MARC4 – 4-bit Microcontroller with 128 Segment LCD

The M44C588 is a member of the TEMIC family of 4-bit single chip microcontrollers. It contains ROM, RAM, up to 32 digital I/O pins, 8-bit synchronous serial I/O, 32 LCD segment drivers, 8 maskable external interrupt sources, 5 maskable internal interrupts, a watchdog timer, 32-kHz oscillator with programmable interval timer, 2 x 8-bit multi-function timer/counter and a versatile on-chip system clock generation module.

Features

- 9K x 8-bit application ROM
- 512 x 4-bit RAM (256 x 4-bit direct accessible)
- Bitwise maskable prioritised interrupts
- Up to 8 external and 5 internal interrupt sources
- Up to 36 I/O lines – bit or nibblewise I/O
- High drive port (4 mA, V_{DD} = 2.2 V)
- 2 x 8-bit multifunction timer/counters
- 32-kHz on-chip oscillator with 2 programmable interval timer / prescaler
- 8-bit synchronous serial I/O for 2- /3-wire communication
- 4-MHz crystal, 4-MHz ceramic resonator, external resistor or fully integrated RC oscillator as options

Benefits

- Extremely low power consumption
- Minimal external components
- Coded reset and watchdog timer option
- 8 hardware and software interrupt priority levels
- Power on reset, “brown out” function
- Power down modes
- 1.8 V to 6.2 V supply voltage
- 2 level battery low detect (2.2 V / 2.4 V)
- Data retention down to 1.8 V in SLEEP mode
- Efficient, hardware controlled interrupt handling
- High level programming language in qFORTH
- Comprehensive library of useful routines
- PC based development tools

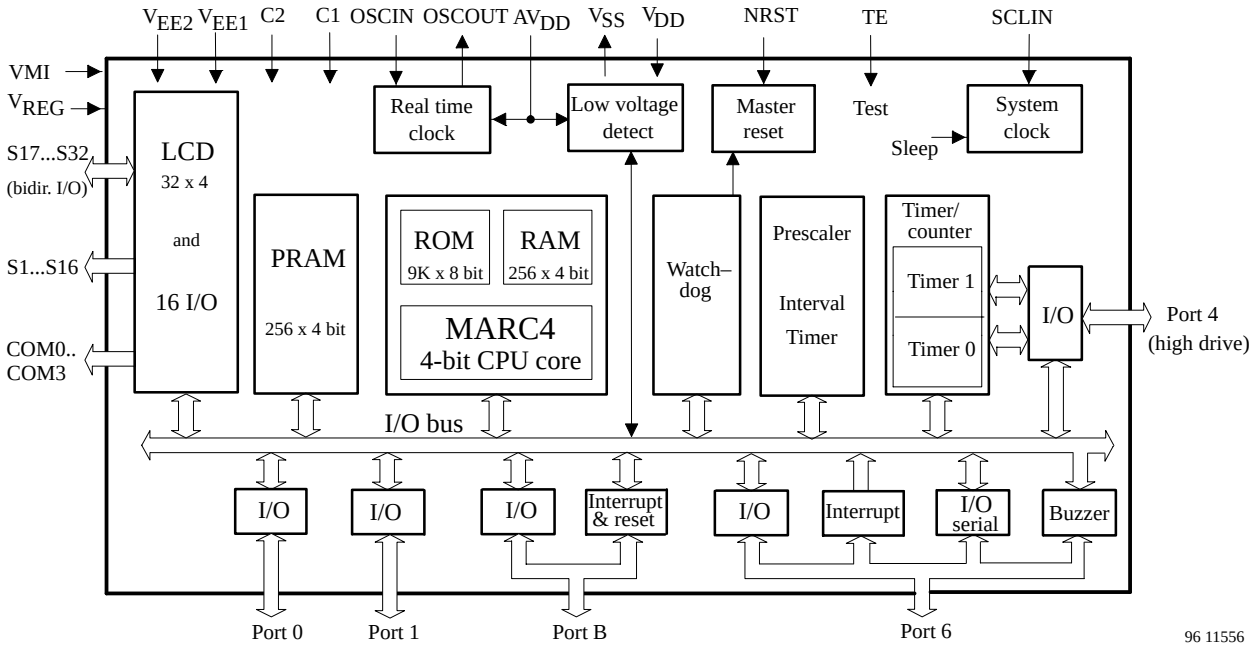


Figure 1. Block diagram

Figure 2. Pin connections

Table 1. Pin description

Name	Function
V_{DD}	Power supply voltage +2.2 V to +6.2 V
AV_{DD}	Analogue power supply voltage +2.2 V to +6.2V
V_{SS}	Circuit ground
BP00 – BP03	4 bidirectional I/O lines of Port 0 – automatic nibblewise configurable I/O
BP10 – BP13	4 bidirectional I/O lines of Port 1 – automatic nibblewise configurable I/O
BP40-T0OUT0	I/O line BP40 of Port 4(*) – configurable I/O or Timer/Counter 0 I/O T0OUT0
BP41-T0OUT1	I/O line BP41 of Port 4(*) – configurable I/O or Timer/Counter 0 I/O T0OUT1
BP42-T1OUT0	I/O line BP42 of Port 4(*) – configurable I/O or Timer/Counter 1 I/O T1OUT0
BP43-T1OUT1	I/O line BP43 of Port 4(*) – configurable I/O or Timer/Counter 1 I/O T1OUT1
BPB0 – BPB3	4 bidirectional I/O lines of Port B – bitwise configurable I/O
BP60	I/O line BP60 of Port 6 – configurable I/O or serial clock output or external interrupt source
BP61	I/O line BP61 of Port 6 – configurable I/O or serial data I/O or external interrupt source
BP62	I/O line BP62 of Port 6 – configurable I/O or buzzer output BZ or external interrupt source
BP63	I/O line BP63 of Port 6 – configurable I/O or buzzer output NBZ or external interrupt source
SCLIN	4-MHz quartz crystal/ceramic resonator or trimming resistor pin (mask option dependent)
SCLOUT	4-MHz quartz crystal/ceramic resonator pin (mask option dependent)
OSCIN	32-kHz quartz crystal pin (mask option dependent)
OSCOUT	32-kHz quartz crystal pin (mask option dependent)
TE	Testmode input. This input is used to control the test modes (internal pull-down)
COM0 - COM3	LCD backplane outputs
S01 – S16	LCD segment output lines
S17 – S32	LCD output lines or bidirectional 2 bitwise configurable digital I/O
V_{EE1}	LCD doubler voltage (2 x V_{REG})
V_{EE2}	LCD tripler voltage (3 x V_{REG})
V_{REG}	Regulated LCD supply voltage
C1, C2	LCD tripler capacitor
NRST	Reset input (/output), a logic low on this pin resets the device. An internal watchdog or coded reset is indicated by a low pulse on this pin.

(*) For mask options please see the ordering information.

Table of Contents

1 MARC4 Architecture

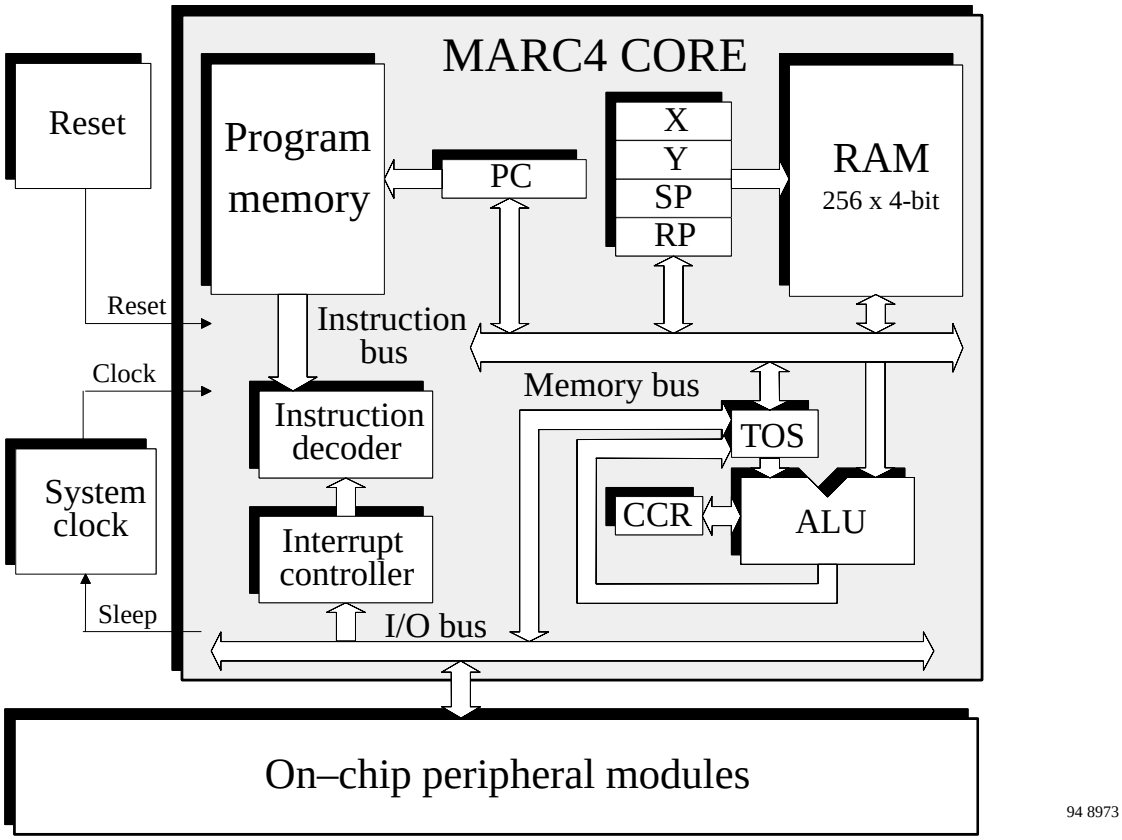


Figure 3. MARC4 core

1.1 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 an expression and a return stack. This architecture allows high level language programming without any loss in efficiency or code density.

1.2 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:

1.2.1 ROM

The program memory (ROM) is mask programmed with the customer application program during the fabrication of the microcontroller. The ROM is addressed by a 12-bit wide program counter, thus predefining a maximum program bank size of 4 Kbytes which can be addressed directly without bank switching.

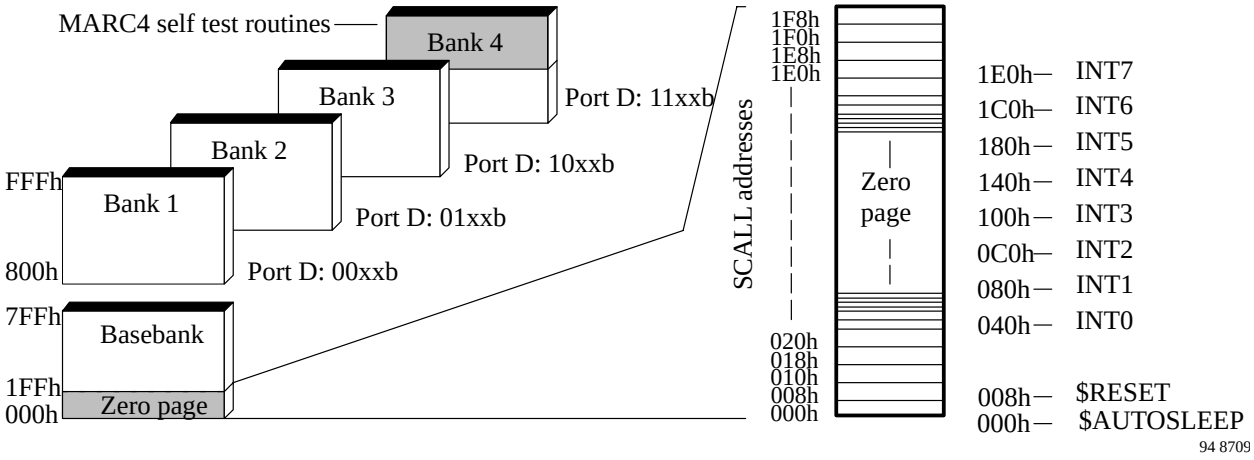


Figure 4. ROM map of M44C588

Test ROM

An additional 1 Kbyte of ROM exists in the 4th ROM bank which is accessible using a ROM bank switch. Of this program space a section is reserved for quality control self-test software, the remainder is available for application program.

The lowest 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.

ROM Banking

For customers programming with qFORTH the bank switching is fully supported by the compiler. The MARC4 switches from one ROM bank to another by writing the new bank number to the ROM Bank Register (RBR). Conventional program space (power up bank) resides in ROM bank 0. Each ROM bank consists of a 4 Kbyte address space whereby the lowest 2 Kbyte is common to all banks, so that addresses between 000h and 7FFh always accesses the same ROM data (see figure 4). When ROM banking is used, the compiler will, if necessary insert program code to save and restore the condition of the RBR on bank switching.

1.2.2 RAM

The M4C588 contains 512 x 4-bit wide static random access memory (RAM). This RAM area consists of two separate blocks. The MARC4 core internal RAM is 256 x 4-bit wide and is used for the expression stack, the return stack and data memory for variables and arrays. The internal RAM is addressed by any of the four 8-bit wide RAM address registers SP, RP, X and Y. The additional block of 256 x 4-bit RAM is I/O-mapped and addressed through the peripheral bus. This PRAM should be used for arrays which are accessed seldomly, when doing heavy duty math routines.

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 result 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 like 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.

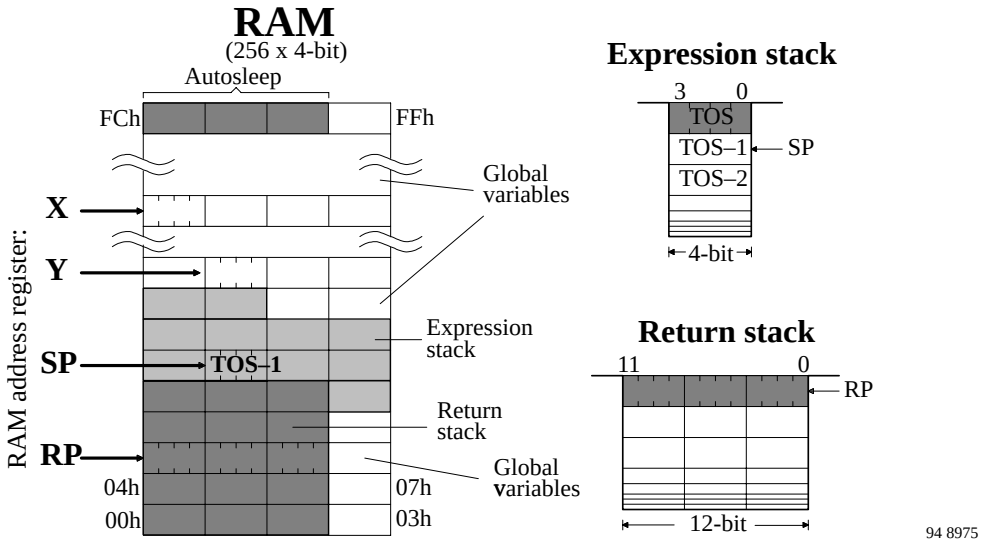


Figure 5. RAM map

1.2.3 Registers

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

Program Counter (PC)

The program counter (PC) is a 12-bit register that contains the address of the next instruction to be fetched from the 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.

ROM Banking Register (RBR)

The ROM banking register is a 4-bit register whereby in the M44C588 only bit 2 and bit 3 are used. These indicate which ROM bank is presently being addressed. The RBR is accessed with a standard peripheral read or write instruction (IN or OUT, port address ‘D’ hex) and is fully supported by the qFORTH compiler.

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 (SP) 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 are left unwritten. These location are used by the qFORTH compiler to allocate 4-bit variables. After a reset the return stack pointer has to be initialized with “>RP FCh ”.

RAM Address Register (X and Y)

The X and Y registers are used to address any 4-bit item in the RAM. A fetch operation moves the addressed nibble onto the TOS. A store operation moves the TOS to the addressed RAM location. Using either the pre-increment or post-decrement addressing mode 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 indicates 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 a direct manipulation of the condition code register.

Carry/Borrow (C)

The carry/borrow flag indicates that borrow or carry out of 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 on executing the DI instruction the interrupt enable flag is reset thus disabling all interrupts. The core will not accept any further interrupt requests until the interrupt enable flag has been set again by either executing an EI, RTI or SLEEP instruction.

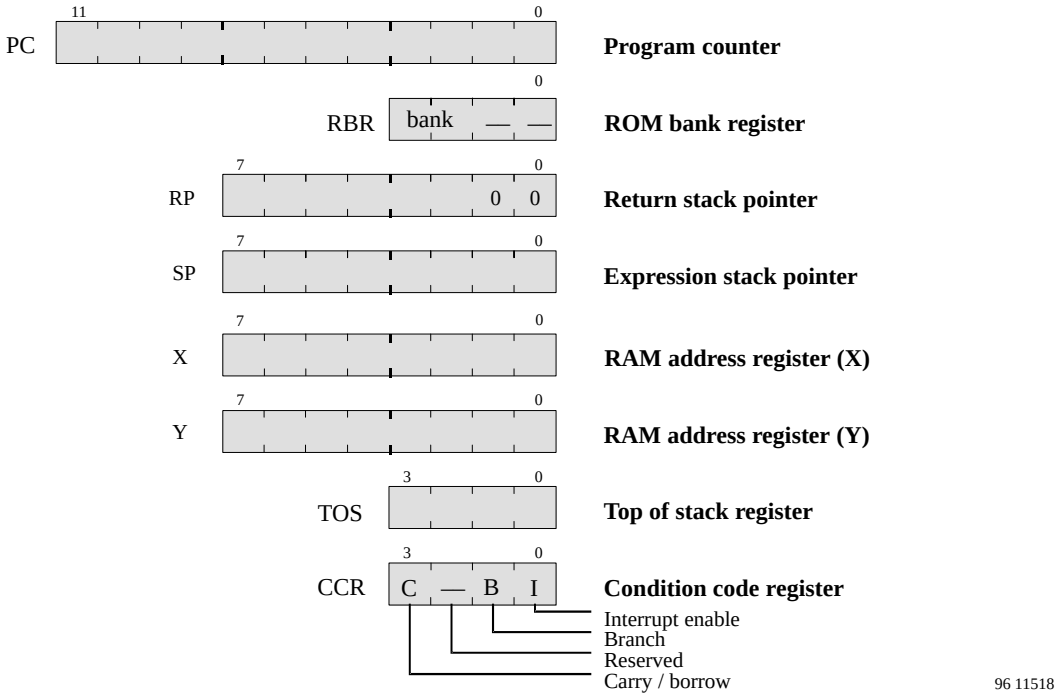


Figure 6. Programming model

1.2.4 ALU

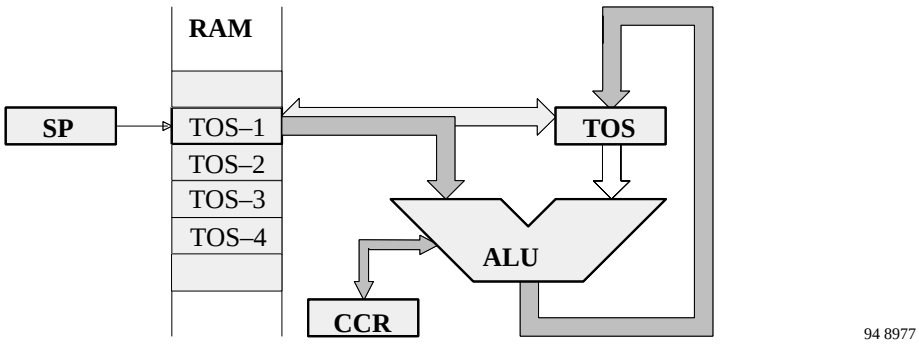


Figure 7. ALU zero address operations

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 returning the result to the TOS. The ALU operations affect the carry/borrow and branch flag in the condition code register (CCR).

1.2.5 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 fast and compact program code. The CPU has an instruction pipeline allowing the controller to prefetch instruction from ROM at the same time as the present instruction is being executed. The MARC4 is a zero address machine, the instructions containing only the operation to be performed and no source or destination address fields. The operations are implicitly performed on the data placed on the stack. There are one and two byte instructions which are executed within 1 to 4 machine cycles. A MARC4 machine cycle is made up of two system clock (SYSCL) cycles. Most of the instructions are only one byte long and are executed in a single machine cycle. For more information see section “MARC4 instruction set overview”.

1.2.6 I/O Bus

The I/O ports and the registers of the peripheral modules (Timer 0, Timer 1, Watch timer, Watchdog etc.) 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 also the section “Emulation”).

1.3 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 2). 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 started 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”).

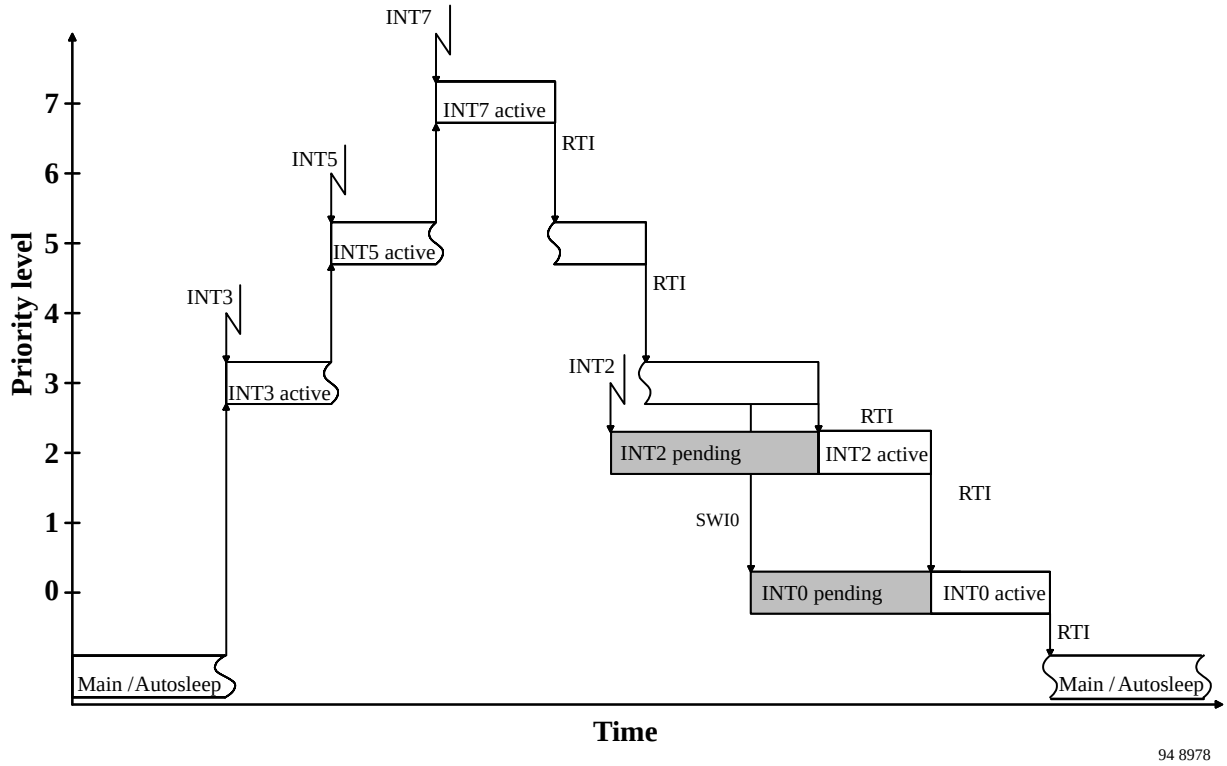


Figure 8. Interrupt handling

Interrupt Processing

For processing the eight interrupt levels the MARC4 includes an interrupt controller with two 8-bit wide “interrupt pending” and “interrupt active” registers. The interrupt controller samples all interrupt requests during every non-I/O instruction cycle and latches these in the interrupt pending register. If no higher priority interrupt is present in the interrupt active register it signals the CPU to interrupt the current program execution. If the interrupt enable bit is set the processor enters an interrupt acknowledge cycle. During this cycle a short call (SCALL) instruction to the service routine is executed and the current PC is saved on the return stack. An interrupt service routine is finished with the RTI instruction. This instruction sets the interrupt enable flag, 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 will be 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 realised that automatic stacking of the RBR is not carried out by the hardware and so if ROM banking is used, the RBR must be stacked on the expression stack by the application program and restored before the RTI. After a master reset (power-on, external 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 the MARC4 this is extremely short taking between 3 to 5 machine cycles depending on the state of the core.

Table 2. Interrupt priority table

Interrupt	Priority	ROM Address	Maskable	Interrupt Opcode
INT0	lowest	040h	Yes	C8h (SCALL 040h)
INT1	|	080h	Yes	D0h (SCALL 080h)
INT2	|	0C0h	Yes	D8h (SCALL 0C0h)
INT3	|	100h	Yes	E8h (SCALL 100h)
INT4	|	140h	Yes	E8h (SCALL 140h)
INT5	|	180h	Yes	F0h (SCALL 180h)
INT6	↓	1C0h	Yes	F8h (SCALL 1C0h)
INT7	highest	1E0h	Yes	FCh (SCALL 1E0h)

1.3.1 Hardware Interrupts

Table 3. Hardware interrupts

Interrupt Source	Possible Interrupt Priorities								RST	Interrupt Mask		Function
	0	1	2	3	4	5	6	7		Register	Bit	
NRST external									X	–	–	low level active
Watchdog									#	–	–	1/2 – 2 sec. time out
Port B coded reset									#	–	–	level any inputs
Port B monitor		*		*		*		*		PBIPR	3	any edge, any input
Port 6 external INTX		*		*		*		*		–	0	any edge, any input
Port 6 external INTY	*		*		*		*			–	0	any edge, any input
Serial I/O	*		*		*		*			SIM0	0	SSI receive buffer full, transmit buffer empty
Interval timer INTA		*				*				ITIPR	0	1 of 8 frequencies (1 – 128Hz)
Interval timer INTB			*				*			ITIPR	1	1 of 8 frequencies (8 – 8192Hz)
Timer 0		*		*		*		*		T0CR	0	overflow/compare/ end measurement
Timer 1	*		*		*		*			T1CR	0	overflow/compare/ end measurement

X = hardwired (neither optional or software configurable)
# = customer mask option (see “Ordering Information”)
* = software configurable (see “Peripheral Modules” section for further details)

In the M44C588 there are eleven hardware interrupt sources which can be programmed to occupy a variety of priority levels. Each source can be individually masked by mask bits in the corresponding control registers. An overview of the possible hardware configurations is shown in table 3.

1.3.2 Software Interrupts

The programmer can generate interrupts 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. Thus using the SWI instruction, interrupts can be re-prioritised or lower priority processes scheduled for later execution.

1.4 Master Reset

The master reset forces the CPU into a well-defined condition, 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, a watchdog time out, activation of the NRST input or the occurrence of a coded reset on Port B (see figure 9). A master reset activation will reset the interrupt enable flag, the interrupt pending register and the interrupt active register. During the reset phase the I/O bus control signals are set to 'reset mode' thereby initializing all on-chip peripherals.

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 initialises all necessary RAM variables, stack pointers and peripheral configuration registers.

Power-on Reset

The fully integrated power-on reset circuit ensures that the core is held in a 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 supply.

External Reset (NRST)

An external reset can be triggered with the NRST pin. To activate an external reset the pin should be low for a minimum of two machine cycles.

Coded Reset (Port B)

The coded reset circuit is connected directly to the Port B terminals. Using a mask option, the user can define a hardwired code combination (e.g. all pins low) which, if occurring on the Port B will generate a reset in the same way as the NRST pin.

Note that if this option is used, the reset is not maskable and will also trigger if the predefined code is written on to the Port B by the CPU itself. Care should also be taken not to generate an unwanted reset by inadvertently passing through the reset code on input transitions. This applies especially if the pins have a high capacitive loading.

Watchdog Reset

The watchdog can be activated by using a mask option and triggers a reset with every watchdog counter overflow. To suppress the watchdog reset, the counter must be regularly reset by reading the watchdog register address (WDRES).

The CPU reacts in exactly the same manner as a reset stimulus from any of the above sources

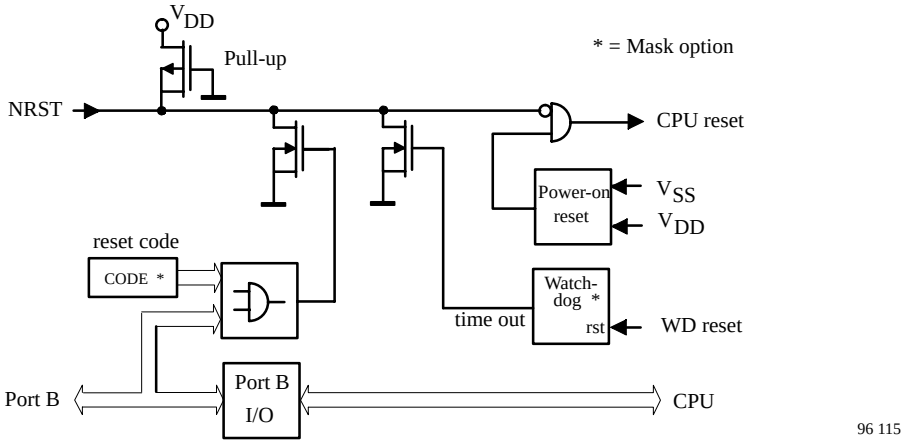


Figure 9. Reset configuration

1.5 Clock Generation

1.5.1 Clock Module

The M44C588 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 OSCIN and OSCOUT are the interface to connect a crystal either to the 4-MHz, or to the 32-kHz crystal oscillator. SCLIN 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 2 is more precise whereby the oscillator frequency can be trimmed with an external resistor attached between SCLIN and V_{DD}. In this configuration, the RC-oscillator 2 frequency can be maintained stable to within a tolerance of ± 10% over the full operating temperature and voltage range.

The clock module is programmable via software with the clock manage register (CM) and the system configuration register (SC). The required oscillator configuration has to be selected with the OS1[1:0]-bits in the SC-register. A programmable 4-bit divider stage allows the adjustment of the system clock speed. A special feature of the clock management is that an external oscillator may be used and switched on and off via a port pin for the power-down mode. Before the external clock is switched off, the internal RC-oscillator 1 must be selected with the CCS-bit and then the SLEEP mode may be activated. In this state an interrupt can wake up the controller with the RC-oscillator, and the external oscillator can be activated and selected by software. A synchronization stage avoids too short clock periods if the clock source or the clock speed is changed.

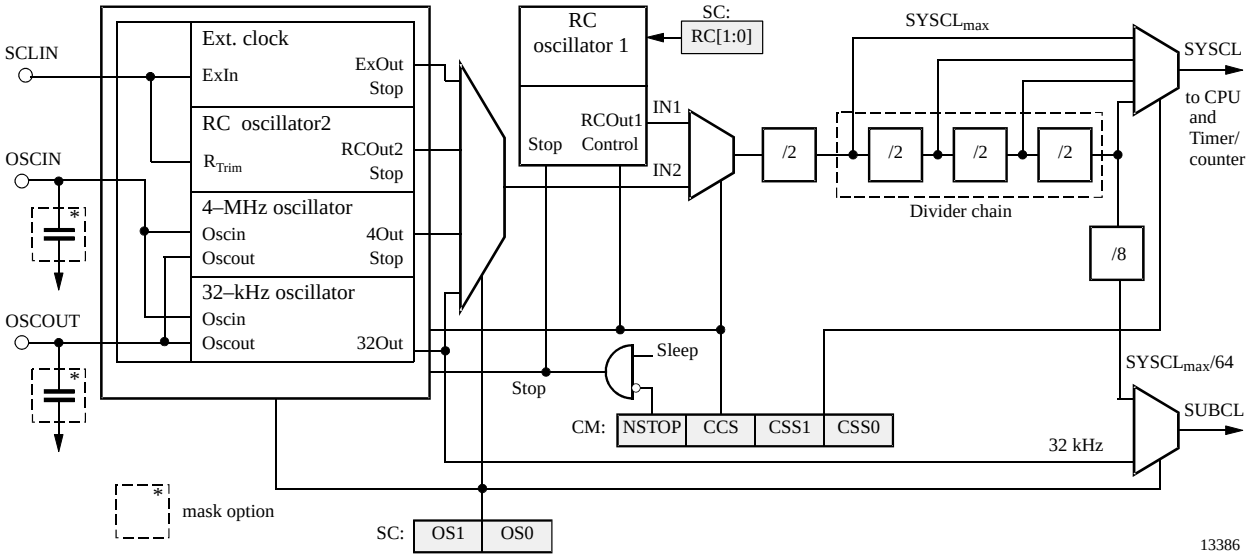


Figure 10. Clock module

Table 4. Clock modes

Mode			Clock Source for SYSCL		Clock Source for SUBCL	
	OS1	OS0	CCS = 1	CCS = 0	CCS = 1	CCS = 0
1	1	1	RC-oscillator 1 (intern)	External input clock	$SYSCL_{max}/64$	SCLIN / 128
2	0	1	RC-oscillator 1 (intern)	RC-oscillator 2 with external trimming resistor	$SYSCL_{max}/64$	$SYSCL_{max}/64$
3	1	0	RC-oscillator 1 (intern)	4-MHz oscillator	$SYSCL_{max}/64$	f_{xtal} / 128
4	0	0	RC-oscillator 1 (intern)	32-kHz oscillator	32 kHz	

The clock module generates two output clocks. One is the system clock (SYSCL) and the other the periphery clock (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.

1.5.2 Oscillator Circuits and External Clock Input Stage

The M44C588 clock system consists of four different internal oscillators: two RC-oscillators, one 4-MHz crystal oscillator, one 32-kHz crystal oscillator and an 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 ±25% over the full temperature and voltage range. The frequency tolerance is better than ±10% for a given voltage and temperature (see diagram). The basic center frequency of the RC-oscillator 1 is programmable with the RC1– and the RC0-bit in the SC register (see page 15).

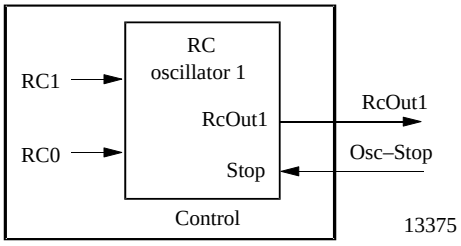


Figure 11. RC-oscillator 1

External Input Clock

The SCLIN pin can be driven by an external clock source provided it meets the specified input levels, duty cycle, rise and fall times. The maximum system clock $SYSC_{max}$ the core operates will therefore be SCLIN/2 (see figure 10).

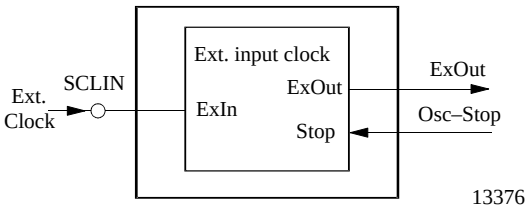


Figure 12. External input clock

RC-Oscillator 2 with External Trimming Resistor

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

For example: An $SYSCL_{max}$ frequency of 2 MHz, can be obtained by connecting a resistor R_{ext} = 136 kΩ (see figures 13 and 47).

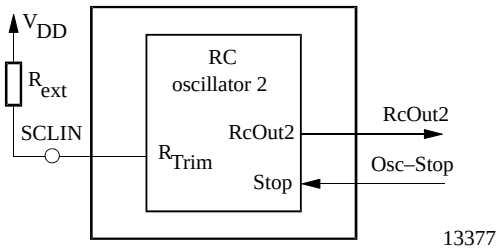


Figure 13. RC-oscillator 2

4-MHz Oscillator

The integrated system clock oscillator expects a crystal or ceramic resonator connected to the OSCIN and OSCOUT pins to establish oscillation. All the necessary oscillator circuitry, with the exception of the actual crystal, resonator and the optional C3 and C4 are integrated on-chip.

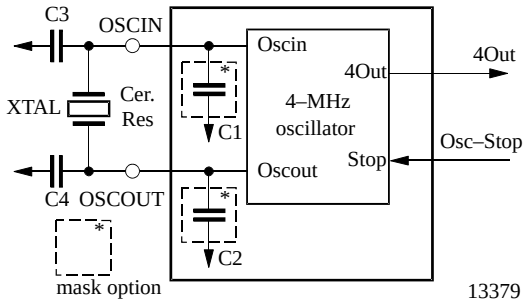


Figure 14. System clock oscillator

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/or the SYSCL. In this mode, power consumption is greatly reduced. The 32-kHz crystal oscillator is still operating (not stopped) during any power-down/SLEEP mode.

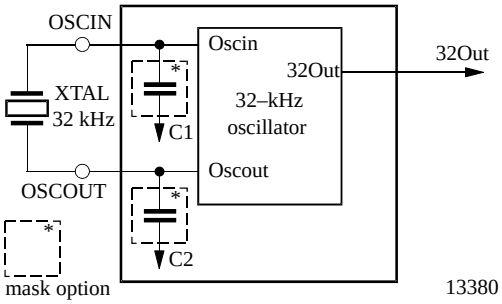


Figure 15. 32-kHz crystal oscillator

1.5.3 Clock Management Register (CM)

The clock management register (CM) controls the system clock divider chain, as well as the peripheral clock in the power-down modes.

Auxiliary register address: ’E’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 (SUBCL) during the core is in SLEEP mode, with the exception of the 32-kHz crystal oscillator SUBCL clock.
NSTOP = 1, enables the peripheral clock (SUBCL) during the core 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) will generate SYSCL dependent on the setting of OS0 and OS1 in the system configuration register

CSS[1:0] **C**ore **S**peed **S**elect
These two bits control the system clock divider chain

Auxiliary register address: ’E’hex

CSS1	CSS0	Divider	Note
0	0	16	$SYSCL_{max}/8$
0	1	8	$SYSCL_{max}/4$
1	0	4	$SYSCL_{max}/2$
1	1	2	Reset value = $SYSCL_{max}$

System Configuration Register (SC)

Primary register address: 'E'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
SC: write	**RC1**	**RC0**	**OS1**	**OS0**	**Reset value: 1111b**

RC1, RC0 Internal RC oscillator 1 frequency select ($SYSCL_{max}$)

RC1	RC0	SYSCLmax @ 25°C, V_{DD} = 5 V	Note
0	0	7.5 MHz	
0	1	3.2 MHz	
1	0	2.0 MHz	
1	1	0.8 MHz	Reset value

OS1, OS0 Oscillator selection bits (in conjunction with the CCS-bit)

CCS	OS1	OS0	SUBCL	System Oscillator Selection
0	1	1	$SYSCL_{max}$/64	External input clock at SCLIN
0	0	1		RC-oscillator 2 with R_{ext}
0	1	0		4-MHz crystal oscillator
0	0	0	32 kHz	32-kHz crystal oscillator
1	x	x	$SYSCL_{max}$/64 or 32 kHz	RC-oscillator 1

If CCS = 0 in the CM-register, the RC-oscillator 1 is stopped.

1.5.4 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 µC is not fully utilized. In this mode, the system clock is stopped. During the sleep mode the peripheral modules remain active and are able to generate interrupts. The µC exits the sleep mode by carrying out any interrupt or a reset condition.

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

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

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

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

The M44C588 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 continously independent on the NSTOP-bit. If the oscillator is stopped or the 32 kHz oscillator is selected, power consumption is extremely reduced.

Table 5. Power-down modes

Mode	CPU Core	NSTOP	RC-Oscillator 1 RC-Oscillator 2 4-MHz Oscillator	32-kHz Oscillator	External Input Clock at SCLIN
Active	RUN	1	RUN	RUN	Enabled
Power-down	SLEEP	1	RUN	RUN	Enabled
SLEEP	SLEEP	0	STOP	RUN	Disabled

1.5.5 Clock Monitor Mode

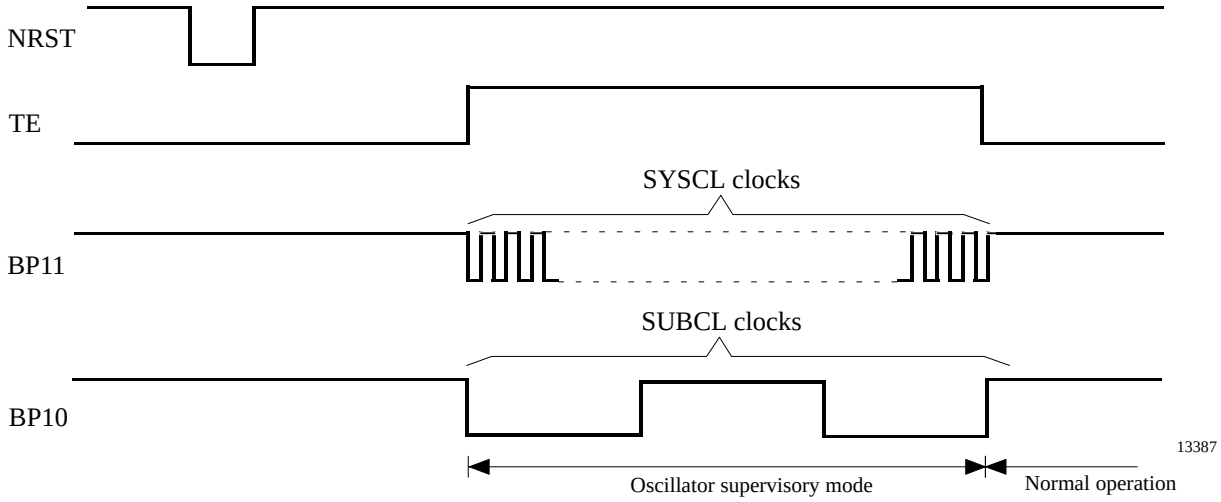


Figure 16. Clock monitoring

For trimming purposes, the M44C588 can be put into a clock monitor mode. The test input (TE) has to be forced high, whereupon the SYSCL clock will appear on BP11 (Port 1, bit 1) and SUBCL clock on Port BP10 (Port 1, bit 0). To put BP10 and BP11 back into normal operation, the TE-pin has to be released again (see figure 17).

2 Peripheral Modules

2.1 Addressing Peripherals

Accessing the peripheral modules takes place via the I/O bus (see figure 12). The IN or OUT instructions allow direct addressing of up to 16 I/O modules. A dual register addressing scheme has been adopted, with 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 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 access 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 being initially written into the auxiliary register.

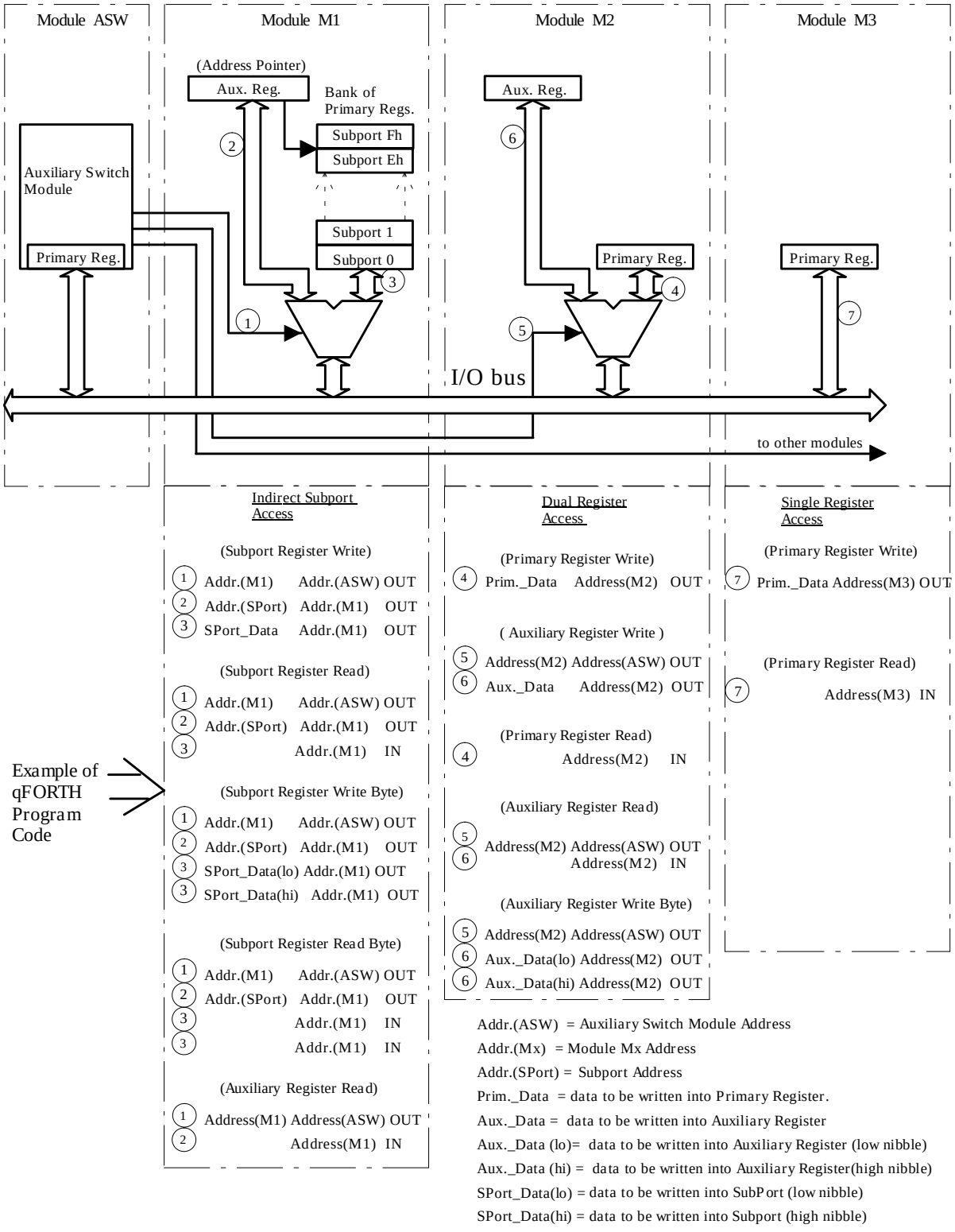


Figure 17. Example of I/O addressing

Table 6. M44C588 peripheral addresses

Address		Name	Write /Read	Function
0		P0DAT	W/R	Port 0 – data register/input data
1		P1DAT	W/R	Port 1 – data register/input data
2		LCD	W/R	LCD segment data shadow register port
	Aux.	LCR	W	LCD control register
3		SC	W	System configuration register
		CWD	R	Watchdog timer reset
	Aux.	CM	W/R	Clock management register
4		P4DAT	W/R	Port 4 – data register/pin data
	Aux.	P4DDR	W	Port 4 – data direction register
5		–	–	Reserved
6		P6DAT	W/R	Port 6 – data register/pin data
	Aux	P6DDR	W	Port 6 – data direction register
7		SSI	W/R	Data to/from subport addressed by SIX
	Aux.	SIX	W	Serial interface subport address register
SSI subport addresses				
	0	SISR	R	Serial interface status register
		SIMO	W	Serial interface mode register
	1	SICC	W	Serial interface clock control register
	2	SICR	W	Serial interface control register
	3	P6IOR	W	Port 6 (SSI/buzzer) I/O control register
	4	BZCR	W	Buzzer control register
	5	P6IPR	W	Port 6 (INTX/INTY) interrupt priority register
	6	IRXCR	W	External interrupt X source select
	7	IRYCR	W	External interrupt Y source select
	8	RxBUF	R	Receive buffer (byte)
		TxBUF	W	Transmit buffer (byte)
8		ASW	W	Auxiliary Switch register
9		TCM	W/R	Data to/from subport addressed by TCX
	Aux.	T0SR	R	Timer 0 interrupt status register
		TCX	W	Timer/counter subport address pointer
TCM subport addresses				
	0	T0MO	W	Timer 0 mode register
	1	T0CR	W	Timer 0 control register
	2	T1MO	W	Timer 1 mode register
	3	T1CR	W	Timer 1 control register
	4	TCMO	W	Timer/counter mode register
	5	TCIO	W	Timer/counter I/O control register
	6	TCCR	W	Timer/counter clock control register
	7	TCIP	W	Timer/counter interrupt priority
	8	T0CP1	W	Timer 0 compare register 1 (byte)
		T0CA	R	Timer 0 capture register (byte)
	9	T1CP1	W	Timer 1 compare register 1 (byte)
		T1CA	R	Timer 1 capture register (byte)
	A	T0CP2	W	Timer 0 compare register 2 (byte)
	B	T1CP2	W	Timer 1 compare register 2 (Byte)

Address		Name	Write /Read	Function
	C	T0ICR	W	Timer 0 interrupt source control register
	D	T1ICR	W	Timer 1 interrupt source control register
	E	P4ICR	W	Port 4 external interrupt control register
	F	P4IOR	W	Port 4 (T0/T1) I/O control register
A		PBIPR	W	Port B – interrupt priority register
Aux.		PBICR	W	Port B – interrupt control register
B		PBDAT	W/R	Port B – data register/pin data
Aux.		PBDDR	W	Port B – data direction register
C		–	–	Reserved
D		RBR	W	Rom bank switch register
E		PRAM	W/R	256 x 4-bit peripheral RAM data access port
Aux.		PRADR	W	Peripheral RAM address pointer register (Byte)
F		ITFSR	W	Interval timer frequency select register
Aux.		ITIPR	W	Interval timer interrupt priority register

2.2 Bidirectional Ports

All bidirectional ports 0, 1, 4, 6 and B are 4-bits wide. All these ports may be used for data input or output. All inputs are equipped with Schmitt-trigger inputs together with a variety of mask options for open drain, open source, full complementary outputs as well as different kinds of pull-up and pull-down transistors. The optional pull-up/ pull-down transistors are only active when the port is in input mode. All Port Data Registers (PxDAT) are I/O mapped to the primary address register of the respective port address and the Port Data Direction Register (PxDDR) to the corresponding auxiliary register.

Port Data Register (PxDAT)

Primary register address: 'Port address'hex

	Bit 3*	Bit 2	Bit 1	Bit 0	
PxDAT	**PxDAT3**	**PxDAT2**	**PxDAT1**	**PxDAT0**	**Reset value: 1111b**

* Bit 3 → MSB, bit 0 → LSB

Port Data Direction Register (PxDDR)

Auxiliary register address: 'Port address'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
PxDDR	**PxDDR3**	**PxDDR2**	**PxDDR1**	**PxDDR0**	**Reset value: 1111b**

Value: 1111b means all pins in input mode

Table 7. Port Data Direction Register (PxDDR)

Code 3 2 1 0	Function
x x x 1	BPx0 input mode
x x x 0	BPx0 output mode
x x 1 x	BPx1 input mode
x x 0 x	BPx1 output mode
x 1 x x	BPx2 input mode
x 0 x x	BPx2 output mode
1 x x x	BPx3 input mode
0 x x x	BPx3 output mode

There are four different types of bidirectional ports:

- Type 1 (Ports 0 and 1) – 4-bit wide bidirectional ports with automatic full bus width direction switching.
- Type 2 (Port B) – 4-bit wide bitwise programmable bidirectional port with static pull-ups/ -downs.
- Type 3 (Port 6) – 4-bit wide bitwise programmable bidirectional port with static pull-ups/ -downs and versatile interrupt control logic.
- Type 4 (Port 4) – 4-bit wide bitwise programmable bidirectional port also provides the I/O interface for Timer 0 and Timer 1.

2.2.1 Bidirectional Port 0 and Port 1

In this port type, the data direction register is not independently software programmable, the direction of the complete port being switched automatically when an I/O instruction occurs (see figure 18). The port is switched to output mode with an OUT instruction and to input with an IN instruction. The data written to a port will be stored into the output data latches and appears immediately at the port pin following the OUT instruction. After RESET all output latches are set to ’1’ and the ports are switched to input mode. An IN instruction reads the condition of the associated pins.

Note

Care must be taken when switching these bidirectional ports from output to input. The capacitive pin loading at this port in conjunction with the high resistance pull-up may cause the CPU to read the contents of the output data register rather than the external input state. To avoid this, one should use either of the following programming techniques:

- Use two IN instructions and DROP the first data nibble. The first IN switches the port from output to input and the DROP removes the first invalid nibble. The second IN reads the valid pin state.
- Use an OUT instruction followed by an IN instruction. With the OUT instruction the capacitive load is charged or discharged depending on the optional pull-up /pull-down configuration. Write a “1” for pins with pull-up resistors and a “0” for pins with pull-down resistors.

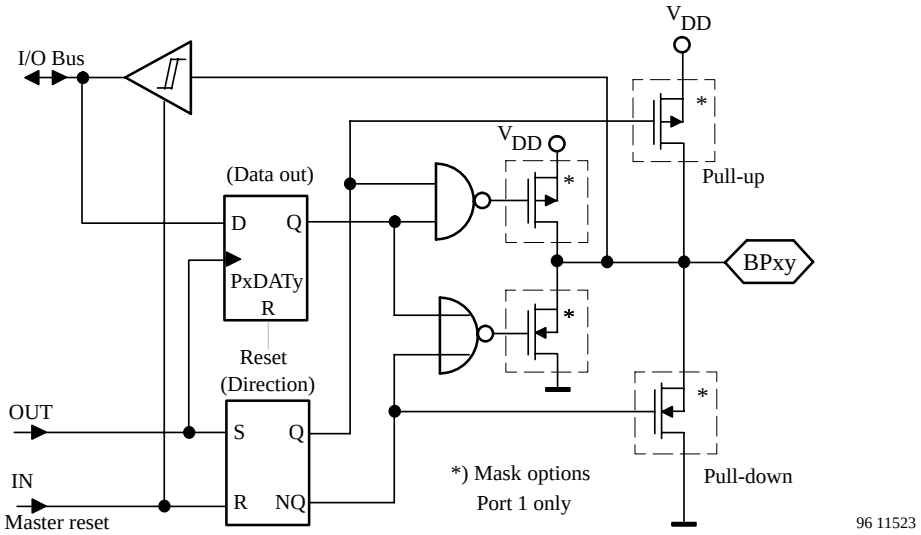


Figure 18. Bidirectional Ports 0 and 1

2.3 Bidirectional Port B

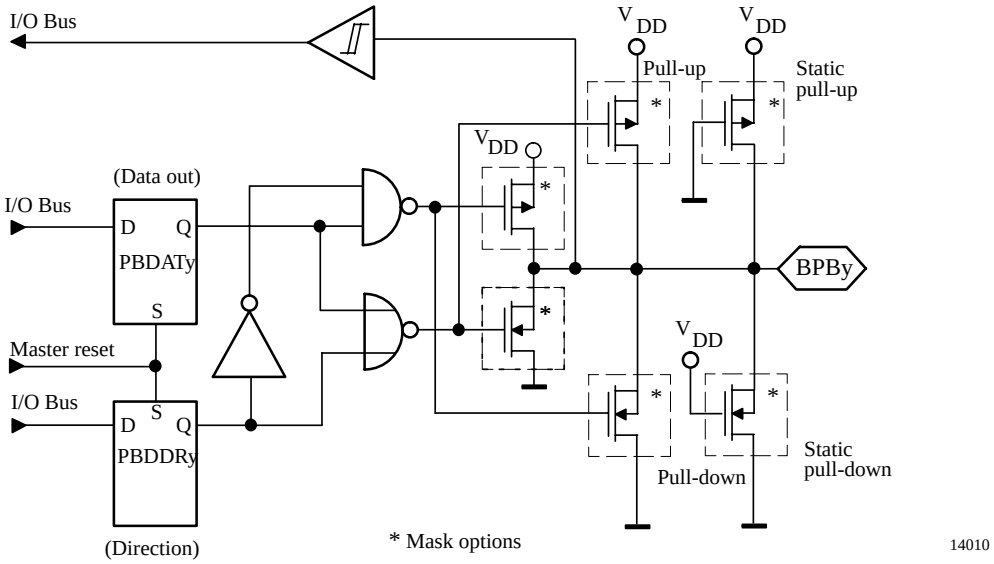


Figure 19. Bidirectional Port B

Port B includes a bitwise programmable Data Direction Register (PBDDR), which allows 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 data bus applications.

2.3.1 Port Monitor Module

In addition to the standard I/O functions, Port B (BPB3 – BPB0) is equipped with port monitor module. This module is connected across all four port pins (see figure 20) and generates an interrupt should a preprogrammed transition occur on any of the pins. This allows interrupt driven port scanning without the power consuming task of continuously polling the port inputs.

Using the Port Interrupt Control Register (PBICR), each pin can be selected individually. A non-selected pin cannot generate an interrupt. The Port Interrupt Priority Register (PBIPR) allows masking of each interrupt, definition of the interrupt generating signal transition and programming of the interrupt priority level.

The Port Interrupt Priority Registers PBIPR is I/O mapped to the primary address registers of the Port Monitor Module addresse ’A’h. The Port Interrupt Control Registers PBICR is mapped to the corresponding auxiliary register.

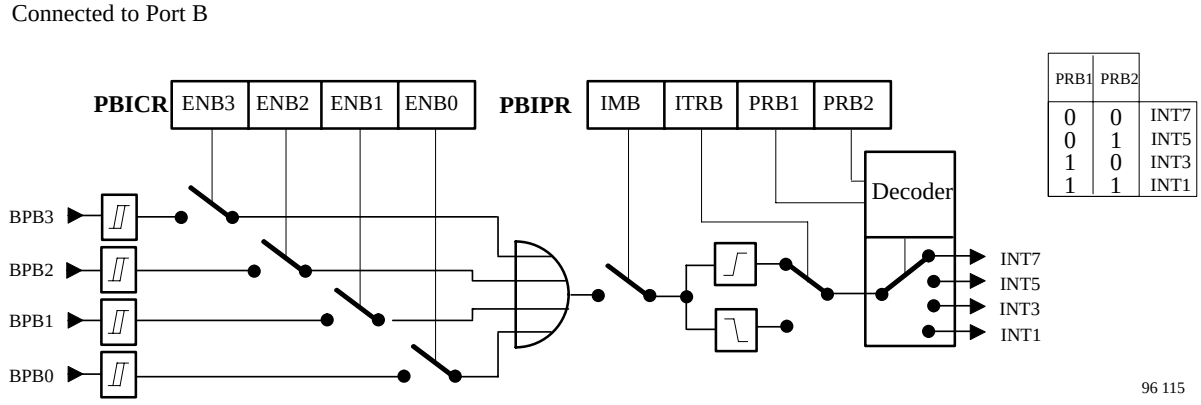


Figure 20. Port B monitor input module

Port B Monitor Interrupt Priority Register (PBIPR)

Primary register address: ’A’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
PBIPR	**IMB**	**ITRB**	**PRB2**	**PRB1**	**Reset value: 1111b**

IMB – Interrupt Mask

ITRB – Interrupt Transition

PRB2..1 – Interrupt Priority code

Table 8. Port B Monitor Interrupt Priority Register (PBIPR)

Code 3 2 1 0	Function
x x 0 0	Port monitor interrupt priority 7
x x 0 1	Port monitor interrupt priority 5
x x 1 0	Port monitor interrupt priority 3
x x 1 1	Port monitor interrupt priority 1
x 1 x x	Port monitor interrupt on rising edge
x 0 x x	Port monitor interrupt on falling edge
0 x x x	Port monitor interrupt enabled
1 x x x	Port monitor interrupt masked

Port B Monitor Interrupt Control Register (PBICR)

Auxiliary register address: ’A’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
PBICR	**ENB3**	**ENB2**	**ENB1**	**ENB0**	**Reset value: 1111b**

ENB3 ... 0 Port B monitor input ENable code

Table 9. Port B Monitor Interrupt Control Register (PBICR)

Code 3 2 1 0	Function
x x x 0	Bit 0 can generate an interrupt
x x x 1	Bit 0 cannot generate an interrupt
x x 0 x	Bit 1 can generate an interrupt
x x 1 x	Bit 1 cannot generate an interrupt
x 0 x x	Bit 2 can generate an interrupt
x 1 x x	Bit 2 cannot generate an interrupt
0 x x x	Bit 3 can generate an interrupt
1 x x x	Bit 3 cannot generate an interrupt

2.4 Bidirectional Port 6

This 4-bit bidirectional port can be used as bitwise programmable I/O. The port pins can also be used as external interrupt inputs (see figures 21 and 22). Both interrupts can be masked or independently configured to trigger on either edge. The interrupt priority levels are also programmable. The interrupt configuration is controlled by the Port 6 Interrupt Priority Register (P6IPR), the External Interrupt X Source Select Register (IRXCR) and the External Interrupt Y Source Register (IRYCR). The port direction is controlled by the Port 6 Data Direction Register (P6DDR) An additional low resistance pull-up transistor provides an internal static pull-up for serial bus applications (mask option).

In output mode (P6DDR bit = 0), the respective Port Data Register (P6DAT) bit will appear on the port pin, driven by an output port driver stage which can be mask programmed as open drain, or full complementary CMOS. With an IN instruction the actual pin state can be read back at any time into the controller without changing the port directional mode. So, for example should the output port be mask configured as an open drain driver, as long as the output transistor is off, the controller is able to receive external data on this pin without switching into input mode.

In input mode (P6DDR bit = 1), the output driver stage is deactivated, so that an IN instruction will directly read the pin state which can be driven from an external source. In this case the state of the Port Data Register (P6DAT), although not appearing at the pin itself remains unchanged. High resistance mask selectable pull-up or pull-down transistors are automatically switched onto the port pin in input mode. The Port Data Register is written with an OUT instruction to the respective port address.

The Port 6 Data Register (P6DAT) is I/O mapped to the primary address register of address ’6’hex and the Port 6 Control Register (P6CR) to the corresponding auxiliary register. The Interrupt Priority Register (P6IPR) and the External Interrupt X/Y Priority Registers IRXCR and IRYCR) are indirectly addressed by using extended addressing mode as descibed in section ”Addressing Peripherals” and I/O mapped to the Port 7 subport register addresses ’5’hex, ’6’hex and ’7’hex (see table 6).

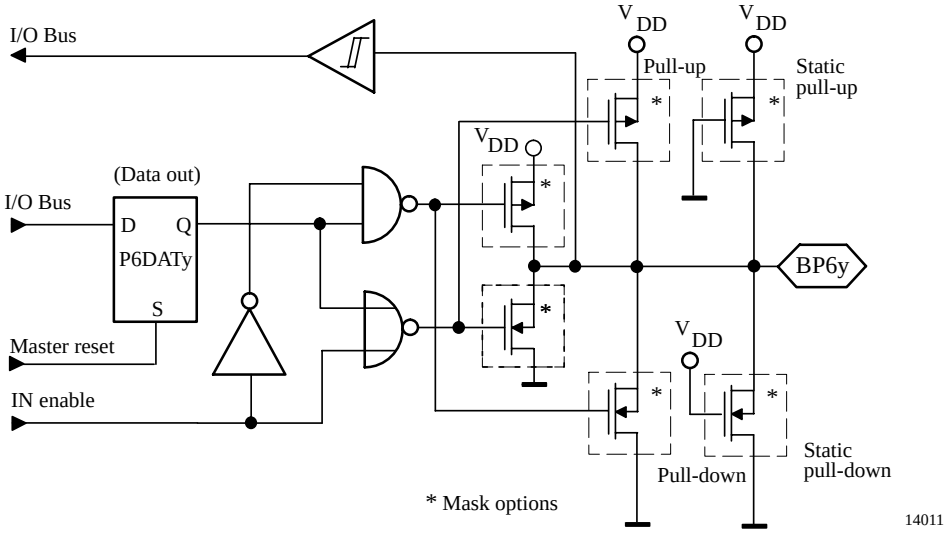
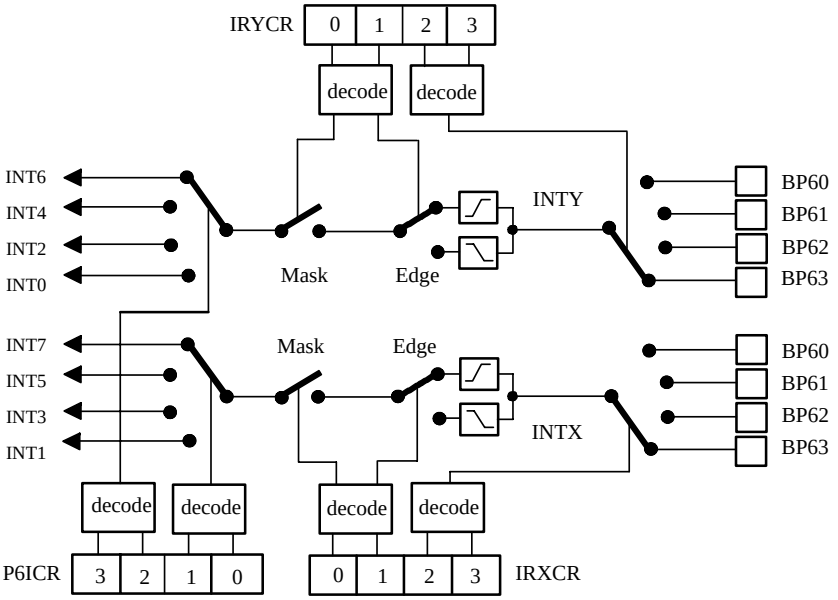


Figure 21. Bidirectional Port 6



P6ICR					
3	2	INTY	1	0	INTX
0	0	INT6	0	0	INT7
0	1	INT4	0	1	INT5
1	0	INT2	1	0	INT3
1	1	INT0	1	1	INT1

IRXCR			IRYCR			
3	2	INT source	1	0	INT edge	INT masked
0	0	BP60	0	0	–	yes
1	0	BP61	0	1	⌐	no
0	1	BP62	1	0	¬	no
1	1	BP63	1	1	–	yes

96 1152

Figure 22. Port 6 external interrupts

Port 6 Interrupt Priority Register (P6IPR)

Auxiliary register address: ’5’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
P6IPR	**PRY2**	**PRY1**	**PRX2**	**PRX1**	**Reset value: 1111b**

PRY2, PRY1 – Interrupt Y priority code

PRX2, PRX1 – Interrupt X priority code

Table 10.Port 6 interrupt priority register (P6IPR)

Code 3 2 1 0	Function
x x 1 1	Interrupt X = priority 1
x x 1 0	Interrupt X = priority 3
x x 0 1	Interrupt X = priority 5
x x 0 0	Interrupt X = priority 7
1 1 x x	Interrupt Y = priority 0
1 0 x x	Interrupt Y = priority 2
0 1 x x	Interrupt Y = priority 4
0 0 x x	Interrupt Y = priority 6

Interrupt X Control Register (IRXCR)

Auxiliary register address: ’6’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
IRXCR	**ISX2**	**ISX1**	**ITRX**	**IMX**	**Reset value: 1111b**

ISX2, ISX1 – Interrupt X source code

ITRX – Interrupt X transition

IMX – Interrupt X mask

Table 11. Interrupt X control register (IRXCR)

Code 3 2 1 0	Function
x x x 0	Interrupt X enabled
x x x 1	Interrupt X masked
x x 0 x	Interrupt X = rising edge
x x 1 x	Interrupt X = falling edge
1 1 x x	Interrupt source = BP63
0 1 x x	Interrupt source = BP62
1 0 x x	Interrupt source = BP61
0 0 x x	Interrupt source = BP60

Interrupt Y Control Register (IRYCR)

Auxiliary register address: ’7’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
IRYCR	**ISY2**	**ISY1**	**ITRY**	**IMY**	**Reset value: 1111b**

ISY2, ISY1 – Interrupt Y source code

ITRY – Interrupt Y transition

IMY – Interrupt Y mask

Table 12. Interrupt Y control register (IRYCR)

Code 3 2 1 0	Function
x x x 0	Interrupt Y enabled
x x x 1	Interrupt Y masked
x x 0 x	Interrupt Y = rising edge
x x 1 x	Interrupt Y = falling edge
1 1 x x	Interrupt source = BP63
0 1 x x	Interrupt source = BP62
1 0 x x	Interrupt source = BP61
0 0 x x	Interrupt source = BP60

2.5 Bidirectional Port 4

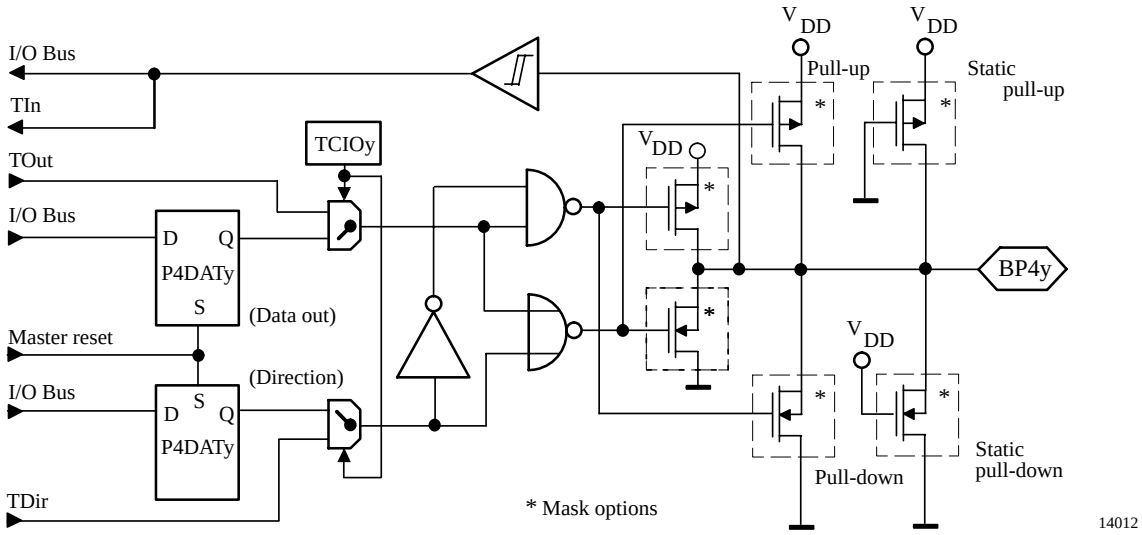


Figure 23. Bidirectional Port 4

The bidirectional Port 4 is both a bitwise configurable I/O port and provides the external pins for both the Timer 0 and the Timer 1. As a normal port, it performs in exactly the same way as bidirectional port type 2 (see figure 14). Two additional multiplexers allow data and port direction control to be passed over to other internal modules (Timer 0 or Timer 1). Each of the four Port 4 pins can be individually switched by the Timer/Counter I/O Register (TCIO). Figure 23 shows the internal interfaces to Port 4.

2.6 Interval Timers / Prescaler

The interval timers are based on a frequency divider for generating two independent time base interrupts. It is driven by SUBCL generated by the clock module (see figure 10) and consists of a 15-stage binary divider and two programmable multiplexers for selecting the appropriate interrupt frequencies for each interrupt source (see figure 24). Each multiplexer is completely independent and is controlled by the common Interval Timer Frequency Select Register (ITFSR). Buffer registers store the respective frequency select codes and ensure complete programming independence of each interrupt channel.

Interrupt masking and programming of the interrupt priority levels is performed with the aid of the Interval Timer Interrupt Priority Register (ITIPR).

2.6.1 Interval Timer Registers

The Interval Timer Frequency Select Register (ITFSR) is I/O mapped to the primary address register of the prescaler/ interval timer address (’F’hex) and the Interval Timer Interrupt Priority Register (ITIPR) to the corresponding auxiliary register.

The interrupt masks MIA and MIB enable interrupt masking of INTA and INTB respectively. Each interrupt source can be programmed with PRA and PRB to one of two interrupt priority levels. Disabling both interrupts resets the interval timer and it’s divider chain.

Interval Timer Interrupt Priority Register (ITIPR)

Auxiliary register address (write only): 'F'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
ITIPR	**PRB**	**PRA**	**MIB**	**MIA**	**Reset value: 1111b**

PRB – Priority select Interval Timer Interrupt INTB

PRA – Priority select Interval Timer Interrupt INTA

MIB – Mask Interval Timer Interrupt INTB

MIA – Mask Interval Timer Interrupt INTA

Table 13.Interval Timer Interrupt Priority Register (ITIPR)

Code 3 2 1 0	Function
x x x 1	Mask interrupt A
x x x 0	Enable interrupt A
x x 1 x	Mask interrupt B
x x 0 x	Enable interrupt B
x 1 x x	Interrupt A => priority 1
x 0 x x	Interrupt A => priority 5
1 x x x	Interrupt B => priority 2
0 x x x	Interrupt B => priority 6
x x 1 1	Reset prescaler and halt

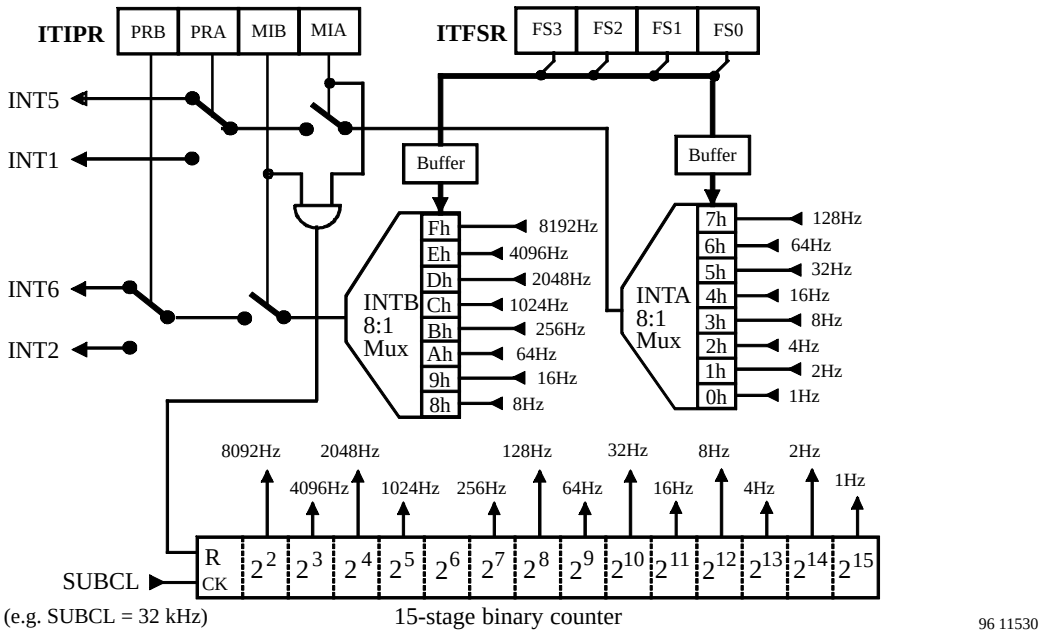


Figure 24. Interval timers / prescaler

Interval Timer Frequency Select Register (ITFSR)

Primary register address (write only): 'F'hex

	Bit 3	Bit 2	Bit 1	Bit 0	
ITFSR	**FS3**	**FS2**	**FS1**	**FS0**	**Reset value: 1111b**

FS3 ... 0 – Frequency select code

Table 14.Interval Timer Frequency Select Register (ITFSR)

Code 3 2 1 0	Function	SUBCL divide by	SUBCL = 32 kHz		Code 3 2 1 0	Function	SUBCL divide by	SUBCL = 32 kHz
0 0 0 0	INTA	2^{15}	Select 1 Hz		1 0 0 0	INTB	2^{12}	Select 8 Hz
0 0 0 1		2^{14}	Select 2 Hz		1 0 0 1		2^{11}	Select 16 Hz
0 0 1 0		2^{13}	Select 4 Hz		1 0 1 0		2^{9}	Select 64 Hz
0 0 1 1		2^{12}	Select 8 Hz		1 0 1 1		2^{7}	Select 256 Hz
0 1 0 0		2^{11}	Select 16 Hz		1 1 0 0		2^{5}	Select 1024 Hz
0 1 0 1		2^{10}	Select 32 Hz		1 1 0 1		2^{4}	Select 2048 Hz
0 1 1 0		2^{9}	Select 64 Hz		1 1 1 0		2^{3}	Select 4096 Hz
0 1 1 1		2^{8}	Select 128 Hz		1 1 1 1		2^{2}	Select 8192 Hz

The control bit FS3 determines whether the INTA or the INTB buffer register is loaded with the select code (FS2–FS0). This allows independent programming of interval times for INTA and INTB.

2.7 Watchdog Timer

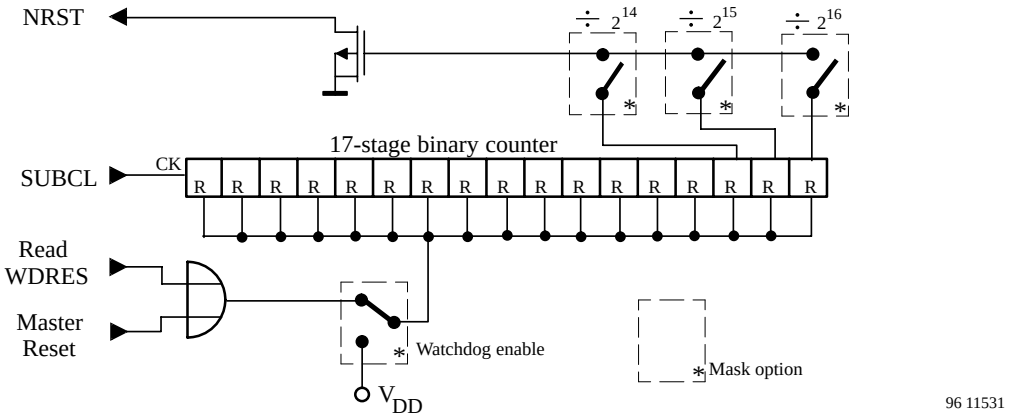


Figure 25. Watchdog timer

The watchdog timer is a 17-stage binary divider clocked by SUBCL generated within the clock module (see figures 10 and 25). It can only be enabled as a mask option whereby it must be periodically reset from the application program. The program cannot disable the watchdog. If the CPU find itself for an extended length of time in SLEEP mode or in a section of program that includes no watchdog reset, then the watchdog will overflow, thus forcing the NRST pin low. This initiates a master reset. The timeout period can be set to 0.5, 1 or 2 seconds (if SUBCL = 32 kHz) by using a mask option.

To reset the watchdog, the program must perform an IN instruction on the address CWD ('3'hex). No relevant data is received. The operation is therefore normally followed by a DROP to flush the data from the stack.

2.8 Timer/ Counter Registers

All timer/ counter registers are indirectly addressed using extended addressing as described in section ‘Addresing Perpherals‘. An overview of all register and subport addresses is shown in table 6. The Timer/ Counter auxiliary register (TCX) holds the subport address of the particular register to be accessed.

Timer 0 Interrupt Status Register (T0SR)

Subport address (read access): ’0’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T0SR	**T0EOM**	**T0OFL**	**T0CMP2**	**T0CMP1**	**Reset value: 0000b**

Note: The Timer 0 status register is reset automatically when read and also when Timer 0 is reset.

T0CMP1, T0CMP2 – Timer 0 compare 1/ compare 2 interrupt status flag
T0OFL – Timer 0 overflow status flag
T0EOM – Timer 0 end of measurement status flag

Table 15.Timer 0 interrupt status register (T0SR)

Code 3 2 1 0	Function
0 0 0 0	No interrupt
x x x 1	Timer 0 compare 1 interrupt event (Timer 0 = T0CP1)
x x 1 x	Timer 0 compare 2 interrupt event (Timer 0 = T0CP2) or external interrupt on BP40
x 1 x x	Timer 0 overflow/ underflow interrupt or external interrupt on BP41
1 x x x	Timer 0 measurement completed

Timer 1 Interrupt Status Register (T1SR)

Subport address (read access): ’1’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T1SR	**T1EOM**	**T1OFL**	**T1CMP2**	**T1CMP1**	**Reset value: 0000b**

Note: The Timer 1 status register is reset automatically when read and also when Timer 1 is reset.

T1CMP1, T1CMP2 – Timer 1 compare 1/ compare 2 interrupt status flag
T1OFL – Timer 1 overflow status flag
T1EOM – Timer 1 end of measurement status flag

Table 16.Timer 0 interrupt status register (T0SR)

Code 3 2 1 0	Function
0 0 0 0	No interrupt
x x x 1	Timer 1 compare 1 interrupt event (Timer 1 = T1CP1)
x x 1 x	Timer 1 compare 2 interrupt event (Timer 1 = T1CP2) or external interrupt on BP42
x 1 x x	Timer 1 overflow/ underflow interrupt or external interrupt on BP43
1 x x x	Timer 1 measurement completed

For both interrupt status registers (T0SR and T1SR) the interrupt flag will be set whenever the assiciated condition occurs irrespective of whether the corresponding interrupt is triggered. So, when the interrupt is masked the status flags will still be set if the interrupt condition occurs. To see exactly when the flags are set, see T0MO and T1MO control tables.

Timer 0 Compare Register 1 (T0CP1)

Subport address (write access): ’8’hex

		Bit 3	Bit 2	Bit 1	Bit 0	
T0CP1	**First write cycle**	**T0CP13**	**T0CP12**	**T0CP11**	**T0CP10**	**Reset value: xxxxb**
		Bit 7	Bit 6	Bit 5	Bit 4	
	Second write cycle	**T0CP17**	**T0CP16**	**T0CP15**	**T0CP14**	**Reset value: xxxxb**

T0CP13 ... T0CP10 – Timer 0 compare register 1 data (low nibble) – first write cycle

T0CP17 ... T0CP14 – Timer 0 compare register 1 data (high nibble) – second write cycle

Timer/ Counter Compare Register (T0CP1)
Subport address: ‘8‘hex Timer/ counter subport pointer (TCX) address: ‘9‘hex

Timer 0 Compare Register 2 (T0CP2)

Subport address (write access): ’A’hex

		Bit 3	Bit 2	Bit 1	Bit 0	
T0CP2	**First write cycle**	**T0CP23**	**T0CP22**	**T0CP21**	**T0CP20**	**Reset value: xxxxb**
		Bit 7	Bit 6	Bit 5	Bit 4	
	Second write cycle	**T0CP27**	**T0CP26**	**T0CP25**	**T0CP24**	**Reset value: xxxxb**

T0CP23 ... T0CP20 – Timer 0 compare register 2 data (low nibble) – first write cycle

T0CP27 ... T0CP24 – Timer 0 compare register 2 data (high nibble) – second write cycle

Timer/ Counter Compare Register (T0CP2)
Subport address: ‘A‘hex Timer/ counter subport pointer (TCX) address: ‘9‘hex

Timer 1 Compare Register 1 (T1CP1)

Subport address (write access): ’9’hex

		Bit 3	Bit 2	Bit 1	Bit 0	
T1CP1	**First write cycle**	**T1CP13**	**T1CP12**	**T1CP11**	**T1CP10**	**Reset value: xxxxb**
		Bit 7	Bit 6	Bit 5	Bit 4	
	Second write cycle	**T1CP17**	**T1CP16**	**T1CP15**	**T1CP14**	**Reset value: xxxxb**

T1CP13 ... T1CP10 – Timer 0 compare register 1 data (low nibble) – first write cycle

T1CP17 ... T1CP14 – Timer 0 compare register 1 data (high nibble) – second write cycle

Timer/ Counter Compare Register (T1CP1)
Subport address: ‘9‘hex Timer/ counter subport pointer (TCX) address: ‘9‘hex

Timer 1 Compare Register 2 (T0CP2)

Subport address (write access): ’B’hex

		Bit 3	Bit 2	Bit 1	Bit 0	
T1CP2	**First write cycle**	**T1CP23**	**T1CP22**	**T1CP21**	**T1CP20**	**Reset value: xxxxb**
		Bit 7	Bit 6	Bit 5	Bit 4	
	Second write cycle	**T1CP27**	**T1CP26**	**T1CP25**	**T1CP24**	**Reset value: xxxxb**

T1CP23 ... T1CP20 – Timer 1 compare register 2 data (low nibble) – first write cycle
T1CP27 ... T1CP24 – Timer 1 compare register 2 data (high nibble) – second write cycle

Timer/ Counter Compare Register (T1CP2)
Subport address: ‘B‘hex Timer/ counter subport pointer (TCX) address: ‘9‘hex

The compare registers (T0CP1, T0CP2, T1CP1 and T1CP2) are all 8-bit wide and must accessed as byte wide subports (see section ‘Addressing Peripherals‘). They are writen low nibble first followed by the high nibble. Any time interrupts are suppressed automatically until the complete compare value has been transferred.

Timer 0 Capture Register (T0CA)

Subport address (read access): ’8’hex

		Bit 3	Bit 2	Bit 1	Bit 0	
T0CA	**Second read cycle**	**T0CA3**	**T0CA2**	**T0CA1**	**T0CA0**	**Reset value: 0000b**
		Bit 7	Bit 6	Bit 5	Bit 4	
	First read cycle	**T0CA7**	**T0CA6**	**T0CA5**	**T0CA4**	**Reset value: 0000b**

T0CA7. .. T0CA4 – Timer 0 capture register data (high nibble) – first read cycle
T0CA3 ... T0CA0 – Timer 0 capture register data (low nibble) – second read cycle

Timer/ Counter Compare Register (T0CP1)
Subport address: ‘8‘hex Timer/ counter subport pointer (TCX) address: ‘9‘hex

Timer 1 Capture Register (T1CA)

Subport address (indirect read access): ’9’hex

		Bit 3	Bit 2	Bit 1	Bit 0	
T1CA	**Second read cycle**	**T1CA3**	**T1CA2**	**T1CA1**	**T1CA0**	**Reset value: 0000b**
		Bit 7	Bit 6	Bit 5	Bit 4	
	First read cycle	**T1CA7**	**T1CA6**	**T1CA5**	**T1CA4**	**Reset value: 0000b**

T1CA7 ... T1CA4 – Timer 1 Capture Register Data (high nibble) – first read cycle
T1CA3 ... T1CA0 – Timer 1 Capture Register Data (low nibble) – second read cycle

Timer/ Counter Compare Register (T0CP1)
Subport address: ‘9‘hex Timer/ counter subport pointer (TCX) address: ‘9‘hex

The 8-bit capture registers (T0CA and T1CA) are read as byte wide subports. Note, however, unlike the writing to the compare registers, the high nibble is read first followed by the low nibble. The 8-bit timer state is captured on reading the first nibble and held until the complete byte has been read. During this transfer the timer is free to continue counting.

Port 4 I/O Control Register (P4IOR)

Subport address (write access): ’F’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
P4IOR	**P4IO3**	**P4IO2**	**P4IO1**	**P4IO0**	**Reset value: 0000b**

P4IO3...P4IO0 – Port 4 I/O select

Table 17.Port 4 I/O control register (P4IOR)

Code 3 2 1 0	Function
x x x 0	BP40 – Timer 0 clock input (T0IN0) or Timer 0 output (T0OUT0)
x x x 1	BP40 – standard port mode
x x 0 x	BP41 – Timer 0 gate input (T0IN1) or Timer 0 output (T0OUT1)
x x 1 x	BP41 – standard port mode
x 0 x x	BP42 – Timer 1 clock input (T1IN0) or Timer 1 output (T1OUT0)
x 1 x x	BP42 – standard port mode
0 x x x	BP43 – Timer 1 gate input (T1IN1) or Timer 1 output (T1OUT1)
1 x x x	BP43 – standard port mode

By using the Port 4 I/O control register (P4IOR) the program can configure the respective Port 4 pins as either standard data I/O ports or as external signal ports for the Timer 0 and Timer 1. It should be noted that if a P4IOR bit is set low, then the corresponding port data direction register (P4DDR) bit not longer influences the port direction. The port direction is then controlled by the corresponding Timer 0/1 mode of operation (T0MO, T1MO).

Timer/ Counter Control Register (TCCR)

Subport address (write access): ’5’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
TCCR	**T0GS**	**TC8**	**TCIO1**	**TCIO0**	**Reset value: 1111b**

TCIO1, TCIO0 – Timer/ counter mode select
TC8 – Timer/ counter in 8-/ 16-bit mode
T0GS – Timer 0 gate select

Table 18.Timer/ counter control register (TCCR)

Code 3 2 1 0	Function
x x x 0	Non-inverted output BP41 appears on BP40 (BP40 = BP41)
x x x 1	Inverted output BP41 appears on BP40 (BP40 = NOT BP41)
x x 0 x	Non-inverted output BP43 appears on BP42 (BP42 = BP43)
x x 1 x	Inverted output BP43 appears on BP42 (BP42 = NOT BP43)
x 0 x x	16-bit mode
x 1 x x	8-bit mode
0 x x x	Timer 0 internal gated by Timer 1
1 x x x	Timer 0 external gated by Port 4

By using the Timer/ counter control register (TCCR) the program can configure the Port 4 pins. With the TCIO0/ TCIO1 bits the Port 4 can programmed as non-inverted or inverted outputs of Timer 0/ Timer 1. In 16-bit mode , Timer 0 and Timer 1 are cascaded thus forming a 16-bit counter (see figure ??) whereby irrespective of the state of Timer 0 interrupt mask bit (T0IM), the Timer 1 counts both Timer 0 overflow and compare interrupt events. These are generated according to the state of the Timer 0 mode register as described in the T0MO table. The comparators are also cascaded so that when both Timer 0 and Timer 1 match their respective compare registers, the Timer 1 generates both an output signal and a compare interrupt (if unmasked).

Port 4 External Interrupt Control Register (P4ICR)

Subport address (write access): ’E’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
P4ICR	**ITR3**	**ITR2**	**ITR1**	**ITR0**	**Reset value: 1111b**

ITR3...ITR0 – Interrupt transition on BP40...BP43

Table 19.Port 4 external interrupt control register (P4ICR)

Code 3 2 1 0	Function
x x x 0	Falling edge on BP40 generates an interrupt
x x x 1	Rising edge on BP40 generates an interrupt
x x 0 x	Falling edge on BP41 generates an interrupt
x x 1 x	Rising edge on BP41 generates an interrupt
x 0 x x	Falling edge on BP42 generates an interrupt
x 1 x x	Rising edge on BP42 generates an interrupt
0 x x x	Falling edge on BP43 generates an interrupt
1 x x x	Rising edge on BP43 generates an interrupt

Timer 0 Interrupt Source Control Register (T0ICR)

Subport address (write access): ’C’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T0ICR	**ITR3**	**ITR2**	**ITR1**	**ITR0**	**Reset value: 1111b**

ITR3...ITR0 – Interrupt transition on BP40...BP43

Table 20.Timer 0 interrupt source control register (T0ICR)

Code 3 2 1 0	Function
x 1 1 1	No interrupt
x x x 0	Compare 1
0 x 0 x	Compare 2
0 0 x x	Overflow or end of measurement
1 x 0 x	External interrupt on BP40/42
1 0 x x	External interrupt on BP41/43

Timer 1 Interrupt Source Control Register (T1ICR)

Subport address (write access): ’D’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
T0ICR	**ITR3**	**ITR2**	**ITR1**	**ITR0**	**Reset value: 1111b**

ITR3...ITR0 – Interrupt transition on BP40...BP43

Table 21.Timer 1 interrupt source control register (T1ICR)

Code 3 2 1 0	Function
x 1 1 1	No interrupt
x x x 0	Compare 1
0 x 0 x	Compare 2
0 0 x x	Overflow or end of measurement
1 x 0 x	External interrupt on BP40/42
1 0 x x	External interrupt on BP41/43

Timer/ Counter Mode Register (TCMO)

Subport address (write access): ’4’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
TCMO	**T1N**	**T0N**	**T1RST**	**T0RST**	**Reset value: 1111b**

T0RST – Timer 0 reset/ run
T1RST – Timer 1 reset/ run
T0N – Timer 0 normal/ autostop
T1N –Timer 1 normal/ autostop

Table 22.Timer/ counter mode register (TCMO)

Code 3 2 1 0	Function
x x x 0	Timer 0 running
x x x 1	Timer 0 reset and halted
x x 0 x	Timer 1 running
x x 1 x	Timer 1 reset and halted
x 0 x x	Timer 0 autostop
x 1 x x	Timer 0 normal
0 x x x	Timer 1 autostop
1 x x x	Timer 1 normal

3 Liquid Crystal Display Driver

This chapter describes the function and the programming of the integrated LCD driver. It also includes

- Information about the relationship between a typical 7 segment display, the segment and backplane outputs (for 3:1 and 4:1 multiplex drive modes)
- Waveform examples for the different LCD drive modes

Figure 26 is a functional block diagram of the LCD driver circuitry. The internal I/O bus is connected to the LCD control register (Port 2 auxiliary register) and the LCD data register (Port 2).

The LCD driver circuitry offers the following features:

- Drives up to 128 display segments
- Supports 3 V or 5 V LCD panels over the full supply voltage range
- Built-in LCD voltage generation with temperature compensation (constant LCD contrast)
- Current consumption of LCD panel adaptable to display size
- Display continues when µC in SLEEP mode
- Programmable multiplex rate (1/3 or 1/4 duty)
- 16 segment drivers configurable by software as bidirectional ports (2-bit wise)

3.1 Display Data Register

The LCD data register receives the data from the µC and writes the data in the shadow register addressed by the address pointer. After any write access the pointer is decremented and the next data can be written in the next data register.

The data in the display buffer is displayed at the LCD. A logical 1 in the display buffer's bit-map indicates the ON state of the corresponding LCD segment. Similiarly a logical 0 indicates the OFF state. There is a 1:1 correspondence between each stage of the buffer register and the segment outputs, and between the individual bits of a buffer nibble and the backplane outputs. The LSB of each nibble corresponds to the 32 segments operated with respect of backplane COM0. In multiplexed LCD applications the segment data of the second, third and fourth column of the display buffer are time multiplexed with COM1, COM2 and COM3 respectively. The LCD specific segment decoding is done via qFORTH software routines, thus omitting the need for separate decoding circuitry.

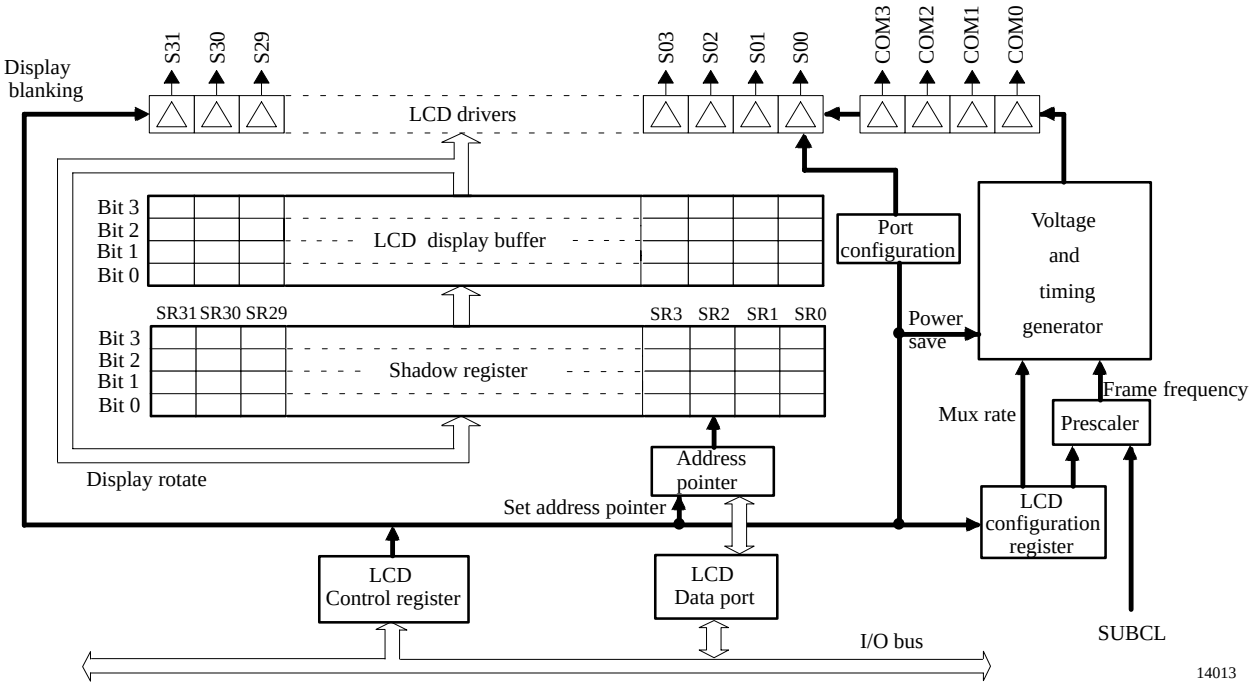


Figure 26. LCD driver – functional block diagram

3.2 Display Control Register

LCD Control Register (LCR)

The LCD control register receives the operation mode at the Port 2 auxiliary register to configure the LCD driver circuitry. The control register also loads the LCD configuration register and port configuration register. The LCD control register is 4-bit wide. Only the upper 3 bits, if non-zero, are significant. If the upper 3 bits are zero the following 5 bits define the address loaded into the address pointer, that means 8 bits must be written.

Auxiliary register address: ’2’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
LCR	**LCR2**	**LCR1**	**LCR0**	**ADR4**	**Reset value: 1111b**

LCR2...LCR0 – LCD Control Register data

ADR4 – Highest address bit if set address pointer

Table 23.LCD Control Register (LCR)

Code 3 2 1 0	Function
1 1 1 1	Powersave
0 0 1 1	Load all segments
0 1 1 1	Display rotate
0 1 0 1	Display normal
1 0 0 1	Blanking
1 0 1 1	Clear/Init
1 1 0 1 c c c c p p p p	Setup LCD/ port configuration with two control nibble –1. LCD configuration register LCFR –2. LCD port configuration register LPCR
0 0 0 a a a a a	Set LCD segment address pointer to binary address ‘aaaaa‘

LCD Frequency Configuration Register (LFCR)

The LCD configuration register is loaded with the SET LCD/PORT CONFIGURATION term (see table 23). The first nibble following this term is loaded in the LCD frequency configuration register as described in table 24. Note, a second nibble must follow (see table 25).

Primary register address: ’2’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
LFCR	**LPF1**	**LPF0**	**LFF**	**LMR**	**Reset value: 0000b**

LPF1...LPF0 – LCD pump frequency

LFF – LCD frame frequency

LMR – LCD multiplex rate

Table 24. LCD Configuration Register (LCFR)

Code 3 2 1 0	Function
x x x 0	Mux 3:1
x x x 1	Mux 4:1
x x 0 x	Frame frequency high (64 Hz at 1/4 duty, 85 Hz at 1/3 duty)
x x 1 x	Frame frequency low (32 Hz at 1/4 duty, 43 Hz at 1/3 duty)
0 0 x x	Pump frequency 2048 Hz
0 1 x x	Pump frequency 1024 Hz
1 0 x x	Pump frequency 512 Hz
1 1 x x	Pump frequency 256 Hz

LCD Port Configuration (LPCR)

The LCD port configuration register is loaded with the SET LCD/PORT CONFIGURATION term (see table 23). The second nibble following this term at the LCD control register (LCR) defines the port configuration as described in table 25.

Primary register address: ’2’hex

	Bit 3	Bit 2	Bit 1	Bit 0	
LPCR	**LPCR3**	**LPCR2**	**LPCR1**	**LPCR0**	**Reset value: 1111 b**

LPCR3...LPCR0 – LCD port configuration

Table 25. LCD Port configuration (LPCR)

Code 3 2 1 0	Function
0 0 0 0	Segment 32 ... 17 acts as bidirectional port
0 0 0 1	Segment 32 ... 19 acts as bidirectional port
0 0 1 0	Segment 32 ... 21 acts as bidirectional port
0 0 1 1	Segment 32 ... 23 acts as bidirectional port
0 1 0 0	Segment 32 ... 25 acts as bidirectional port
0 1 0 1	Segment 32 ... 27 acts as bidirectional port
0 1 1 1	Segment 32 ... 30 acts as bidirectional port
1 x x x	LCD output only

3.2.1 Initializing the LCD Driver

After power-on the LCD driver circuitry is set automatically into 3:1 multiplex drive, LCD-segment output and powersave mode. This means all LCD outputs (COM0...COM3, S01...S32) are at V_{SS} level. The contents of the LCD display buffer and the shadow register are set to 0. The CLEAR/INIT term will initialise the LCD control logic after power-on into a well-defined state.

3.2.2 LCD operating modes

After power-on or the CLEAR/INIT term the LCD display buffer and the shadow register are loaded with 0. With the LOAD ALL SEGMENTS term the whole shadow register is loaded with a data nibble following the term. To display the data at the LCD panel there are two terms available – the DISPLAY NORMAL and the DISPLAY ROTATE term. With the DISPLAY NORMAL term the data available in the shadow register is copied to the LCD display buffer and then displayed at the LCD panel. The DISPLAY ROTATE works in a different way. The data from the shadow register is moved into the LCD display buffer and displayed. The previous data of the LCD display buffer will be available in the shadow register. This term is useful by displaying alternating display information.

The BLANKING term causes a blank display. Note that the contents of LCD display buffer and the shadow register are not influenced by this term.
By using the address pointer a direct read/write access to any nibble of the shadow register is possible. On any write access to the shadow register the address pointer is post-decremented. A shadow register read access after an address pointer setup will suppress the pre-increment cycle for the first data nibble read.

The POWERSAVE term blanks the LCD by switching all the LCD levels to V_{SS}, thus causing a reduction of display power consumption. This mode is only effective if the display is generally to be blanked for periods of more then 5 seconds. The LCD display buffer and the shadow register are unchanged.

This function may also be helpful in radio-controlled clock applications to increase the sognal sensitivity by turning off the noise (EMI) of the switching LCD backplane/ segment outputs.

Example for a LCD shadow register read/write access (see also table):

```
: TogFlags    2h 8h OUT          \ select auxiliary switch register of Port 2
              0001b 2h OUT       \ upper 3 bits = 000, that means the lowest bit of this nibble and the next
              0110b 2h OUT       \ nibble define the address –> load address pointer with 10110b = 16h
              2h IN              \ read the nibble from address 16h without preincr. on first access
              2h IN              \ read the nibble from address 17h (pointer is preincremented)
              0101b XOR          \ XOR with 0101b and nibble of address 17h from the TOS
              2h OUT             \ write result to the address 17h (pointer now 16h)
              0101b XOR          \ XOR with 0101b and nibble of address 16h from the TOS
              2h OUT             \ write result to the address 17h (pointer now 15h)
;
```

3.2.3 Programming the LCD Frequency and Port Configuration Register

To modify the LCD frequency and the LCD port configuration register the following command sequence must be executed without being interrupted (see tables 23...25):

- select the auxiliary switch register of Port 2 (LCR)
- write the SET LCD/PORT CONFIGURATION (110x) command
- output the nibble defining the LCD frequency configuration (LFCR)
- write the nibble defining the Port configuration to the Port 2 address (LPCR)

Example:

```
: SetConfiguration   CCR@  DI          \ disable interrupts
                     2h 8h OUT         \ select the auxiliary switch register of Port 2 (LCR)
                     1100b 2h OUT      \ 2 control nibble following
                     0101b 2h OUT      \ mux 4:1, frame frequency high, pump frequency 1024 Hz
                     0100b 2h OUT      \ seg 31:24 acts as bidirectional ports
                     CCR!              \ restore interrupt status
;
```

3.2.4 Reduction of LCD Charge Pump and Frame Frequency

After any power-on or hardware reset, the LCD voltage generator is in the fast charge mode and the frame frequency is high. This mode is used to quickly charge the capacitance of large displays by using a high charge pumping frequency of the LCD voltage generator. For smaller displays and to reduce the overall system current the M44C588 allows the modification of the charge frequency and frame frequency under software control (see also table 12).

3.2.5 LCD Port Configuration

The M44C588 has 32 segment outputs to drive a LCD panel. The upper 16 segment drivers of the LCD32 module can be configured as bidirectional port pins. The port configuration register allows the reconfiguration of segments pairwise into I/O pins, WHEREBY THE DATA DIRECTION OF EACH I/O pin is bitwise programmable

(see table 25). The port data and direction information is mapped into the LCD shadow register cells SR31 down to SRn as shown in figure 27. Because each block of 4 I/O pins only require two register cells (4 bit for data register and 4 bit for data direction register), only the upper two cells are used, the upper cell acting as data register and the one below as the data direction register. For example the MSBs of these two cells provide the data direction information for the upper I/O pin.

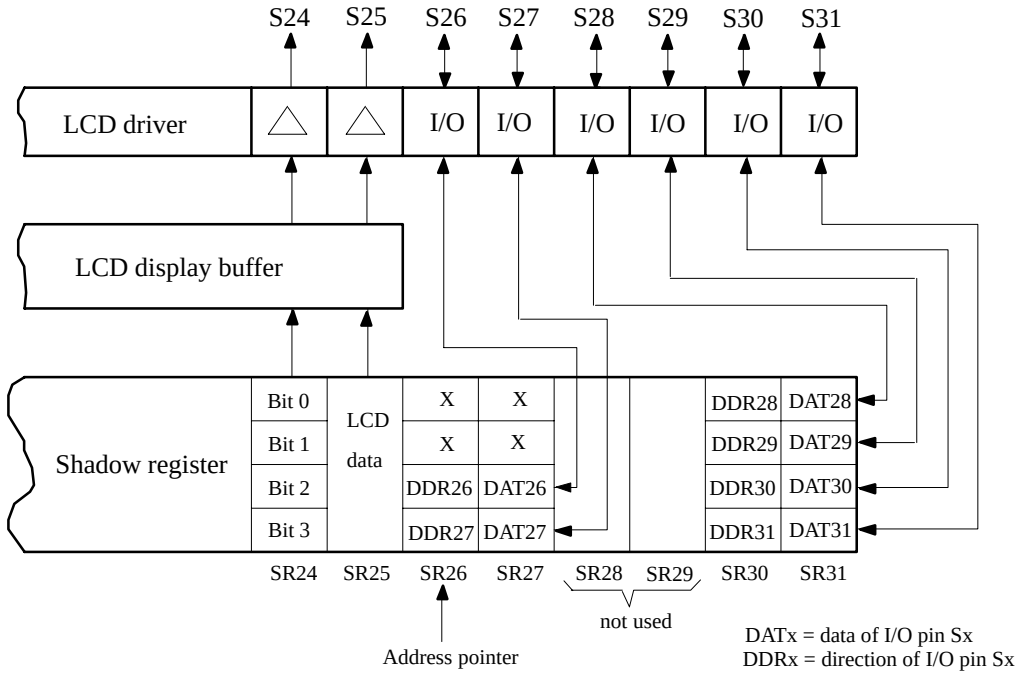


Figure 27. LCD 32 – 6 segment outputs used as I/O ports

If two I/O pins are used, the upper two cells are used as data direction and data register and the shadow register cells below are used for driving the LCD segments. In this case only the upper two bits are significant for the port and port data direction (see figure 27).

The access to the port direction and data register is identical to the LCD shadow register read/write access. A data direction (logical '1') bit sets the corresponding I/O pin into output mode and a logical '0' into input mode

3.3 LCD Voltage and Timing Generator

The LCD voltage generator circuitry boosts the regulated liquid crystal display voltage ($\mathbf{V_{REG}}$) to the doubled and tripled voltage components ($\mathbf{V_{EE1}}$, $\mathbf{V_{EE2}}$) required by multiplexed liquid crystal displays. These voltage levels are applied to the driver circuitry (see figure 28).

Most low voltage (3 V) LCD panels have a temperature coefficient of –6 mV/°C. The temperature compensated reference for the LCD voltage booster circuitry ($\mathbf{V_{REG}}$), has the task of meeting this requirement directly, so that the user gets the best LCD contrast over the full operating temperature and supply voltage range. The external components for the LCD voltage generation (one pump and two storage capacitors) should be connected to the µC as shown in figure 28. For very small LCD panels the capacitor values and charge pumping frequency may be reduced to save costs and system current. The capacitor values may be reduced from 100 nF to 47 nF. The user has to connect the µC and the LCD as it will be in the final product in order to select the capacitor value. To examine the LCD driver waveforms, an oscilloscope with a low capacitance probe should be used.

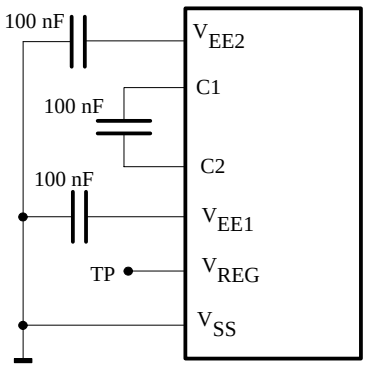


Figure 28. External components

3.4 3:1 Multiplex Drive Mode

Figure 27 shows the connection of a 3:1 multiplex LCD panel having the numeric display pattern shown in figure 29, the segment outputs (**S00-S31**), and the backplane outputs (**COM0-COM2**).

Backplane and segment drive waveforms for this mode are shown in figure 31.

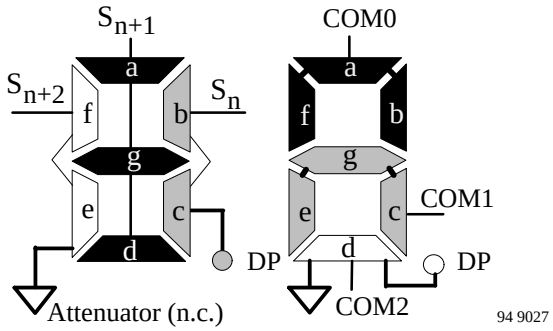


Figure 29. 3:1 multiplex 7 segment digit

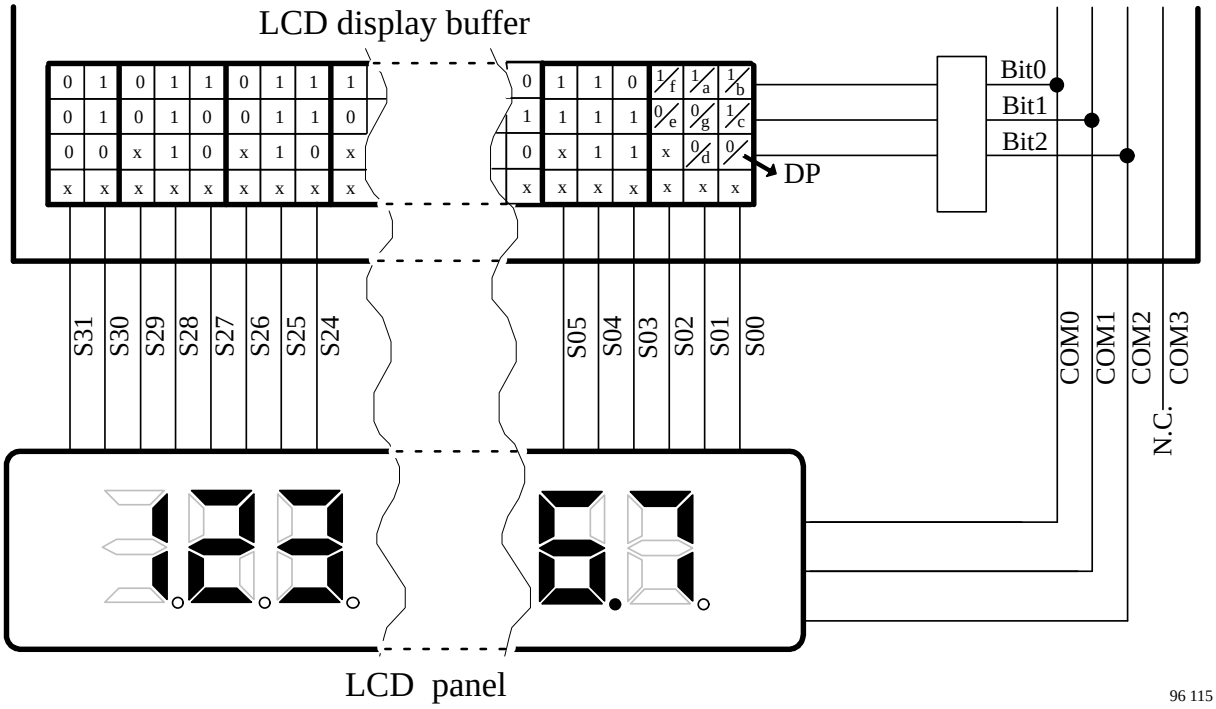


Figure 30. 3:1 multiplex LCD panel connection

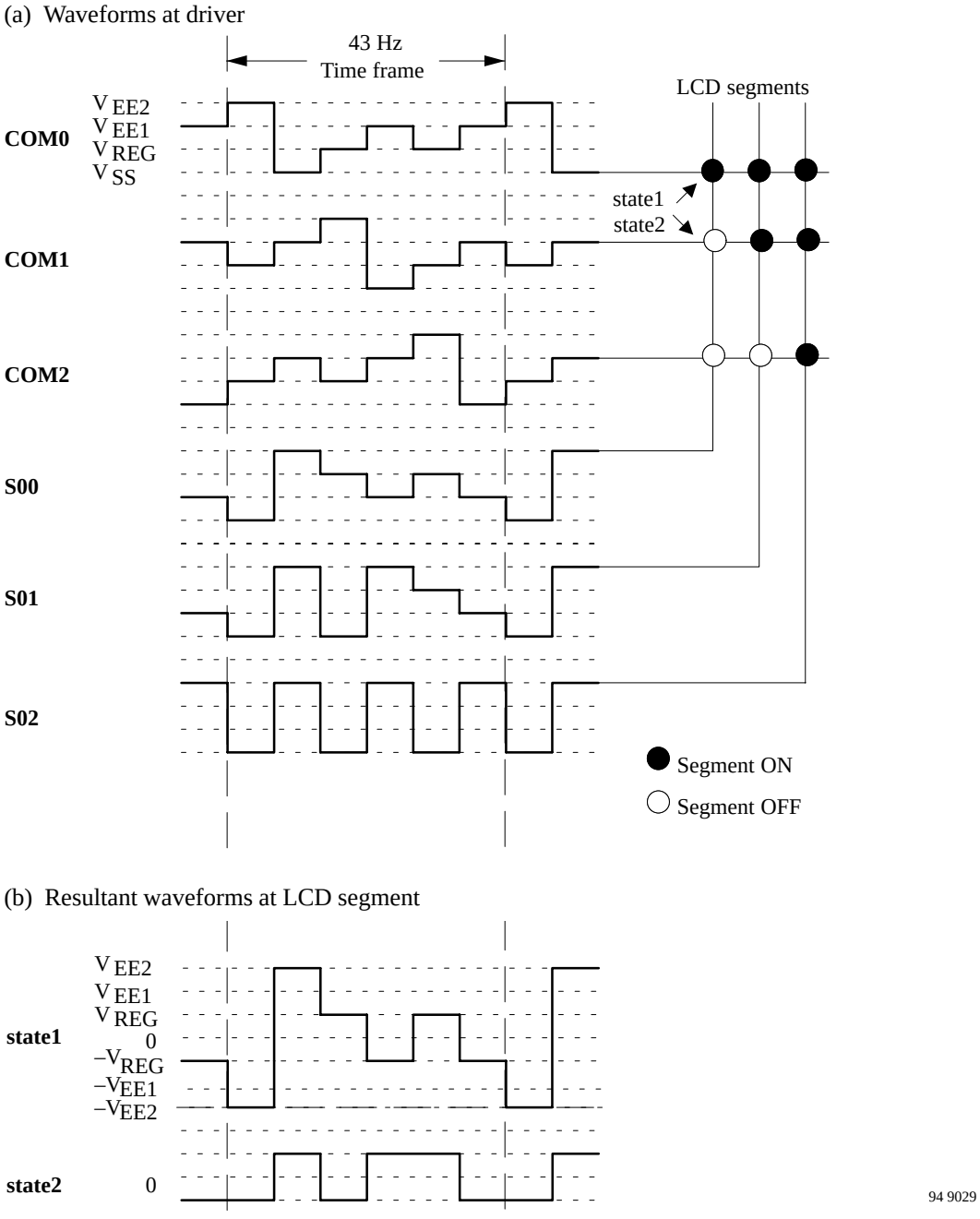


Figure 31. Waveforms for 3:1 multiplex drive mode

The following formulas are valid in the 3:1 multiplex drive mode at any instant (t):

$V_{State1}(t) = V_{S01}(t) - V_{COM0}(t)$ and $V_{State2}(t) = V_{S02}(t) - V_{COM1}(t)$

$V_{ON(rms)} = \frac{V_{EE2}}{9}\sqrt{33} = 0.638\ V_{EE2}$ and $V_{OFF(rms)} = \frac{V_{EE2}}{3}$

Contrast ratio = $V_{ON(rms)} / V_{OFF(rms)} = 1.915$

3.5 4:1 Multiplex Drive Mode

Figure 33 shows the connection of a 4:1 multiplex 16 digit LCD panel having the numeric display pattern shown in figure 32, the segment outputs (**S00-S31**), and the back-plane outputs (**COM0-COM3**).

Backplane and segment drive waveforms for this mode are shown in figure 34.

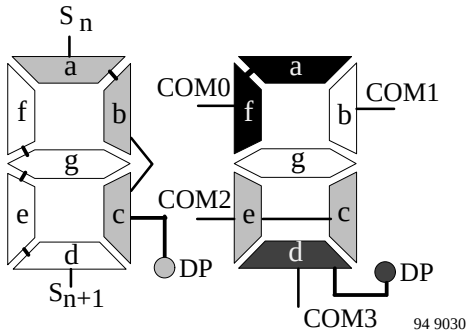


Figure 32. 4:1 multiplex 7 segment digit

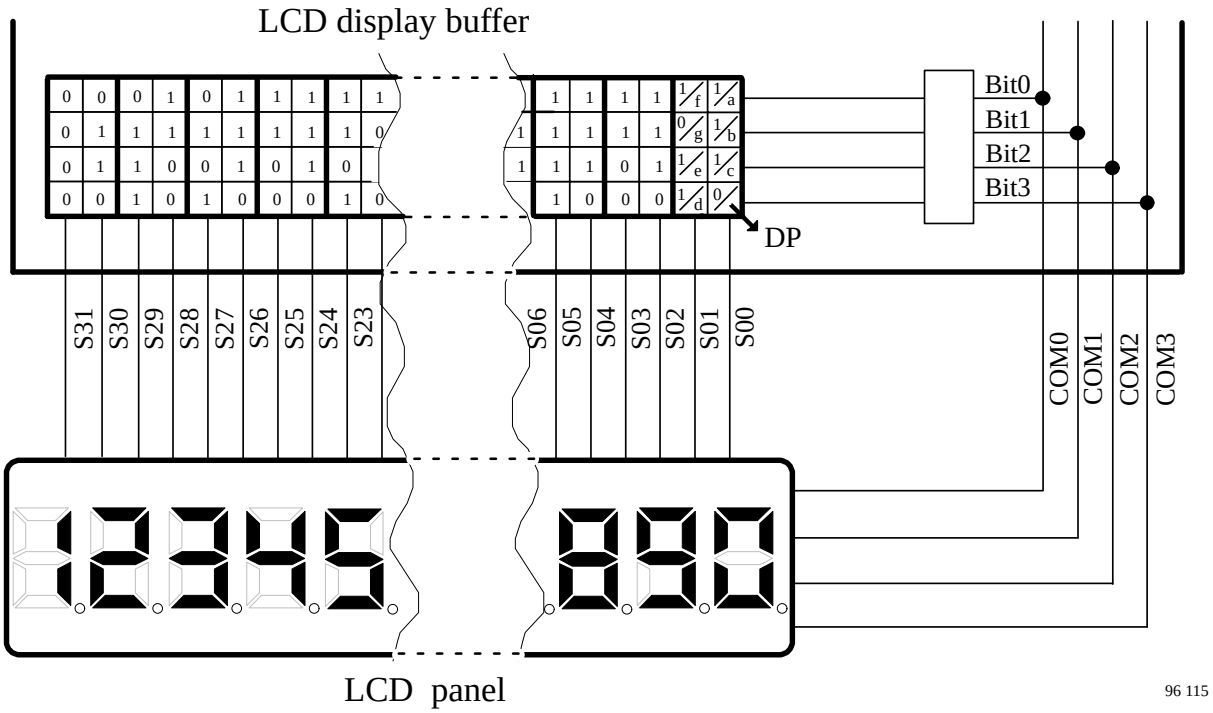
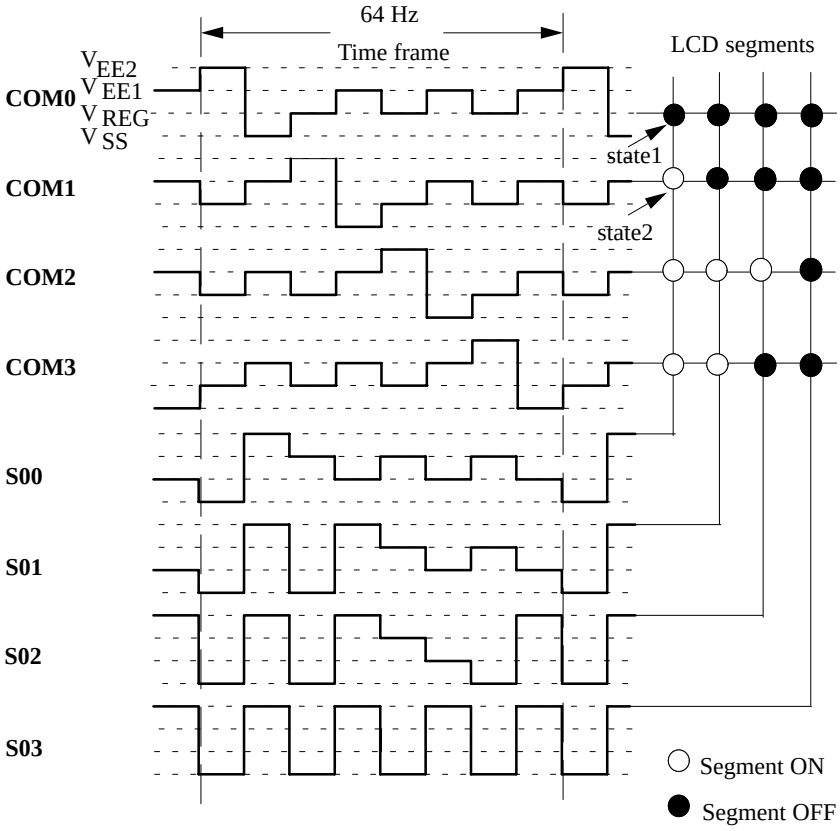
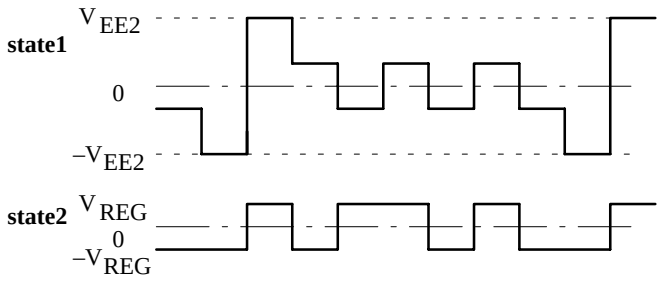


Figure 33. 4:1 multiplex LCD panel connection

(a) Waveforms at driver



(b) Resultant waveforms at LCD segment



94 9032

Figure 34. Waveforms for 4:1 multiplex drive mode

The following formulas are valid in the 4:1 multiplex drive mode at any instant (t):

$V_{State1}(t) = V_{S01}(t) - V_{COM0}(t)$ and $V_{State2}(t) = V_{S02}(t) - V_{COM1}(t)$

$V_{ON(rms)} = \frac{V_{EE2}}{3}\sqrt{3} = 0.577\ V_{EE2}$ and $V_{OFF(rms)} = \frac{V_{EE2}}{3}$

Contrast ratio $= V_{ON(rms)} / V_{OFF(rms)} = 1.732$

4 Extended RAM

The M44C588 contains a 256 × 4-bit wide extended random access memory (XRAM). The extended RAM is addressed by the 8-bit wide peripheral RAM address register. This is loaded trough the auxiliary switch register of port address C.
The extended RAM allows random access to any of the 256 data nibbles and supports postdecrement after any WRITE access and pre-increment after any READ access.

A peripheral RAM read access after a new address setup/ write cycle will suppress the pre-increment cycle before the addressed data nibble is read. This will deliver the specified data nibble on TOS.

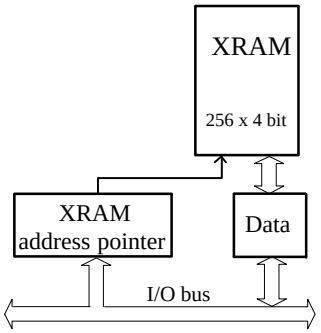


Figure 35. Extended RAM

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Voltages are given relative to V_{SS} .

Parameters	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to + 6.5	V
Input voltage (on any pin)	V_{IN}	V_{SS} –0.3 ≤V_{IN} ≤V_{DD} +0.3	V
Output short circuit duration	t_{short}	indefinite	sec
Operating temperature range	T_{amb}	–40 to +85	°C
Storage temperature range	T_{stg}	–40 to +130	°C
Thermal resistance (DIP40)	R_{thJA}	110	K/W
Soldering temperature (t ≤ 10 sec)	T_{sd}	260	°C

Stresses greater than 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 any condition above those indicated in the operational section of these specification is not implied. Exposure to absolute maximum rating condition for an extended period may affect device reliability. All inputs and outputs are protected against high electrostatic voltages or electric fields. However, precautions to minimize built-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}).

TEMIC TELEFUNKEN microelectronic GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany
Telephone: 49 (0)7131 67 2831, Fax Number: 49 (0)7131 67 2423