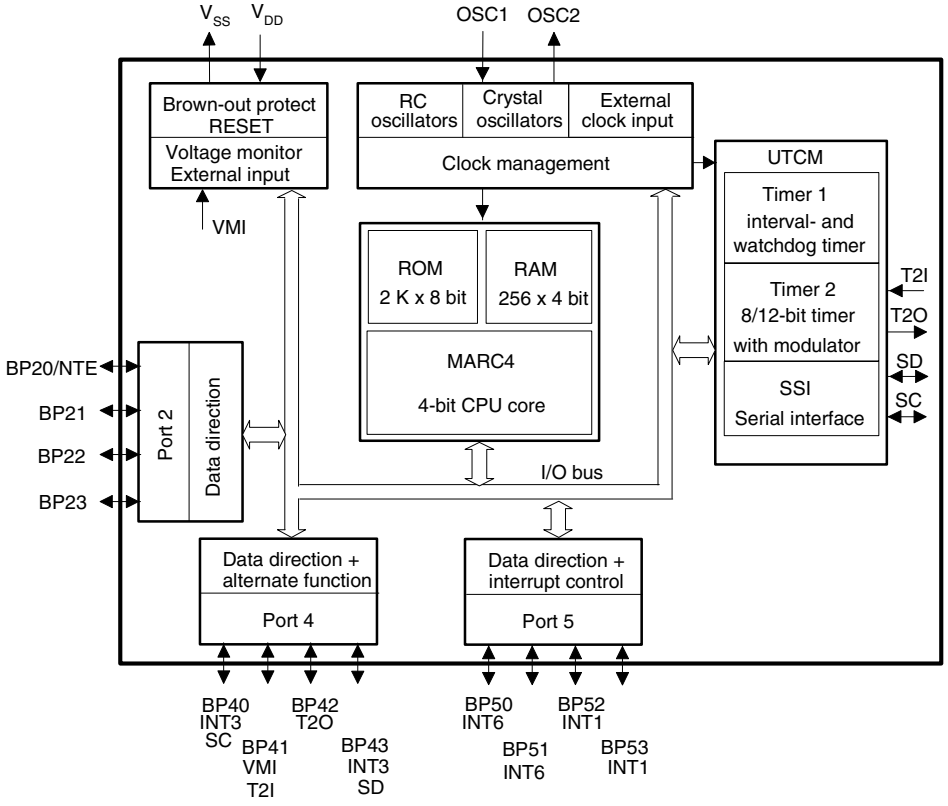
Features

- **2-Kbyte ROM, 256 × 4-bit RAM**
- **12 Bi-directional I/Os**
- **Up to 6 External/Internal Interrupt Sources**
- **Multifunction Timer/Counter with**
  - **– IR Remote Control Carrier Generator**
  - **– Bi-phase-, Manchester- and Pulse-width Modulator and Demodulator**
- **Programmable System Clock with Prescaler and Five Different Clock Sources**
- **Wide Supply-voltage Range (1.8 V to 6.5 V)**
- **Very Low Sleep Current (< 1 µA)**
- **32 × 16-bit EEPROM (ATAR890 only)**
- **Synchronous Serial Interface (2-wire, 3-wire)**
- **Watchdog, POR and Brown-out Function**
- **Voltage Monitoring Inclusive Lo_BAT Detection**
- **Flash Controller ATAM893 Available (SSO20)**

Description

The ATAR090 and ATAR890 are members of Atmel's family of 4-bit single-chip microcontrollers. They offer the highest integration for IR and RF data communication and remote-control. The ATAR090 and ATAR890 are suitable for the transmitter side. They contain ROM, RAM, parallel I/O ports, two 8-bit programmable multifunction timer/counters with modulator and demodulator function, voltage supervisor, interval timer with watchdog function and a sophisticated on-chip clock generation with external clock input, integrated RC-, 32-kHz crystal- and 4-MHz crystal-oscillators. The ATAR890 has an additional EEPROM as a second chip in one package.

Figure 1. Block Diagram



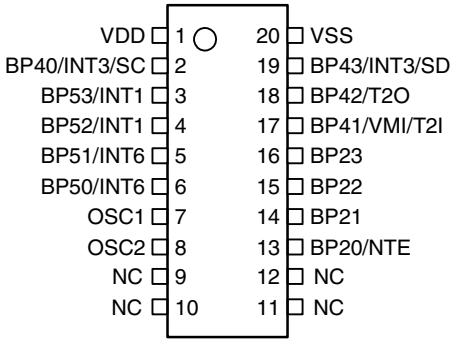
Low-current Microcontroller for Wireless Communication

ATAR090
ATAR890

Rev. 4696C–4BMCU–02/04

Pin Configuration

Figure 2. Pinning SSO20



Pin Description

Name	Type	Function	Alternate Function	Pin No.	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 section “Master Reset”	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	–
OSC1	I	Oscillator input	4-MHz crystal input or 32-kHz crystal input or external clock input or external trimming resistor input	7	Input
OSC2	O	Oscillator output	4-MHz crystal output or 32-kHz crystal output or external clock input	8	NA

Introduction

The ATAR090/ATAR890 are members of Atmel's family of 4-bit single-chip microcontrollers. They contain ROM, RAM, parallel I/O ports, one 8-bit programmable multifunction timer/counters, voltage supervisor, interval timer with watchdog function and a sophisticated on-chip clock generation with integrated RC-, 32-kHz crystal- and 4-MHz crystal oscillators. Table 2 provides an overview of the available variants.

Table 1. Available Variants of ATAxx9x

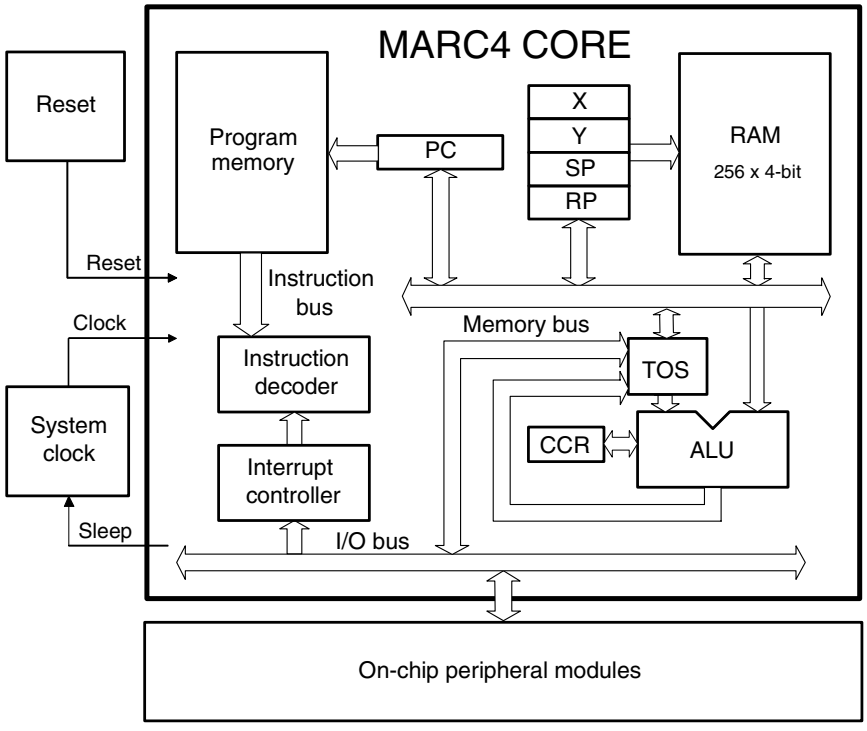
Version	Type	ROM	E2PROM Peripheral	Packages
Flash device	ATAM893	4-Kbyte EEPROM	64 byte	SSO20
Production	ATAR090	2-Kbyte mask ROM	–	SSO20
Production	ATAR890	2-Kbyte mask ROM	64 byte	SSO20

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 separate 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. MARC4 Core



Components of MARC4 Core

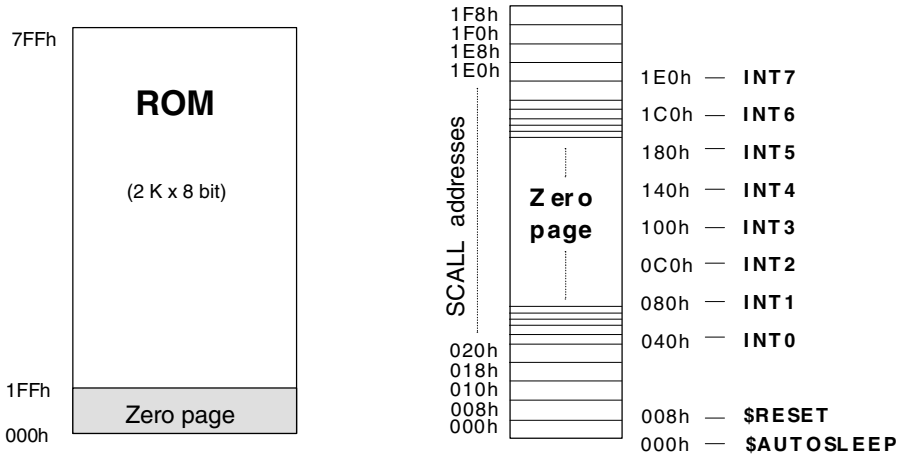
The core contains ROM, RAM, ALU, program counter, RAM address registers, instruction decoder and interrupt controller. The following sections describe each functional block in more detail.

ROM

The program memory (ROM) is mask programmed with the customer application program during 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 4 Look-up tables of constants can also be held in ROM and are accessed via the MARC4's built-in table instruction.

Figure 4. ROM Map



RAM

The ATAR090 and ATAR890 contain 256 x 4-bit wide static random access memory (RAM). It is used for the expression stack, the return stack and data memory for variables and arrays. The RAM is addressed by any of the four 8-bit wide RAM address registers SP, RP, X and Y.

Expression Stack

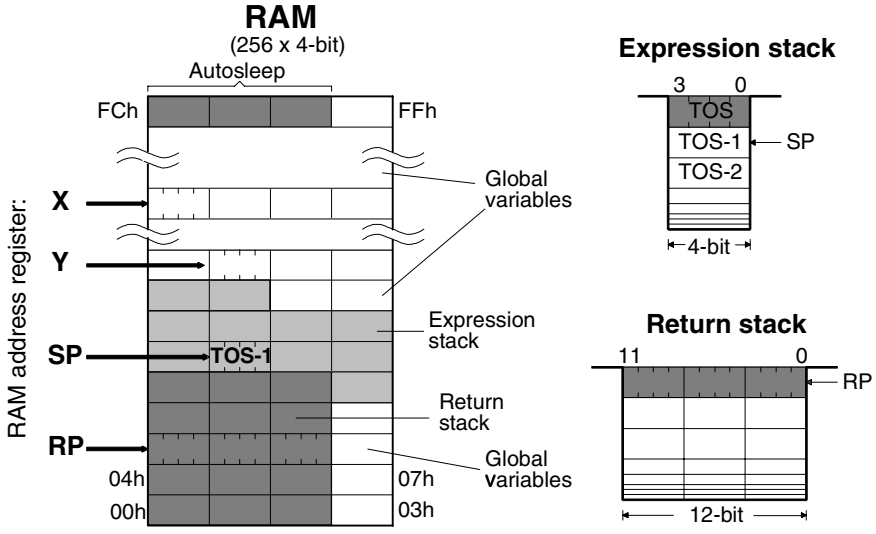
The 4-bit wide expression stack is addressed with the expression stack pointer (SP). All arithmetic, I/O and memory reference operations take their operands from, and return their results to the expression stack. The MARC4 performs the operations with the top of stack items (TOS and TOS-1). The TOS register 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.

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.

Figure 5. RAM Map



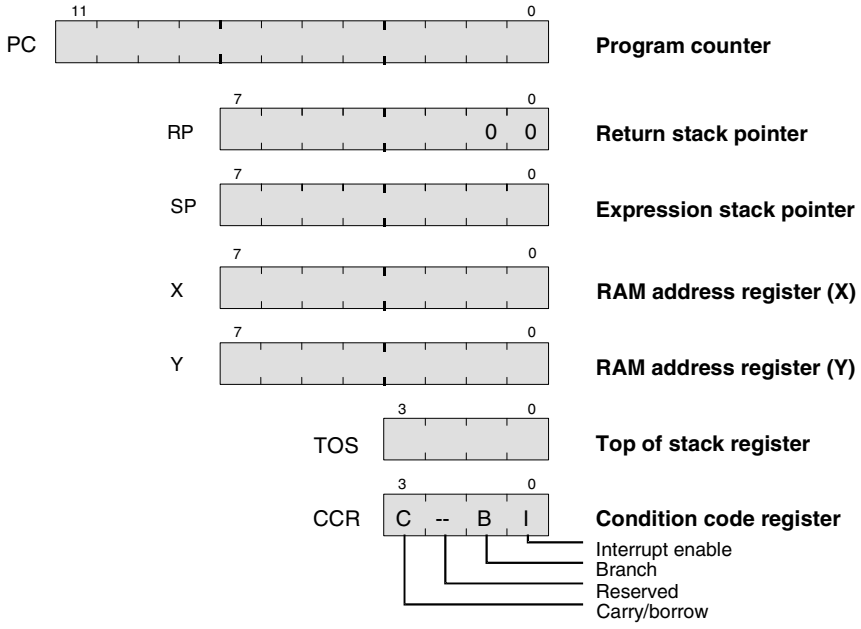
Registers

The MARC4 controller has seven programmable registers and one condition code register. They are shown in the following programming model (see Figure 6).

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 6. Programming Model



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.

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.

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.

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 modes arrays in the RAM can be compared, filled or moved

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.

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.

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 affect on the C-flag.

Branch (B)

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

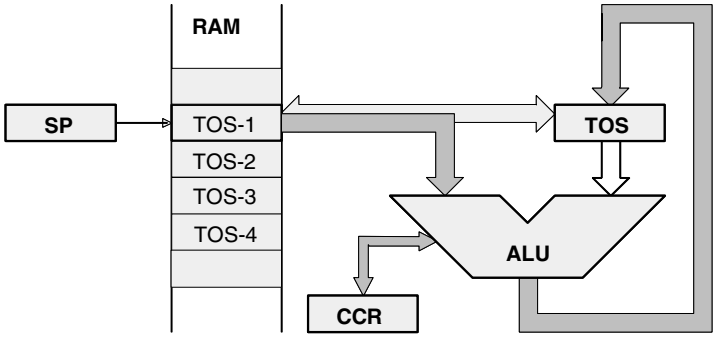
Interrupt Enable (I)

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 while 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.

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 affect the carry/borrow and branch flag in the condition code register (CCR).

Figure 7. ALU Zero-address Operations



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 takes 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 “Peripheral Modules”. 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 “Emulation”).

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 contain 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”.

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 the ROM (see Table 1 on page 3). 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 “Peripheral Modules”).

Interrupt Processing

In order to be able to process eight interrupt levels, the MARC4 contains 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 also be noted that automatic stacking of the RBR is not carried out by the hardware, therefore, 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.

Interrupt Latency

The interrupt latency is the time from the occurrence of the interrupt to the interrupt service routine being activated. In MARC4 this is extremely short (taking between 3 to 5 machine cycles depending on the state of the core).

Figure 8. Interrupt Handling

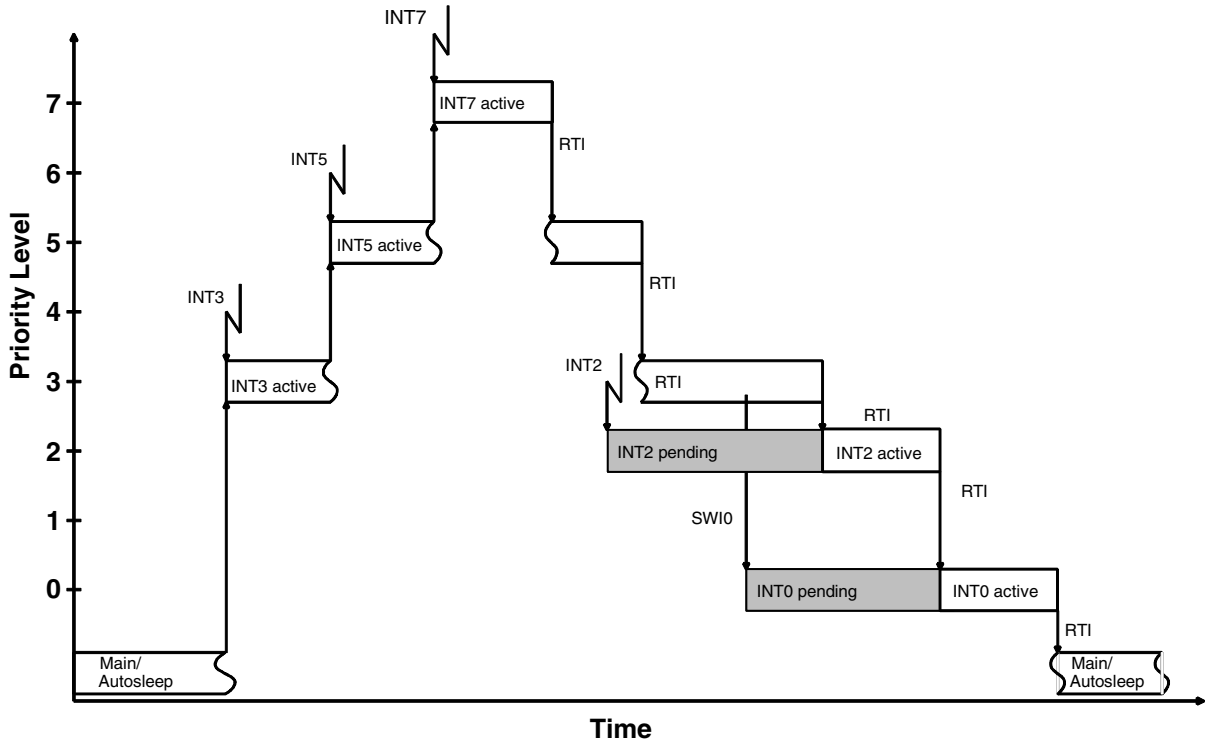


Table 2. Interrupt Priority Table

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. Hardware Interrupts

Interrupt	Interrupt Mask		Interrupt Source
	Register	Bit	
INT1	P5CR	P52M1, P52M2 P53M1, P53M2	Any edge at BP52 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 P51M1, P51M2	Any edge at BP50 Any edge at BP51
INT7	VCM	VIM	External/internal voltage monitoring

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.

Hardware Interrupts

In the ATAR090, 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.

Master Reset

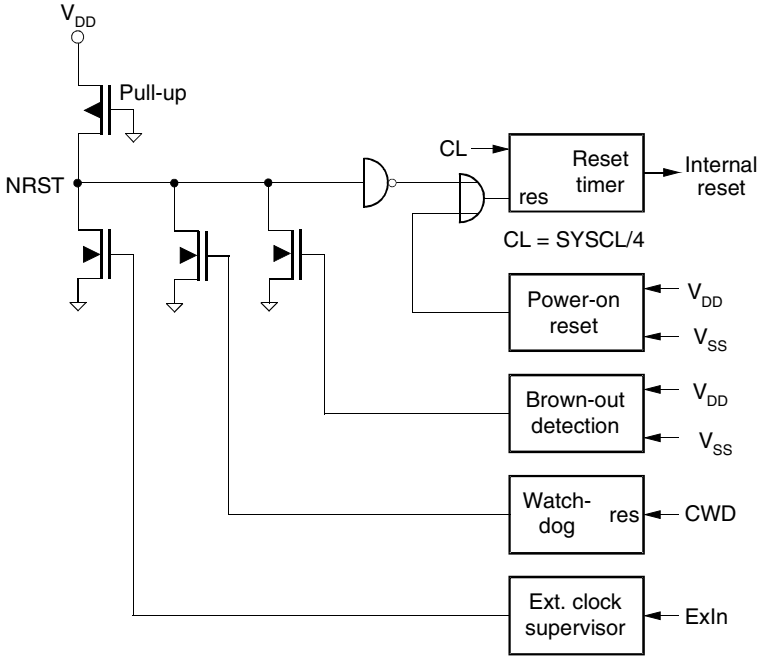
The master reset forces the CPU into a well-defined condition. It is unmaskable and is activated independent of the current program state. It can be triggered by either initial supply power-up, a short collapse of the power supply, the brown-out detection circuitry, a watchdog time-out, or an external input clock supervisor stage (see Figure 9).

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 150 kΩ.

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 9. Reset Configuration



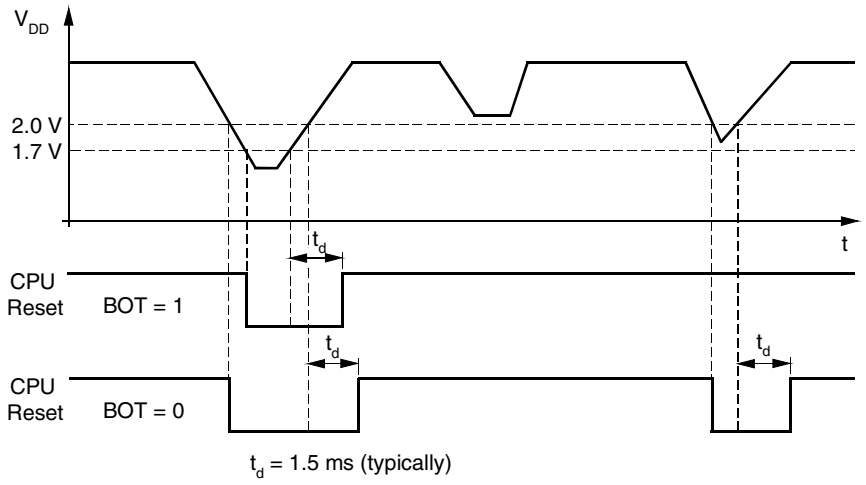
Power-on Reset and Brown-out Detection

The ATAR090/ATAR890 have 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 (1.7 V). 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 10. Brown-out Detection



Note: BOT = 1, low brown-out voltage threshold 1.7 V (is reset value).
BOT = 0, high brown-out voltage threshold 2.0 V.

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.

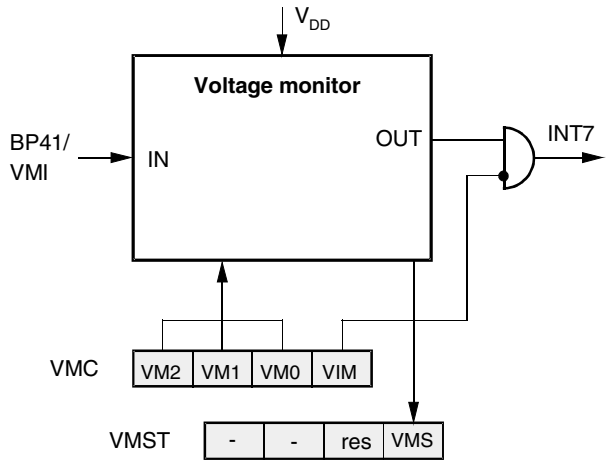
External Clock Supervisor

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.

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 three internal programmable thresholds: one lower threshold (2.2 V), one middle threshold (2.6 V). and one higher threshold (3.0 V). For external voltages at the VMI-pin, the comparator threshold is set to V_{BG} = 1.3 V. 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 11. Voltage Monitor



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: Voltage monitor **M**ode bit 2
VM1: Voltage monitor **M**ode bit 1
VM0: Voltage monitor **M**ode bit 0

Table 4. Voltage Monitor Modes

VM2	VM1	VM0	Function
1	1	1	Disable voltage monitor
1	1	0	External (VIM input), internal reference threshold (1.3 V), interrupt with negative slope
1	0	1	Not allowed
1	0	0	External (VMI input), internal reference threshold (1.3 V), interrupt with positive slope
0	1	1	Internal (supply voltage), high threshold (3.0 V), interrupt with negative slope
0	1	0	Internal (supply voltage), middle threshold (2.6 V), interrupt with negative slope
0	0	1	Internal (supply voltage), low threshold (2.2 V), interrupt with negative slope
0	0	0	Not allowed

VIM Voltage **I**nterrupt **M**ask bit

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

VMS Voltage **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 12. Internal Supply Voltage Supervisor

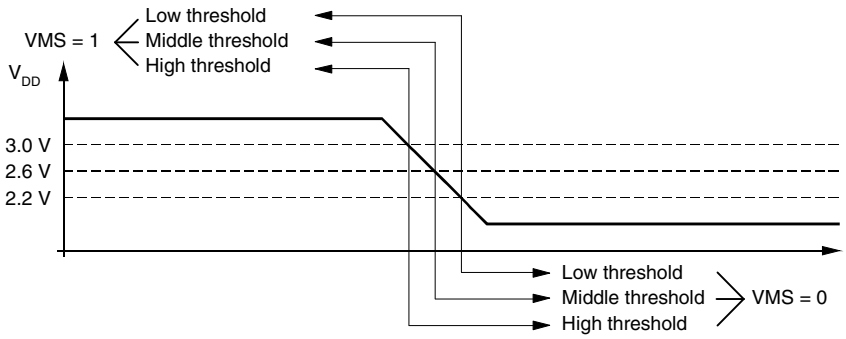
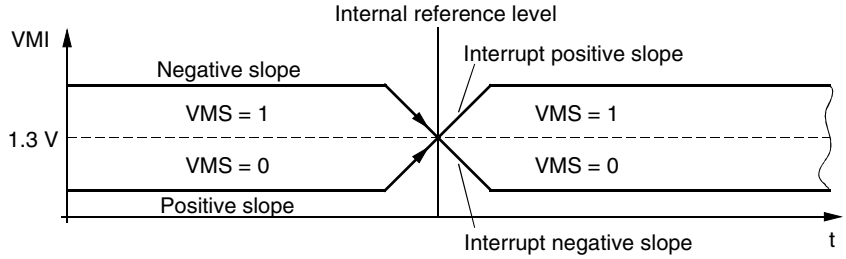


Figure 13. External Input Voltage Supervisor



Clock Generation

Clock Module

The ATAR090/ATAR890 contains a clock module with 4 different internal oscillator types: two RC-oscillators, one 4-MHz crystal oscillator and one 32-kHz crystal oscillator. The pins OSC1 and OSC2 are the interface to connect a crystal either to the 4-MHz, or to the 32-kHz crystal oscillator. OSC1 can be used as input for external clocks or to connect an external trimming resistor for the RC-oscillator 2. All necessary circuitry except the crystal and 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%. The RC-oscillator 2 is a trimmable oscillator whereby the oscillator frequency can be trimmed with an external resistor attached between OSC1 and V_{DD}. In this configuration, the RC-oscillator 2 frequency can be maintained stable with 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 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 500 kHz for more than 1 ms.

Figure 14. Clock Module

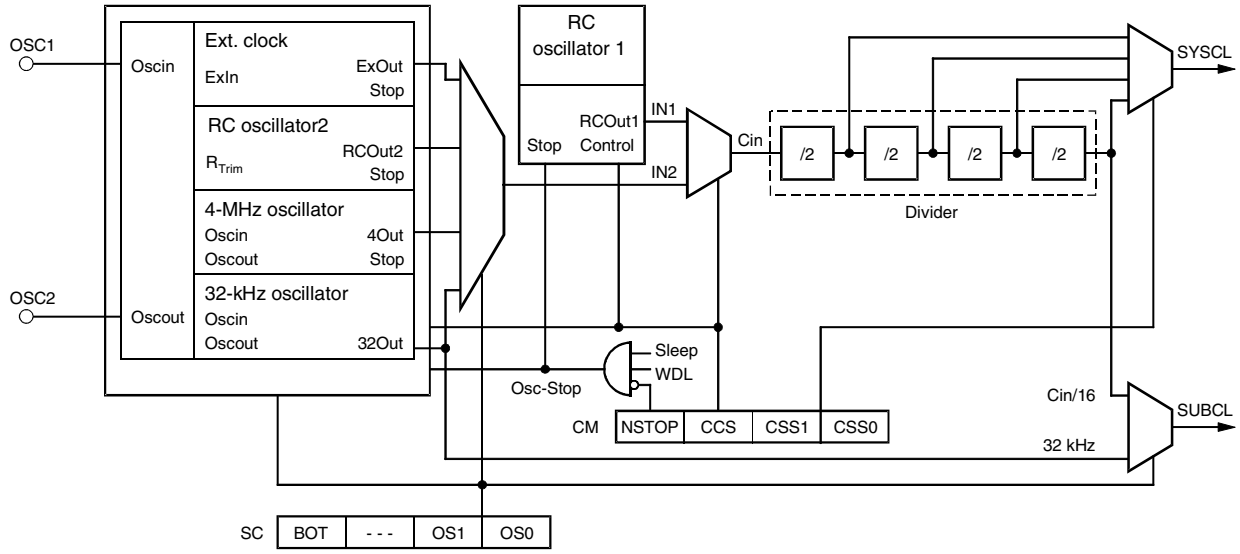


Table 5. Clock Modes

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
3	1	0	RC-oscillator 1 (internal)	4-MHz oscillator	C_{in}/16
4	0	0	RC-oscillator 1 (internal)	32-kHz oscillator	32 kHz

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.

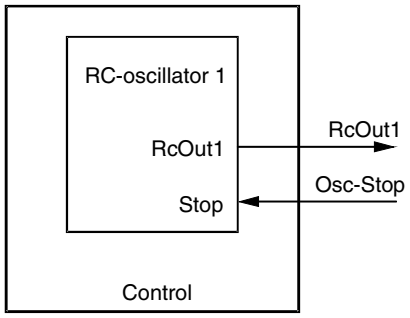
Oscillator Circuits and External Clock Input Stage

The ATAR090/ATAR890 series consists of four different internal oscillators: two RC-oscillators, one 4-MHz crystal oscillator, one 32-kHz crystal oscillator and one external clock input stage.

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 $f_0 \approx$ 3.8 MHz. The RC oscillator 1 is selected by default after power-on reset.

Figure 15. RC-oscillator 1



External Input Clock

The OSC1 or OSC2 (mask option) 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 is missing for more than 1 ms 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 500 kHz for more than 1 ms.

Figure 16. External Input Clock

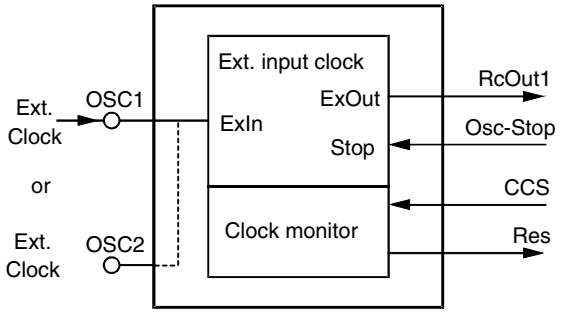


Table 6. Supervisor Function Control Bits

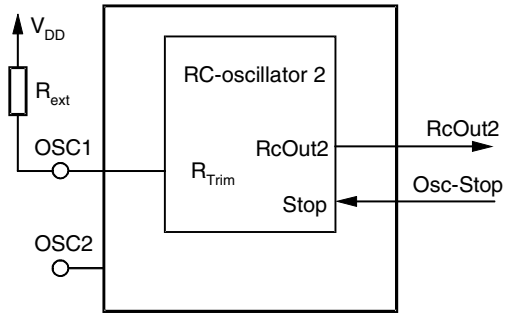
OS1	OS0	CCS	Supervisor Reset Output (Res)
1	1	0	Enable
1	1	1	Disable
x	0	x	Disable

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 with a tolerance of ±10% over the full operating temperature and a voltage range of V_{DD} from 2.5 V to 6.0 V.

For example: An output frequency at the RC-oscillator 2 of 2 MHz can be obtained by connecting a resistor R_{ext} = 360 kΩ (see Figure 17).

Figure 17. RC-oscillator 2



4-MHz Oscillator

The ATAR090/ATAR890 4-MHz oscillator options need a crystal or ceramic resonator connected to the OSC1 and OSC2 pins to establish oscillation. All the necessary oscillator circuitry is integrated, except the actual crystal, resonator, C_1 and C_2.

Figure 18. 4-MHz Crystal Oscillator

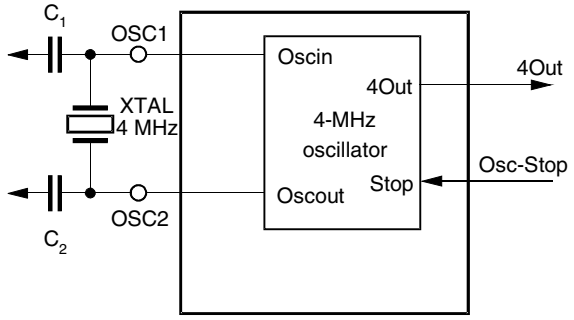
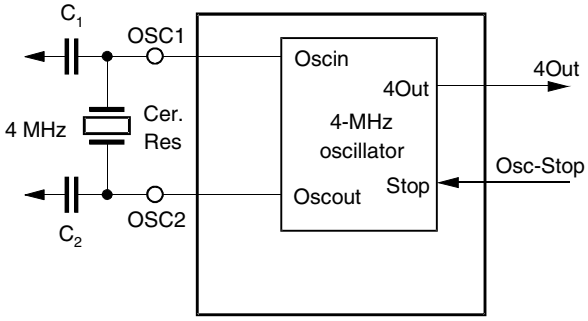


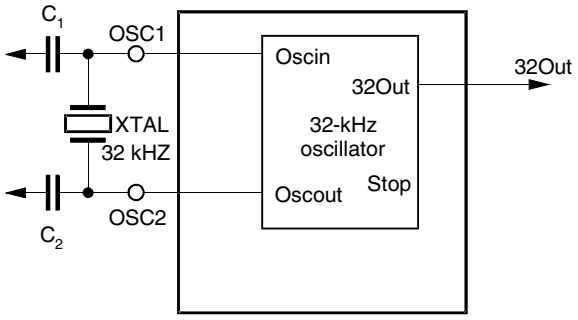
Figure 19. Ceramic Resonator



32-kHz Oscillator

Some applications require long-term time keeping or low resolution timing. In this case, an on-chip, low power 32-kHz crystal oscillator can be used to generate both the SUBCL and the SYSCL. In this mode, power consumption is greatly reduced. The 32-kHz crystal oscillator can not be stopped while the power-down mode is in operation.

Figure 20. 32-kHz Crystal Oscillator



Clock Management

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

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, the 4-Mhz crystal oscillator, the 32-kHz crystal oscillator, 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 7. Core Speed Select

CSS1	CSS0	Divider	Note
0	0	16	
1	1	8	Reset value
1	0	4	
0	1	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 (1.7 V)
BOT = 0, high brown-out voltage threshold (2.0 V)

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

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

Table 8. 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
3	1	0	C_{in}/16	RC-oscillator 1 and 4-MHz crystal oscillator
4	0	0	32 kHz	RC-oscillator 1 and 32-kHz crystal oscillator

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

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 maintained while 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. For standard applications use the $AUTOSLEEP routine to enter the power-down mode. Using the SLEEP instruction instead of the $AUTOSLEEP following an I/O instruction requires the insertion of 3 non I/O instruction cycles (for example NOP NOP NOP) between the IN or OUT command and the SLEEP command.

The total power consumption is directly proportional to the active time of the microcontroller. For a rough estimate 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 ATAR090/ATAR890 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. An exception is the 32-kHz oscillator, if it is selected it runs continuously independent of the NSTOP-bit. If the oscillator is stopped or the 32-kHz oscillator is selected, power consumption is extremely low.

Table 9. Power-down Modes

Mode	CPU Core	Osc-Stop[1]	Brown-out Function	RC-Oscillator 1 RC-Oscillator 2 4-MHz Oscillator	32-kHz Oscillator	External Input Clock
Active	RUN	NO	Active	RUN	RUN	YES
Power-down	SLEEP	NO	Active	RUN	RUN	YES
SLEEP	SLEEP	YES	STOP	STOP	RUN	STOP

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

Peripheral Modules

Addressing Peripherals

Accessing the peripheral modules takes place via the I/O bus (see Figure 21). 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 module's 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 21. Example of I/O Addressing

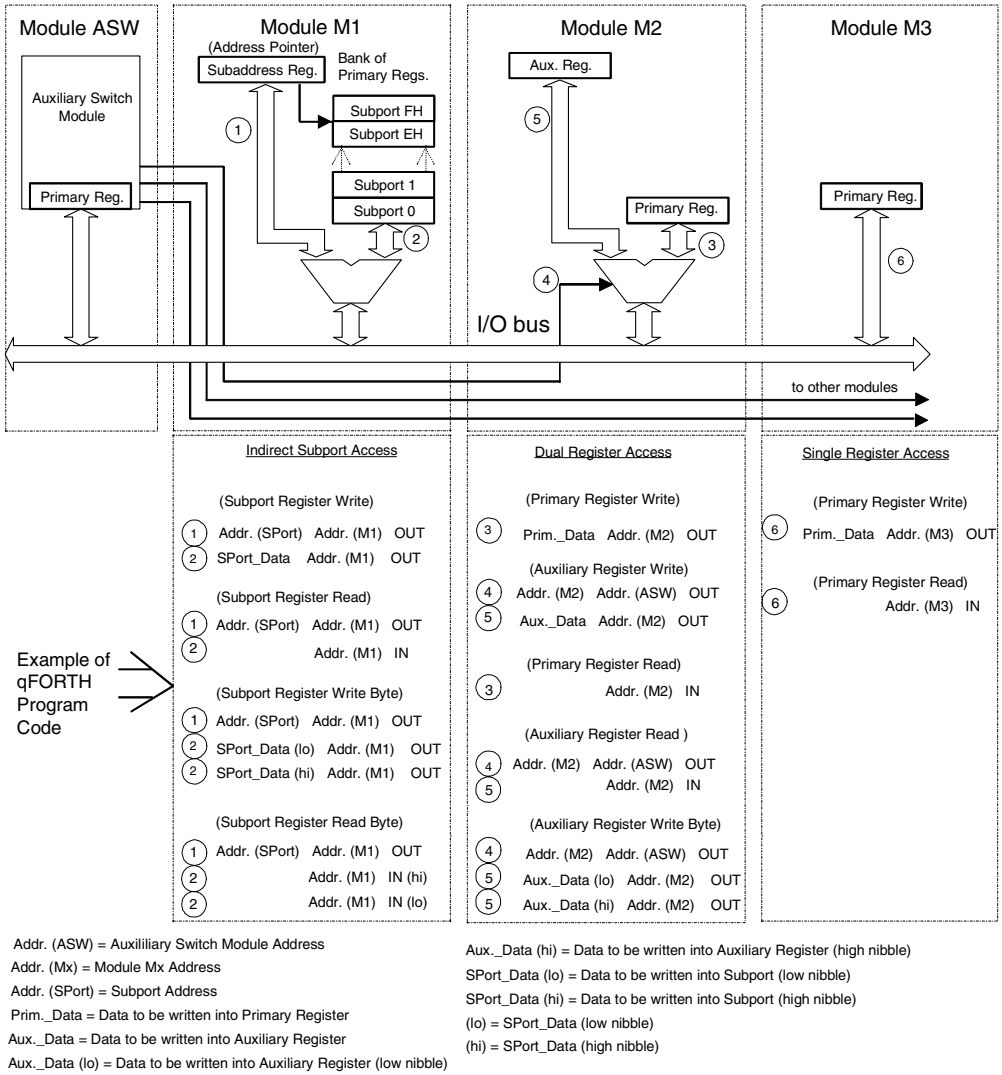


Table 10. Peripheral Addresses

Port Address			Name	Write/ Read	Reset Value	Register Function	Module Type	See Page
1			–	–	–	Reserved		
2			P2DAT	W/R	1111b	Port 2 - data register/pin data	M2	23
	Auxiliary		P2CR	W	1111b	Port 2 - control register		23
3			SC	W	1x11b	System configuration register	M3	18
			CWD	R	xxxxb	Watchdog reset	M3	11
	Auxiliary		CM	W	1111b	Clock management register	M2	17
4			P4DAT	W/R	1111b	Port 4 - data register/pin data	M2	26
	Auxiliary		P4CR	W	1111 1111b	Port 4 - control register (byte)		26
5			P5DAT	W/R	1111b	Port 5 - data register/pin data	M2	25
	Auxiliary		P5CR	W	1111 1111b	Port 5 - control register (byte)		25
6			–	–	–	Reserved		
7			T12SUB	W	–	Data to Timer 1/2 subport	M1	20
		Subport 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	29
		9	T1C2	W	x111b	Timer 1 control register 2	M1	29
		A	WDC	W	1111b	Watchdog control register	M1	30
		B-F				Reserved		
8			ASW	W	1111b	Auxiliary/switch register	ASW	20
9			STB	W	xxxx xxxxb	Serial transmit buffer (byte)	M2	51
			SRB	R	xxxx xxxxb	Serial receive buffer (byte)		51
	Auxiliary		SIC1	W	1111b	Serial interface control register 1		49
A			SISC	W/R	1x11b	Serial interface status/control register	M2	50
	Auxiliary		SIC2	W	1111b	Serial interface control register 2		49
B		–			–	Reserved		
C		–			–	Reserved		
D		–			–	Reserved		
E		–			–	Reserved		
F		VMC		W	1111b	Voltage monitor control register	M3	12
		VMST		R	xx11b	Voltage monitor status register	M3	12

Bi-directional Ports

Ports (2, 4, 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 strong 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.

Bi-directional Port 2

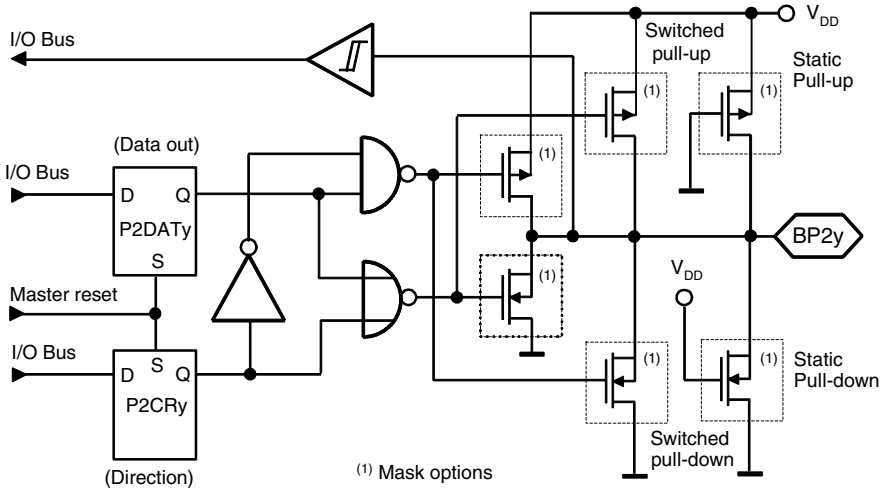
As all other bi-directional ports, this port includes a bitwise programmable Control Register (P2CR), which enables the individual programming of each port bit as input or output. It also opens up the possibility of reading the pin condition when in output mode. This is a useful feature for self-testing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

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

Note: 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 150 kΩ 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 22. Bi-directional Port 2



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

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 11. Port 2 Control Register

Code 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

Bi-directional Port 5

As all other bi-directional ports, this port includes 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 23 on page 24 and Figure 24 on page 24). 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 and then the high nibble (see section “Addressing Peripherals”).

Figure 23. Bi-directional Port 5

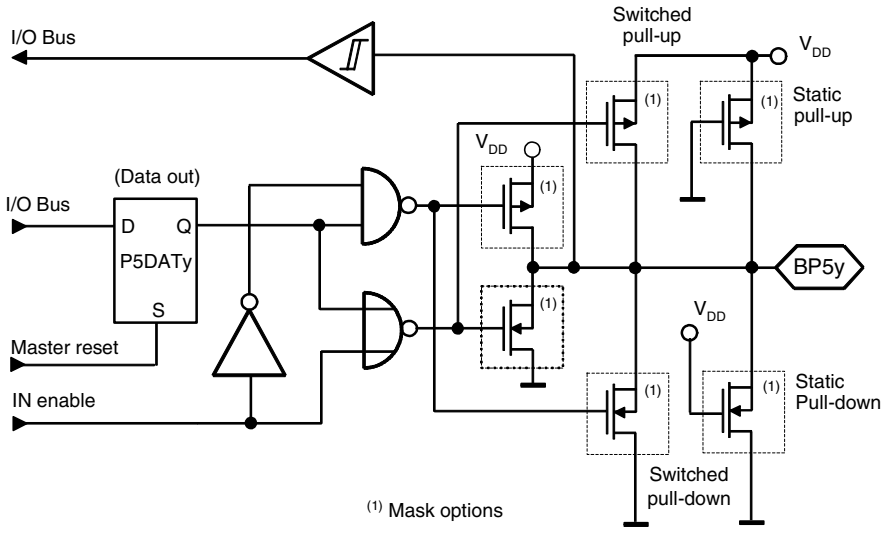
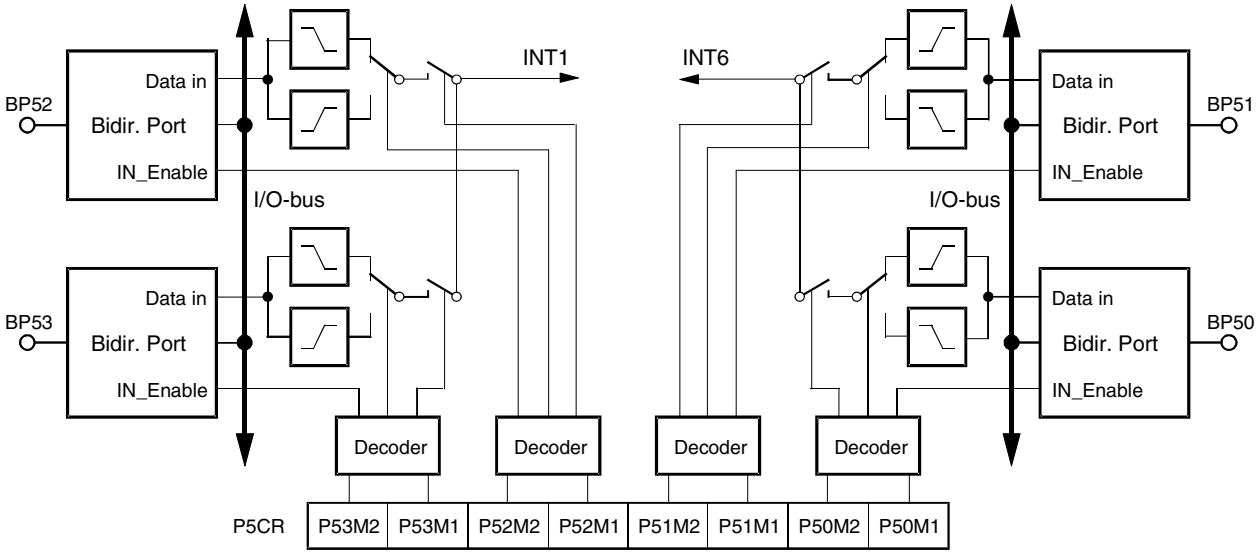


Figure 24. Port 5 External Interrupts



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**

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 12. Port 5 Control Register

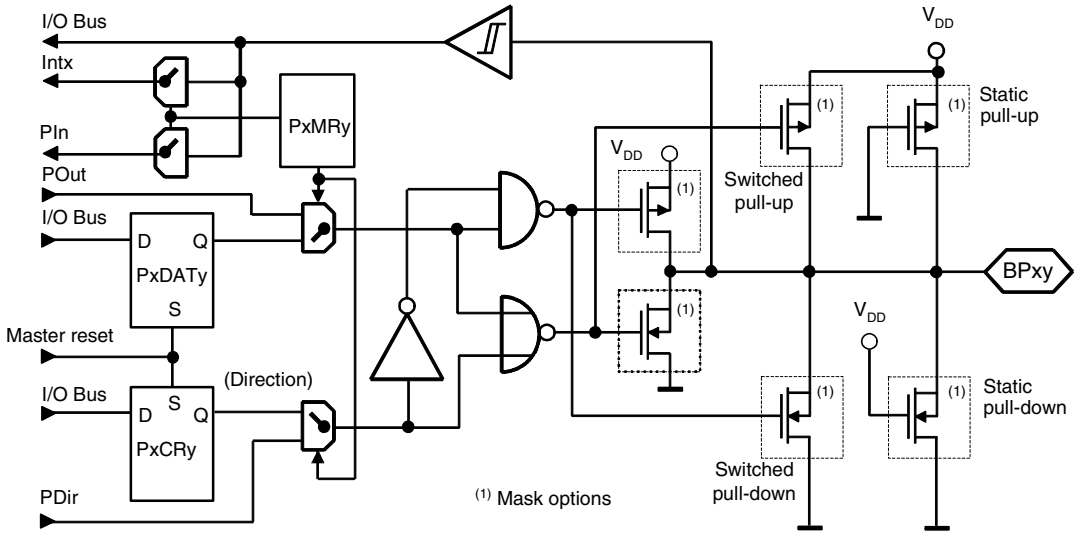
Auxiliary Address: '5'hex First Write Cycle		**Second Write Cycle**	
Code 3 2 1 0	**Function**	**Code 3 2 1 0**	**Function**
x x 1 1	BP50 in input mode – interrupt disabled	x x 1 1	BP52 in input mode – interrupt disabled
x x 0 1	BP50 in input mode – rising edge interrupt	x x 0 1	BP52 in input mode –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

Bi-directional Port 4

The bi-directional Port 4 is 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 26 on page 27). 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 26 on page 27 shows the internal interfaces to bi-directional Port 4.

Figure 25. Bi-directional Port 4 and Port 6



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**

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 13. 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)	–	–

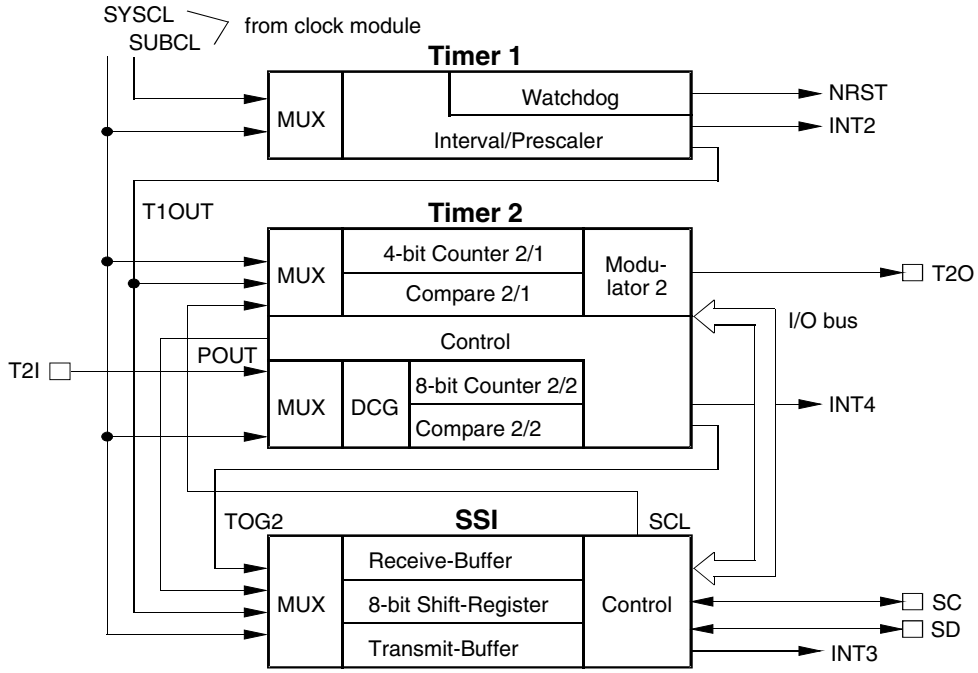
Universal Timer/Counter/ Communication Module (UTCM)

The Universal Timer/Counter/Communication Module (UTCM) consists of three timers (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 shift register for modulation and demodulation. The modulator units work together with the timers and shift the data bits into or out of the shift register.

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

Figure 26. UTCM Block Diagram



Timer 1

Timer 1 is an interval timer which can be used to generate periodic interrupts and as a prescaler for Timer 2, 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 a a 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 27. Timer 1 Module

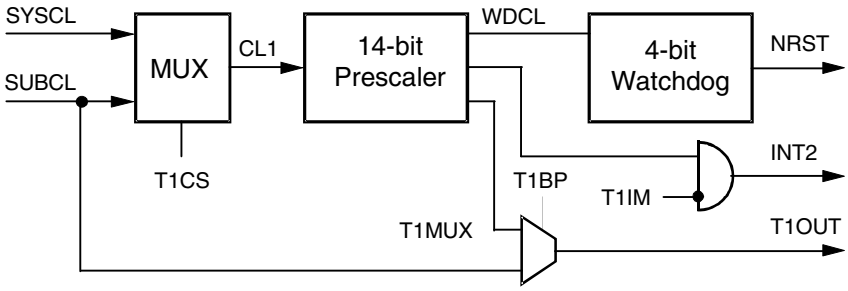
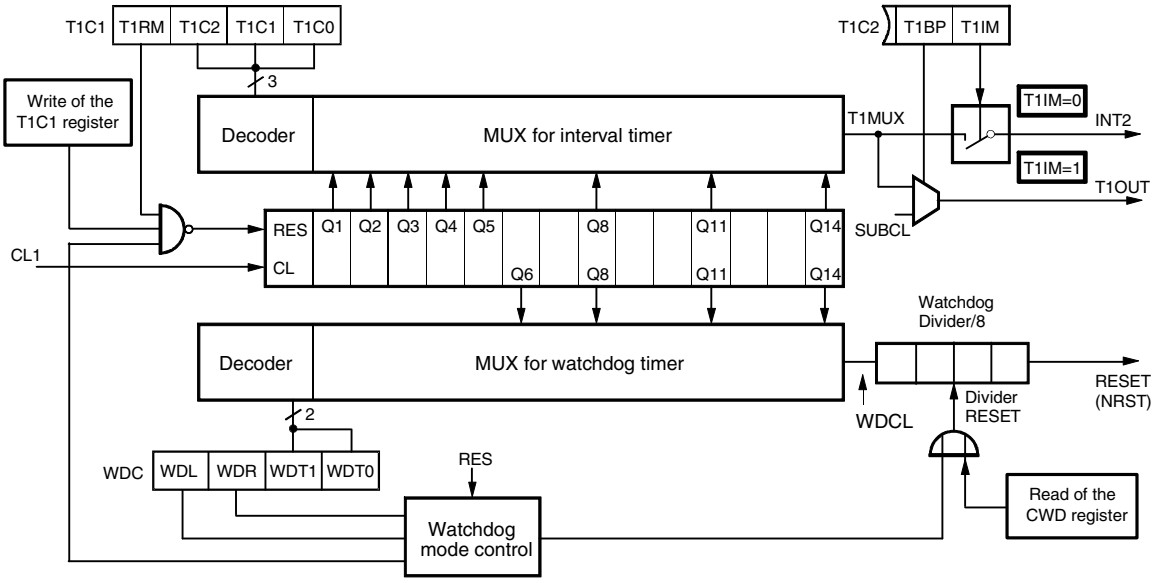


Figure 28. Timer 1 and Watchdog



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 T1RM = 0, write access without Timer 1 restart
T1RM = 1, write access with Timer 1 restart
Note: if WDL = 0, Timer 1 restart is impossible

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, the 32-kHz oscillator or via clock management. If the clock management generates the SUBCL, the selected input clock from the RC-oscillator, 4-MHz oscillator or an external clock is divided by 16.

Table 14. Timer 1 Control Bits

T1C2	T1C1	T1C0	Divider	Time Interval with SUBCL	Time Interval with SUBCL = 32 kHz	Time Interval with SYSCL = 2/1 MHz
0	0	0	2	SUBCL/2	61 µs	1 µs/2 µs
0	0	1	4	SUBCL/4	122 µs	2 µs/4 µs
0	1	0	8	SUBCL/8	244 µs	4 µs/8 µs
0	1	1	16	SUBCL/16	488 µs	8 µs/16 µs
1	0	0	32	SUBCL/32	0.977 ms	16 µs/32 µs
1	0	1	256	SUBCL/256	7.812 ms	128 µs/256 µs
1	1	0	2048	SUBCL/2048	62.5 ms	1024 µs/2048 µs
1	1	1	16384	SUBCL/16384	500 ms	8192 µs/16384 µs

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 **By****P**assed
T1BP = 1, TIOUT = T1MUX
T1BP = 0, T1OUT = SUBCL

T1CS **T**imer **1** input **C**lock **S**elect
T1CS = 1, CL1 = SUBCL (see Figure 28 on page 28)
T1CS = 0, CL1 = SYSCL (see Figure 28 on page 28)

T1IM **T**imer **1** **I**nterrupt **M**ask
T1IM = 1, disables Timer 1 interrupt
T1IM = 0, enables Timer 1 interrupt

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
WDL = 1, the watchdog can be enabled and disabled by using the WDR-bit
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
WDR = 1, the watchdog is stopped/disabled
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 15. Watchdog Time Control Bits

WDT1	WDT0	Divider	Delay Time to Reset with t_{in} = 1/32 kHz	Delay Time to Reset with t_{in} = 1/(2/1 MHz)
0	0	512	15.625 ms	0.256 ms/0.512 ms
0	1	2048	62.5 ms	1.024 ms/2.048 ms
1	0	16384	0.5 s	8.2 ms/16.4 ms
1	1	131072	4 s	65.5 ms/131 ms

Timer 2

Timer 2 is an 8-/12-bit timer used 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 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 12-bit timer or as 8-bit timer with 4-bit prescaler. The timer can also be configured as 8-bit timer and 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, 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 supply 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 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.

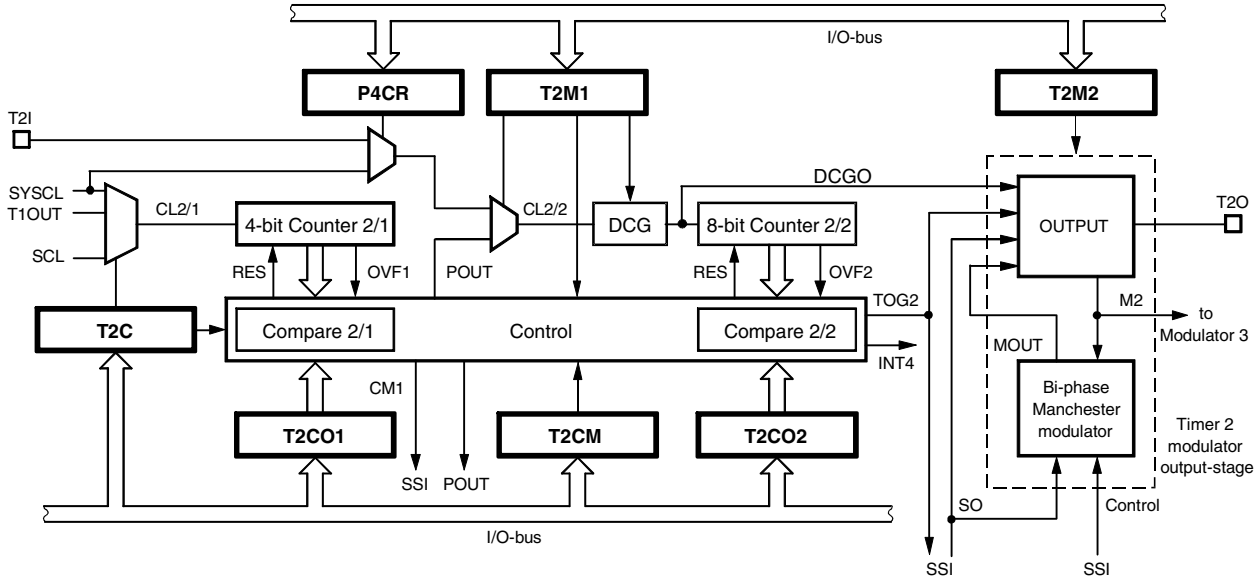
If the timer is used for carrier frequency modulation, the 4-bit stage works together with an additional 2-bit duty cycle generator like a 6-bit prescaler to generate carrier frequency and duty cycle. The 8-bit counter is used to enable and disable the modulator output for a programmable count of pulses.

The timer has a 4-bit and an 8-bit compare register for programming the time interval. 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 to function for various modes.

The 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	$0 \leq x \leq 4095$
For 8-bit compare data value:	n = y + 1	$0 \leq y \leq 255$
For 4-bit compare data value:	l = z + 1	$0 \leq z \leq 15$

Figure 29. Timer 2

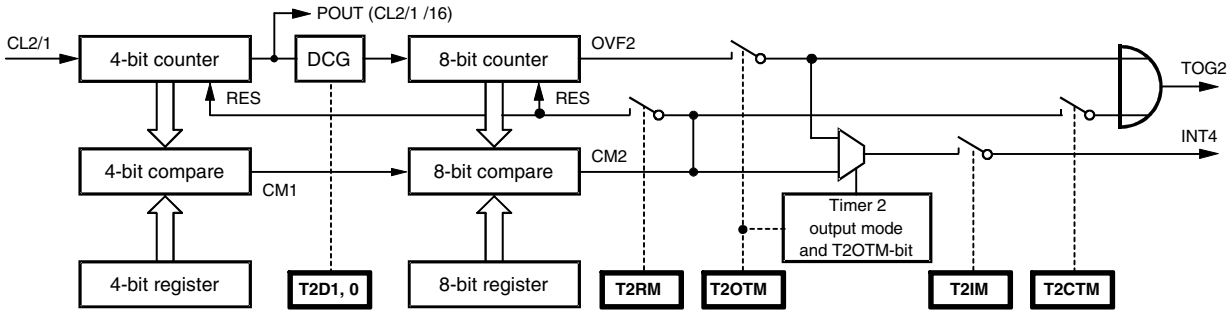


Timer 2 Modes

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.

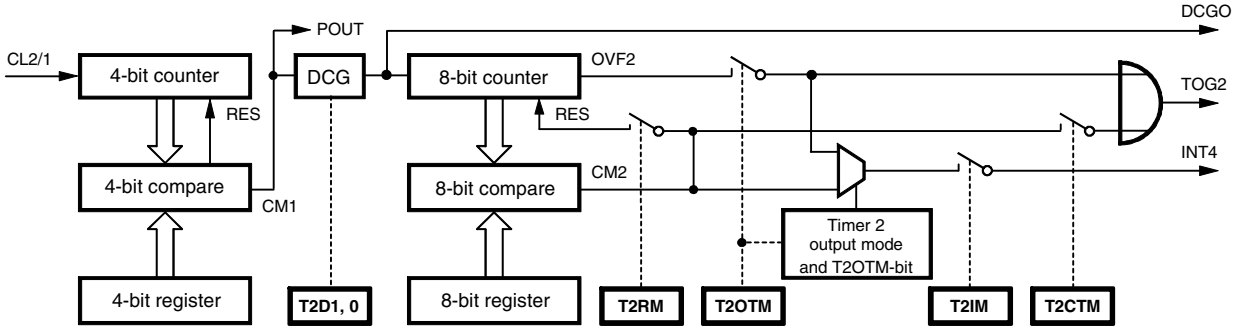
Figure 30. 12-bit Compare Counter



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.

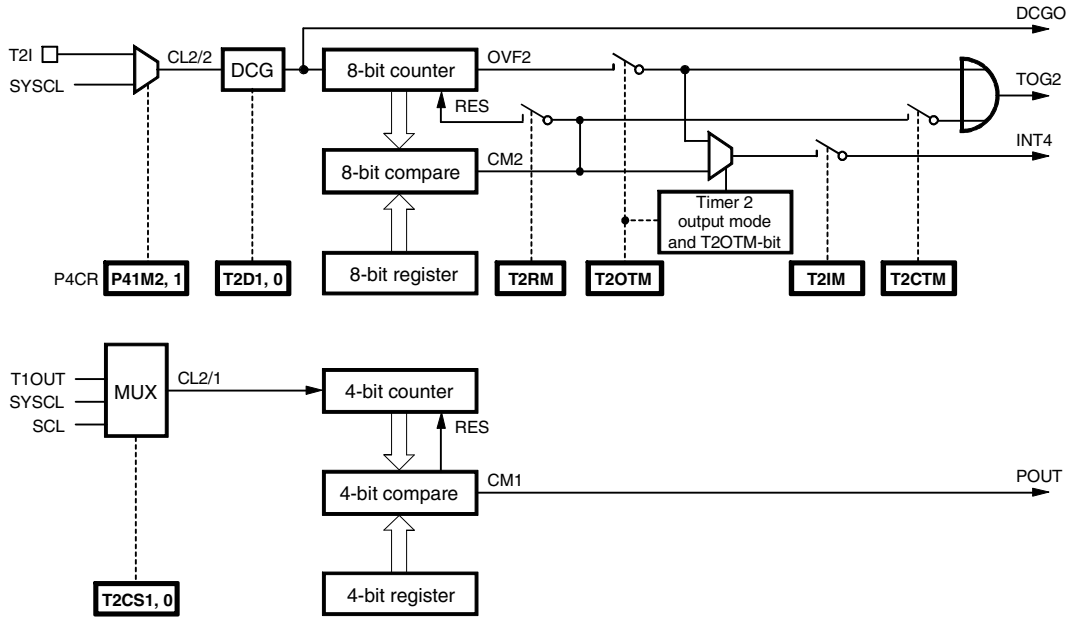
Figure 31. 8-bit Compare Counter



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 32. 4-/8-bit Compare Counter

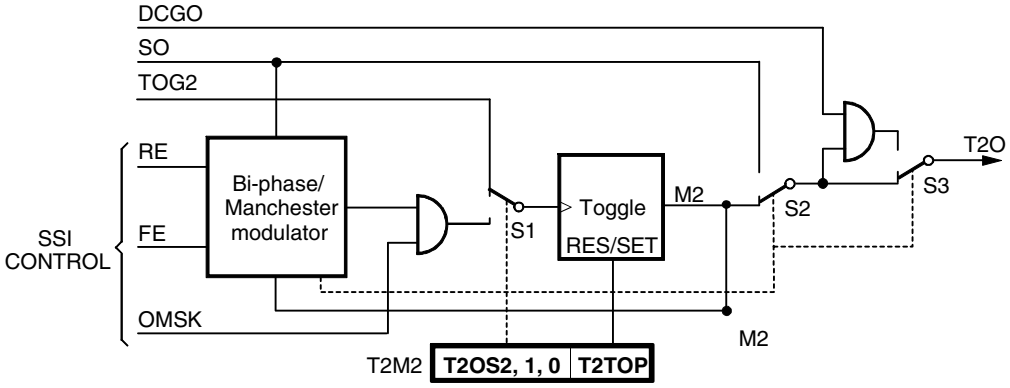


Timer 2 Output Modes

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 33. Timer 2 Modulator Output Stage

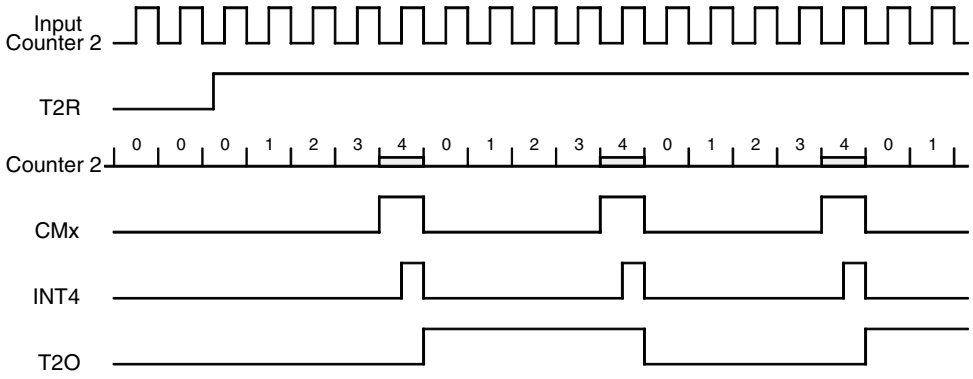


Timer 2 Output Signals

Timer 2 Output Mode 1

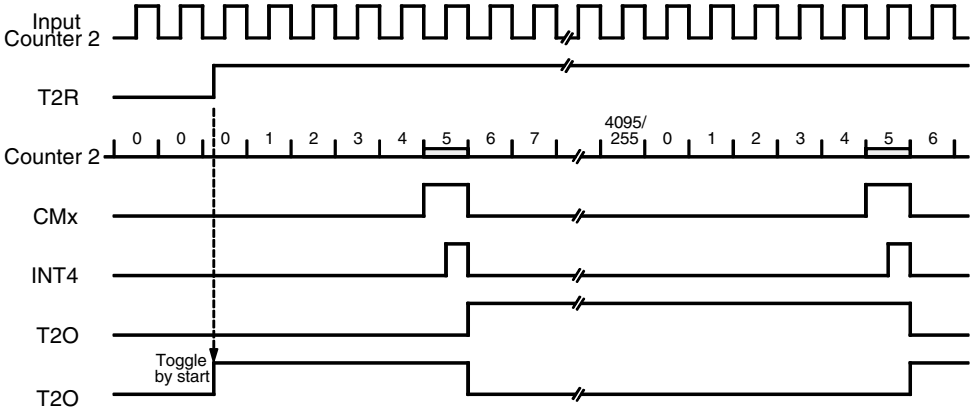
Toggle Mode A: A Timer 2 compare match toggles the output flip-flop (M2) -> T2O

Figure 34. Interrupt Timer/Square Wave Generator – Output Toggles with Each Edge Compare Match Event



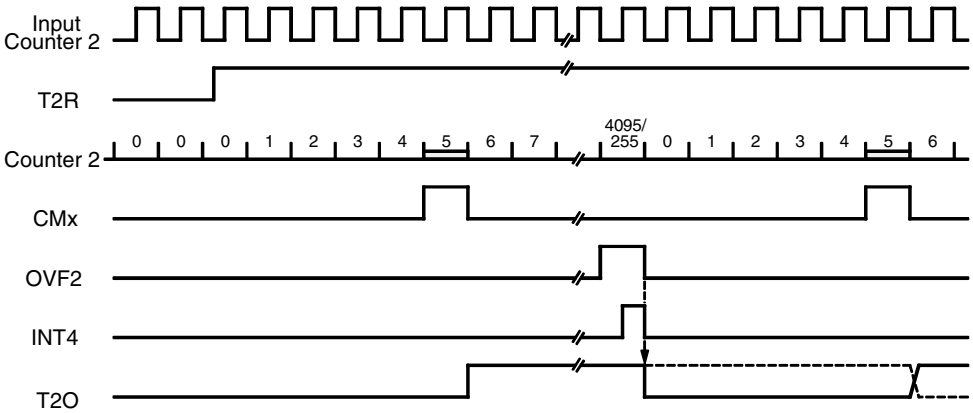
Toggle Mode B: A Timer 2 compare match toggles the output flip-flop (M2) -> T2O

Figure 35. Pulse Generator – Timer Output Toggles with the Timer Start if the T2TS-bit is Set



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

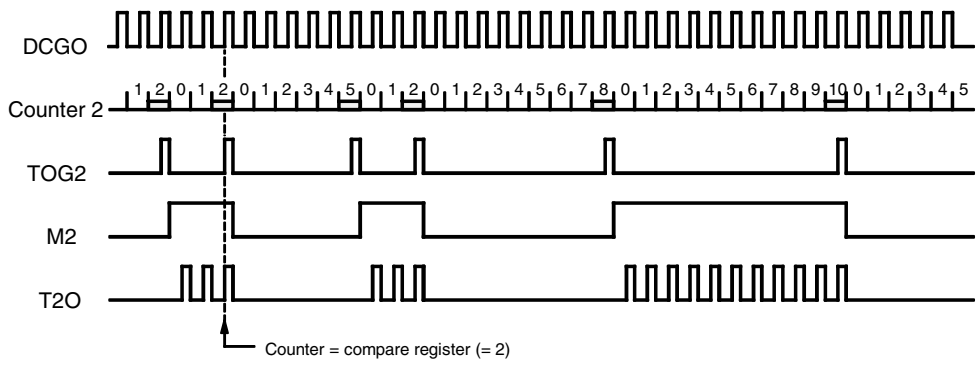
Figure 36. Pulse Generator – Timer Toggles with Timer Overflow and Compare Match



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).

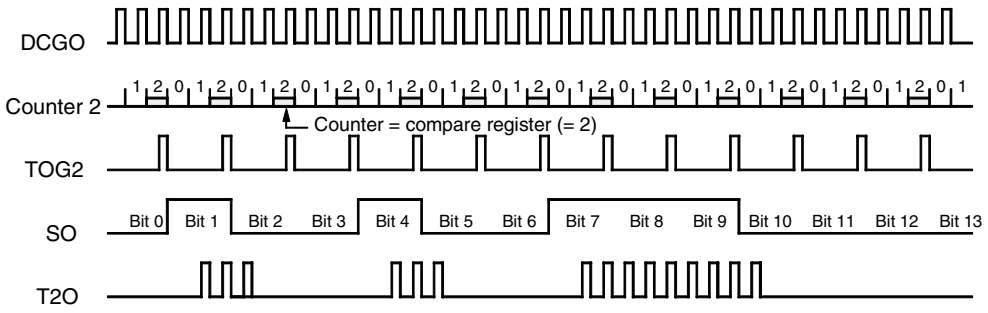
Figure 37. Carrier Frequency Burst Modulation with Timer 2 Toggle Flip-flop Output



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).

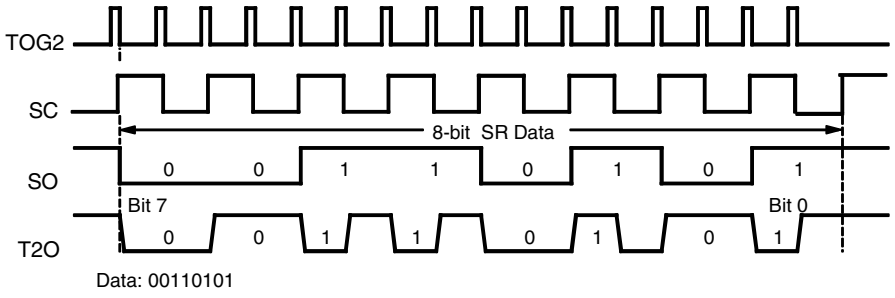
Figure 38. Carrier Frequency Burst Modulation with the SSI Data Output



Timer 2 Output Mode 4

Bi-phase Modulator: Timer 2 Modulates the SSI Internal Data Output (SO) to Bi-phase Code.

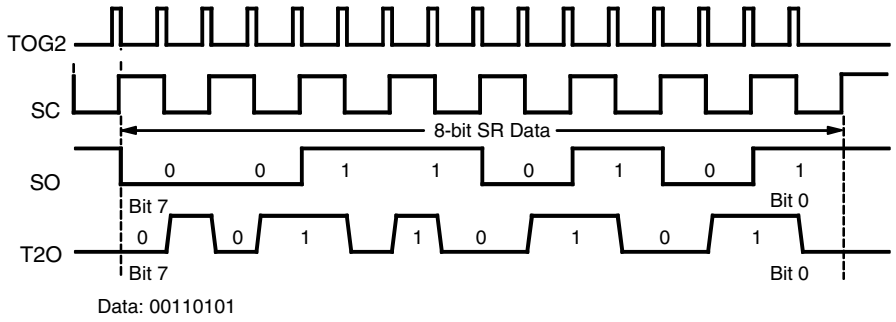
Figure 39. Bi-phase Modulation



Timer 2 Output Mode 5

Manchester Modulator: Timer 2 Modulates the SSI internal data output (SO) to Manchester code.

Figure 40. Manchester Modulation

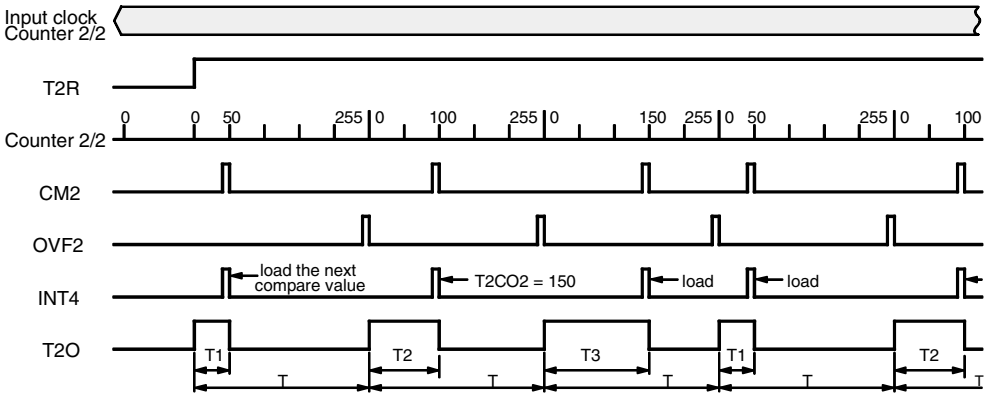


Timer 2 Output Mode 7

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 occurs 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 41. PWM Modulation



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 “Addressing Peripherals”. 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.

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 16. 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

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 17. 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
0	0	Duty cycle 1/3 (DCG03)	/4

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

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

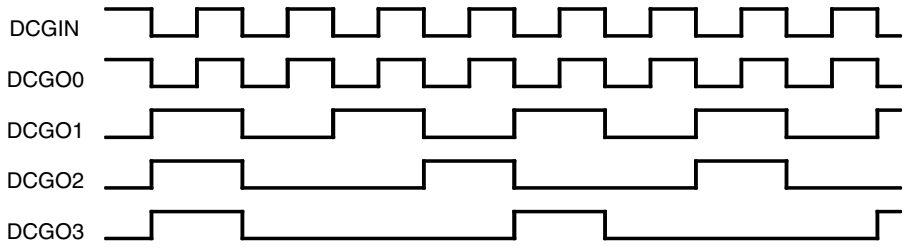
Table 18. Timer 2 Mode Select Bits

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

Duty Cycle Generator

The duty cycle generator generates duty cycles of 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 42. DCG Output Signals



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
This bit allows the programmer to preset the Timer 2 output T2O.
T2TOP = 0, resets the toggle outputs with the write cycle (M2 = 0)
T2TOP = 1, sets toggle outputs with the write cycle (M2 = 1)
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 19. 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

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

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- and 8-bit compare register. When assigned to the compare register a compare event will be suppressed.

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 20. 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)

Timer 2 COmpare Register 1 (T2CO1)

Address: '7'hex -Subaddress: '4'hex

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

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

Timer 2 COmpare Register 2 (T2CO2) Byte Write

Address: '7'hex - Subaddress: '5'hex

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

Synchronous Serial Interface (SSI)

SSI Features

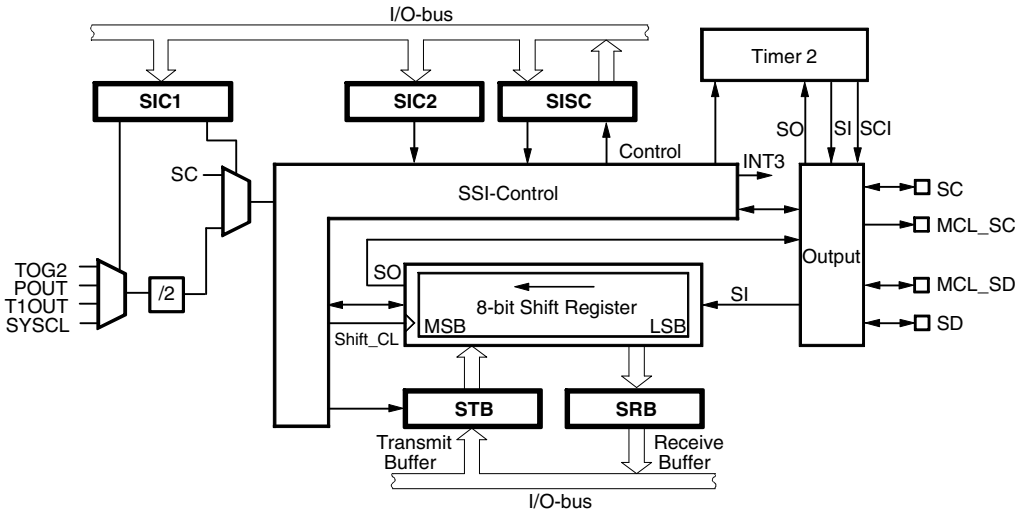
- 2- and 3-wire NRZ
- 2-wire mode, additional internal 2-wire link for multi-chip packaging solutions
- With Timer 2
  - Bi-phase modulation
  - Manchester modulation
  - Pulse-width demodulation
  - Burst modulation

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 (BP4),a 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 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 demodulation functions (see section “Timer”). 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.
4. Multi-chip link (MCL) – the SSI can also be used as an interchip data interface for use in single package multi-chip modules or hybrids. For such applications, the SSI is provided with two dedicated pads (MCL_SD and MCL_SC) which act as a two-wire chip-to-chip link. The MCL can be activated by the MCL control bit. Should these MCL pads be used by the SSI, the standard SD and SC pins are not required and the corresponding Port 4 ports are available as conventional data ports.

Figure 43. Block Diagram of the Synchronous Serial Interface



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 accessable 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) from one of several on-chip clock sources or it can 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”).

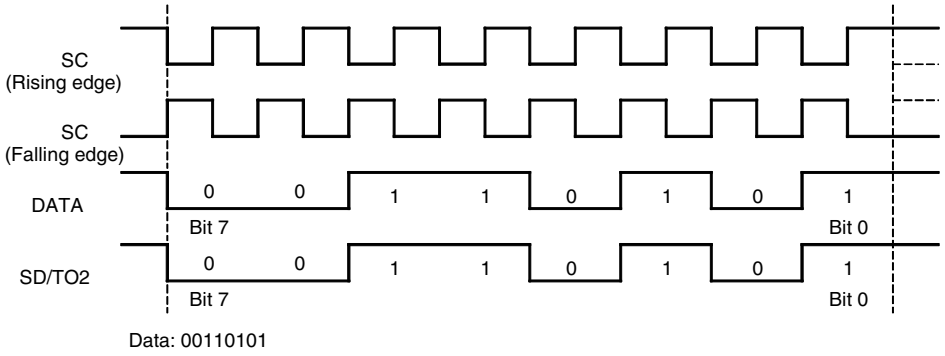
At the beginning of every telegram, the SSI control loads the transmit buffer into the shift register and proceeds immediately to shift data serially out. At the same time, incoming data is shifted into the shift register input. This incoming data is automatically loaded into the receive buffer when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

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.

8-bit Synchronous Mode

Figure 44. 8-bit Synchronous Mode



In the 8-bit synchronous mode, the SSI can operate as either a 2- or 3-wire interface (see section "SSI Peripheral Configuration"). 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 45. Example of 8-bit Synchronous Transmit Operation

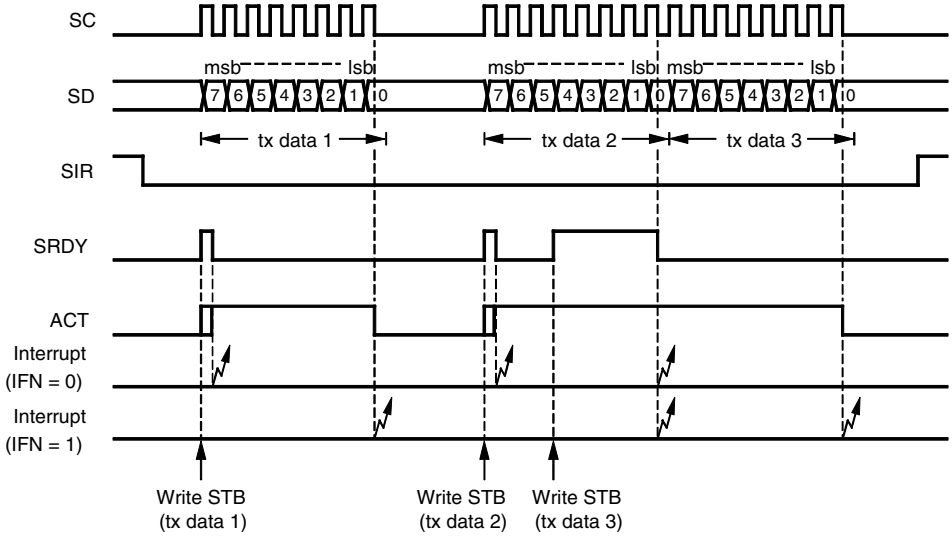
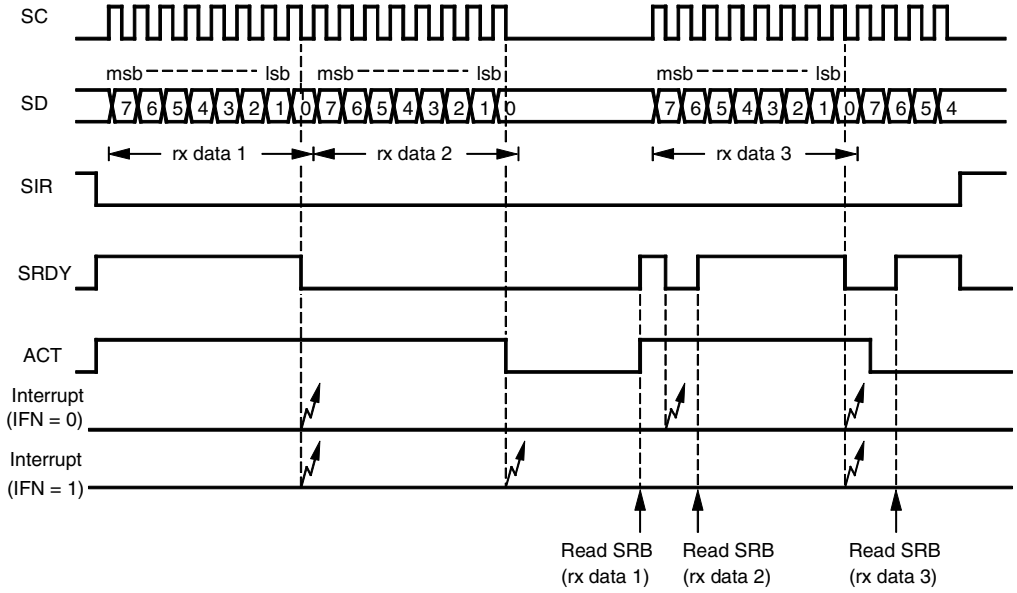


Figure 46. Example of 8-bit Synchronous Receive Operation



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. In 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.

Once started, a 9-bit telegram 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 mid telegram, the SSI will complete the current transfer and terminate the dialog with an MCL stop condition.

Figure 47. Example of MCL Transmit Dialog

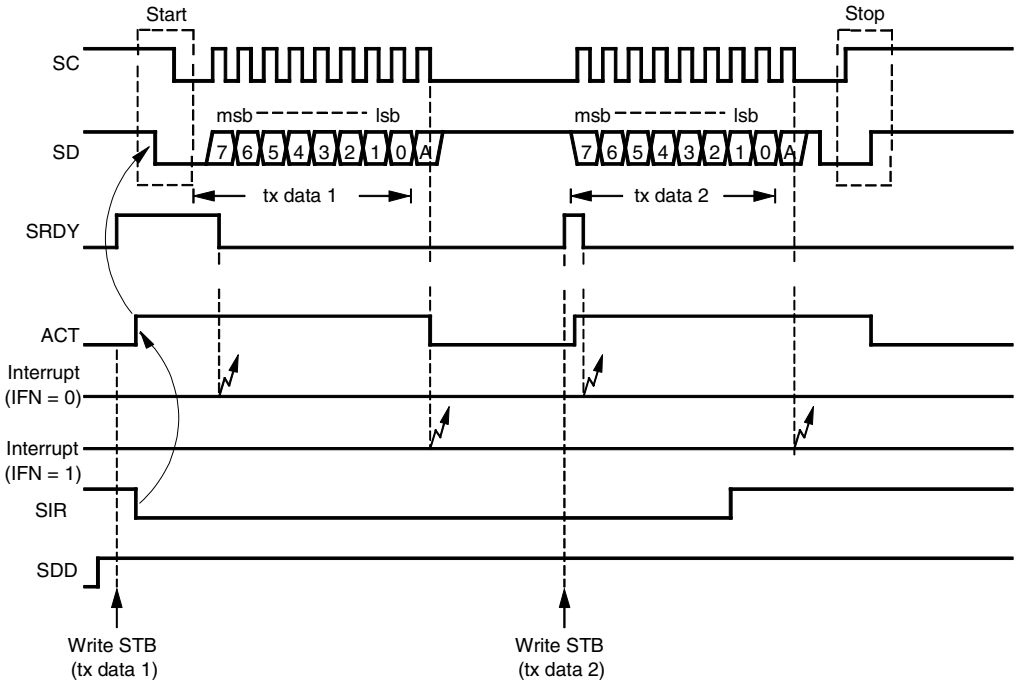
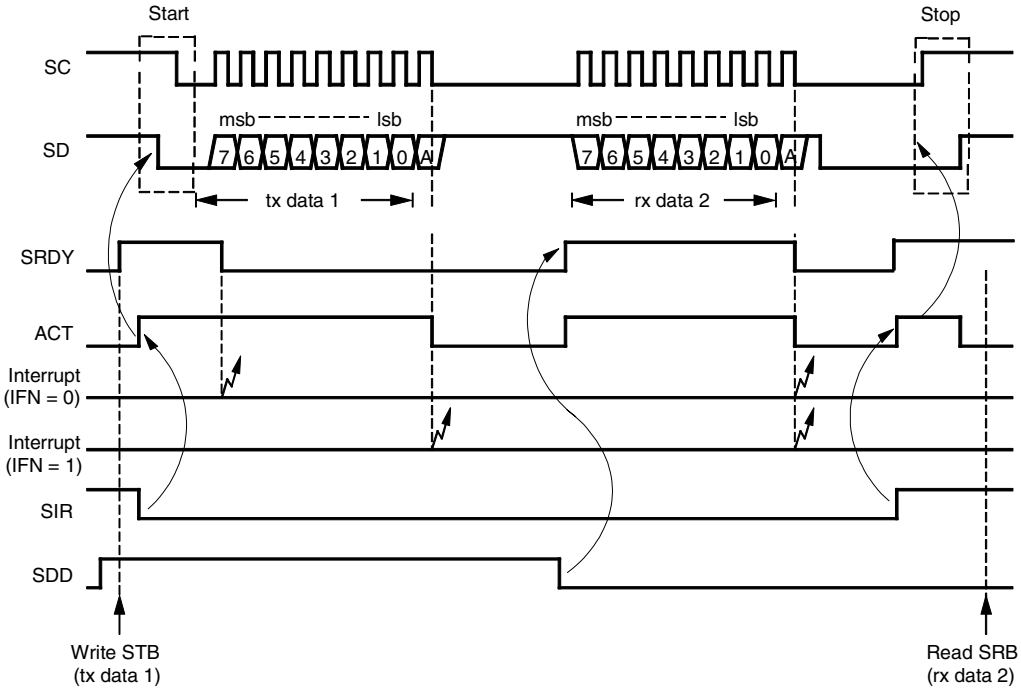


Figure 48. Example of MCL Receive Dialog



8-bit Pseudo MCL Mode

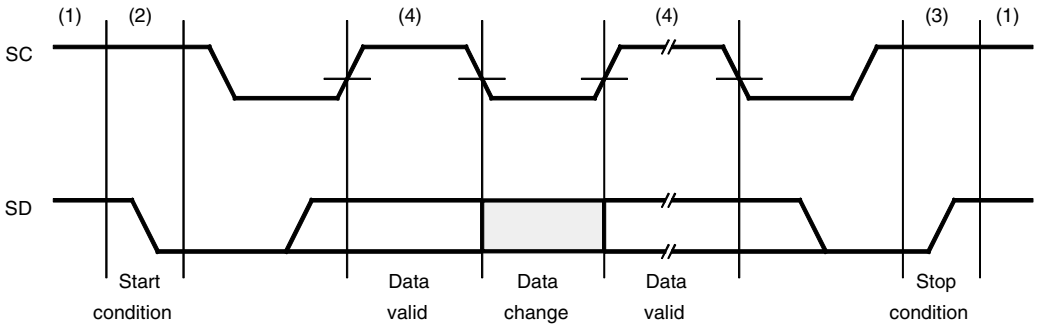
In this mode, the SSI exhibits all the typical MCL operational features except for the acknowledge-bit which is never expected or transmitted.

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 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 49. MCL Bus Protocol 1



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

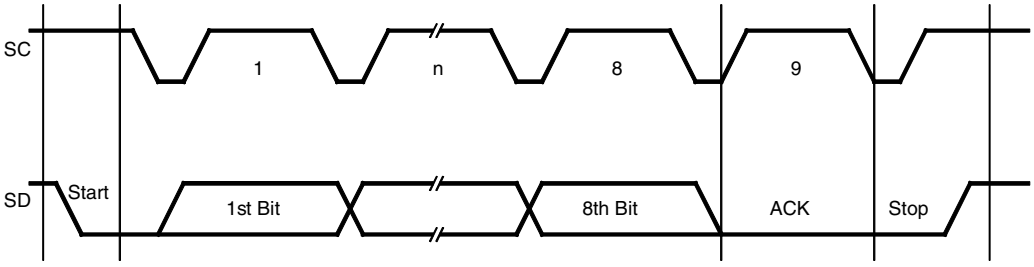
Start data transfer (2) A HIGH to LOW transition of the SD line while the clock (SC) is HIGH defines a START condition

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 a 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 50. MCL Bus Protocol 2



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), the end of a 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.

Modulation

If the shift register is used together with Timer 2 for modulation, 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 and demodulation stages, if enabled, operate as soon as the SSI is activated (SIR = 0) and cease 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, 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.

Internal 2-wire Multi-chip Link

Two additional on-chip pads (MCL_SC and MCL_SD) for the SC and the SD line can be used as chip-to-chip link for multi-chip applications. These pads can be activated by setting the MCL-bit in the SISC register.

Figure 51. Multi-chip Link

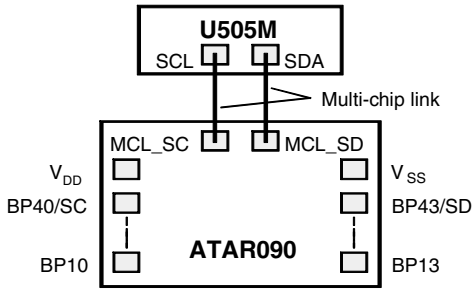
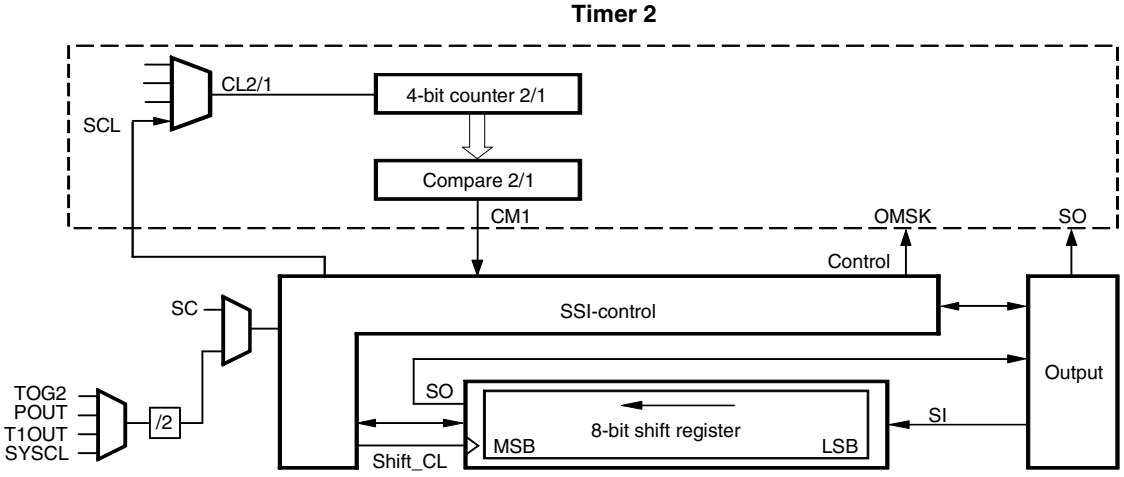


Figure 52. SSI Output Masking Function



Serial Interface Registers

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 21. 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.

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 bits.

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

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

Table 22. 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 mode
4	0	0	8-bit two-wire pseudo MCL mode (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

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

Serial Interface Status and Control Register (SISC)

Primary register address: 'A'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
Write	**MCL**	**RACK**	**SIM**	**IFN**	**Reset value: 1111b**
Read	**–**	**TACK**	**ACT**	**SRDY**	**Reset value: xxxxb**

MCL **M**ulti-**C**hip **L**ink activation
MCL = 1, multi-chip link disabled. **This bit has to be set to 0 during transactions to/from the EEPROM of the ATAR890**
MCL = 0, connects SC and SD additionally to the internal multi-chip link pads

RACK **R**eceive **ACK**nowledge status/control bit for MCL mode
RACK = 0, transmit acknowledge in next receive telegram
RACK = 1, transmit no acknowledge in last receive telegram

TACK **T**ransmit **ACK**nowledge status/control bit for MCL mode
TACK = 0, acknowledge received in last transmit telegram
TACK = 1, no acknowledge received in last transmit telegram

SIM **S**erial **I**nterrupt **M**ask
SIM = 1, disable interrupts
SIM = 0, enable serial interrupt. An interrupt is generated.

IFN **I**nterrupt **F**u**N**ction
IFN = 1, the serial interrupt is generated at the end of the telegram
IFN = 0, the serial interrupt is generated when the SRDY goes low (i.e., buffer becomes empty/full in transmit/receive mode)

SRDY **S**erial interface buffer **R**ea**DY** status flag
SRDY = 1, in receive mode: receive buffer empty
in transmit mode: transmit buffer full
SRDY = 0, in receive mode: receive buffer full
in transmit mode: transmit buffer empty

ACT Transmission **ACT**ive status flag
ACT = 1, transmission is active, i.e., serial data transfer. Stop or start conditions are currently in progress.
ACT = 0, transmission is inactive

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.

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**

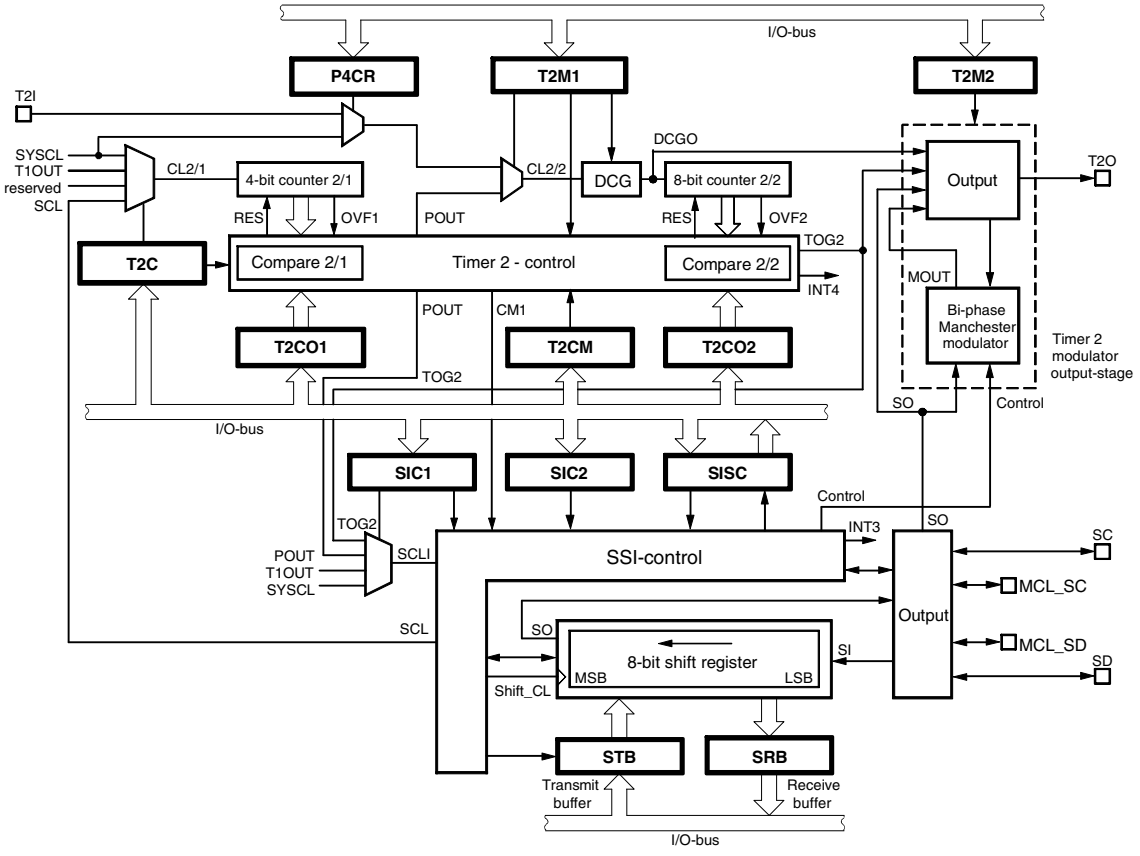
Combination Modes

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 and demodulation. The modulator and demodulator units work together with the timers and shift the data bits into or out of the shift register.

Combination Mode Timer 2 and SSI

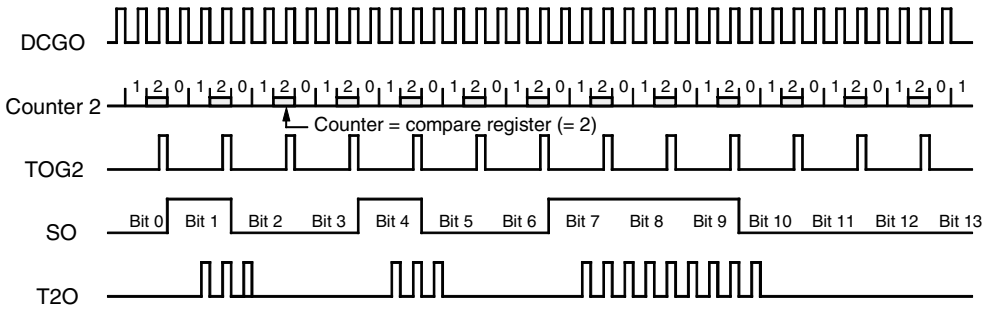
Figure 53. Combination Timer 2 and SSI



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

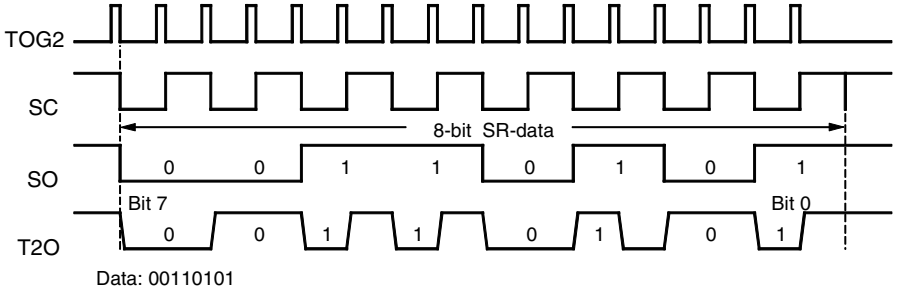
Figure 54. Carrier Frequency Burst Modulation with the SSI Internal Data Output



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:	Modulator 2 of Timer 2 modulates the SSI internal data output to Bi-phase code

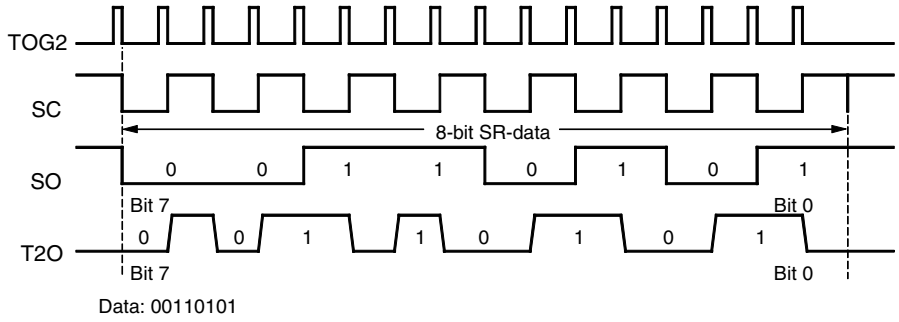
Figure 55. Bi-phase Modulation 1



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:	Modulator 2 of Timer 2 modulates the SSI internal data output to Manchester code

Figure 56. Manchester Modulation 1

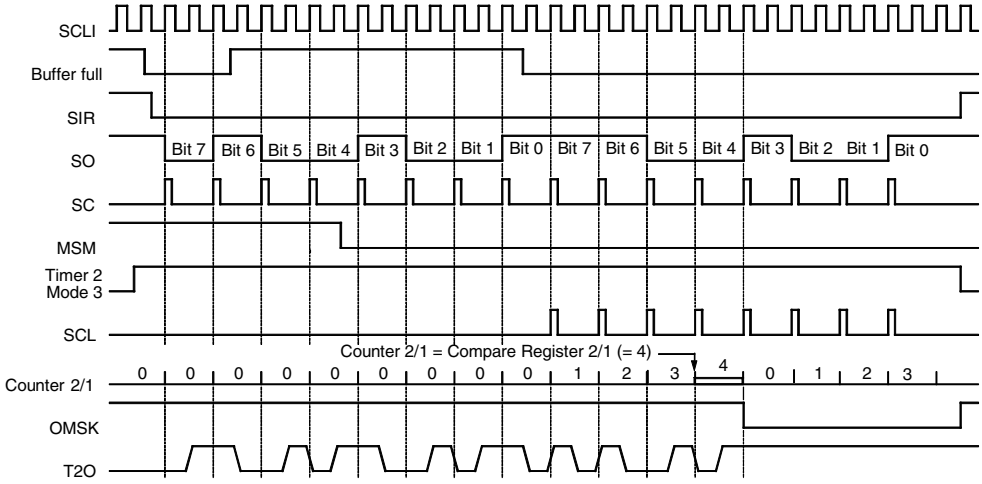


Combination Mode 4: Manchester Modulation 2

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 5:	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 a stop signal for the modulator. Figure 57 shows an example for a 12-bit Manchester telegram.

Figure 57. Manchester Modulation 2

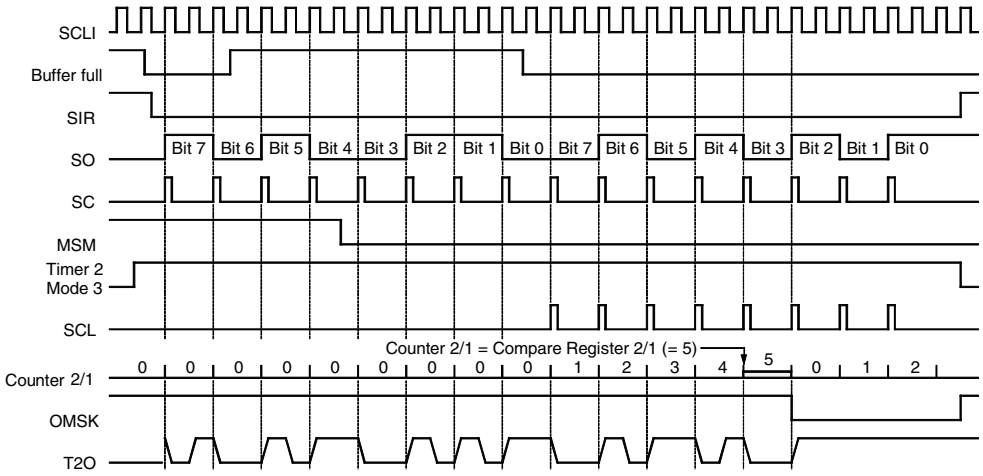


Combination Mode 5: Bi-phase Modulation 2

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 4:	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 58 shows an example for a 13-bit Bi-phase telegram.

Figure 58. Bi-phase Modulation 2



ATAR890

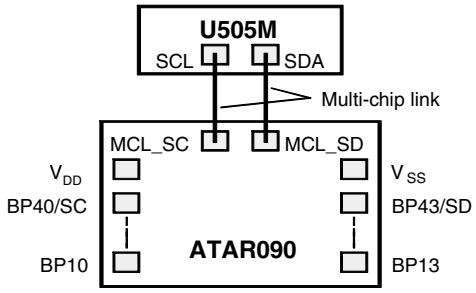
The ATAR890 is a multichip device which offers a combination of a MARC4-based microcontroller and a serial E2PROM data memory in a single package. The ATAR090 is used as a microcontroller and the U505M is used as a serial E2PROM. Two internal lines can be used as chip-to-chip link in a single package. The maximum internal data communication frequency between the ATAR090 and the U505M over the chip link (MCL_SC and MCL_SD) is f_{SC_MCL} = 500 kHz.

The microcontroller and the EEPROM portions of this multi-chip device are equivalent to their respective individual component chips, except for the electrical specification.

Internal 2-wire Multi-chip Link

Two additional on-chip pads (MCL_SC and MCL_SD) for the SC and the SD line can be used as chip-to-chip link for multi-chip applications. These pads can be activated by setting the MCL-bit in the SISC register.

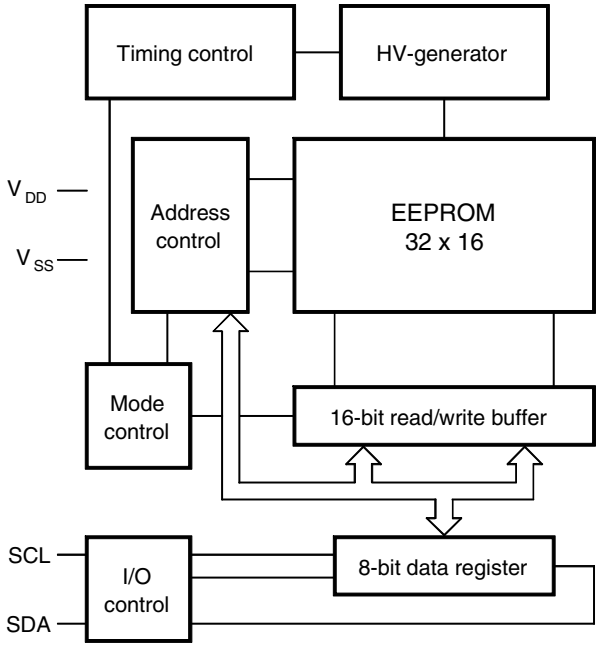
Figure 59. Multi-chip Link



U505M EEPROM

The U505M is a 512-bit EEPROM internally organized as 32 x 16 bits. The programming voltage as well as the write-cycle timing is generated on-chip. The U505M features a serial interface allowing operation on a simple two-wire bus with an MCL protocol. Its low power consumption makes it well suited for battery applications.

Figure 60. Block Diagram EEPROM



Serial Interface

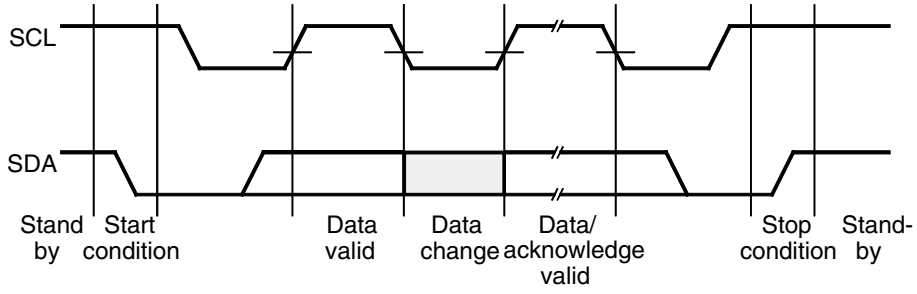
The U505M has a two-wire serial interface to the microcontroller for read and write accesses to the EEPROM. The U505M is considered to be a slave in all these applications. That means, the controller has to be the master that initiates the data transfer and provides the clock for transmit and receive operations.

The serial interface is controlled by the ATAR890 microcontroller which generates the serial clock and controls the access via the SCL-line and SDA-line. SCL is used to clock the data into and out of the device. SDA is a bi-directional line that is used to transfer data into and out of the device. The following serial protocol is used for the data transfers.

Serial Protocol

- Data states on the SDA-line change only while SCL is low.
- Changes on the SDA-line while SCL is high are interpreted as START or STOP condition.
- A START condition is defined as a high to low transition on the SDA-line while the SCL-line is high.
- A STOP condition is defined as a low to high transition on the SDA-line while the SCL-line is high.
- Each data transfer must be initialized with a START condition and terminated with a STOP condition. The START condition wakes the device from standby mode and the STOP condition returns the device to standby mode.
- A receiving device generates an acknowledge (A) after the reception of each byte. This requires an additional clock pulse, generated by the master. If the reception was successful the receiving master or slave device pulls down the SDA-line during that clock cycle. If an acknowledge is not detected (N) by the interface in transmit mode, it will terminate further data transmissions and go into receive mode. A master device must finish its read operation by a non-acknowledge and then send a stop condition to bring the device into a known state.

Figure 61. MCL Protocol



- Before the START condition and after the STOP condition the device is in standby mode and the SDA line is switched as an input with a pull-up resistor.
- The control byte that follows the START condition determines the following operation. It consists of the 5-bit row address, 2 mode control bits and the READ/ NWRITE bit that is used to control the direction of the following transfer. A ‘0’ defines a write access and a ‘1’ a read access.
- Control byte format

	EEPROM Address					Mode Control Bits		Read/ NWrite	
Start	A4	A3	A2	A1	A0	C1	C0	R/NW	Ackn

- Control byte format

Start	Control byte	Ackn	Data byte	Ackn	Data byte	Ackn	Stop

EEPROM

The EEPROM has a size of 512 bits and is organized as 32 x 16-bit matrix. To read and write data to and from the EEPROM the serial interface must be used. The interface supports one and two byte write accesses and one to n-byte read accesses to the EEPROM.

EEPROM – Operating Modes

The operating modes of the EEPROM are defined via the control byte. The control byte contains the row address, the mode control bits and the read/not-write bit that is used to control the direction of the following transfer. A ‘0’ defines a write access and a ‘1’ a read access. The five address bits select one of the 32 rows of the EEPROM memory to be accessed. For all accesses the complete 16-bit word of the selected row is loaded into a buffer. The buffer must be read or overwritten via the serial interface. The two mode control bits C_1 and C_2 define in which order the accesses to the buffer are performed: High byte – low byte or low byte – high byte. The EEPROM also supports auto-increment and auto-decrement read operations. After sending the start address with the corresponding mode, consecutive memory cells can be read row by row without transmission of the row addresses.

Two special control bytes enable the complete initialization of EEPROM with ‘0’ or with ‘1’.

Write Operations

The EEPROM permits 8-bit and 16-bit write operations. A write access starts with the START condition followed by a write control byte and one or two data bytes from the master. It is completed via the STOP condition from the master after the acknowledge cycle.

The programming cycle consists of an erase cycle (write ‘zeros’) and the write cycle (write ‘ones’). Both cycles together take about 10 ms.

Acknowledge Polling

If the EEPROM is busy with an internal write cycle, all inputs are disabled and the EEPROM will not acknowledge until the write cycle is finished. This can be used to detect the end of the write cycle. The master must perform acknowledge polling by sending a start condition followed by the control byte. If the device is still busy with the write cycle, it will not return an acknowledge and the master has to generate a stop condition or perform further acknowledge polling sequences. If the cycle is complete, it returns an acknowledge and the master can proceed with the next read or write cycle.

Write One Data Byte

Start	Control byte	A	Data byte 1	A	Stop

Write Two Data Bytes

Start	Control byte	A	Data byte 1	A	Data byte 2	A	Stop

Write Control Byte Only

Start	Control byte	A	Stop

Write Control Bytes

	MSB							LSB
Write low byte first	A4	A3	A2	A1	A0	C1	C0	R/NW
	Row address					0	1	0

Byte order	LB(R)	HB(R)

	MSB							LSB
Write high byte first	A4	A3	A2	A1	A0	C1	C0	R/NW
	Row address					1	0	0

Byte order	HB(R)	LB(R)

A: acknowledge; HB: high byte; LB: low byte; R: row address

Read Operations

The EEPROM allows byte, word and current address read operations. The read operations are initiated in the same way as write operations. Every read access is initiated by sending the START condition followed by the control byte which contains the address and the read mode. When the device has received a read command, it returns an acknowledge, loads the addressed word into the read/write buffer and sends the selected data byte to the master. The master has to acknowledge the received byte if it wants to proceed with the read operation. If two bytes are read out from the buffer the device increments respectively decrements the word address automatically and loads the buffer with the next word. The read mode bits determines if the low or high byte is read first from the buffer and if the word address is incremented or decremented for the next read access. If the memory address limit is reached, the data word address will 'roll over' and the sequential read will continue. The master can terminate the read operation after every byte by not responding with an acknowledge (N) and by issuing a stop condition.

Read One Data Byte

Start	Control byte	A	Data byte 1	N	Stop

Read Two Data Bytes

Start	Control byte	A	Data byte 1	A	Data byte 2	N	Stop

Read n Data Bytes

Start	Control byte	A	Data byte 1	A	Data byte 2	A	- - -	Data byte n	N	Stop

Read Control Bytes

Read low byte first, address increment

MSB							LSB
A4	A3	A2	A1	A0	C1	C0	R/NW
Row address					0	1	1

Byte order	LB(R)	HB(R)	LB(R+1)	HB(R+1)	- - -	LB(R+n)	HB(R+n)

Read high byte first, address decrement

MSB							LSB
A4	A3	A2	A1	A0	C1	C0	R/NW
Row address					1	0	1

Byte order	HB(R)	LB(R)	HB(R-1)	LB(R-1)	- - -	HB(R-n)	LB(R-n)

A: acknowledge, N: no acknowledge; HB: high byte; LB: low byte, R: row address

Initialization After a Reset Condition

The EEPROM with the serial interface has its own reset circuitry. In systems with microcontrollers that have their own reset circuitry for power-on reset, watchdog reset or brown-out reset, it may be necessary to bring the U505M into a known state independent of its internal reset. This is performed by writing to the serial interface.

Start	Control byte	A	Data byte 1	N	Stop

If the U505M acknowledges this sequence it is in a defined state. It may be necessary to perform this sequence twice.

Absolute Maximum Ratings

Voltages are given relative to V_{SS}.

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}).

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
Soldering temperature (t ≤10 s)	T_{sld}	260	°C

Thermal Resistance

Parameter	Symbol	Value	Unit
Thermal resistance (SSO20)	R_{thJA}	140	K/W

DC Operating Characteristics

V_{SS} = 0 V, T_{amb} = -40 to 85°C unless otherwise specified

Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
Power Supply						
Operating voltage at V_{DD}		V_{DD}	V_{POR}		6.5	V
Active current CPU active	f_{SYSCL} = 1 MHz V_{DD} = 1.8 V V_{DD} = 3.0 V V_{DD} = 6.5 V	I_{DD}		150 220 600	350	µA µA µA
Power down current (CPU sleep, RC oscillator active, 4-MHz quartz oscillator active)	f_{SYSCL} = 1 MHz V_{DD} = 1.8 V V_{DD} = 3.0 V V_{DD} = 6.5 V	I_{PD}		30 50 150	100	µA µA µA
Sleep current (CPU sleep, 32-kHz quartz oscillator active 4-MHz quartz oscillator inactive)	V_{DD} = 1.8 V V_{DD} = 3.0 V V_{DD} = 6.5 V	I_{Sleep}		0.4 0.6 0.8	1.3 1.8	µA µA µA
Sleep current (CPU sleep, 32-kHz quartz oscillator inactive 4-MHz quartz oscillator inactive)	V_{DD} = 1.8 V for ATAR090 V_{DD} = 3.0 V for ATAR090 V_{DD} = 6.5 V for ATAR090 V_{DD} = 6.5 V for ATAR890	I_{Sleep}		0.1 0.3 0.5 0.6	0.5 0.8 1.0	µA µA µA µA
Pin capacitance	Any pin to V_{SS}	C_L		7	10	pF

DC Operating Characteristics (Continued)

V_{SS} = 0 V, T_{amb} = -40 to 85°C unless otherwise specified

Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
Power-on Reset Threshold Voltage						
POR threshold voltage	BOT = 1	V_{POR}	1.6	1.7	1.8	V
POR threshold voltage	BOT = 0	V_{POR}	1.75	1.9	2.05	V
POR hysteresis		V_{POR}		50		mV
Voltage Monitor Threshold Voltage						
VM high threshold voltage	V_{DD} > VM, VMS = 1	V_{MThh}		3.0	3.25	V
VM high threshold voltage	V_{DD} < VM, VMS = 0	V_{MThh}	2.8	3.0		V
VM middle threshold voltage	V_{DD} > VM, VMS = 1	V_{MThm}		2.6	2.8	V
VM middle threshold voltage	V_{DD} < VM, VMS = 0	V_{MThm}	2.4	2.6		V
VM low threshold voltage	V_{DD} > VM, VMS = 1	V_{MThl}		2.2	2.4	V
VM low threshold voltage	V_{DD} < VM, VMS = 0	V_{MThl}	2.0	2.2		V
External Input Voltage						
VMI	V_{VMI} > VBG, VMS = 1	V_{VMI}		1.3	1.4	V
VMI	V_{VMI} > VBG, VMS = 0	V_{VMI}	1.2	1.3		V
All Bi-directional Ports						
Input voltage LOW	V_{DD} = 1.8 to 6.5 V	V_{IL}	V_{SS}		$0.2 \times V_{DD}$	V
Input voltage HIGH	V_{DD} = 1.8 to 6.5 V	V_{IH}	$0.8 \times V_{DD}$		V_{DD}	V
Input LOW current (switched pull-up)	V_{DD} = 2.0 V, V_{DD} = 3.0 V, V_{IL}= V_{SS} V_{DD} = 6.5 V	I_{IL}	-2 -10 -50	-4 -20 -100	-12 -40 -200	µA µA µA
Input HIGH current (switched pull-down)	V_{DD} = 2.0 V, V_{DD} = 3.0 V, V_{IH} = V_{DD} V_{DD} = 6.5 V	I_{IH}	2 10 50	4 20 100	12 40 200	µA µA µA
Input LOW current (static pull-up)	V_{DD} = 2.0 V V_{DD} = 3.0 V, V_{IL}= V_{SS} V_{DD} = 6.5 V	I_{IL}	-20 -80 -300	-50 -160 -600	-100 -320 -1200	µA µA µA
Input LOW current (static pull-down)	V_{DD} = 2.0 V V_{DD} = 3.0 V, V_{IH}= V_{DD} V_{DD} = 6.5 V	I_{IH}	20 80 300	50 160 600	100 320 1200	µA µA µA
Input leakage current	V_{IL}= V_{SS}	I_{IL}			100	nA
Input leakage current	V_{IH}= V_{DD}	I_{IH}			100	nA
Output LOW current	$V_{OL} = 0.2 \times V_{DD}$ V_{DD} = 2.0 V V_{DD} = 3.0 V, V_{DD} = 6.5 V	I_{OL}	0.6 3 8	1.2 5 15	2.5 8 22	mA mA mA
Output HIGH current	$V_{OH} = 0.8 \times V_{DD}$ V_{DD} = 2.0 V V_{DD} = 3.0 V, V_{DD} = 6.5 V	I_{OH}	-0.6 -3 -8	-1.2 -5 -16	-2.5 -8 -24	mA mA mA

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

AC Characteristics

Supply voltage V_{DD} = 1.8 to 6.5 V, V_{SS} = 0 V, T_{amb} = 25°C unless otherwise specified.

Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
Operation Cycle Time						
System clock cycle	V_{DD} = 1.8 to 6.5 V T_{amb} = -40 to 85°C	t_{SYSCL}	500		2000	ns
	V_{DD} = 2.4 to 6.5 V T_{amb} = -40 to 85°C	t_{SYSCL}	250		2000	ns
Timer 2 Input Timing Pin T2I						
Timer 2 input clock		f_{T2I}			5	MHz
Timer 2 input LOW time	Rise/fall time < 10 ns	t_{T2IL}	100			ns
Timer 2 input HIGH time	Rise/fall time < 10 ns	t_{T2IH}	100			ns
Interrupt Request Input Timing						
Interrupt request LOW time	Rise/fall time < 10 ns	t_{IRL}	100			ns
Interrupt request HIGH time	Rise/fall time < 10 ns	t_{IRH}	100			ns
External System Clock						
EXSCL at OSC1, ECM = EN	Rise/fall time < 10 ns	f_{EXSCL}	0.5		4	MHz
EXSCL at OSC1, ECM = DI	Rise/fall time < 10 ns	f_{EXSCL}	0.02		4	MHz
Input HIGH time	Rise/fall time < 10 ns	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}		3.8		MHz
Stability	V_{DD} = 2.0 to 6.5 V T_{amb} = -40 to 85°C	Δf/f			±50	%
RC Oscillator 2 – External Resistor						
Frequency	R_{ext} = 170 kΩ	f_{RcOut2}		4		MHz
Stability	V_{DD} = 2.0 to 6.5 V T_{amb} = -40 to 85°C	Δf/f			±15	%
Stabilization time		t_S			10	µs
4-MHz Crystal Oscillator (Operating Range V_{DD} = 2.2 V to 6.5 V)						
Frequency		f_X		4		MHz
Start-up time		t_{SQ}		5		ms
Stability		Δf/f	-10		10	ppm

AC Characteristics (Continued)

Supply voltage V_{DD} = 1.8 to 6.5 V, V_{SS} = 0 V, T_{amb} = 25°C unless otherwise specified.

Parameters	Test Conditions	Symbol	Min.	Typ.	Max.	Unit
32-kHz Crystal Oscillator (Operating Range V_{DD} = 2.0 V to 6.5 V)						
Frequency		f_X		32.768		kHz
Start-up time		t_{SQ}		0.5		s
Stability		Δf/f	-10		10	ppm
External 32-kHz Crystal Parameters						
Crystal frequency		f_X		32.768		kHz
Serial resistance		RS		30	50	kΩ
Static capacitance		C0		1.5		pF
Dynamic capacitance		C1		3		fF
External 4-MHz Crystal Parameters						
Crystal frequency		f_X		4.0		MHz
Serial resistance		RS		40	150	Ω
Static capacitance		C0		1.4	3	pF
Dynamic capacitance		C1		3		fF
External 4-MHz Ceramic Resonator Parameters						
Frequency		f_X		4.0		MHz
Serial resistance		RS		8	20	Ω
Static capacitance		C0		36	45	pF
Dynamic capacitance		C1		4.4		fF
EEPROM						
Operating current during erase/write cycle		I_{WR}		600	1300	µA
Endurance	Erase-/write cycles	E_D	500,000	1,000,000		Cycles
Data erase/write cycle time	For 16-bit access	t_{DEW}		9	12	ms
Data retention time		t_{DR}	10			Years
Power-up to read operation		t_{PUR}			0.2	ms
Power-up to write operation		t_{PUW}			0.2	ms
Serial Interface						
SCL clock frequency		f_{SC_MCL}		100	500	kHz

Crystal Characteristics

Figure 62. Crystal Equivalent Circuit

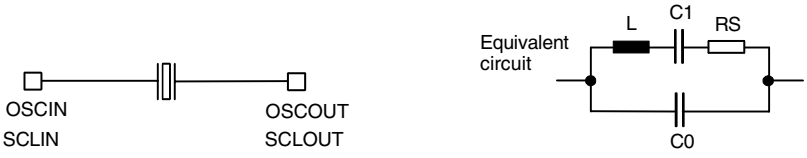


Figure 63. Active Supply Current versus Frequency

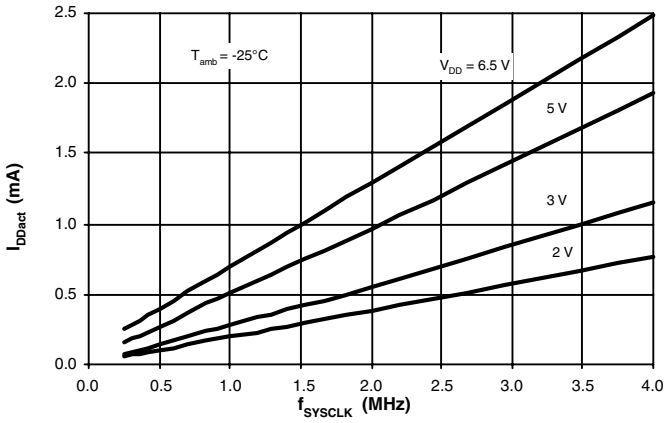


Figure 64. Power-down Supply Current versus Frequency

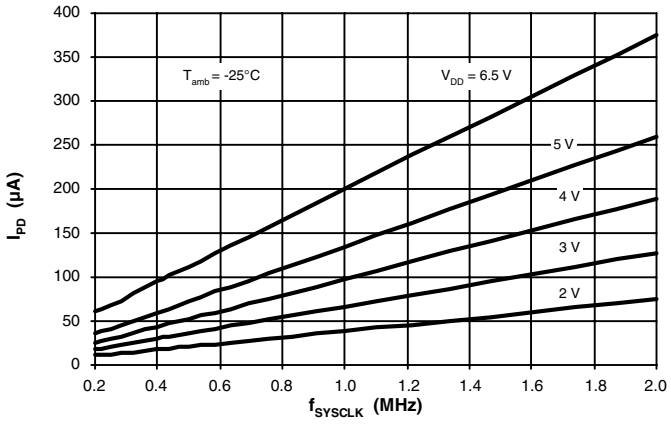


Figure 65. Sleep Current versus T_{amb} ATAR090

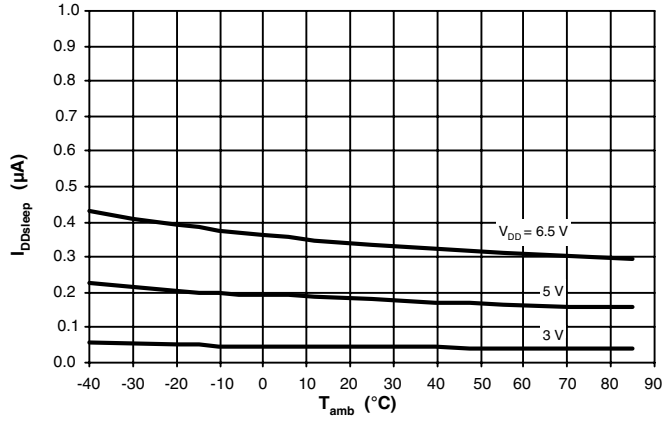


Figure 66. Active Supply Current versus V_{DD}

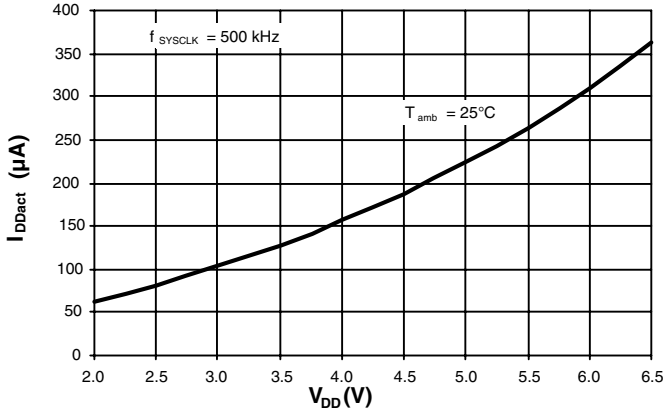


Figure 67. Power-down Supply Current versus V_{DD}

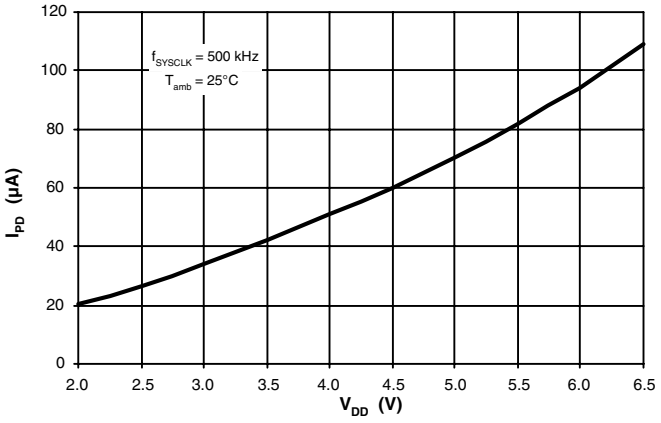


Figure 68. Sleep Current versus T_{amb} – ATAR890

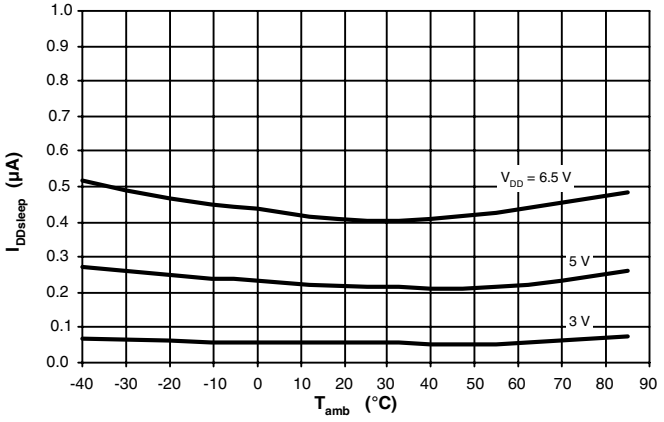


Figure 69. Internal RC Frequency versus V_{DD} – ATAR090

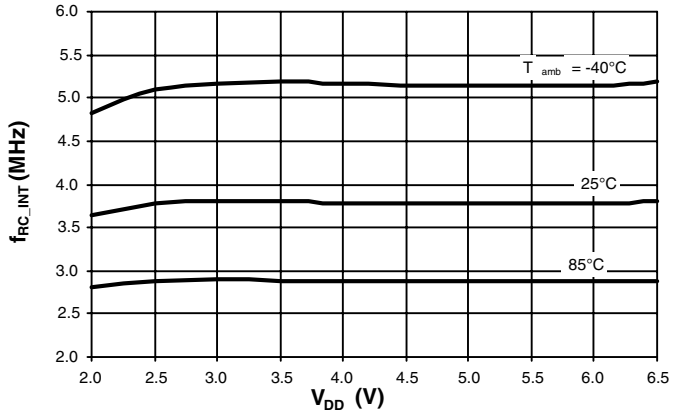


Figure 70. External RC Frequency versus V_{DD}

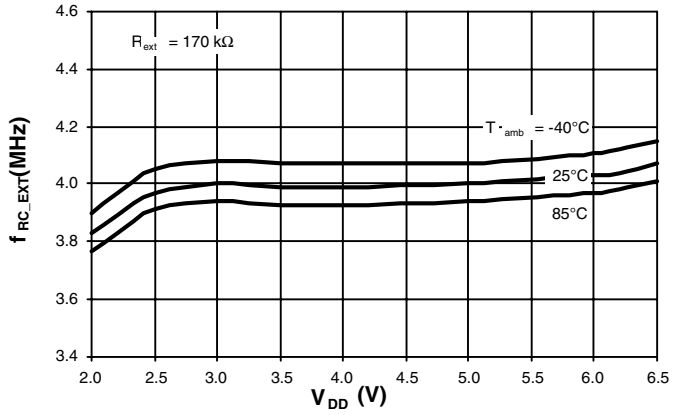


Figure 71. System Clock versus V_{DD}

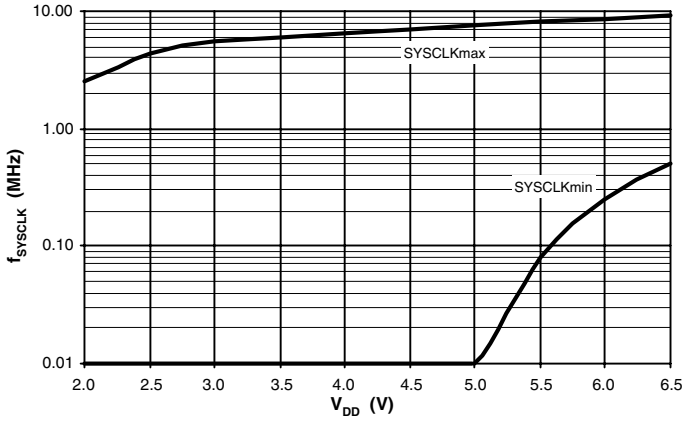


Figure 72. Internal RC Frequency versus T_{amb} – ATAR090

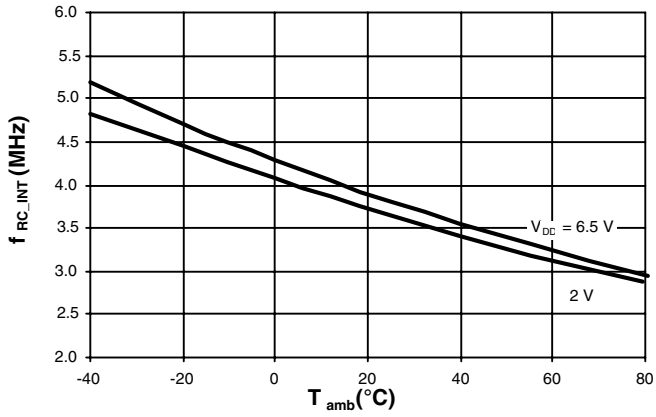


Figure 73. External RC Frequency versus T_{amb}

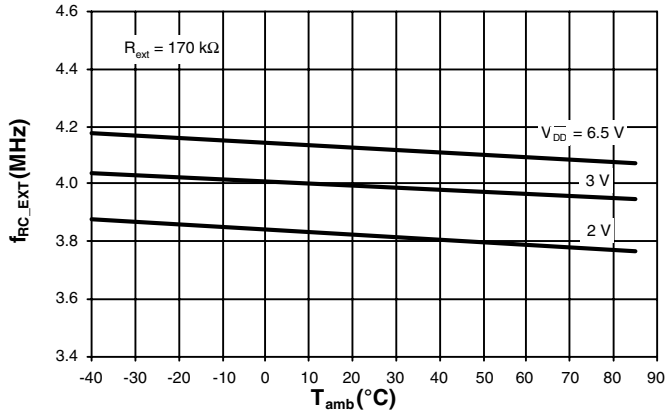


Figure 74. External RC Frequency versus R_{ext}

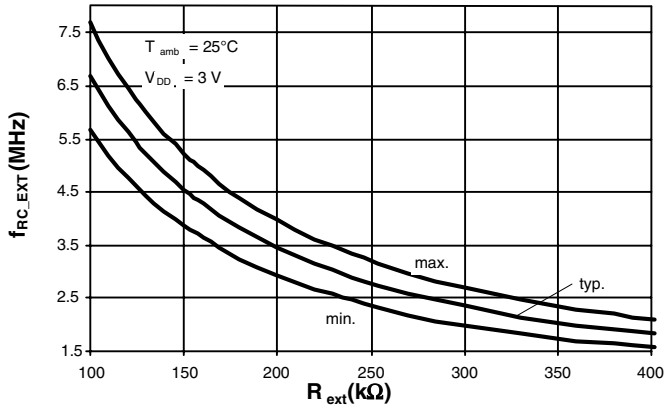


Figure 75. Pull-up Resistor versus V_{DD}

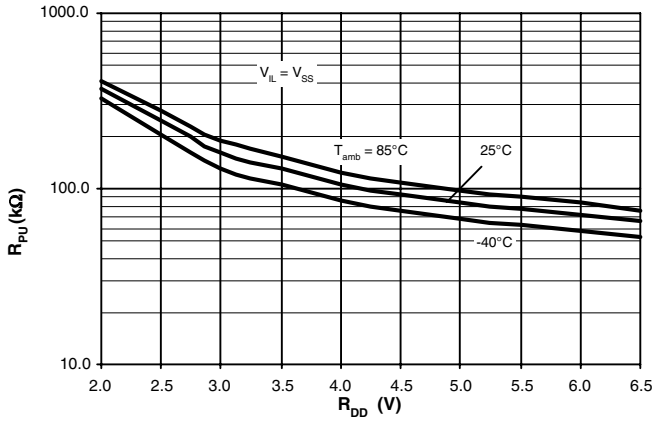


Figure 76. Strong Pull-up Resistor versus V_{DD}

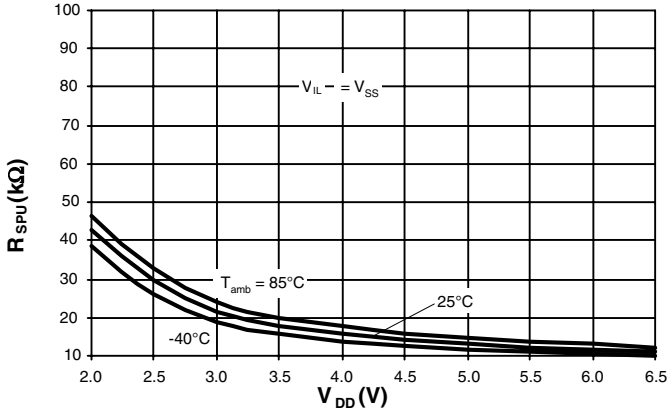


Figure 77. Output High Current versus V_{DD} - Output High Voltage

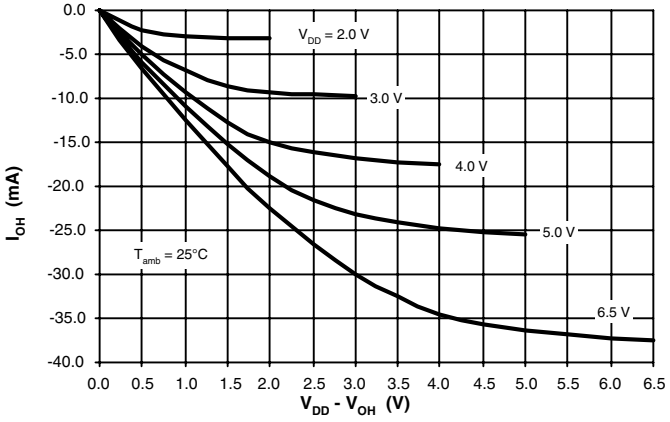


Figure 78. Pull-down Resistor versus V_{DD}

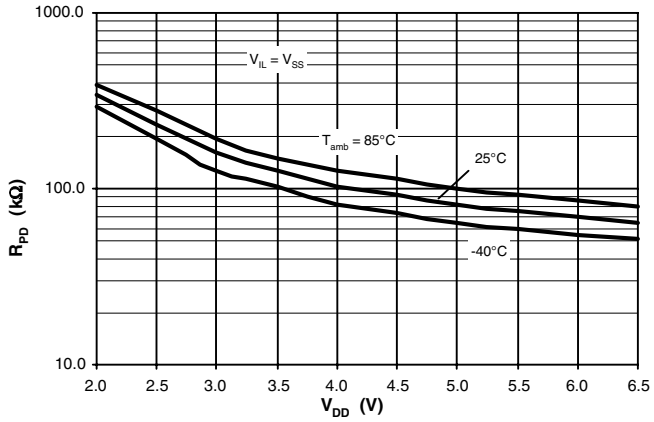


Figure 79. Strong Pull-down Resistor versus V_{DD}

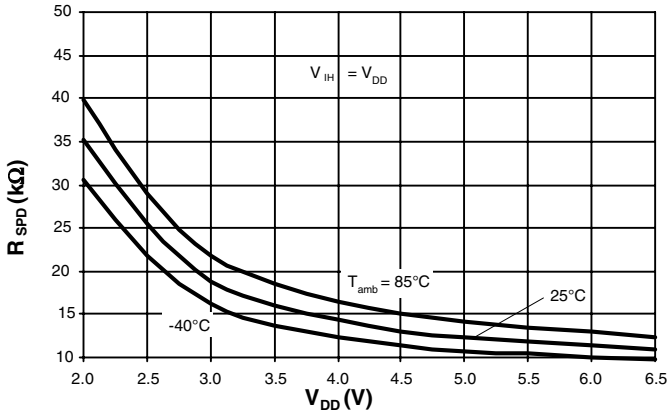


Figure 80. Output Low Current versus Output Low Voltage

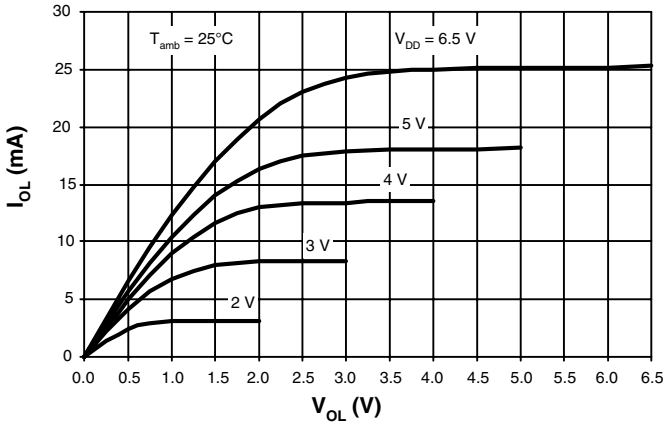


Figure 81. Output High Current versus T_{amb} = 25°C, V_{DD} = 6.5 V, V_{OH} = 0.8 × V_{DD}

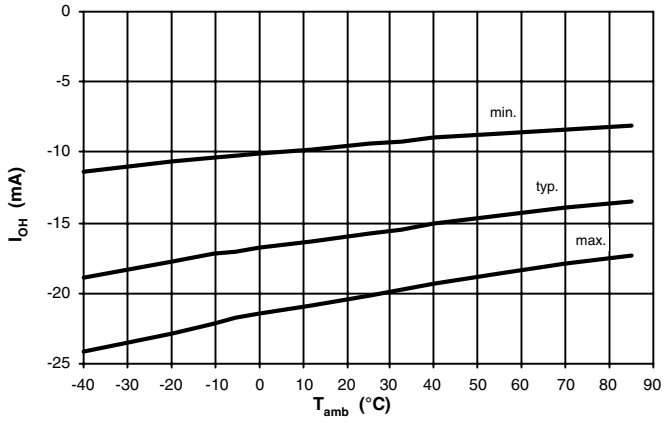
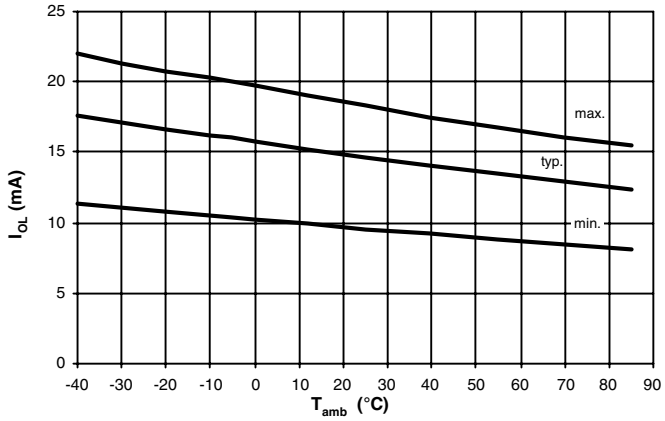


Figure 82. Output Low Current versus T_{amb}, V_{DD} = 6.5 V, V_{OL} = 0.2 × V_{DD}

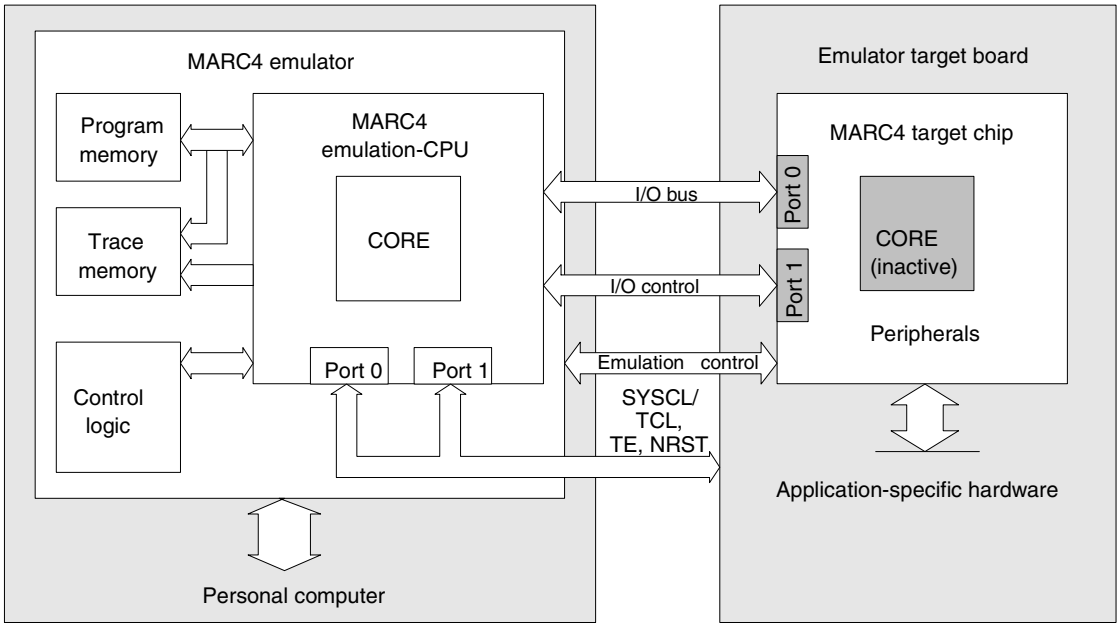


Emulation

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 83. MARC4 Emulation



Option Settings for Ordering

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

	Output	Input
Port 2		
BP20	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP21	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP22	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP23	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
Port 4		
BP40	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP41	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP42	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP43	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down

	Output	Input
Port 5		
BP50	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP51	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP52	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
BP53	[ ] CMOS	[ ] Switched pull-up
	[ ] Open drain [N]	[ ] Switched pull-down
	[ ] Open drain [P]	[ ] Static pull-up
		[ ] Static pull-down
Clock Used	[ ] External resistor	
	[ ] External clock OSC1	
	[ ] External clock OSC2	
	[ ] 32-kHz crystal	
	[ ] 4-MHz crystal	
ECM (External Clock Monitor)		
	[ ] Enable	
	[ ] Disable	
Watchdog		
	[ ] Softlock	
	[ ] Hardlock	

Please attach this page to the approval form.

Date: _____ Signature: _____ Company: _____

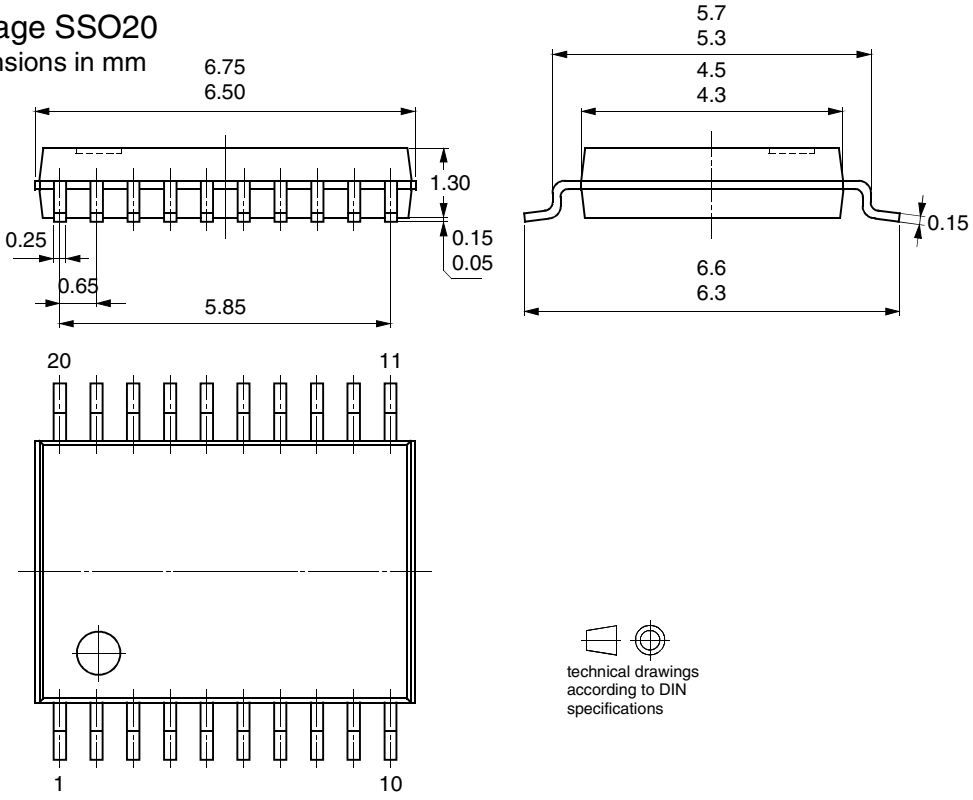
Ordering Information

Extended Type Number	Program Memory	Data-EEPROM	Package	Delivery
ATAR090x-yyy-TKQz	2 kB ROM	No	SSO20	Taped and reeled
ATAR090x-yyy-TKSz	2 kB ROM	No	SSO20	Tubes
ATAR890x-yyy-TKQz	2 kB ROM	512 Bit	SSO20	Taped and reeled
ATAR890x-yyy-TKSz	2 kB ROM	512 Bit	SSO20	Tubes

x = Hardware Revision
yyy = Customer specific ROM-version
z = Operating temperature range
= -- (-40° C to +85° C)
= C (-40° C to +105° C)
= D (-40° C to +125° C)

Package Information

Package SSO20
Dimensions in mm



Revision History

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

Changes from Rev. 4696A - 03/03 to Rev. 4696B - 01/04

1. Put datasheet in a new template.
2. Figure 5 “RAM Map” on page 5 changed.
3. Table 10 “Pheripheral Addresses” on page 21 changed.
4. New heading rows at Table “Absolute Maximum Ratings” on page 60 added.
5. Section “Emulation” on page 71 added.
6. Table “Ordering Information” on page 73 added.
7. Table name on page 72 changed.

Changes from Rev. 4696B - 01/04 to Rev. 4696C - 02/04

1. Figure 4 on page 4 changed.
2. “Ordering Information” on page 73 changed.

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

4696C–4BMCU–02/04