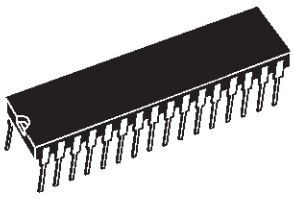
ST72141K

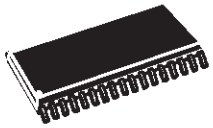
8-BIT MCU WITH ELECTRIC-MOTOR CONTROL, ADC, 16-BIT TIMERS, SPI INTERFACE

PRELIMINARY DATA

- **Memories**
  - 8K Program memory (ROM/OTP/EPROM)
  - 256 Bytes RAM
- **Clock, Reset and Supply Management**
  - Enhanced reset system
  - Low voltage supply supervisor
  - 3 Power saving modes
- **14 I/O Ports**
  - 14 multifunctional bidirectional I/O lines with: External interrupt capability (2 vectors), 13 alternate function lines, 3 high sink outputs
- **Motor Control peripheral**
  - 6 PWM output channels
  - Emergency pin to force outputs to HiZ state
  - 3 analog inputs for rotor position detection with no need for additional sensors
  - Comparator for current limitation
- **3 Timers**
  - Two 16-bit timers with: 2 input captures, 2 output compares, external clock input, PWM and Pulse generator modes
  - Watchdog timer for system integrity
- **Communications Interface**
  - SPI synchronous serial interface
- **Analog Peripheral**
  - 8-bit ADC with 8 input pins



SDIP32



SO34S

- **Instruction Set**
  - 8-bit data manipulation
  - 63 basic instructions
  - 17 main addressing modes
  - 8 x 8 unsigned multiply instruction
  - True bit manipulation
- **Development Tools**
  - Full hardware/software development package

Device Summary

Features	ST72141K2
Program memory - bytes	8K
RAM (stack) - bytes	256 (64)
Peripherals	Motor control, Watchdog, Two 16-bit timers, SPI, ADC
Operating Supply	4V to 5.5V
CPU Frequency	4 or 8 MHz (with 8 or 16 MHz oscillator)
Operating Temperature	-40°C to +85°C
Packages	SO34 / SDIP32

Rev. 1.6

Table of Contents

Table of Contents

Table of Contents

1 GENERAL DESCRIPTION

1.1 INTRODUCTION

The ST72141K devices are members of the ST7 microcontroller family designed specifically for motor control applications and including A/D conversion and SPI interface capabilities. They include an on-chip Moter Controller peripheral for control of electric brushless moters with or without sensors. An example application, for 6-step control of a Permanent Magnet DC motor, is shown in Figure 1.

The ST72141K devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

Under software control, they can be placed in WAIT, SLOW, or HALT mode, reducing power consumption when the application is in idle or standby state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

Figure 1. Example of a 6-step-controlled Motor

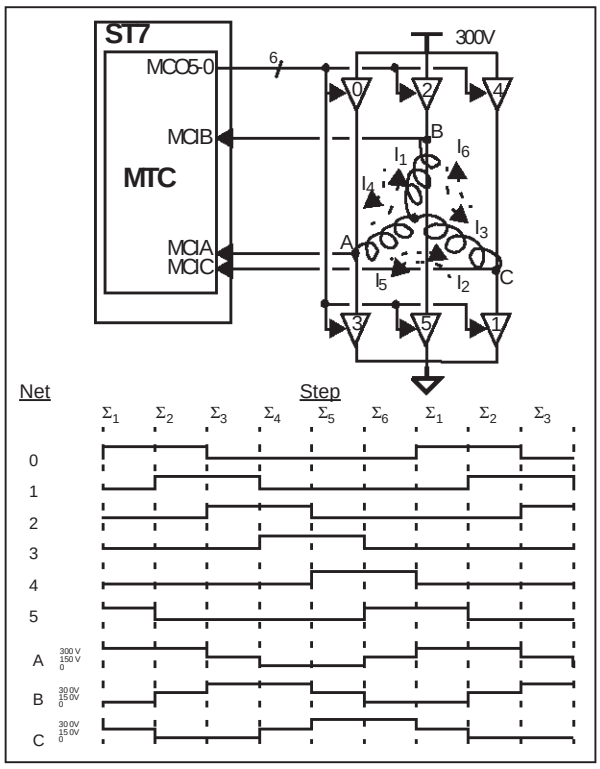
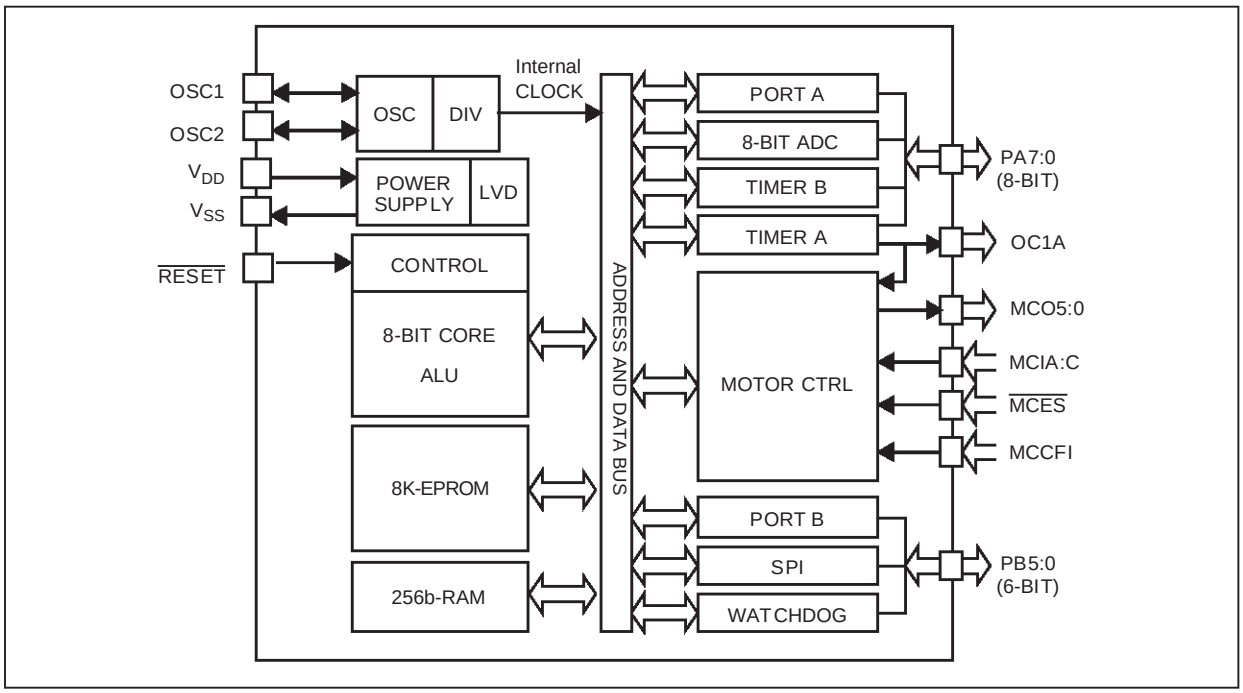


Figure 2. Device Block Diagram



1.2 PIN DESCRIPTION

Figure 3. 34-Pin SO Package Pinout

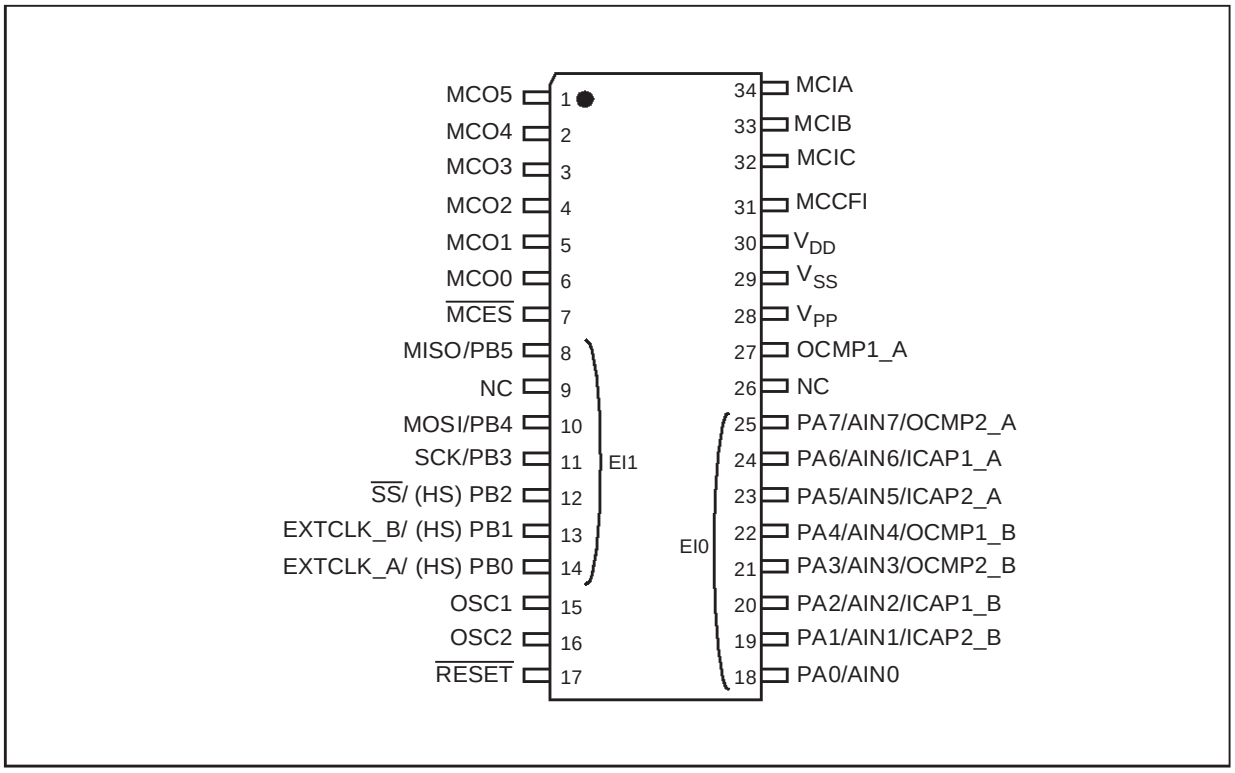
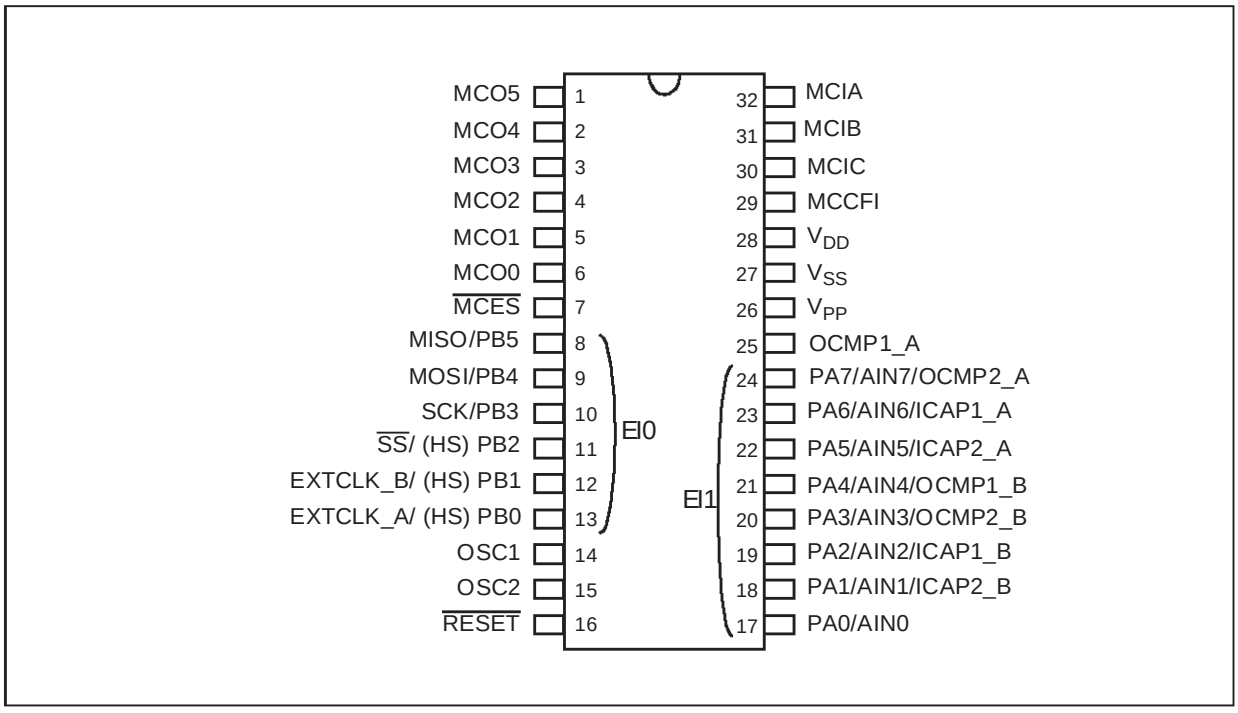


Figure 4. 32-Pin SDIP Package Pinout



PIN DESCRIPTION (Cont'd)

Legend / Abbreviations:

Type: I = input, O = output, S = supply

Input level: A = Dedicated analog input

In/Output level: C = CMOS $0.3V_{DD}/0.7V_{DD}$,
C_T= CMOS $0.3V_{DD}/0.7V_{DD}$ with input trigger

Output level: HS = high sink (on N-buffer only),
R = 70Ω/100Ω ratio of logical levels.
Analog level if used as PWM filtered with an external capacitor

Port configuration capabilities:

– Input: float = floating, wpu = weak pull-up, int = interrupt, ana = analog

– Output: OD = open drain, T = true open drain, PP = push-pull

Note: the Reset configuration of each pin is shown in bold.

Table 1. Device Pin Description

Pin n°		Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
						Input				Output			
SDIP32	SO34			Input	Output	float	wpu	int	ana	OD	PP		
1	1	MCO5	O		C						X	Motor Control Output Channel 5	
2	2	MCO4	O		C						X	Motor Control Output Channel 4	
3	3	MCO3	O		C						X	Motor Control Output Channel 3	
4	4	MCO2	O		C						X	Motor Control Output Channel 2	
5	5	MCO1	O		C						X	Motor Control Output Channel 1	
6	6	MCO0	O		C						X	Motor Control Output Channel 0	
7	7	$\overline{MCES}$	I	C_T			X					Motor Control Emergency Stop Input	
8	8	PB5/MISO	I/O	C_T		**X**	EI1			X	X	Port B5	SPI Master In / Slave Out Data
	9	NC	Not Connected										
9	10	PB4/MOSI	I/O	C_T		**X**	EI1			X	X	Port B4	SPI Master Out / Slave In Data
10	11	PB3/SCK	I/O	C_T		X	EI1			X	X	Port B3	SPI Serial Clock
11	12	PB2/$\overline{SS}$	I/O	C_T	HS	X	EI1			T		Port B2	SPI Slave Select (active low)
12	13	PB1/EXTCLK_B	I/O	C_T	HS	X	EI1			T		Port B1	Timer B Input Clock
13	14	PB0/EXTCLK_A	I/O	C_T	HS	X	EI1			T		Port B0	Timer A Input Clock
14	15	OSC1										These pins connect a crystal or ceramic resonator, or an external RC, or an external source to the on-chip oscillator	
15	16	OSC2											
16	17	$\overline{RESET}$	I/O	C			X			X		Top priority non maskable interrupt (active low)	
17	18	PA0/AIN0	I/O	C_T		**X**	EI0		X	X	X	Port A0	ADC Analog Input 0
18	19	PA1/ICAP2_B/AIN1	I/O	C_T		**X**	EI0		X	X	X	Port A1	Timer B Input Capture 2 or ADC Analog Input 1
19	20	PA2/ICAP1_B/AIN2	I/O	C_T		**X**	EI0		X	X	X	Port A2	Timer B Input Capture 1 or ADC Analog Input 2

Pin n°		Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
						Input				Output			
SDIP32	SO34			Input	Output	float	wpu	int	ana	OD	PP		
20	21	PA3/OCMP2_B/AIN3	I/O	C_T		**X**	EI0		X	X	X	Port A3	Timer B Output Compare 2 or ADC Analog Input 3
21	22	PA4/OCMP1_B/AIN4	I/O	C_T		**X**	EI0		X	X	X	Port A4	Timer B Output Compare 1 or ADC Analog Input 4
22	23	PA5/ICAP2_A/AIN5	I/O	C_T		**X**	EI0		X	X	X	Port A5	Timer A Input Capture 2 or ADC Analog Input 5
23	24	PA6/ICAP1_A/AIN6	I/O	C_T		**X**	EI0		X	X	X	Port A6	Timer A Input Capture 1 or ADC Analog Input 6
24	25	PA7/OCMP2_A/AIN7	I/O	C_T		**X**	EI0		X	X	X	Port A7	Timer A Output Compare 2 or ADC Analog Input 7
	26	NC	Not Connected										
25	27	OCMP1_A	O		R							Timer A Output Compare 1	
26	28	V_{PP}	I									Must be tied low during normal operating mode,EPROM Programming voltage pin.	
27	29	V_{SS}	S									Ground	
28	30	V_{DD}	S									Main power supply	
29	31	MCCFI	I	A								Motor Control Current Feedback Input	
30	32	MCIC	I	A								Motor Control Input C	
31	33	MCIB	I	A								Motor Control Input B	
32	34	MCIA	I	A								Motor Control Input A	

1.3 EXTERNAL CONNECTIONS

The following figure shows the recommended external connections for the device.

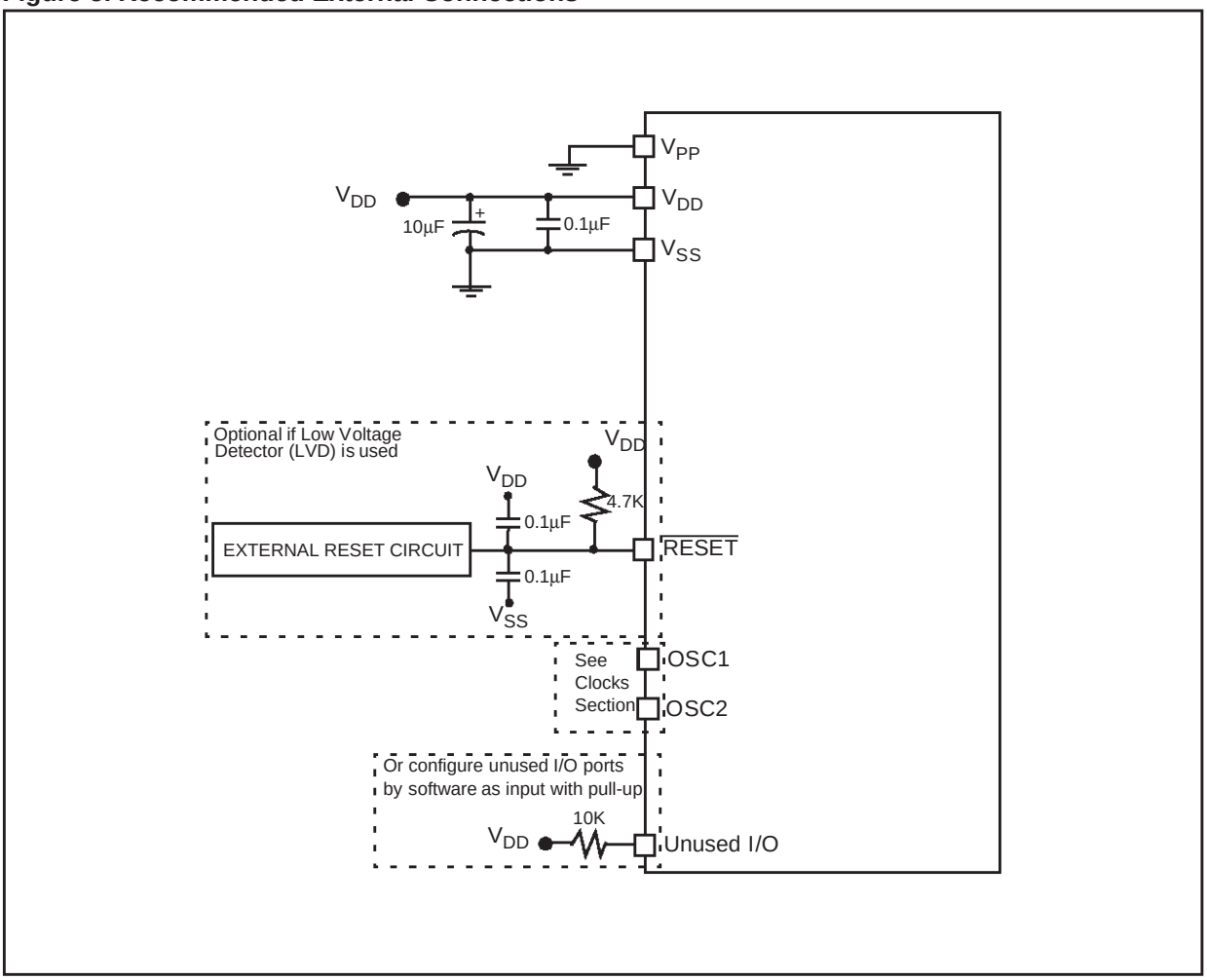
The V_{PP} pin is only used for programming OTP and EPROM devices and must be tied to ground in user mode.

The 10 nF and 0.1 µF decoupling capacitors on the power supply lines are a suggested EMC performance/cost tradeoff.

The external reset network is intended to protect the device against parasitic resets, especially in noisy environments.

Unused I/Os should be tied high to avoid any unnecessary power consumption on floating lines. An alternative solution is to program the unused ports as inputs with pull-up.

Figure 5. Recommended External Connections



1.4 REGISTER & MEMORY MAP

As shown in Figure 6, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, 256 bytes of RAM and 8Kbytes of user program memory. The RAM space includes up to 64 bytes for the stack from 0140h to 017Fh.

The highest address bytes contain the user reset and interrupt vectors.

Figure 6. Memory Map

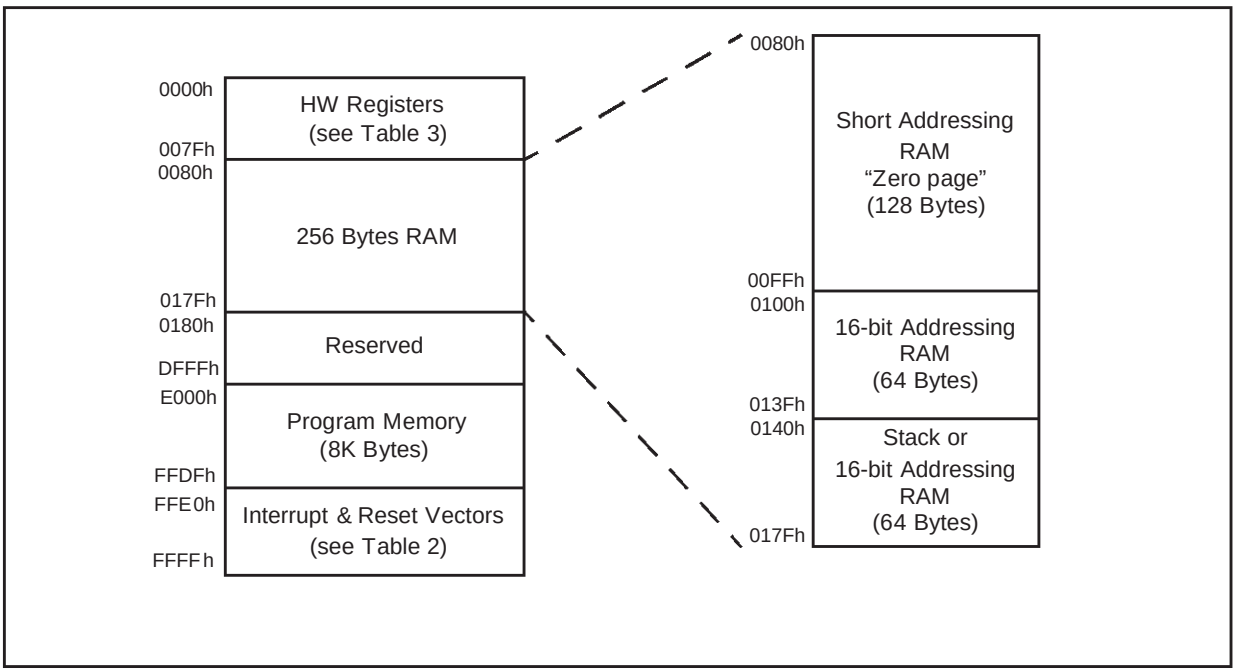


Table 2. Interrupt Vector Map

Vector Address	Description	Remarks
FFE0-FFE1h	Not used	Internal Interrupt
FFE2-FFE3h	Not used	
FFE4-FFE5h	Not used	
FFE6-FFE7h	Not used	
FFE8-FFE9h	Not used	
FFEA-FFEBh	TIMER B interrupt vector	
FFEC-FFEDh	TIMER A interrupt vector	
FFEE-FFEFh	SPI interrupt vector	
FFF0-FFF1h	Motor control interrupt vector (events: E, O)	
FFF2-FFF3h	Motor control interrupt vector (events: C, D)	
FFF4-FFF5h	Motor control interrupt vector (events: R, Z)	
FFF6-FFF7h	External interrupt vector EI1: port B7..0	External Interrupt
FFF8-FFF9h	External interrupt vector EI0: port A7..0	External Interrupt
FFFA-FFFBh	Not used	
FFFC-FFFDh	TRAP (software) interrupt vector	CPU Interrupt
FFFE-FFFFh	RESET vector	

Table 3. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h	Port A	PADR	Port A Data Register	00h	R/W
0001h		PADDR	Port A Data Direction Register	00h	R/W
0002h		PAOR	Port A Option Register	00h	R/W
0003h	Reserved Area (1 Byte)				
0004h	Port B	PBDR	Port B Data Register	00h	R/W
0005h		PBDDR	Port B Data Direction Register	00h	R/W
0006h		PBOR	Port B Option Register	00h	R/W.
0007h to 001F	Reserved Area (24 Byte)				
0020h		MISCR	Miscellaneous Register	00h	R/W
0021h	SPI	SPIDR	SPI Data I/O Register	xxh	R/W
0022h		SPICR	SPI Control Register	0xh	R/W
0023h		SPISR	SPI Status Register	00h	Read Only
0024h	WATCHDOG	WDGCR	Watchdog Control Register	7Fh	R/W
0025h		WDGSR	Watchdog Status Register	x0h	Read Only
0026h to 0030h	Reserved Area (11 Bytes)				
0031h	TIMER A	TACR2	Timer A Control Register 2	00h	R/W
0032h		TACR1	Timer A Control Register 1	00h	R/W
0033h		TASR	Timer A Status Register	xxh	Read Only
0034h		TAIC1HR	Timer A Input Capture 1 High Register	xxh	Read Only
0035h		TAIC1LR	Timer A Input Capture 1 Low Register	xxh	Read Only
0036h		TAOC1HR	Timer A Output Compare 1 High Register	80h	R/W
0037h		TAOC1LR	Timer A Output Compare 1 Low Register	00h	R/W
0038h		TACHR	Timer A Counter High Register	FFh	Read Only
0039h		TACLR	Timer A Counter Low Register	FCh	Read Only
003Ah		TAACHR	Timer A Alternate Counter High Register	FFh	Read Only
003Bh		TAACLR	Timer A Alternate Counter Low Register	FCh	Read Only
003Ch		TAIC2HR	Timer A Input Capture 2 High Register	xxh	Read Only
003Dh		TAIC2LR	Timer A Input Capture 2 Low Register	xxh	Read Only
003Eh		TAOC2HR	Timer A Output Compare 2 High Register	80h	R/W
003Fh		TAOC2LR	Timer A Output Compare 2 Low Register	00h	R/W
0040h	Reserved Area (1 Byte)				

Address	Block	Register Label	Register Name	Reset Status	Remarks
0041h	TIMER B	TBCR2	Timer B Control Register 2	00h	R/W
0042h		TBCR1	Timer B Control Register 1	00h	R/W
0043h		TBSR	Timer B Status Register	xxh	Read Only
0044h		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
0045h		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
0046h		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
0047h		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
0048h		TBCHR	Timer B Counter High Register	FFh	Read Only
0049h		TBCLR	Timer B Counter Low Register	FCh	Read Only
004Ah		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
004Bh		TBACLR	Timer B Alternate Counter Low Register	FCh	Read Only
004Ch		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
004Dh		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
004Eh		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
004Fh		TBOC2LR	Timer B Output Compare 2 Low Register	00h	R/W
0050h to 005Fh	Reserved Area (16 Bytes)				
0060h	MOTOR CONTROL	MTIM	Timer Counter Register	00h	R/W
0061h		MZPRV	Zn-1 Capture Register	00h	R/W
0062h		MZREG	Zn Capture Register	00h	R/W
0063h		MCOMP	C_{n+1}Compare Register	00h	R/W
0064h		MDREG	D capture/Compare Register	00h	R/W
0065h		MWGHT	Weight Register	00h	R/W
0066h		MPRSR	Prescaler and Ratio Register	00h	R/W
0067h		MIMR	Interrupt Mask Register	00h	R/W
0068h		MISR	Interrupt Status Register	00h	R/W
0069h		MCRA	Control Register A	00h	R/W
006Ah		MCRB	Control Register B	00h	R/W
006Bh		MPHST	Phase State Register	00h	R/W
006Ch		MPAR	Output Parity Register	00h	R/W
006Dh		MPOL	Output Polarity Register	00h	R/W
006Eh to 006Fh	Reserved Area (2 bytes)				
0070h	ADC	ADCDR	Data Register	00h	Read Only
0071h		ADCCSR	Control/Status Register	00h	R/W
0072h to 007Fh	Reserved Area (14 Bytes)				

1.5 EPROM PROGRAM MEMORY

The program memory of the OTP and EPROM devices can be programmed with EPROM programming tools available from STMicroelectronics.

EPROM Erasure

EPROM devices are erased by exposure to high intensity UV light admitted through the transparent window. This exposure discharges the floating gate to its initial state through induced photo current.

It is recommended that the EPROM devices be kept out of direct sunlight, since the UV content of sunlight can be sufficient to cause functional failure. Extended exposure to room level fluorescent lighting may also cause erasure.

An opaque coating (paint, tape, label, etc...) should be placed over the package window if the product is to be operated under these lighting conditions. Covering the window also reduces I_{DD} in power-saving modes due to photo-diode leakage currents.

2 CENTRAL PROCESSING UNIT

2.1 INTRODUCTION

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

2.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

2.3 CPU REGISTERS

The 6 CPU registers shown in Figure 7 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

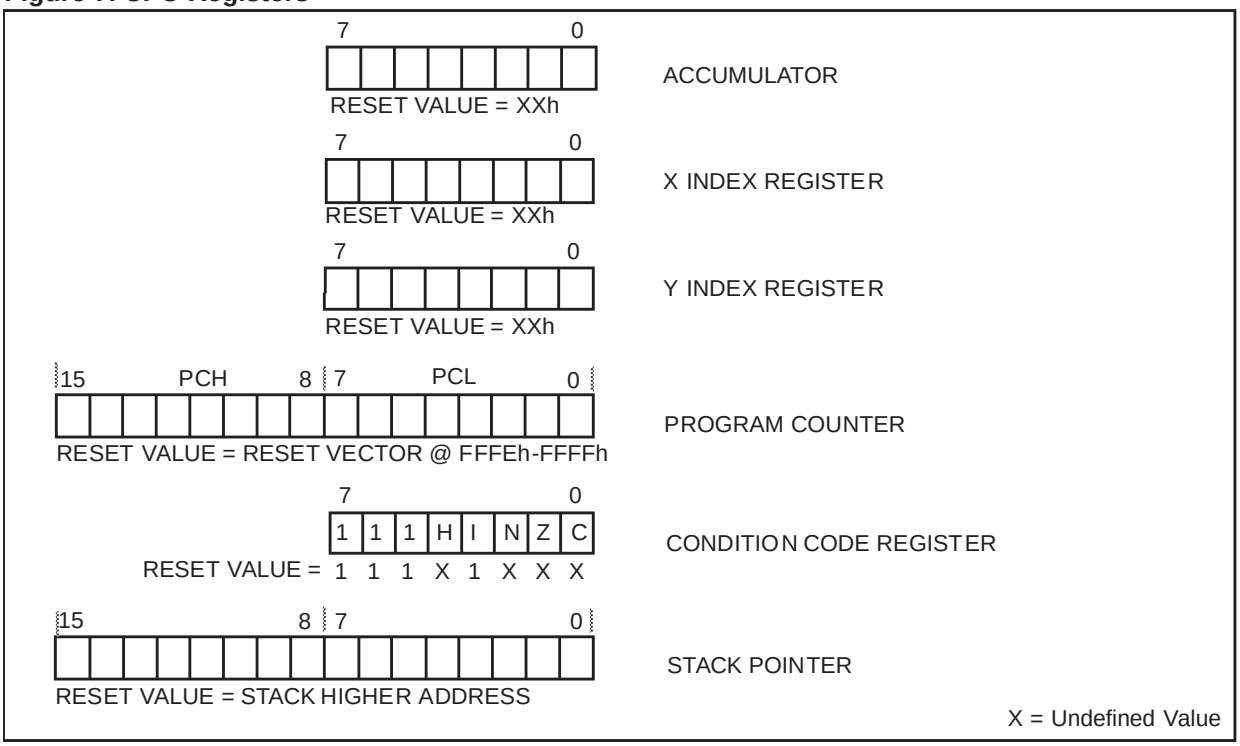
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 7. CPU Registers



CPU REGISTERS (Cont'd)

CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Bit 4 = **H** *Half carry.*

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.
0: No half carry has occurred.
1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 3 = **I** *Interrupt mask.*

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.
0: Interrupts are enabled.
1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptable because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = **N** *Negative.*

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.
0: The result of the last operation is positive or null.
1: The result of the last operation is negative (i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero.*

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.
0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** *Carry/borrow.*

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.
0: No overflow or underflow has occurred.
1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 7Fh

15							8
0	0	0	0	0	0	0	1

7							0
0	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 8).

Since the stack is 128 bytes deep, the 9th most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP6 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

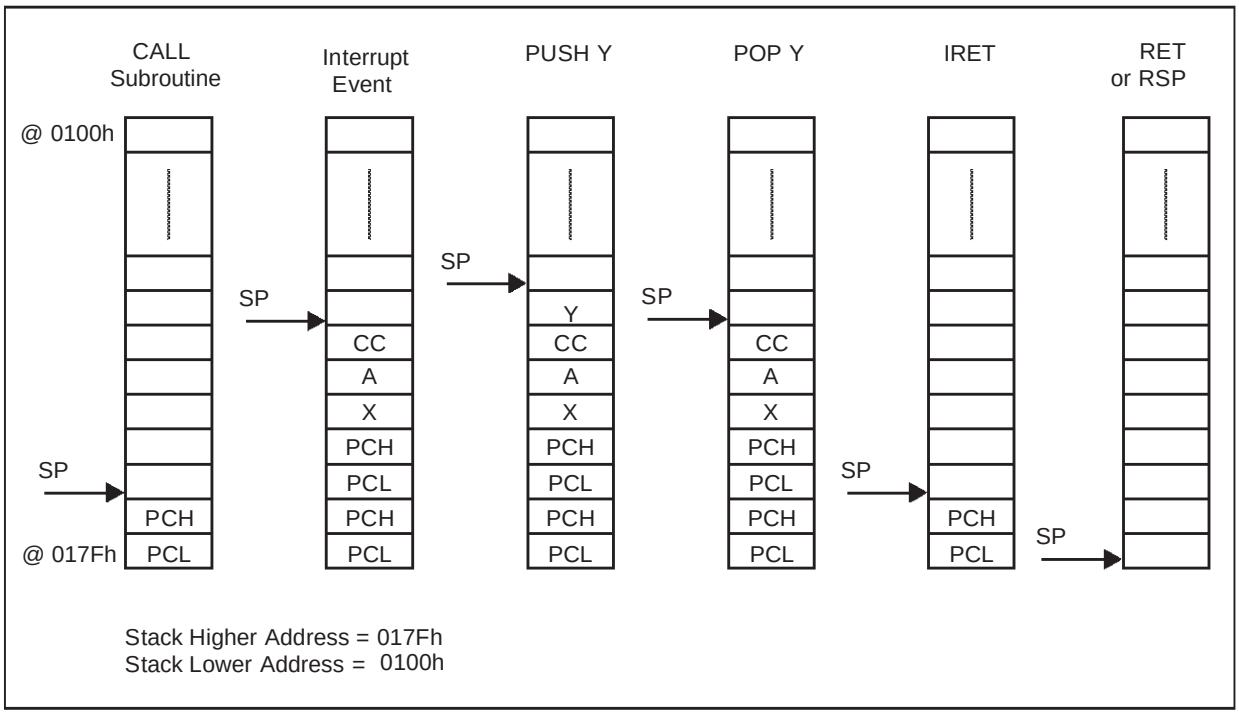
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 8.

– When an interrupt is received, the SP is decremented and the context is pushed on the stack.

– On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 8. Stack Manipulation Example



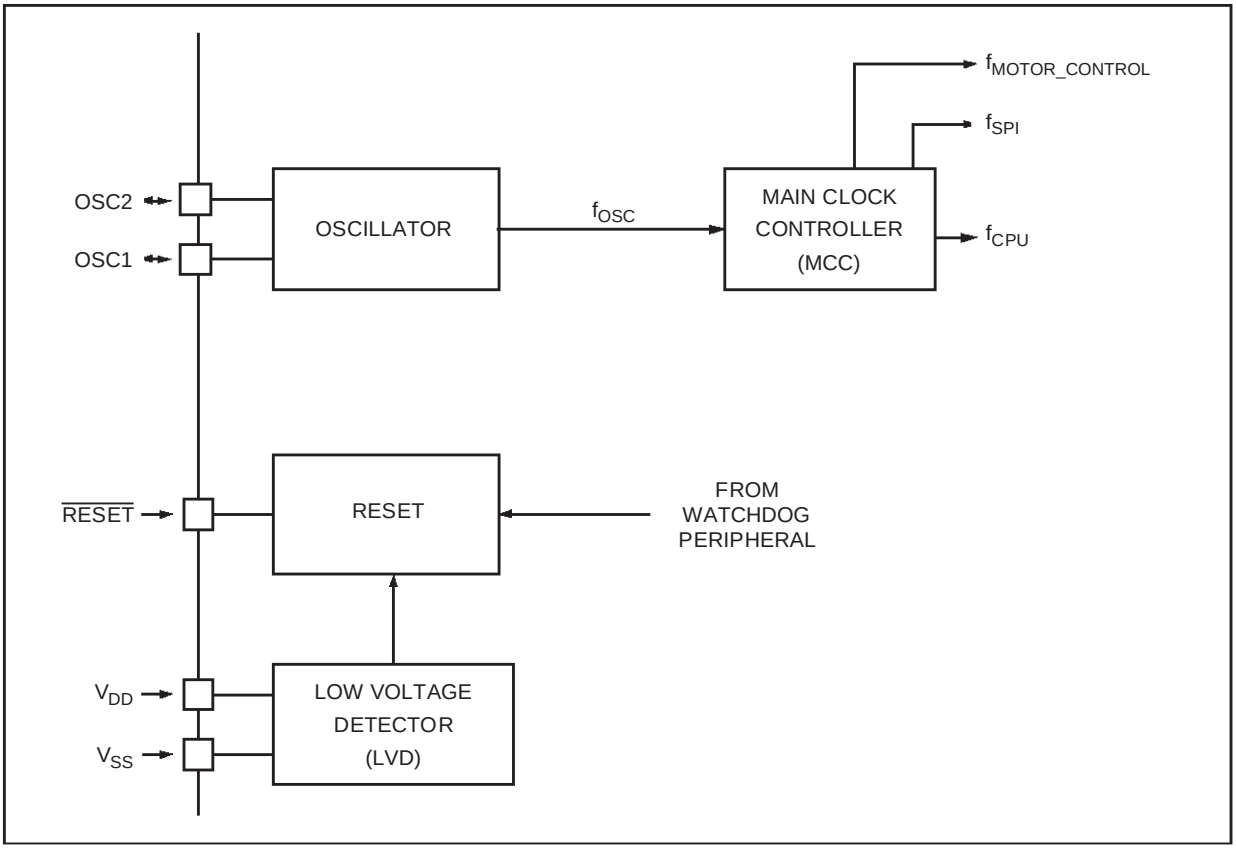
3 SUPPLY, RESET AND CLOCK MANAGEMENT

The ST72141K includes a range of utility features for securing the application in critical situations (for example in case of a power brown-out), and reducing the number of external components. An overview is shown in Figure 9.

Main Features

- Main supply low voltage detection (LVD)
- RESET Manager
- Low consumption resonator oscillator
- Main clock controller (MCC)

Figure 9. Clock, RESET, Option and Supply Management Overview



3.1 LOW VOLTAGE DETECTOR (LVD)

To allow the integration of power management features in the application, the Low Voltage Detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a V_{LVDf} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{LVDf} reference value for a voltage drop is lower than the V_{LVDr} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD Reset circuitry generates a reset when V_{DD} is below:

– V_{LVDr} when V_{DD} is rising
– V_{LVDf} when V_{DD} is falling

The LVD function is illustrated in Figure 10.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is below V_{LVDf}, the MCU can only be in two modes:

– under full software control
– in static safe reset

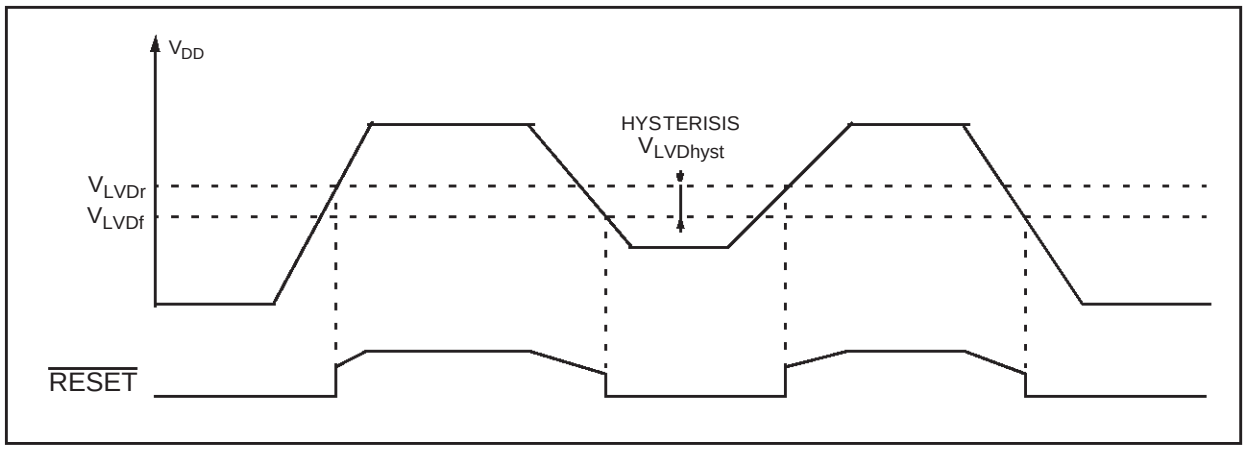
In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

During a Low Voltage Detector Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Notes:

The LVD allows the device to be used without any external RESET circuitry.

Figure 10. Low Voltage Detector vs Reset



3.2 RESET MANAGER

The RESET block includes three RESET sources as shown in Figure 11:

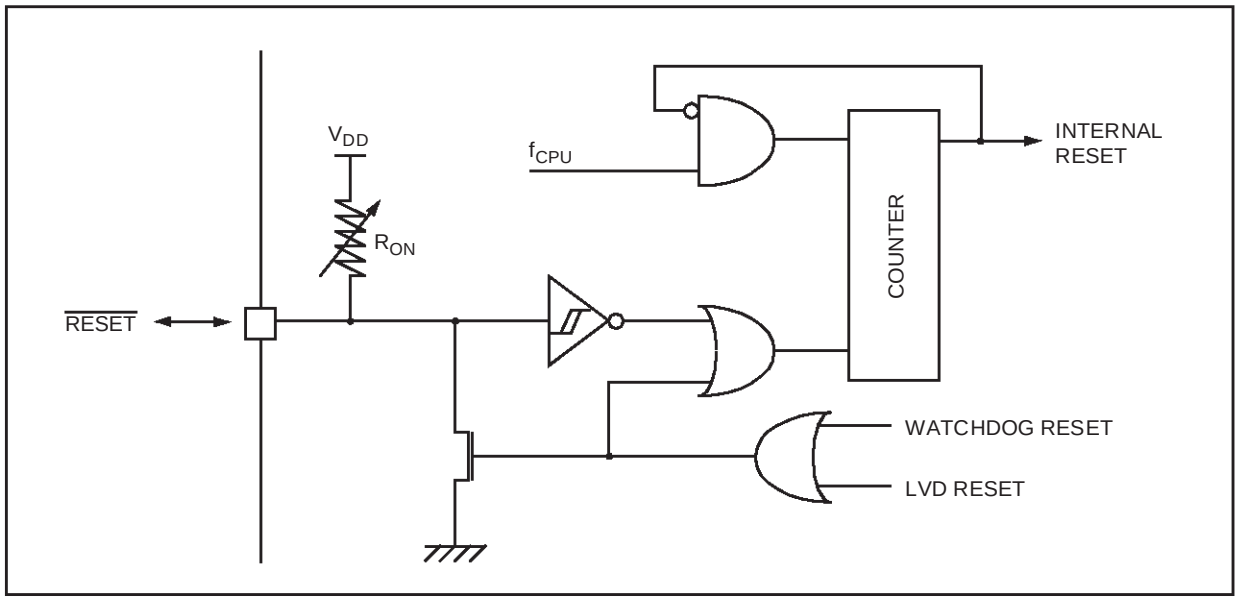
- External $\overline{\text{RESET}}$ source pulse
- Internal LVD RESET (Low Voltage Detection)
- Internal WATCHDOG RESET

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

A 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state.

The RESET vector fetch phase duration is 2 clock cycles.

Figure 11. Reset Block Diagram



RESET MANAGER (Cont'd)

External RESET pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor (see Figure 11). This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device.

A RESET signal originating from an external source must have a duration of at least t_{PULSE} in order to be recognized. Two RESET sequences can be associated with this RESET source as shown in Figure 12.

When the RESET is generated by a internal source, during the two first phases of the RESET sequence, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low.

Generic Power On RESET

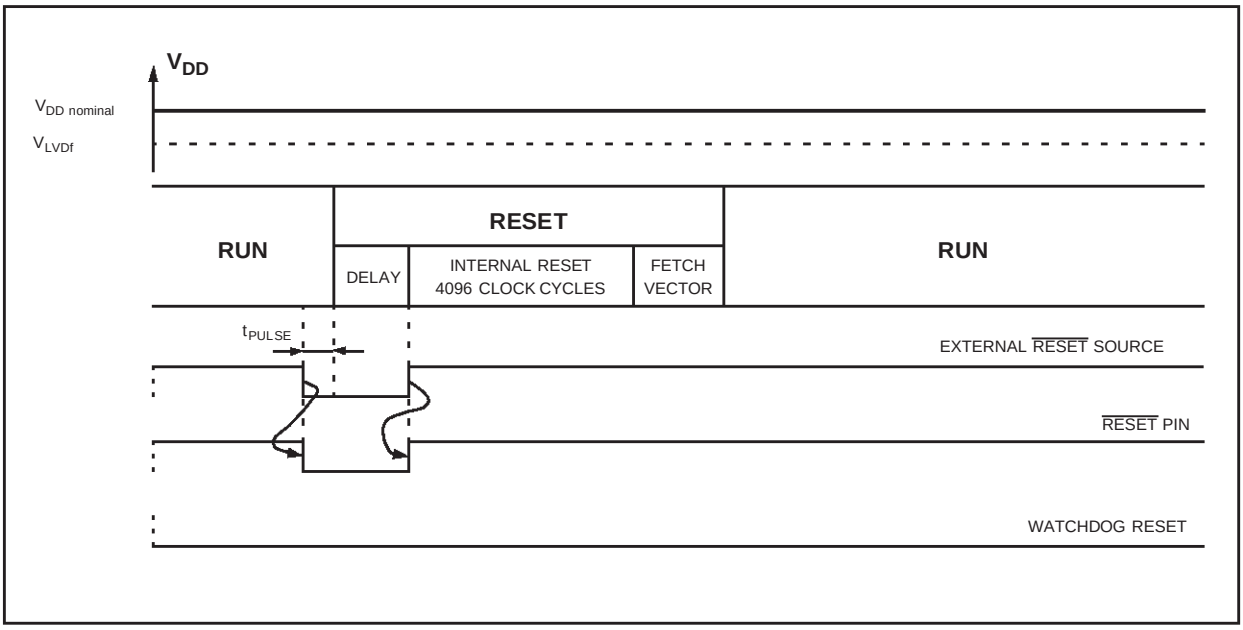
The function of the POR circuit consists of waking up the MCU by detecting (at around 2V) a dynamic (rising edge) variation of the V_{DD} Supply. At the beginning of this sequence, the MCU is configured in the RESET state. When the power supply voltage rises to a sufficient level, the oscillator starts to operate, whereupon an internal 4096 CPU cycles delay is initiated, in order to allow the oscillator to fully stabilize before executing the first instruction. The initialization sequence is executed immediately following the internal delay.

To ensure correct start-up, the user should take care that the VDD Supply is stabilized at a sufficient level for the chosen frequency (see Electrical Characteristics) before the reset signal is released. In addition, supply rising must start from 0V.

As a consequence, the POR does not allow to supervise static, slowly rising, or falling, or noisy (oscillating) V_{DD} supplies.

An external RC network connected to the $\overline{\text{RESET}}$ pin, or the LVD reset can be used instead to get the best performance.

Figure 12. External RESET Sequences



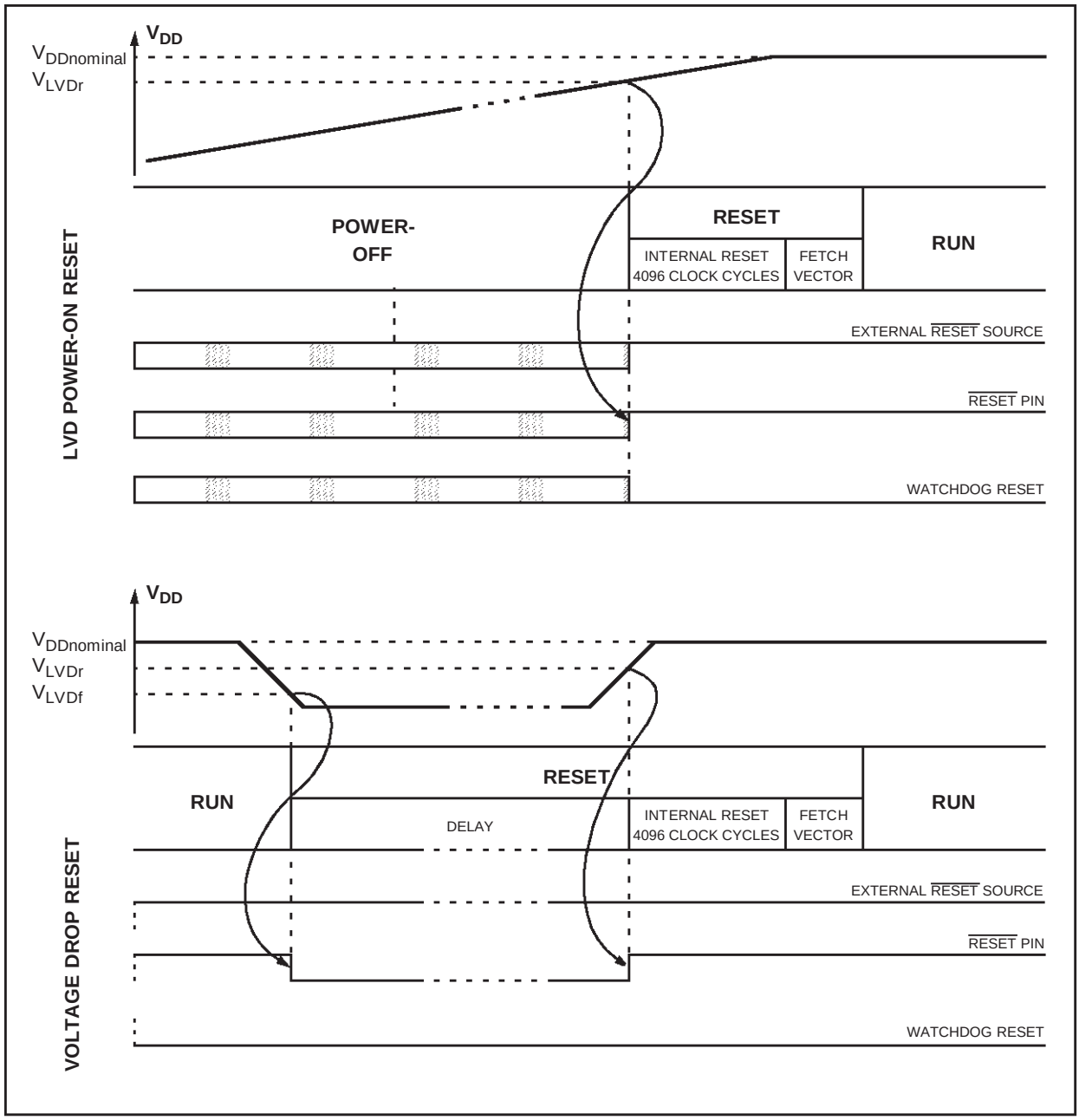
RESET MANAGER (Cont'd)

Internal Low Voltage Detection RESET (option)

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:
- LVD Power-On RESET
- Voltage Drop RESET

In the second sequence, a "delay" phase is used to keep the device in RESET state until V_{DD} rises up to V_{LVDr} (see Figure 13).

Figure 13. LVD RESET Sequences

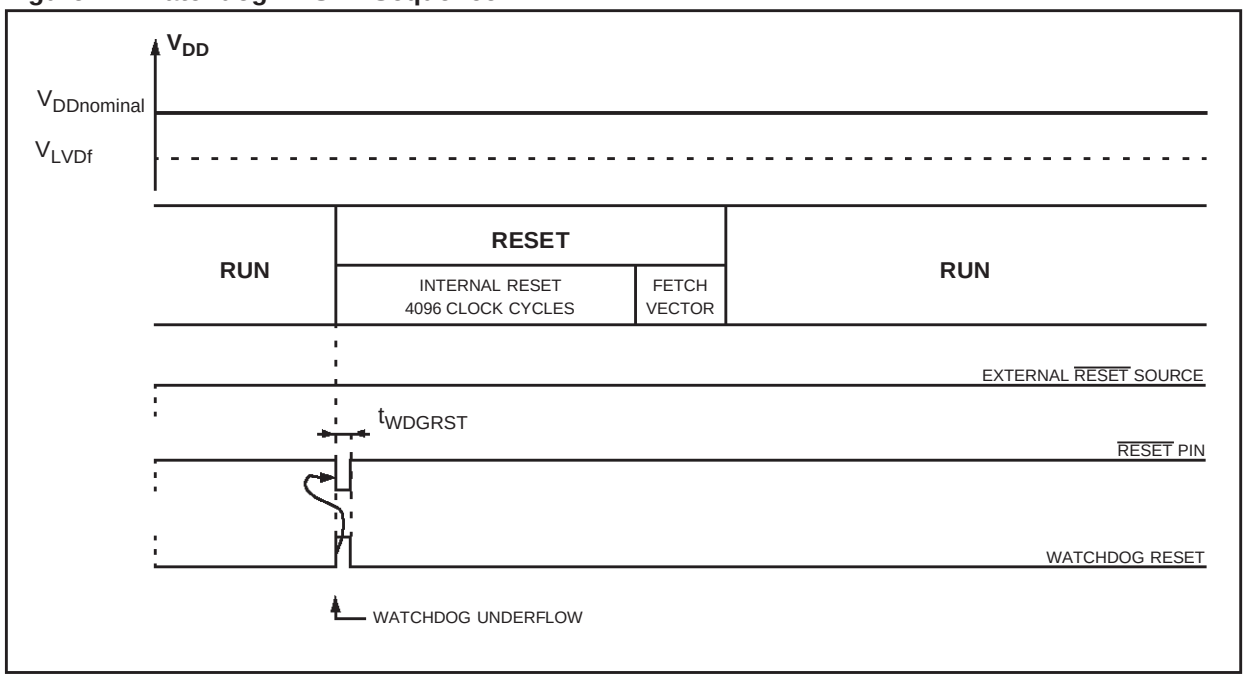


RESET MANAGER (Cont'd)

Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow has the shortest reset phase (see Figure 14).

Figure 14. Watchdog RESET Sequence



3.3 LOW CONSUMPTION OSCILLATOR

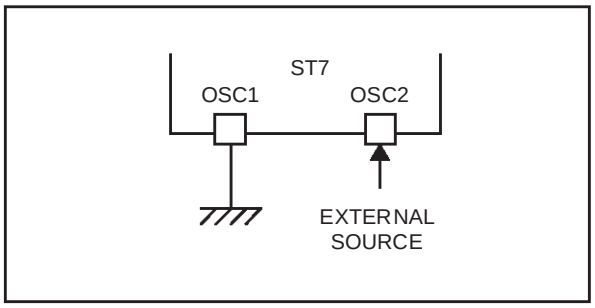
The main clock of the ST7 can be generated by two different sources:

- an external source
- a crystal or ceramic resonator oscillators

External Clock Source

In this mode, a square clock signal with ~50% duty cycle has to drive the OSC2 pin while the OSC1 pin is tied to V_{SS} (see Figure 15).

Figure 15. External Clock

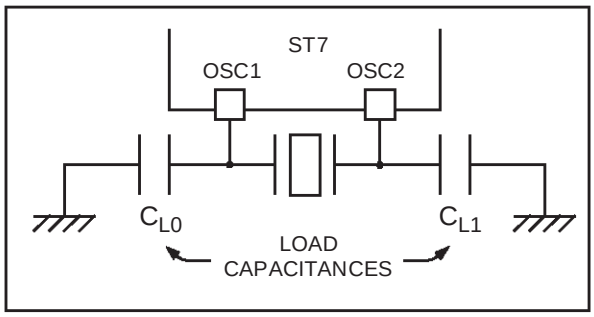


Crystal/Ceramic Oscillators

This oscillator (based on constant current source) is optimized in terms of consumption and has the advantage of producing a very accurate rate on the main clock of the ST7.
When using this oscillator, the resonator and the load capacitances have to be connected as shown in Figure 16 and have to be mounted as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time.

This oscillator is not stopped during the RESET phase to avoid losing time in the oscillator start-up phase.

Figure 16. Crystal/Ceramic Resonator

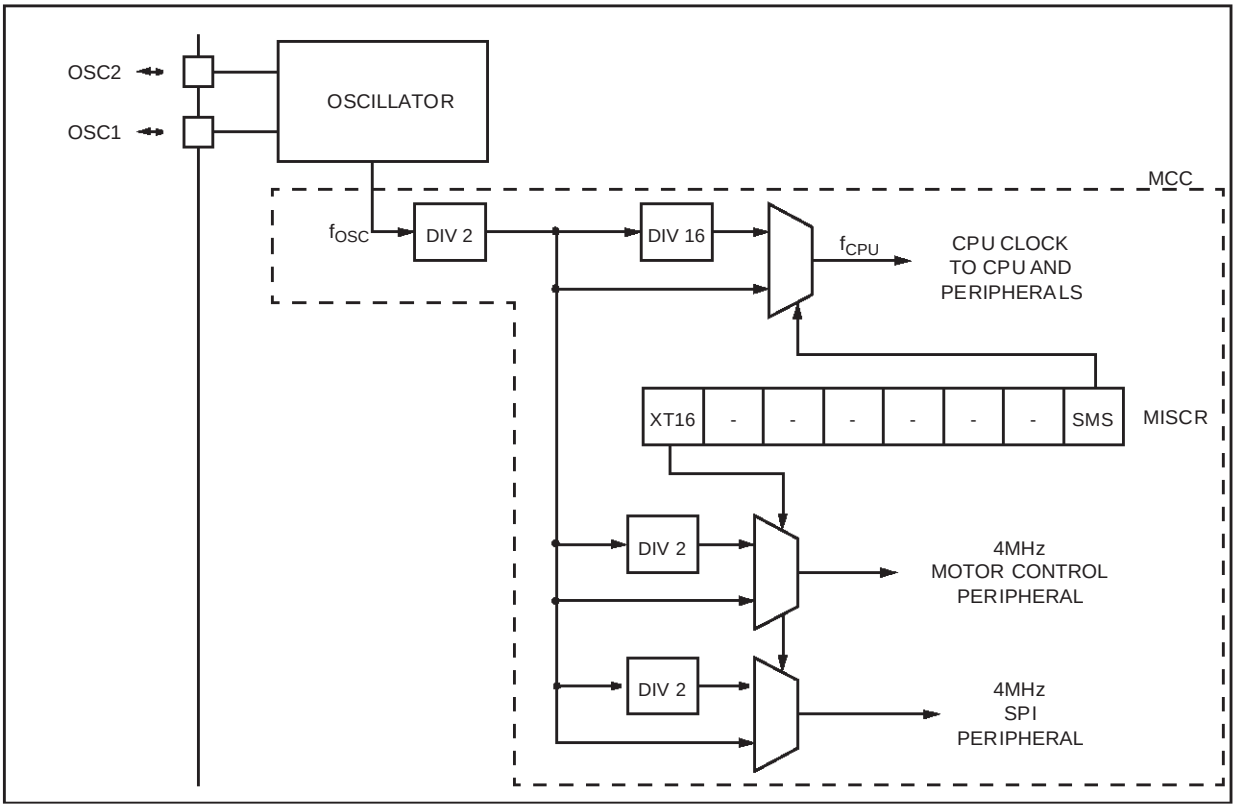


3.4 MAIN CLOCK CONTROLLER (MCC)

The MCC block supplies the clock for the ST7 CPU and its internal peripherals. It allows the SLOW power saving mode and the Motor Contral and SPI peripheral clocks to be managed independently. The MCC functionality is controlled by two bits of the MISCR register: SMS and XT16.

The XT16 bit acts on the clock of the motor control and SPI peripherals while the SMS bit acts on the CPU and the other peripherals.

Figure 17. Main Clock Controller (MCC) Block Diagram



4 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in the Interrupt Mapping Table and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 1.
The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

– Normal processing is suspended at the end of the current instruction execution.

– The PC, X, A and CC registers are saved onto the stack.

– The I bit of the CC register is set to prevent additional interrupts.

– The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to the Interrupt Mapping Table for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I bit will be cleared and the main program will resume.

Priority Management

By default, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case when several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see the Interrupt Mapping Table).

Interrupts and Low Power Mode

All interrupts allow the processor to leave the WAIT low power mode. Only external and specifically mentioned interrupts allow the processor to leave the HALT low power mode (refer to the “Exit from HALT“ column in the Interrupt Mapping Table).

4.1 NON MASKABLE SOFTWARE INTERRUPT

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on Figure 1.

4.2 EXTERNAL INTERRUPTS

External interrupt vectors can be loaded into the PC register if the corresponding external interrupt occurred and if the I bit is cleared. These interrupts allow the processor to leave the Halt low power mode.

The external interrupt polarity is selected through the miscellaneous register or interrupt register (if available).

An external interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.

If several input pins, connected to the same interrupt vector, are configured as interrupts, their signals are logically ANDed before entering the edge/level detection block.

Caution: The type of sensitivity defined in the Miscellaneous or Interrupt register (if available) applies to the ei source. In case of an ANDed source (as described on the I/O ports section), a low level on an I/O pin configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

4.3 PERIPHERAL INTERRUPTS

Different peripheral interrupt flags in the status register are able to cause an interrupt when they are active if both:

– The I bit of the CC register is cleared.

– The corresponding enable bit is set in the control register.

If any of these two conditions is false, the interrupt is latched and thus remains pending.

Clearing an interrupt request is done by:

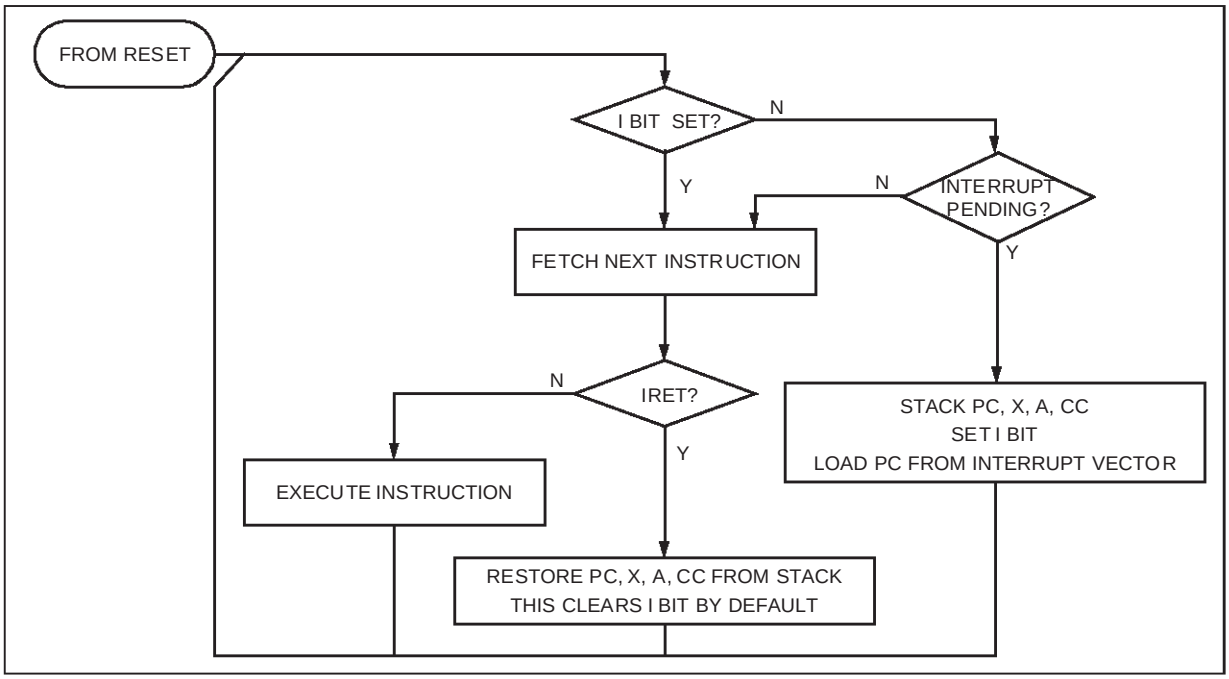
– Writing “0” to the corresponding bit in the status register or

– Access to the status register while the flag is set followed by a read or write of an associated register.

Note: the clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being enabled) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

Figure 18. Interrupt Processing Flowchart



INTERRUPTS (Cont'd)

Table 4. Interrupt Mapping

N°	Source Block	Description	Register Label	Priority Order	Exit from HALT	Address Vector
	RESET	Reset	N/A	Highest Priority	yes	FFFEh-FFFFh
	TRAP	Software Interrupt			no	FFFCh-FFFDh
0		Not used				FFFAh-FFFBh
1	EI0	External Interrupt Port A7..0 (C5..0*)	N/A		yes	FFFAh-FFFBh
2	EI1	External Interrupt Port B7..0 (C5..0*)			yes	FFF8h-FFF9h
3	MTC	Motor Control Interrupt (events: R, Z)	MISR		no	FFF4h-FFF5h
4		Motor Control Interrupt (events: C, D)			no	FFF2h-FFF3h
5		Motor Control Interrupt (events: E, O)			no	FFF0h-FFF1h
6	SPI	SPI Peripheral Interrupts	SPISR		no	FFEEh-FFEFh
7	TIMER A	TIMER A Peripheral Interrupts	TASR		no	FFECh-FFEDh
8	TIMER B	TIMER B Peripheral Interrupts	TBSR		no	FFEAh-FFEBh
9		Not used				FFE8h-FFE9h
10		Not used				FFE6h-FFE7h
11		Not used				FFE4h-FFE5h
12		Not Used		Lowest Priority		FFE2h-FFE3h
13		Not Used				FFE0h-FFE1h

5 POWER SAVING MODES

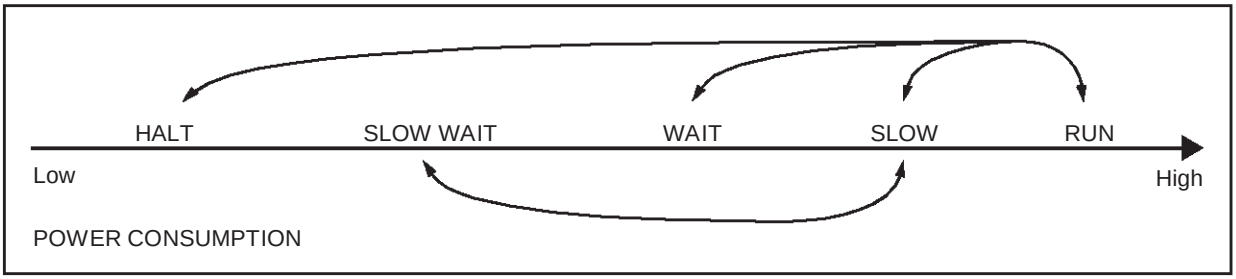
5.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, three main power saving modes are implemented in the ST7 (see Figure 19).

After a RESET the normal operating mode is selected by default (RUN mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided by 2 (f_{CPU}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the the oscillator status.

Figure 19. Power saving mode consumption / transitions



POWER SAVING MODES (Cont'd)

5.2 HALT Mode

The HALT mode is the lowest power consumption mode of the MCU. It is entered by executing the ST7 HALT instruction (see Figure 21).

The MCU can exit HALT mode on reception of either an external interrupt or a reset (see Table 2). When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 4096 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 20).

When entering HALT mode, the I bit in the CC Register is forced to 0 to enable interrupts.

In the HALT mode the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

Figure 20. HALT Mode timing overview

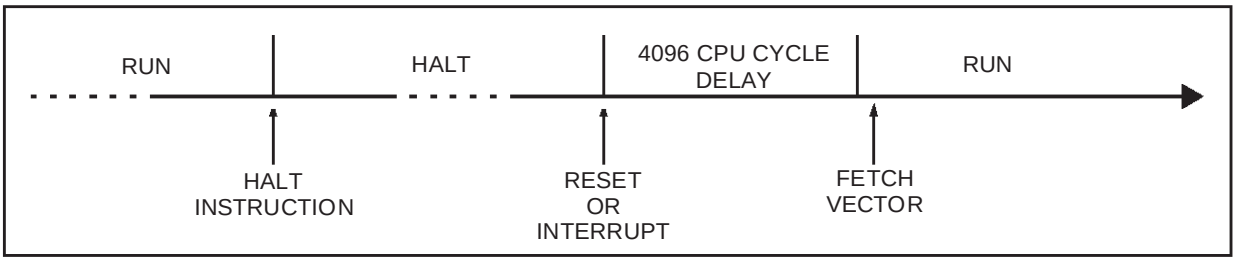
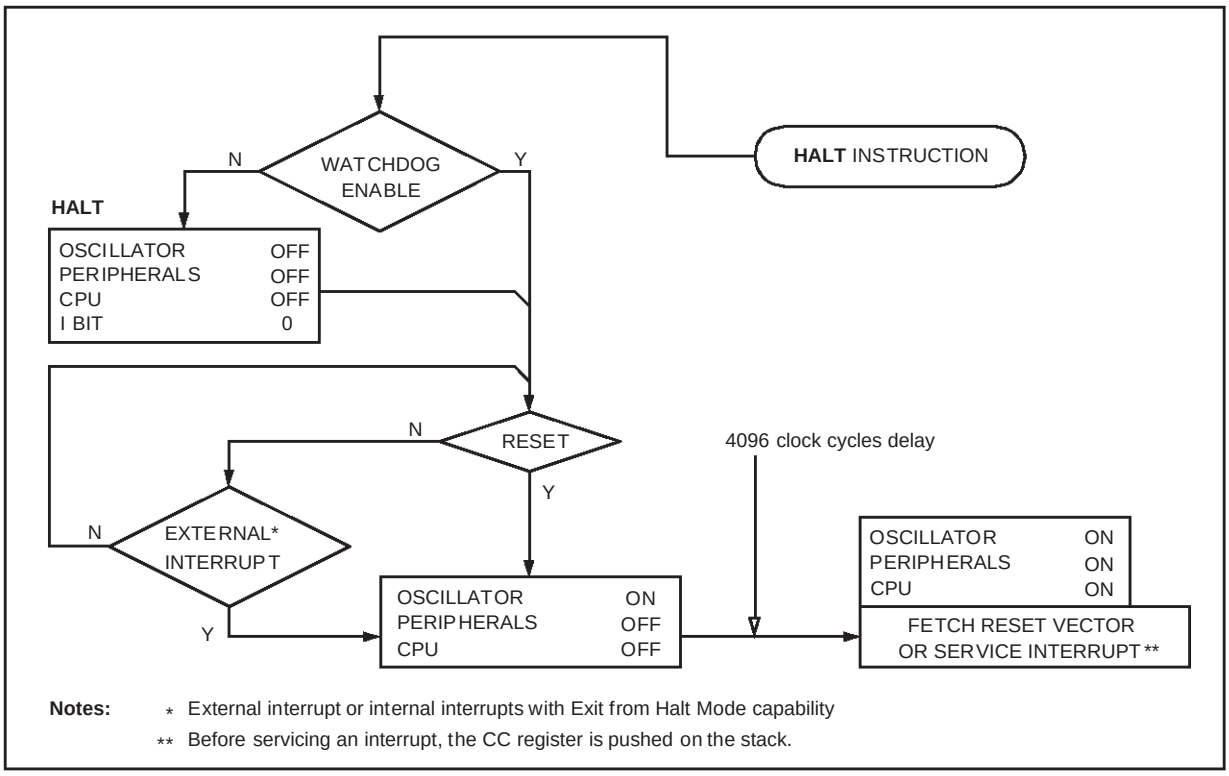


Figure 21. HALT modes flow-chart



Notes: * External interrupt or internal interrupts with Exit from Halt Mode capability

** Before servicing an interrupt, the CC register is pushed on the stack.

POWER SAVING MODES (Cont'd)

5.3 WAIT Mode

WAIT mode places the MCU in a low power consumption mode by stopping the CPU.
This power saving mode is selected by calling the "WFI" ST7 software instruction.
All peripherals remain active. During WAIT mode, the I bit of the CC register are forced to 0, to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in WAIT mode until an interrupt or Reset occurs, whereupon the Program Counter branches to the starting address of the interrupt or Reset service routine.
The MCU will remain in WAIT mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to Figure 22.

5.4 SLOW Mode

This mode has two targets:

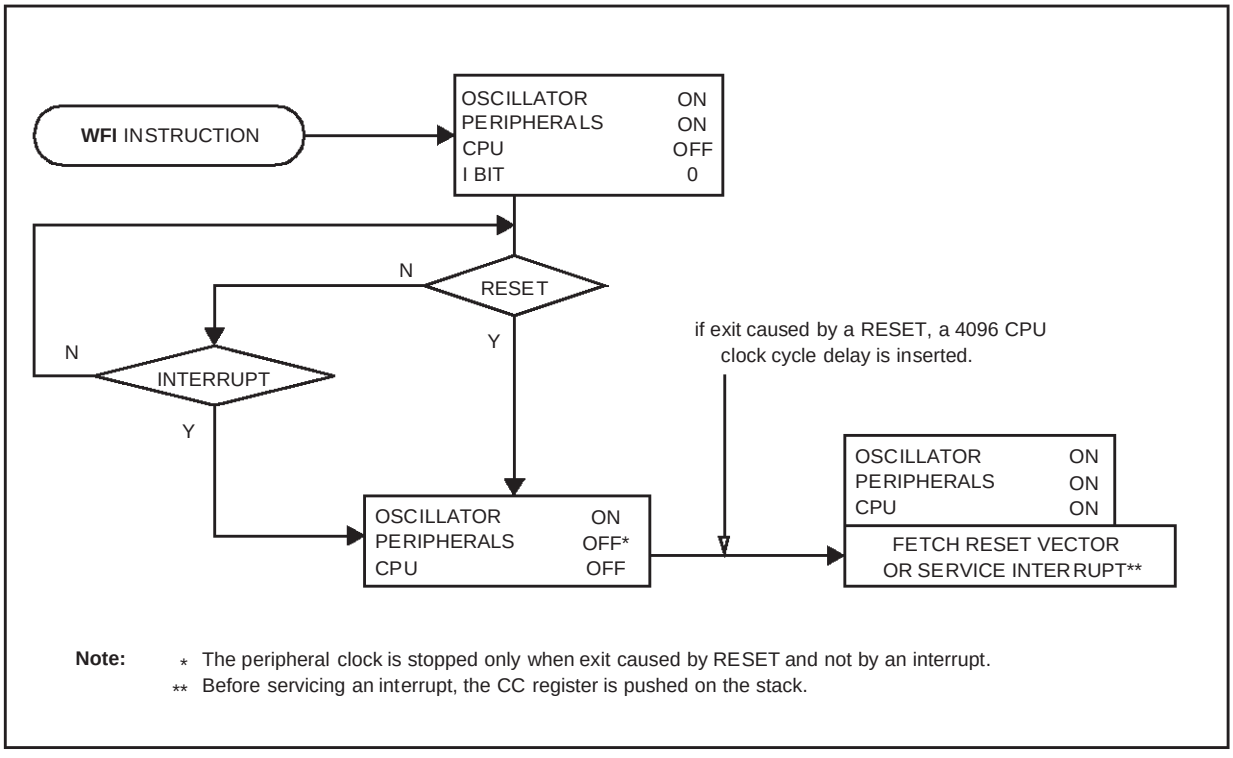
– To reduce power consumption by decreasing the internal clock in the device,

– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by the SMS bit in the MISCR register. This bit enables or disables Slow mode selecting the internal slow frequency (f_{CPU}).

In this mode, the oscillator frequency can be divided by 32 instead of 2 in normal operating mode. The CPU and peripherals are clocked at this lower frequency except the Motor Control and the SPI peripherals which have their own clock selection bit (XT16) in the MISCR register.

Figure 22. WAIT mode flow-chart



Note: * The peripheral clock is stopped only when exit caused by RESET and not by an interrupt.
** Before servicing an interrupt, the CC register is pushed on the stack.

6 I/O PORTS

6.1 INTRODUCTION

The I/O ports offer different functional modes:
– transfer of data through digital inputs and outputs

and for specific pins:
– external interrupt generation
– alternate signal input/output for the on-chip peripherals.

An I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

6.2 FUNCTIONAL DESCRIPTION

Each port has 2 main registers:

– Data Register (DR)

– Data Direction Register (DDR)

and one optional register:

– Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, (for specific ports which do not provide this register refer to the I/O Port Implementation section). The generic I/O block diagram is shown in Figure 23

6.2.1 Input Modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Notes:
1. Writing the DR register modifies the latch value but does not affect the pin status.
2. When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

External interrupt function

When an I/O is configured as Input with Interrupt, an event on this I/O can generate an external interrupt request to the CPU.

Each pin can independently generate an interrupt request. The interrupt sensitivity is independently programmable using the sensitivity bits in the Miscellaneous register.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see pinout description and interrupt section). If several input pins are selected simultaneously as interrupt source, these are logically ANDed. For this reason if one of the interrupt pins is tied low, it masks the other ones.

In case of a floating input with interrupt configuration, special care must be taken when changing the configuration (see Figure 24).

The external interrupts are hardware interrupts, which means that the request latch (not accessible directly by the application) is automatically cleared when the corresponding interrupt vector is fetched. To clear an unwanted pending interrupt by software, the sensitivity bits in the Miscellaneous register must be modified.

6.2.2 Output Modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain
0	V_{SS}	Vss
1	V_{DD}	Floating

6.2.3 Alternate Functions

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over the standard I/O programming.

When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin must be configured in input mode. In this case, the pin state is also digitally readable by addressing the DR register.

Note: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on-chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.

I/O PORTS (Cont'd)

Figure 23. I/O Port General Block Diagram

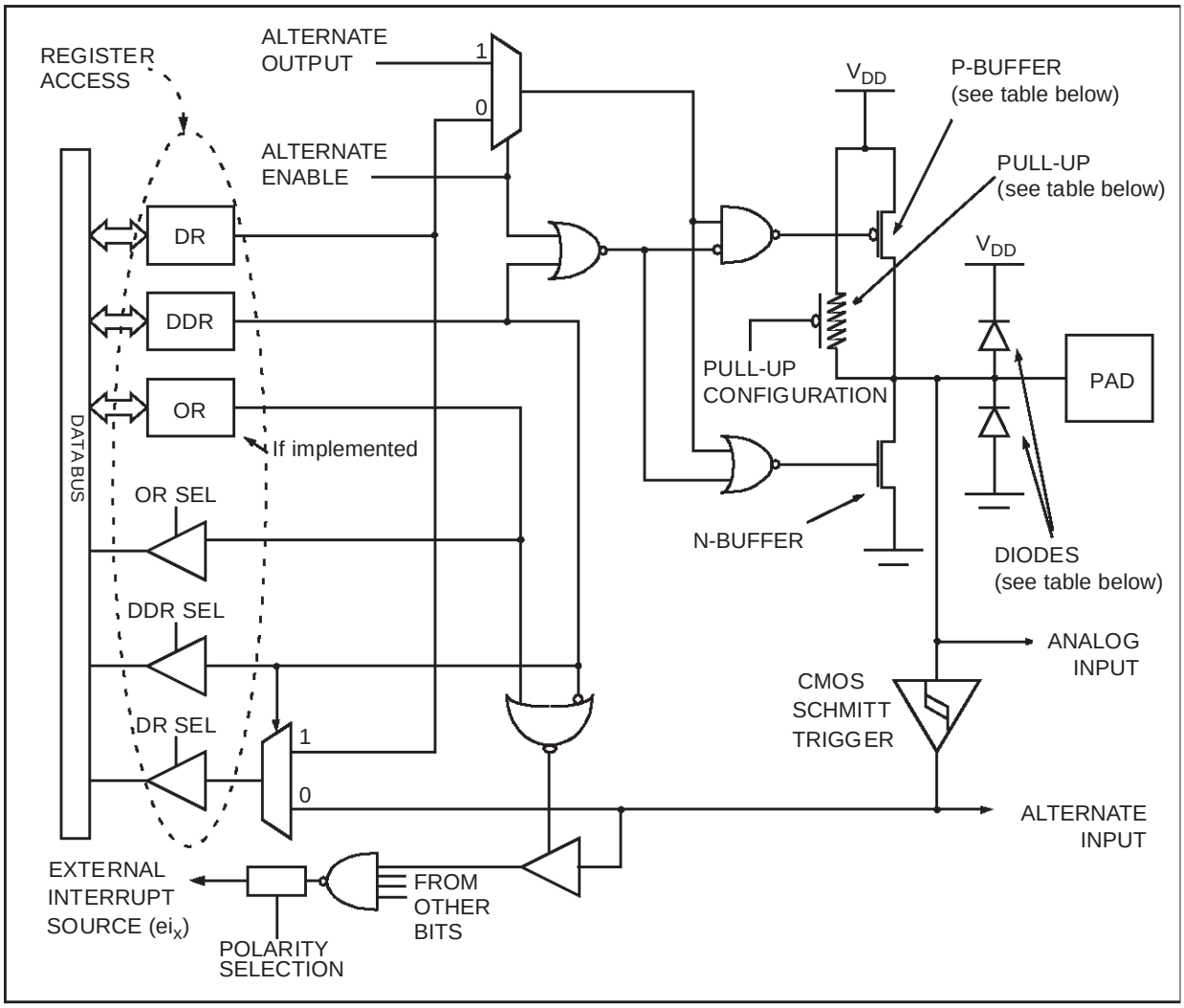


Table 5. I/O Port Mode Options

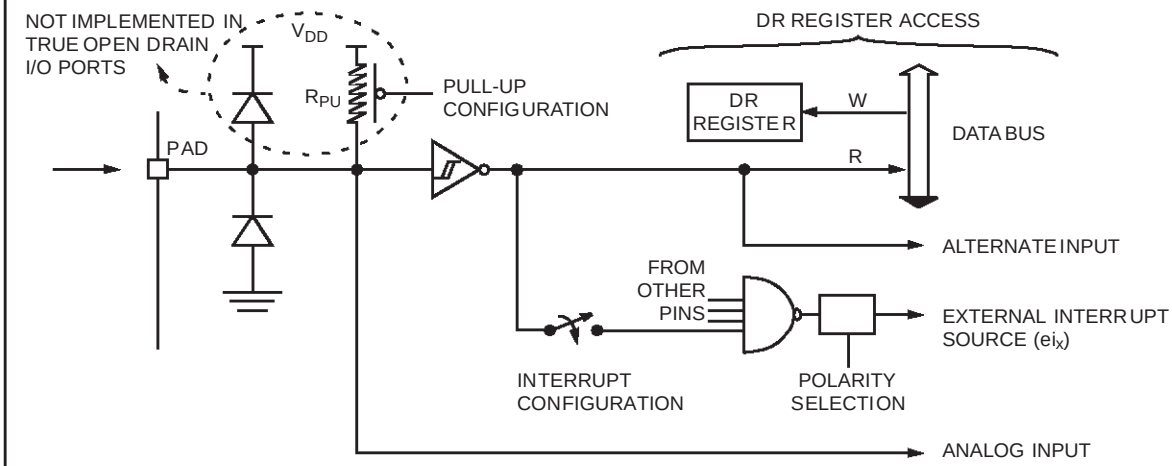
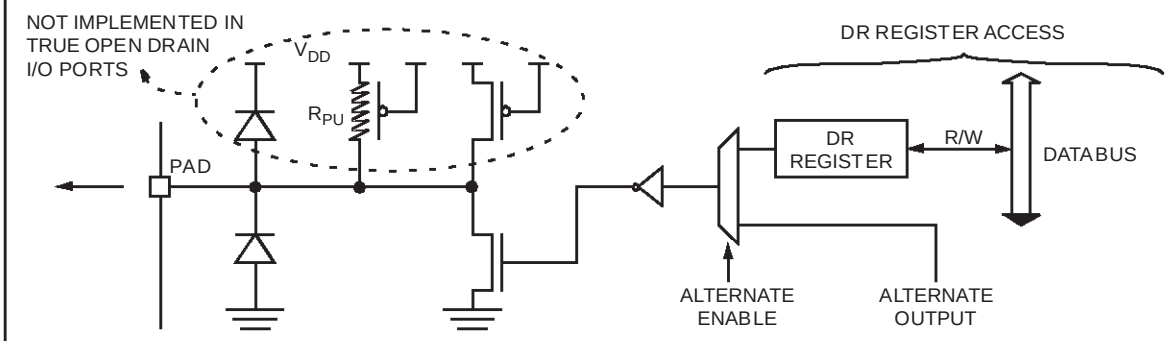
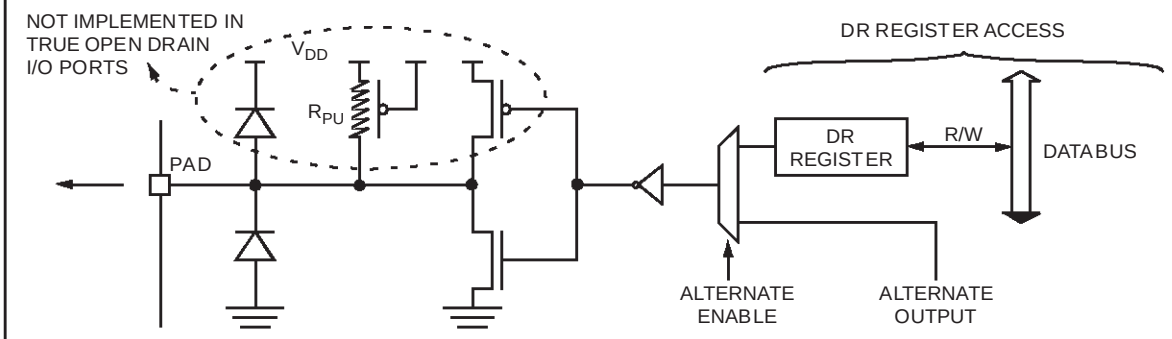
Configuration Mode		Pull-Up	P-Buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open Drain (logic level)		Off		
	True Open Drain	NI	NI	NI (see note)	

Legend: NI - not implemented
Off - implemented not activated
On - implemented and activated

Note: The diode to V_{DD} is not implemented in the true open drain pads. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.

I/O PORTS (Cont'd)

Table 6. I/O Port Configurations

	Hardware Configuration
INPUT [1]	
OPEN-DRAIN OUTPUT [2]	
PUSH-PULL OUTPUT [2]	

Notes:

1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

I/O PORTS (Cont'd)

CAUTION: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

Analog alternate function

When the pin is used as an ADC input, the I/O must be configured as floating input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail which is connected to the ADC input.

It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

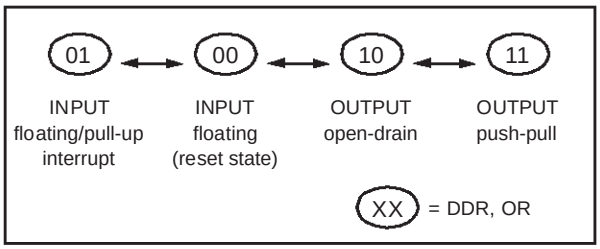
WARNING: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

6.3 I/O PORT IMPLEMENTATION

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific feature of the I/O port such as ADC Input or true open drain.

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 24 Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 24. Interrupt I/O Port State Transitions



The I/O port register configurations are summarized as follows.

I/O PORTS (Cont'd)

Interrupt Ports

PA7:0, PB5:3 (with pull-up)

MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

True Open Drain Interrupt Ports

PA6, PA4 (without pull-up)

MODE	DDR	OR
floating input	0	0
floating interrupt input	0	1
true open drain (high sink ports)	1	X

Table 7. Port Configuration

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:0	floating	pull-up interrupt	open drain	push-pull
Port B	PB5:3	floating	pull-up interrupt	open drain	push-pull
	PB2:0	floating	floating interrupt	true open drain	

I/O PORTS (Cont'd)

6.3.1 Register Description

DATA REGISTER (DR)

Port x Data Register
PxDR with x = A or B.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *Data register 8 bits.*

The DR register has a specific behaviour according to the selected input/output configuration. Writing the DR register is always taken into account even if the pin is configured as an input; this allows to always have the expected level on the pin when toggling to output mode. Reading the DR register returns either the DR register latch content (pin configured as output) or the digital value applied to the I/O pin (pin configured as input).

DATA DIRECTION REGISTER (DDR)

Port x Data Direction Register
PxDDR with x = A or B.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

Bit 7:0 = **DD[7:0]** *Data direction register 8 bits.*

The DDR register gives the input/output direction configuration of the pins. Each bits is set and cleared by software.

0: Input mode
1: Output mode

OPTION REGISTER (OR)

Port x Option Register
PxOR with x = A or B.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
O7	O6	O5	O4	O3	O2	O1	O0

Bit 7:0 = **O[7:0]** *Option register 8 bits.*

For specific I/O pins, this register is not implemented. In this case the DDR register is enough to select the I/O pin configuration.

The OR register allows to distinguish: in input mode if the pull-up with interrupt capability or the basic pull-up configuration is selected, in output mode if the push-pull or open drain configuration is selected.

Each bit is set and cleared by software.

Input mode:
0: floating input
1: pull-up input with or without interrupt

Output mode:
0: output open drain (with P-Buffer unactivated)
1: output push-pull (when available)

I/O PORTS (Cont'd)

Table 8. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Reset Value of all IO port registers		0	0	0	0	0	0	0	0
0000h	**PADR**	MSB							LSB
0001h	**PADDR**								
0002h	**PAOR**								
0004h	**PBDR**	MSB							LSB
0005h	**PBDDR**								
0006h	**PBOR**								

7 MISCELLANEOUS REGISTER

The miscellaneous register allows control over several different features such as the external interrupts or the I/O alternate functions.

7.1 I/O Port Interrupt Sensitivity Description

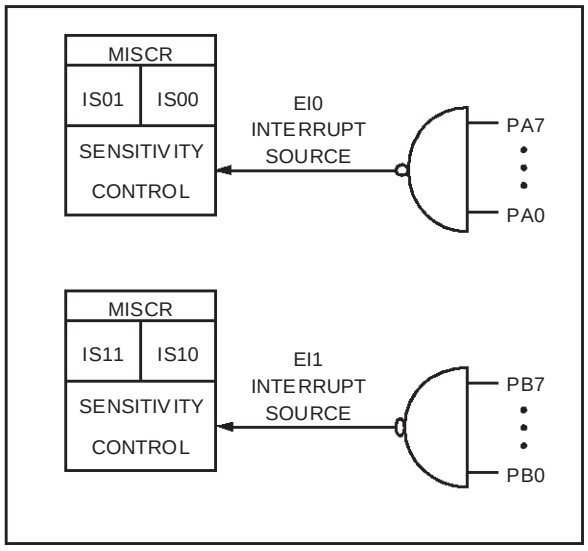
The external interrupt sensitivity is controlled by the ISxx bits of the Miscellaneous register. This control allows to have two fully independent external interrupt source sensitivities as shown in Figure 25.

Each external interrupt source can be generated on four different events on the pin:

- Falling edge
- Rising edge
- Falling and rising edge
- Falling edge and low level

To guaranty correct functionality, a modification of the sensitivity in the MISCR register must be done only when the I bit of the CC register is set to 1 (interrupt masked). See I/O port register and Miscellaneous register descriptions for more details on the programming.

Figure 25. External Interrupt Sensitivity



7.2 I/O Port Alternate Functions

The MISCR register manages the SPI $\overline{SS}$ pin alternate function configuration. This makes it possible to use the PB2 I/O port function while the SPI is active.

These functions are described in detail in Section 7.4 Miscellaneous Register Description.

7.3 Clock Prescaler Selection

The MISCR register is used to select the SLOW mode (see Section 5.4 SLOW Mode for more details) and the SPI and Motor Control peripheral clock prescaler.

MISCELLANEOUS REGISTER (Cont'd)

7.4 Miscellaneous Register Description

MISCELLANEOUS REGISTER (MISCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
XT16	SSM	SSI	IS11	IS10	IS01	IS00	SMS

Bit 7 = **XT16** *MTC and SPI clock selection*
This bit is set and cleared by software. The maximum allowed frequency is 4MHz.
0: MTC and SPI clock supplied with $f_{OSC}/2$
1: MTC and SPI clock supplied with $f_{OSC}/4$

Bit 6 = **SSM** *$\overline{SS}$ mode selection*
This bit is set and cleared by software.
0: Normal mode - the level of the SPI $\overline{SS}$ signal is the external $\overline{SS}$ pin.
1: I/O mode, the level of the SPI $\overline{SS}$ signal is read from the SSI bit.

Bit 5 = **SSI** *$\overline{SS}$ internal mode*
This bit replaces the $\overline{SS}$ pin of the SPI when the SSM bit is set to 1. (see SPI description). It is set and cleared by software.

Bits 4:3 = **IS1[1:0]** *EI1 sensitivity*
The interrupt sensitivity defined using the IS1[1:0] bits combination is applied to the EI1 external interrupts. These two bits can be written only when the I bit of the CC register is set to 1 (interrupt masked).

EI1: Port B

IS11	IS10	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Bits 2:1 = **IS0[1:0]** *EI0 sensitivity*
The interrupt sensitivity defined using the IS0[1:0] bits combination is applied to the EI1 external interrupts. These two bits can be written only when the I bit of the CC register is set to 1 (interrupt masked).

EI0: Port A

IS01	IS00	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Bit 0 = **SMS** *Slow mode select*
This bit is set and cleared by software.
0: Normal mode. $f_{CPU} = f_{OSC} / 2$
1: Slow mode. $f_{CPU} = f_{OSC} / 32$
See sections on low power consumption mode and MCC for more details.

Table 9. Miscellaneous Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0020h	**MISCR**	XT16	SSM	SSI	IS11	IS10	IS01	IS00	SMS
	Reset Value	0	0	0	0	0	0	0	0

8 ON-CHIP PERIPHERALS

8.1 MOTOR CONTROLLER (MTC)

8.1.1 Introduction

The ST7 Motor Controller (MTC) can be seen as a Pulse Width Modulator multiplexed on six output channels, and a Back Electromotive Force (BEMF) zero-crossing detector for sensorless control of Permanent Magnet Direct Current (PMDC) brushless motors.

The MTC is particularly suited to driving synchronous motors and supports operating modes like:

– Commutation step control with motor voltage regulation and current limitation

– Commutation step control with motor current regulation, i.e. direct torque control

– Sensor or sensorless motor phase commutation control

– BEMF zero-crossing detection with high sensitivity. The integrated phase voltage comparator is directly referred to the full BEMF voltage without any attenuation. A BEMF voltage down to 200 mV can be detected, providing high noise immunity and self-commutated operation in a large speed range.

– Realtime motor winding demagnetization detection for fine-tuning the phase voltage masking time to be applied before BEMF monitoring.

– Automatic and programmable delay between BEMF zero-crossing detection and motor phase commutation.

8.1.2 Main Features

- Two on-chip analog comparators, one for BEMF zero-crossing detection with 100 mV hysteresis, the other for current regulation or limitation
- Four selectable reference voltages for the hysteresis comparator (0.2 V, 0.6 V, 1.2 V, 2.5 V)
- 8-bit timer (MTIM) with two compare registers and two capture features
- Measurement window generator for BEMF zero-crossing detection
- Auto-calibrated prescaler with 16 division steps
- 8x8-bit multiplier
- Phase input multiplexer
- Sophisticated output management:
  - The six output channels can be split into two groups (odd & even).
  - The PWM signal can be multiplexed on even, odd or both groups, alternatively or simultaneously.
  - The output polarity is programmable channel by channel.
  - An software enabled bit (active low) forces the outputs in HiZ.
  - An “emergency stop” input pin (active low) asynchronously forces the outputs in HiZ.

Table 10. MTC Registers

Register	Description	Page
MTIM	Timer Counter Register	71
MZPRV	Capture Z_{n-1} Register	71
MZREG	Capture Z_n Register	71
MCOMP	Compare C_{n+1} Register	71
MDREG	Demagnetization Register	71
MWGHT	A_n Weight Register	71
MPRSR	Prescaler & Sampling Register	71
MIMR	Interrupt Mask Register	72
MISR	Interrupt Status Register	72
MCRA	Control Register A	73
MCRB	Control Register B	74
MPHST	Phase State Register	75
MPAR	Parity Register	75
MPOL	Polarity Register	75

8.1.3 Application Example

This example shows a six-step command sequence for a 3-phase permanent magnet DC brushless motor (PMDC motor). Figure 27 shows the phase steps and voltage, while Table 11 shows the relevant phase configurations.

To run this kind of motor efficiently, an autoswitching mode has to be used, i.e. the position of the rotor must self-generate the powered winding commutation. The BEMF zero crossing (Z event) on the non-excited winding is used by the MTC as a rotor position sensor. The delay between this event and the commutation is computed by the MTC and the commutation event C_n is automatically generated after this delay.

After the commutation occurs, the MTC waits until the winding is completely demagnetized by the free-wheeling diode: during this phase the winding is tied to 0V or to the HV high voltage rail and no BEMF can be read. At the end of this phase a new BEMF zero-crossing detection is enabled.

The end of demagnetization event (D), is also detected by the MTC or simulated with a timer compare feature when no detection is possible.

MOTOR CONTROLLER (Cont'd)

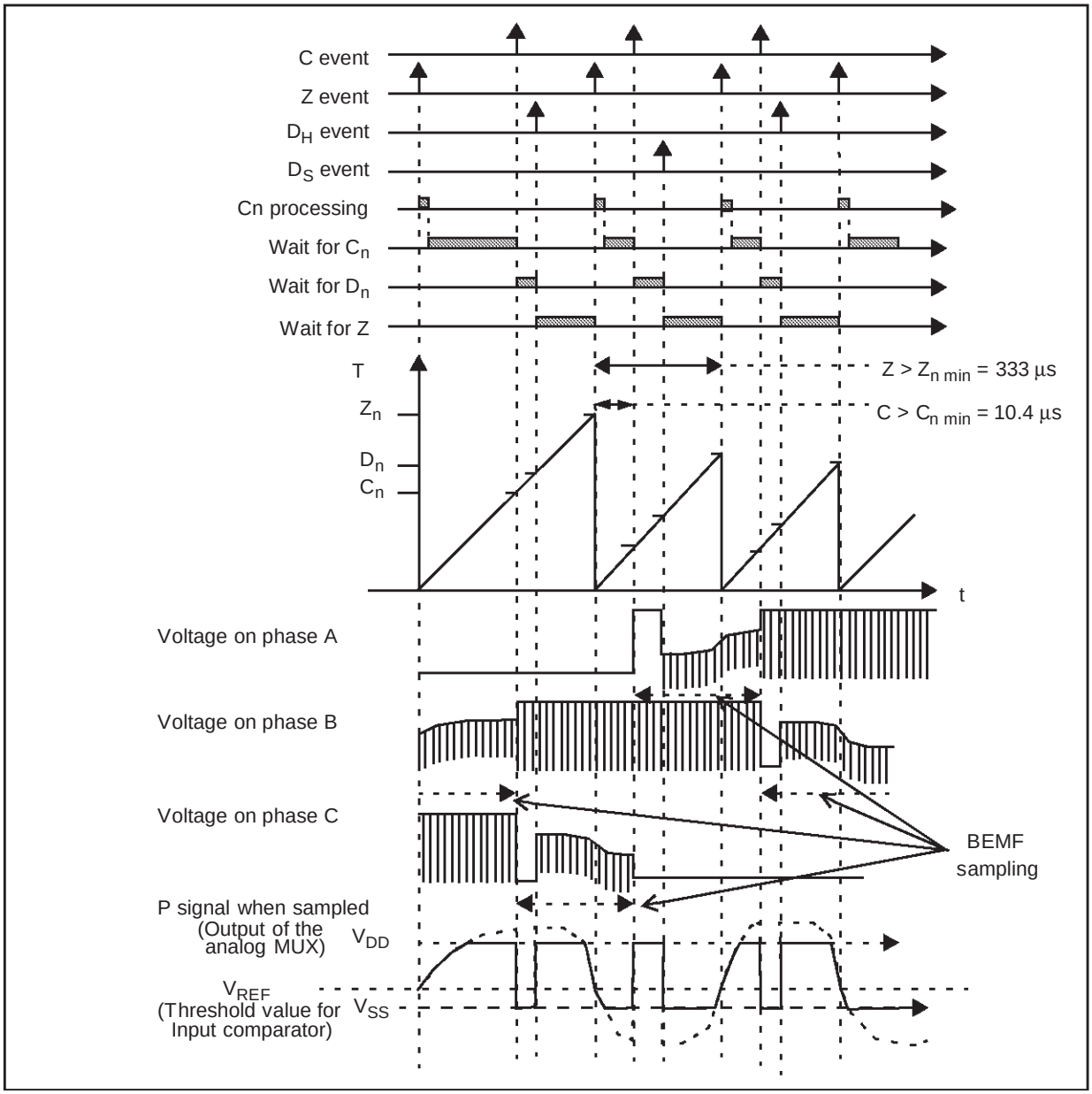
The MTC manages these three events always in the same order: Z generates C after a delay computed in realtime, then waits for D in order to enable the peripheral to detect another Z event.

The speed regulation is managed by the microcontroller, by means of an adjustable reference current level in case of current control, or by direct PWM duty-cycle adjustment in case of voltage control.

All detections of Z_n events are done during a short measurement window while the high side switch is turned off. For this reason the PWM signal is applied on the high side switches.

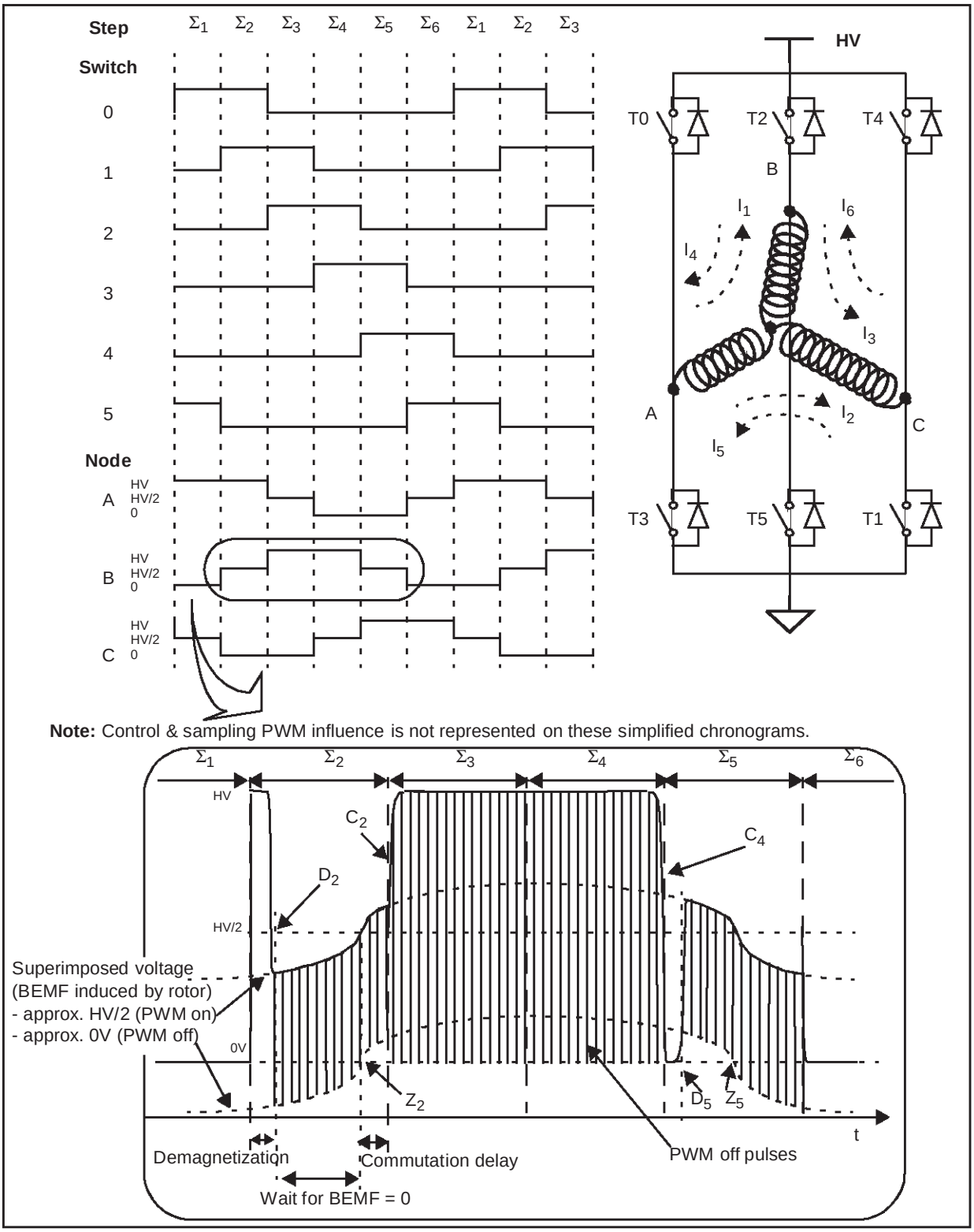
When the high side switch is off, the high side winding is tied to 0V by the free-wheeling diode, the low side winding voltage is also held at 0V by the low side ON switch and the complete BEMF voltage is present on the third winding: detection is then possible.

Figure 26. Chronogram of Events (in Autoswitched Mode)



MOTOR CONTROLLER (Cont'd)

Figure 27. Example of Command Sequence for 6-step Mode (typical 3-phase PMDC Motor Control)



MOTOR CONTROLLER (Cont'd)

Table 11. Step Configuration Summary

Configuration		Step					
		Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6
Phase state register	Current direction	A to B	A to C	B to C	B to A	C to A	C to B
	High side	T0	T0	T2	T2	T4	T4
	Low side	T5	T1	T1	T3	T3	T5
	OO[5:0] bits in MPHST register	100001	000011	000110	001100	011000	110000
BEMF input	Measurement done on:	MCIC	MCIB	MCIA	MCIC	MCIB	MCIA
	IS[1:0] bits in MPHST register	10	01	00	10	01	00
BEMF edge	Back EMF shape	Falling	Rising	Falling	Rising	Falling	Rising
	CPB bit in MCRB register (ZVD bit = 0)	0	1	0	1	0	1
Hardware or Hardware-software demagnetization	Voltage on measured point at the start of demagnetization	0V	HV	0V	HV	0V	HV
	HDM-SDM bits in MCRB register	10	11	10	11	10	11
Demagnetization switch	PWM side selection to accelerate demagnetization	Odd Side	Even Side	Odd Side	Even Side	Odd Side	Even Side
	Driver selection to accelerate de-magnetization	T5	T0	T1	T2	T3	T4

For a detailed description of the MTC registers, see Section 8.1.7.

MOTOR CONTROLLER (Cont'd)

8.1.4 Functional Description

The MTC can be split into four main parts as shown in the simplified block diagram in Figure 28.

– The BEMF ZERO-CROSSING DETECTOR with a comparator and an input multiplexer.

– The DELAY MANAGER with an 8-bit timer (MTIM) and an 8x8 bit multiplier.

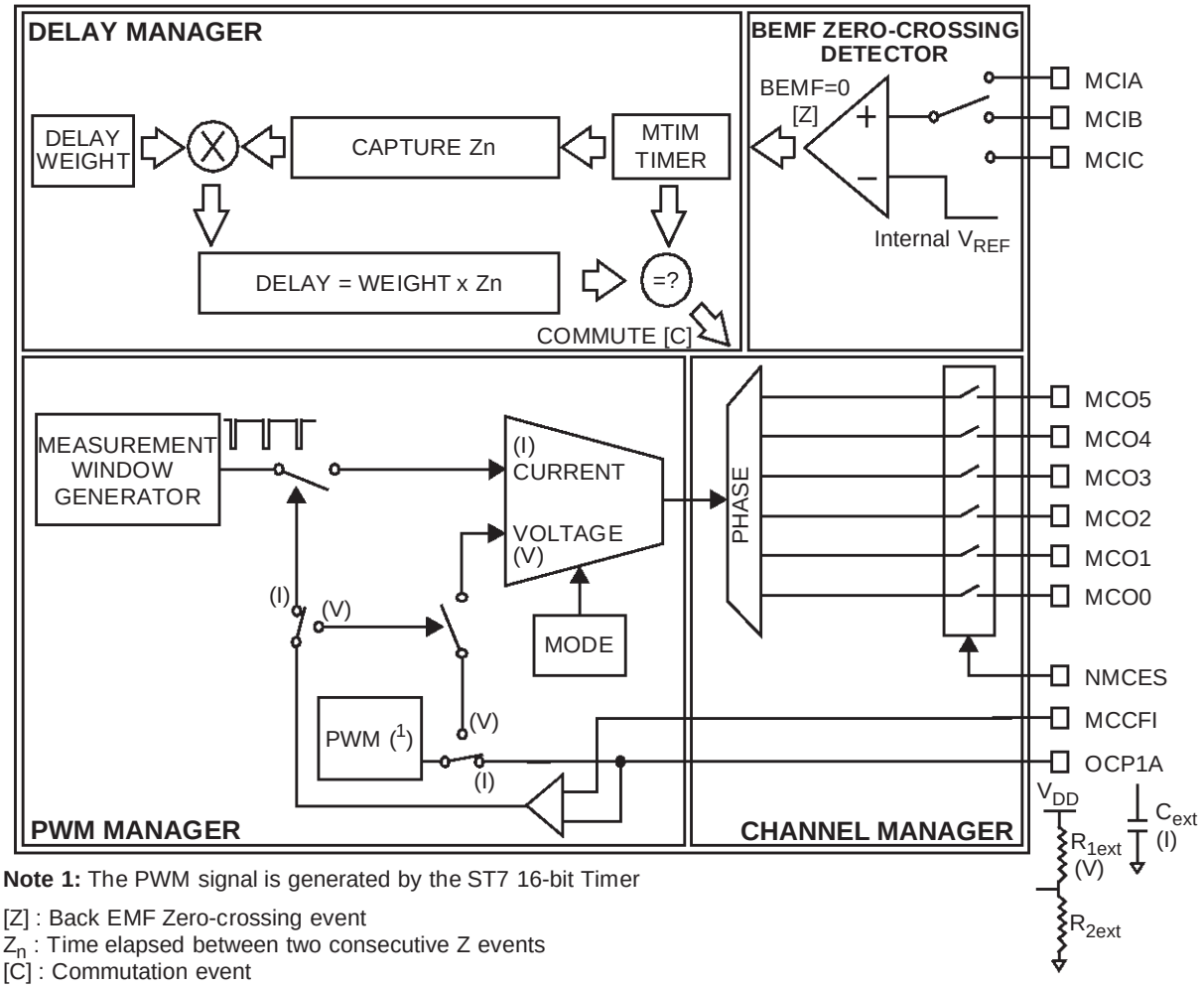
– The PWM MANAGER, including a measurement window generator, a mode selector and a current comparator.

– The CHANNEL MANAGER with the PWM multiplexer, polarity programming capability and emergency HiZ configuration input.

8.1.4.1 Input Detection Block

This block can operate in sensor mode or sensorless mode. The mode is selected via the SR bit in the MCRA register. The block diagram is shown in Figure 29.

Figure 28. Simplified MTC Block Diagram



Note 1: The PWM signal is generated by the ST7 16-bit Timer

[Z] : Back EMF Zero-crossing event
Z_n : Time elapsed between two consecutive Z events
[C] : Commutation event
C_n : Time delayed after Z event to generate C event
(I): Current mode
(V): Voltage mode

MOTOR CONTROLLER (Cont'd)

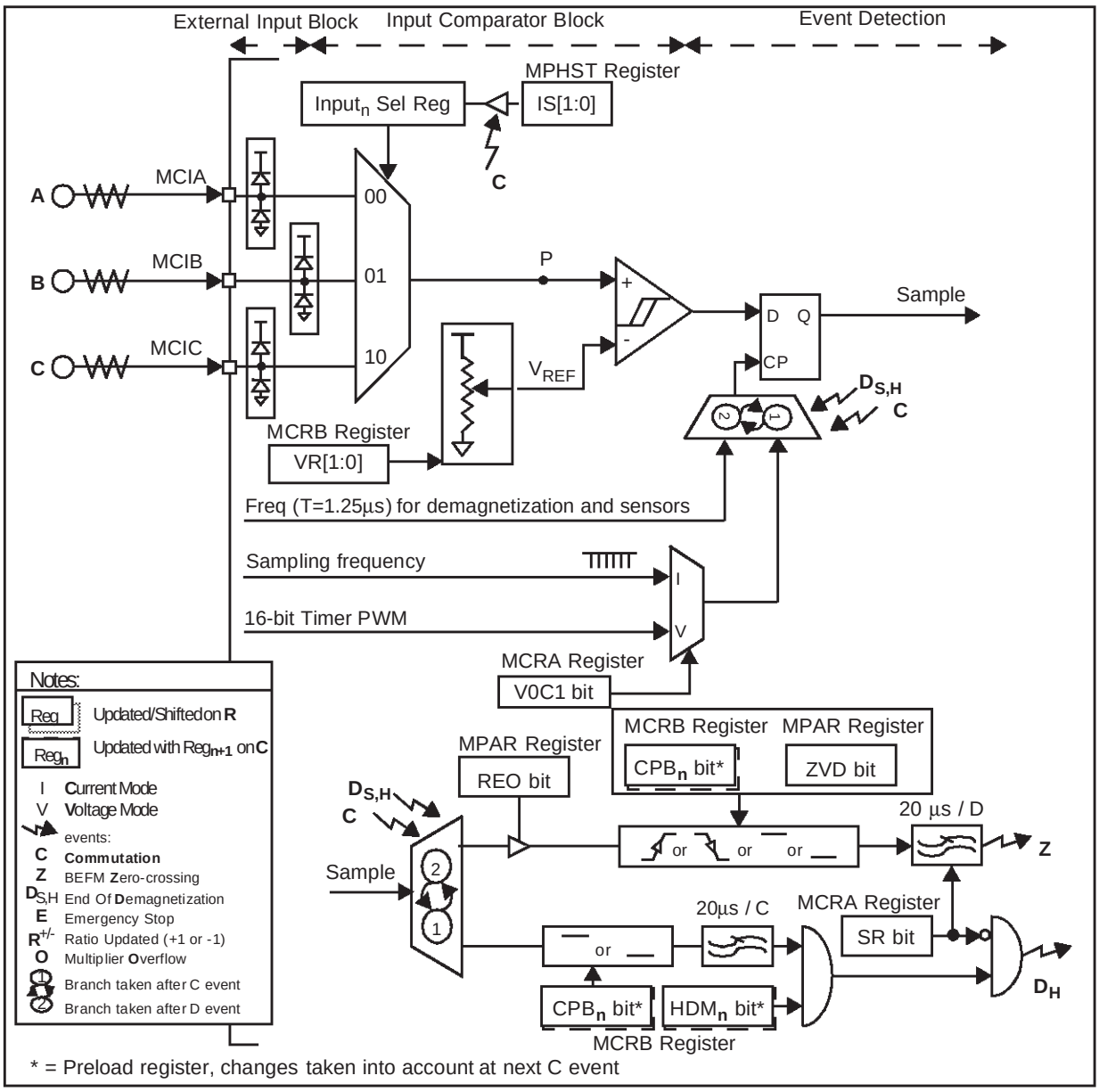
Input Pins

The MCIA, MCIB and MCIC input pins can be used as analog pins in sensorless mode or as digital pins in sensor mode. In sensorless mode, the analog inputs are used to measure the BEMF zero crossing and to detect the end of demagnetization if required. In sensor mode, they are connected to sensor outputs.

Due to the presence of diodes, these pins can permanently support an input current of 5mA. In sensorless mode, this feature enables the inputs to be connected to each motor phase through a single resistor.

A multiplexer, programmed by the IS[1:0] bits in MPHST register selects the input pins and connects them to the rotor position control logic in either sensorless or sensor mode.

Figure 29. Input Stage



MOTOR CONTROLLER (Cont'd)

Sensorless Mode

This mode is used to detect BEMF zero crossing and end of demagnetization events.

The analog phase multiplexer connects the non-excited motor winding to an analog 100mV hysteresis comparator referred to a selectable reference voltage.

The VR[1:0] bits in the MCRB register select the reference voltage from four internal values depending on the noise level and the application voltage supply.

BEMF detections are performed during the measurement window, when the excited windings are free-wheeling through the low side switches and diodes. At this stage the common star connection voltage is near to ground voltage (instead of $V_{DD}/2$ when the excited windings are powered) and the complete BEMF voltage is present on the non-excited winding terminal, referred to the ground terminal.

The zero crossing sampling frequency is then defined, in current mode, by the measurement window generator frequency (SA[3:0] bits in the MPRSR register) or, in voltage mode, by the 16-bit Timer PWM frequency and duty cycle.

During a short period after a phase commutation (C event), the winding is no longer excited but needs a demagnetisation phase during which the BEMF cannot be read. A demagnetization current goes through the free-wheeling diodes and the winding voltage is stuck at the high voltage or to the ground terminal. For this reason an "end of demagnetization event" D must be detected on the winding before the detector can sense a BEMF zero crossing.

For the end-of-demagnetization detection, no special PWM configuration is needed, the comparator sensing is done at a fixed 800kHz sampling frequency.

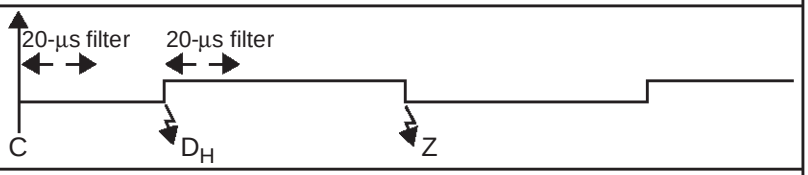
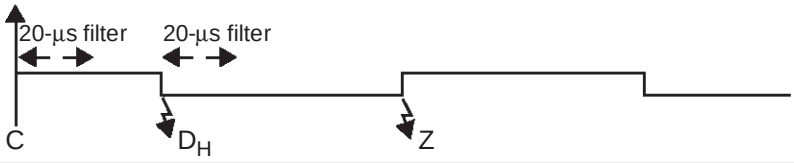
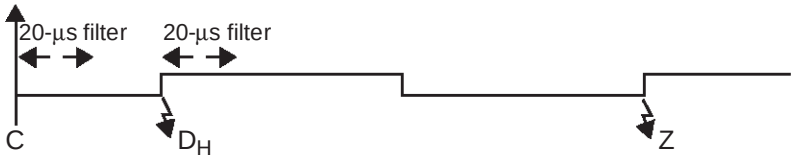
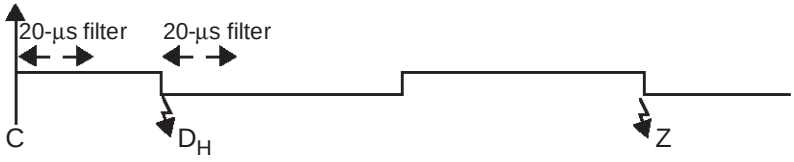
So, the three events: C (commutation), D (demagnetization) and Z (BEMF zero crossing) must always occur in this order.

The comparator output is processed by a detector that automatically recognizes the D or Z event, depending on the CPB or ZVD edge and level configuration bits as described in Table 12.

A 20–µs filter after a C event disables a D event if spurious spikes occur.

Another 20–µs filter after a D event disables a Z event if spurious spikes occur.

Table 12. ZVD and CPB Edge Selection Bits

ZVD bit	CPB bit	Event generation vs input data sampled
0	0	
0	1	
1	0	
1	1	

Note: The ZVD bit is located in the MPAR register, the CPB bit is in the MCRB register.

MOTOR CONTROLLER (Cont'd)

Demagnetization (D) Event

At the end of the demagnetization phase, current no longer goes through the free-wheeling diodes. The voltage on the non-excited winding terminal goes from one of the power rail voltages to the common star connection voltage plus the BEMF voltage. In some cases (if the BEMF voltage is positive and the free-wheeling diodes are at ground for example) this end of demagnetization can be seen as a voltage edge on the selected MCIx input and it is called a hardware demagnetization event D_H. See Table 12.

If enabled by the HDM bit in the MCRB register, the current value of the MTIM timer is captured in register MDREG when this event occurs in order to be able to simulate the demagnetization phase for the next steps.

When enabled by the SDM bit in the MCRB register, demagnetization can also be simulated by comparing the MTIM timer with the MDREG register. This kind of demagnetization is called software demagnetization D_S.

If the HDM and SDM bits are both set, the first event that occurs, triggers a demagnetization event. For this to work correctly, a D_S event must not precede a D_H event because the latter could be detected as a Z event.

Software demagnetization can also be always used if the HDM bit is reset and the SDM bit is set. This mode works as a programmable masking time between the C and Z events. To drive the motor securely, the masking time must be always greater than the real demagnetization time in order to avoid a spurious Z event.

When an event occurs, (either D_H or D_S) the DI bit in the MISR register is set and an interrupt request is generated if the DIM bit of register MIMR is set.

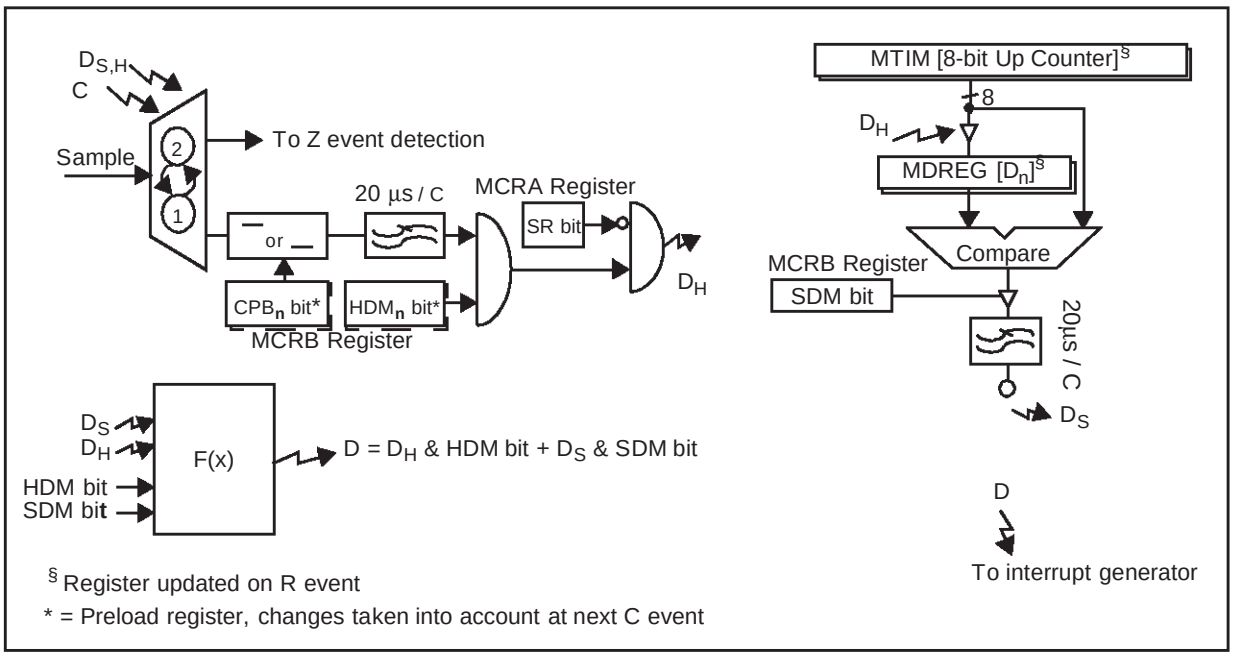
Warning 1: Due to the alternate automatic capture and compare of the MTIM timer with MDREG register by D_H and D_S events, the MDREG register should be manipulated with special care.

Warning 2: To avoid a system stop, the value written to the MDREG register in Soft Demagnetization Mode (SDM = 1) should always be:

– Greater than the MCOMP value of the commutation before the related demagnetization

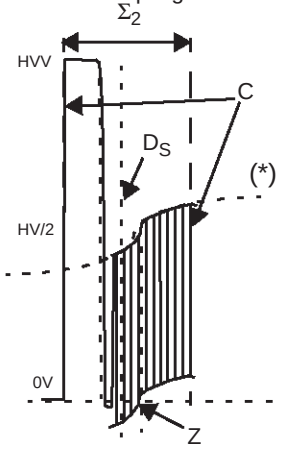
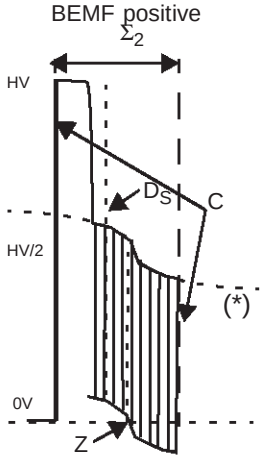
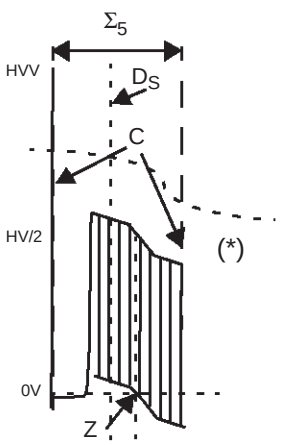
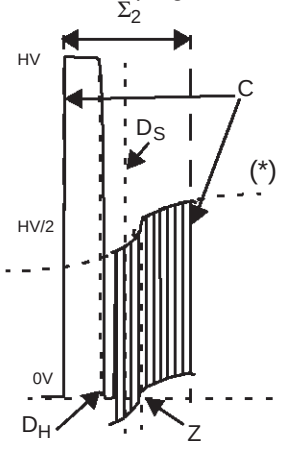
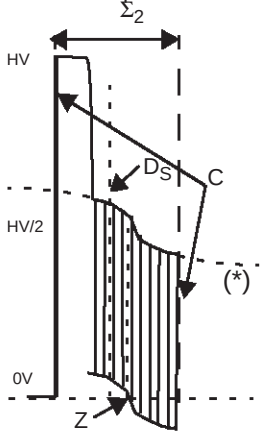
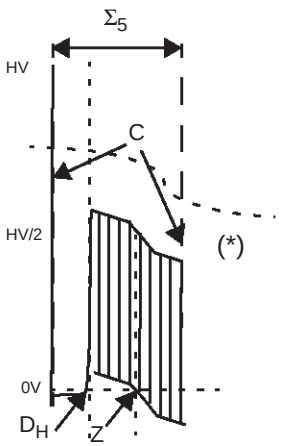
– Greater than the value in the MTIM counter at that moment (when writing to the MDREG register).

Figure 30. D Event Generation Mechanism



MOTOR CONTROLLER (Cont'd)

Table 13. Demagnetisation (D) Event Generation (example for ZVD=0)

HDM bit	Meaning	CPB bit = 1 (Even Σ)	CPB bit = 0 (Odd Σ)
0	Software Mode (SDM bit =1 and HDM bit = 0)	D = D_S = Output Compare [MDREG, MTIM registers] Undershoot due to motor parasite or first sampling  Weak / null undershoot and BEMF positive 	
1	Hardware/Simulated Mode (SDM bit = 1 and HDM bit = 1)	D = D_H + D_S (Hardware detection or Output compare true) Undershoot due to motor parasite or first sampling  Weak / null undershoot and BEMF positive 	D = D_H (Hardware detection only) 
(*) **Note:** This is a zoom to the additional voltage induced by the rotor (Back EMF)			

MOTOR CONTROLLER (Cont'd)

BEMF Zero Crossing (Z) Event

When both C and D events have occurred, the PWM may be switched to another group of outputs (depending on the OS[2:0] bits in the MCRB register) and the real BEMF zero crossing sampling can start (see Figure 32).

A BEMF voltage is present on the non-powered terminal but referred to common star connection of the motor whose voltage is equal to $V_{DD}/2$.

When a winding is free-wheeling (during PWM off-time) its terminal voltage changes to the other power rail voltage, that means if the PWM is applied on the high side driver, free-wheeling will be done through the high side diode and the terminal will be 0V.

This is used to force the common star connection to 0V in order to read the BEMF referred to the ground terminal.

Consequently, BEMF reading (i.e. comparison with a voltage close to 0V) can only be done when the PWM is applied on the high side drivers.

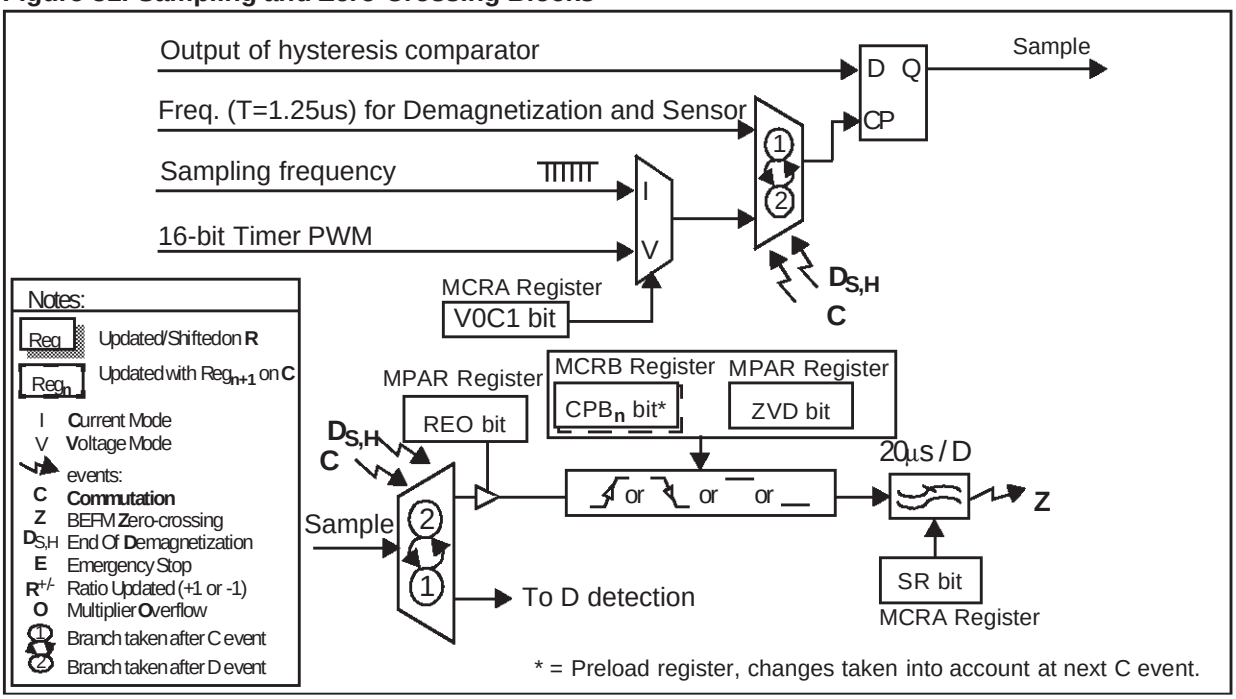
For this reason the MTC outputs can be split in two groups called ODD and EVEN and the BEMF reading will be done only when PWM is applied on one of these two groups. The REO bit in the MPAR register is used to select the group to be used for BEMF sensing (high side group)

Refer to Table 15 for an overview of when a BEMF can be read depending on REO bit, PWM mode and function mode of peripheral.

Depending on the edge and level selection (ZVD and CPB) bits and when PWM is applied on the correct group, a BEMF zero crossing detection sets the ZI bit in the MISR register and generates an interrupt if the ZIM bit is set.

The Z event also triggers some timer/multiplier operations, for more details see Section 8.1.4.2

Figure 31. Sampling and Zero Crossing Blocks



MOTOR CONTROLLER (Cont'd)

Sensor Mode

In sensor mode, the rotor position information is given to the peripheral by means of logical data on the three inputs MCIA, MCIB and MCIC.

For each step one of these three inputs is selected (IS[1:0] bits in register MPHST) in order to detect the Z event.

In this case Demagnetization has no meaning and the relevant features such as the special PWM configuration, D_S or D_H management, 20-µs filter; are not available (see Table 14).

For this configuration the rotor detection doesn't need a particular phase configuration to validate the measurement and a Z event can be read from any detection window. A fixed sampling frequency (800 kHz) is used, that means the Z event and position sensoring is more precise than it is in sensorless mode.

The minimum off time for current control PWM is also reduced to 1.25µs.

Procedure for reading sensor inputs in Direct Access mode: In Direct Access mode, the peripheral clock is disabled as shown in Table 25. As the data present on the selected input is synchronized by a 800 kHz clock, the sensor can't be read directly (the value is latched). To read the sensor data the following steps have to be performed:

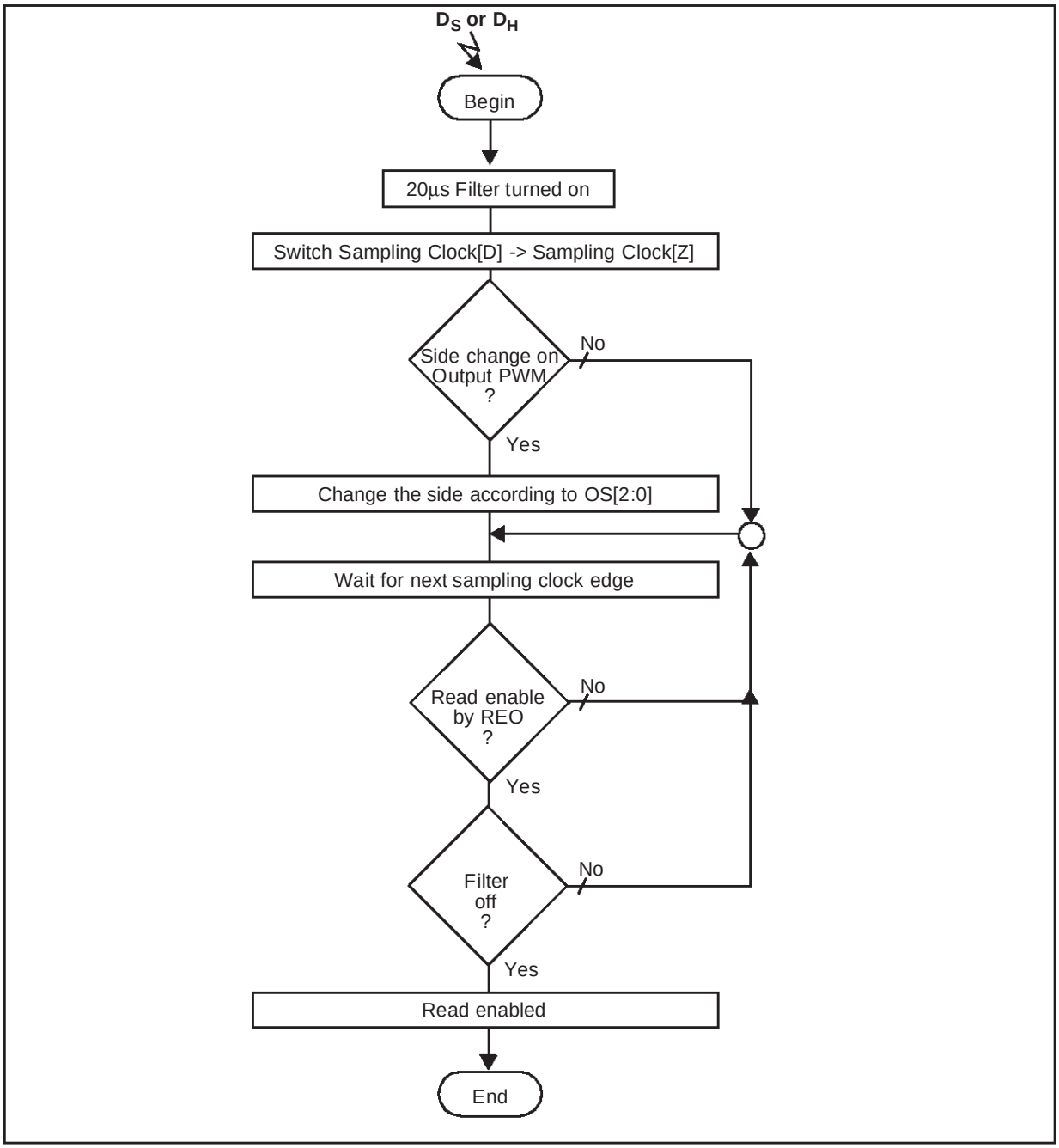
1. Select the appropriate MCIx input pin by means of the IS[1:0] bits in the MPHST register
2. Switch from direct access mode to indirect access mode in order to latch the sensor data (DAC bit in MCRA register).
3. Switch back to direct access mode.
4. Read the comparator output (HST bit in the MIMR register)

Table 14. Sensor mode selection

SR bit	Mode	OS2 bit use	Event detection sampling clock	Filtering	Behaviour of the output PWM
0	Sensors not used	Enabled	D: Clock 1.25µs Z: SA&OT config.	20µs after C for D event 20µs after D for Z event	"Before D" behaviour & "after D" behaviour
1	Sensors used	Disabled	Z: Clock 1.25µs	20µs after C for Z event	Only "after D" behaviour

MOTOR CONTROLLER (Cont'd)

Figure 32. Functional Diagram of Z Detection after D Event



MOTOR CONTROLLER (Cont'd)

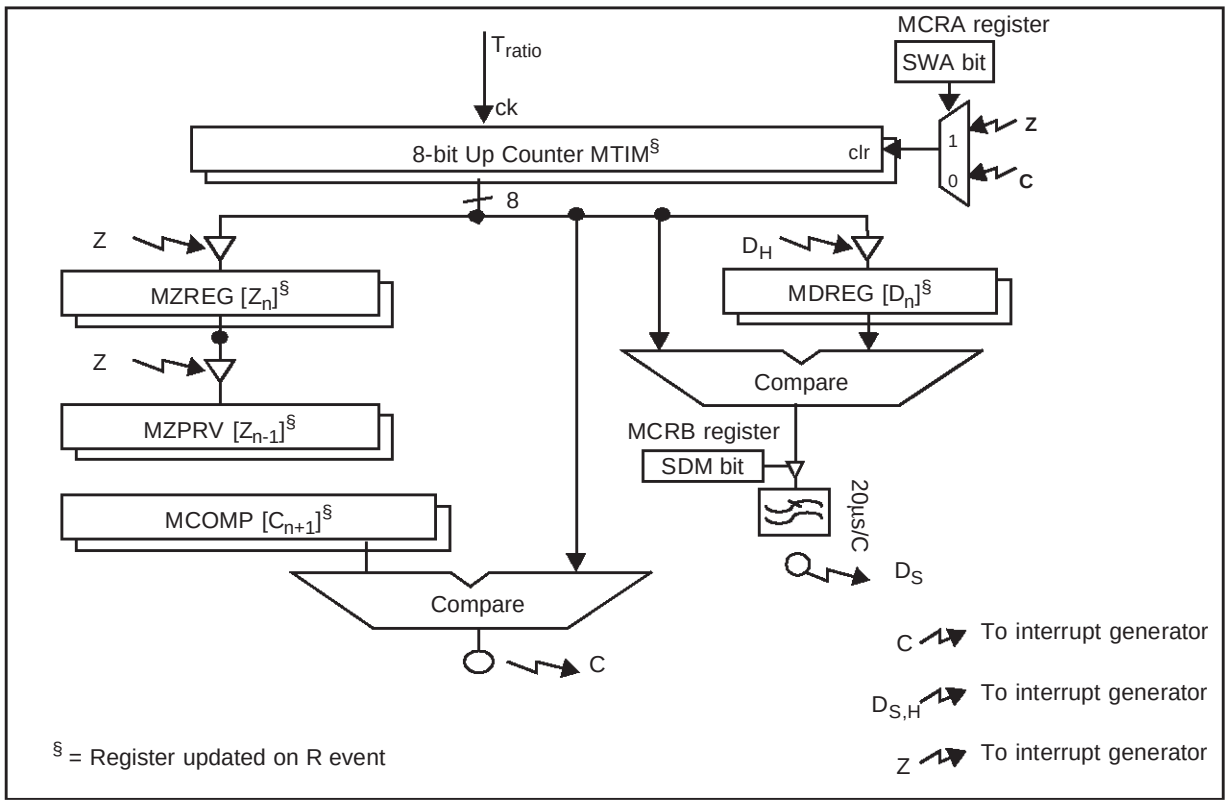
Table 15. Modes permitting BEMF reading after Demagnetization (D event)

SR bit (Sensor/ Sensor-less Mode)	**Demagnet-ization**	**V0C1 bit** (Voltage/ Current Mode)	**OS[2:0] bits** (PWM Output Config.)	**Significant PWM Group**	**REO bit** (Read BEMF on Even/Odd group)	**BEMF reading permitted after D event when:**
0	After D event	1	x00	Even	0	Sensorless mode, Current Mode, PWM output only on Even group and BEMF read on Even group
			x01	Odd	1	Sensorless mode, Current Mode, PWM output only on Odd group and BEMF read on Odd group
			x10	Even	0	Sensorless mode, Current Mode, PWM output on alternate groups but BEMF read only on Even group
			x10	Odd	1	Sensorless mode, Current Mode, PWM output on alternate groups but BEMF read only on Odd group
		0	000	Even	0	Sensorless mode, Voltage Mode, PWM output only on Even group and BEMF read on Even group
			001	Odd	1	Sensorless mode, Voltage Mode, PWM output only on Odd group and BEMF read on Odd group
			100	Even	0	Sensorless mode, Voltage Mode, PWM output only on Even group and BEMF Read on Even group
			101	Odd	1	Sensorless mode, Voltage Mode, PWM output only on Odd group and BEMF Read on Odd group
			110	Even	0	Sensorless mode, Voltage Mode, PWM output on alternate groups but BEMF read only on Even group
			110	Odd	1	Sensorless mode, Voltage Mode, PWM output on alternate groups but BEMF read on Odd group
		x	x11	Even and Odd	x	Sensorless mode, Current or Voltage Mode, PWM output on both groups, BEMF read on either group
1	Not Used	x	xxx	Odd or Even	x	Sensor Mode, in any PWM output configu-ration, BEMF read on either group
Other cases						BEMF reading forbidden

MOTOR CONTROLLER (Cont'd)

8.1.4.2 Delay Manager

Figure 33. Overview of MTIM Timer



This part of the MTC contains all the time-related functions, its architecture is based on an 8-bit shift left/shift right timer shown in Figure 33. The MTIM timer includes:

– An auto-updated prescaler

– A capture/compare register for software demagnetization simulation (MDREG)

– Two cascaded capture register (MZREG and MZPRV) for storing the times between two consecutive BEMF zero crossings (Z events)

– An 8x8 bit multiplier for auto computing the next commutation time

– One compare register for phase commutation generation (MCOMP)

The MTIM timer module can work in two main modes. In switched mode the user must process the step duration and commutation time by software, in autoswitched mode the commutation action is performed automatically depending on the rotor position information and register contents.

Table 16. Switched and Autoswitched Modes

SWA bit	Commutation Type	MCOMP User access
0	Switched mode	Read/Write
1	Autoswitched mode	Read only

Switched Mode

This feature allows the motor to be run step-by-step. This is useful when the rotor speed is still too low to generate a BEMF. It can also run other kinds of motor without BEMF generation such as induction motors or switch reluctance motors. This mode can also be used for autoswitching with all computation for the next commutation time done by software (hardware multiplier not used) and using the powerful interrupt set of the peripheral.

In this mode, the step time is directly written by software in the commutation compare register MCOMP. When the MTIM timer reaches this value a commutation occurs (C event) and the MTIM timer is reset.

MOTOR CONTROLLER (Cont'd)

At this time all registers with a preload function are loaded (registers marked with (*) in Section 8.1.7). The CI bit of MISR is set and if the CIM bit in the MISR register is set an interrupt is generated.

An overflow of the MTIM timer generates an RPI interrupt if the RIM bit is set.

The MTIM timer prescaler (Step ratio bits ST[3:0] in the MPRSR register) is user programmable. Access to this register is not allowed while the MTIM timer is running (access is possible only before the starting the timer by means of the MOE bit) but the prescaler contents can be incremented/decremented at the next commutation event by setting the RMI (decrement) or RPI (increment) bits in the MISR register. When this method is used, at the next commutation event the prescaler value will be updated but also all the MTIM timer-related registers will be shifted in the appropriate direction to keep their value. After it has been taken into account, (at commutation) the RPI or RMI bit is reset. See Table 17.

Only one update per step is allowed, so if both RPI and RMI are set together, RPI is taken into account at the next commutation and RMI is used one commutation latter.

In switched mode, BEMF and demagnetization detection are already possible in order to pass in autoswitched mode as soon as possible but Z and D events do not affect the timer contents.

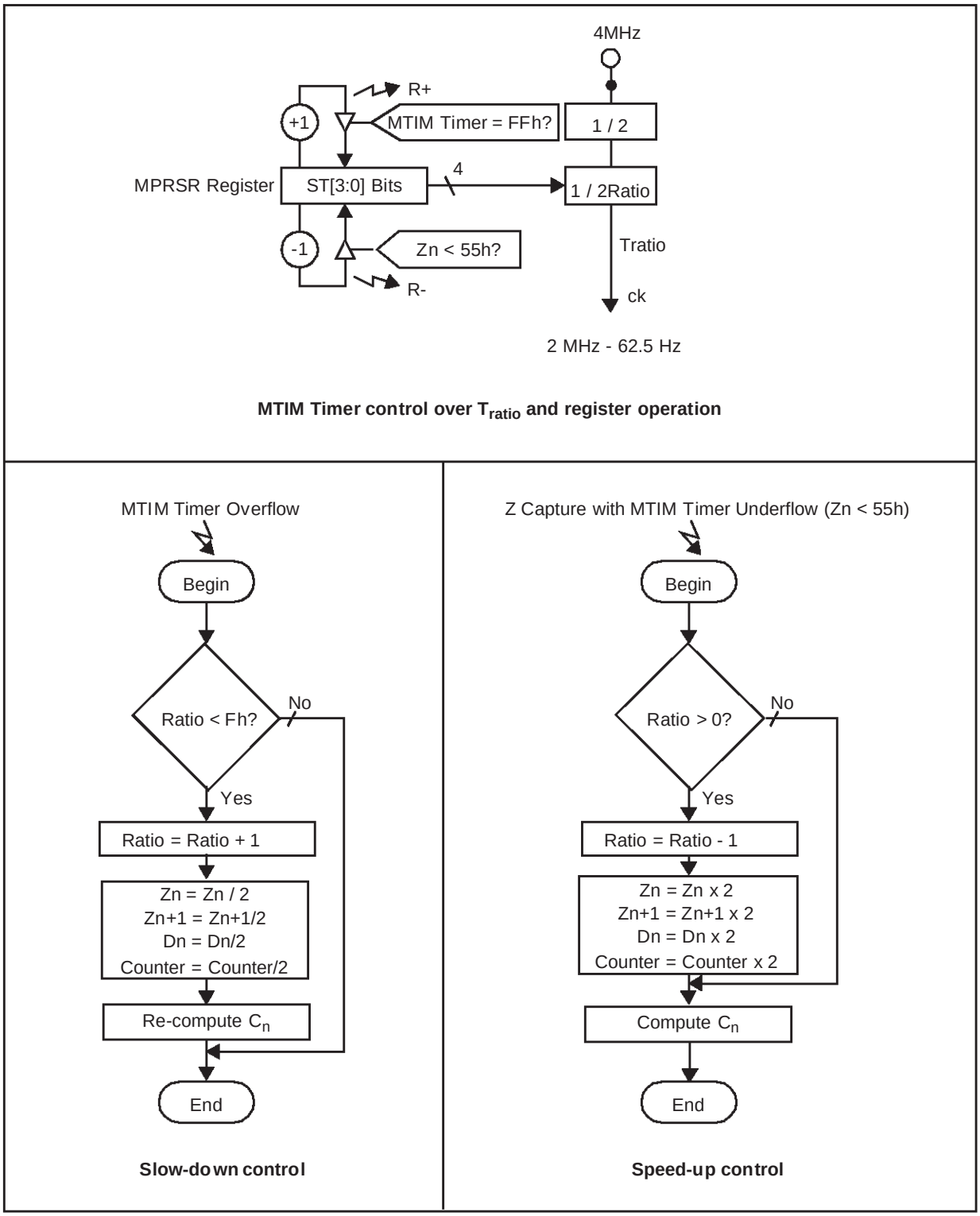
Warning: In this mode, MCOMP must never be written to 0.

Table 17. Step Ratio Update

MOE bit	SWA bit	Clock State	Read	Ratio Increment (Slow Down)	Ratio Decrement (Speed-Up)
0	x	Disabled	Always possible	Write the ST[3:0] value directly in the MPRSR register	
1	0	Enabled		Set RPI bit in the MISR register till next commutation	Set RMI bit in the MISR register till next commutation
1	1	Enabled		Automatically updated according to MZREG value	

MOTOR CONTROLLER (Cont'd)

Figure 34. Step Ratio Functional Diagram



MOTOR CONTROLLER (Cont'd)

Autoswitched Mode

In this mode the MCOMP register content is automatically computed in real time as described below and in Figure 35. This register is READ ONLY.

The C event has no effect on the contents of the MTIM timer.

When a Z event occurs the MTIM timer value is captured in the MZREG register, the previous captured value is shifted into the MZPRV register and the MTIM timer is reset. See Figure 26.

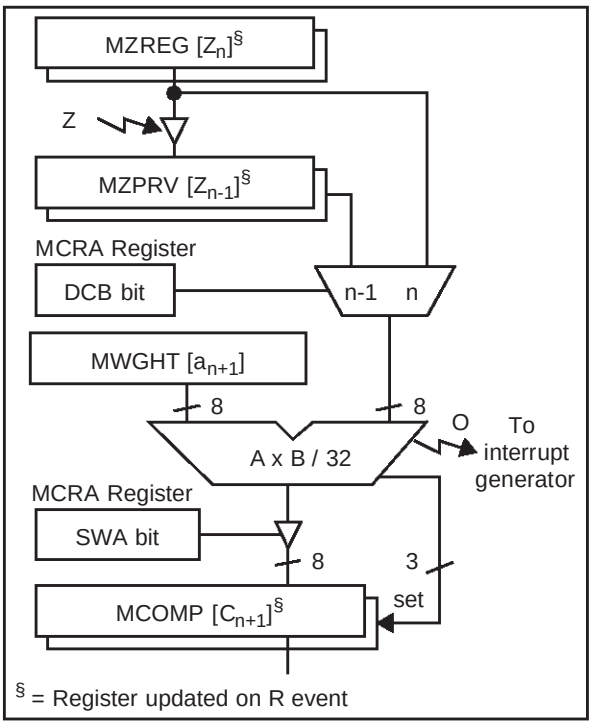
One of these two registers (depending on the DCB bit in the MCRA register) is multiplied with the contents of the MWGHT register and divided by 32. The result is loaded in the MCOMP compare register, which automatically triggers the next commutation (C event)

Table 18. Multiplier Result

DCB bit	Commutation Delay
0	MCOMP = MWGHT x MZPRV / 32
1	MCOMP = MWGHT x MZREG / 32

When an overflow occurs during the multiply operation, FFh is written in the MCOMP register and an interrupt (O event) is generated if enabled by the OIM bit in the MIMR register.

Figure 35. Commutation Processor Block



When the timer reaches this value an RPI interrupt is generated (timer overflow).

After each shift operation the multiply is recomputed for greater precision.

Using either the MZREG or MZPRV register depends on the motor symmetry and type.

The MWGHT register gives directly the phase shift between the motor driven voltage and the BEMF. This parameter generally depends on the motor and on the speed.

Auto-updated Step Ratio Register: In switched mode, the MTIM timer is driven by software only and any prescaler change has to be done by software (see Section 8.1.4.2 for more details).

– In autoswitched mode an auto-updated prescaler always configures the MTIM timer for best accuracy. Figure 34 shows process of updating the Step Ratio bits:

– When the MTIM timer value reaches FFh, the prescaler is automatically incremented in order to slow down the MTIM timer and avoid an overflow. To keep consistent values, the MTIM register and all the relevant registers are shifted right (divided by two). The RPI bit in the MISR register is set and an interrupt is generated (if RIM is set).

– When a Z-event occurs, if the MTIM timer value is below 55h, the prescaler is automatically decremented in order to speed up the MTIM timer and keep precision better than 1.2%. The MTIM register and all the relevant registers are shifted left (multiplied by two). The RMI bit in the MISR register is set and an interrupt is generated if RIM is set.

– If the prescaler contents reach the value 0, it can no longer be automatically decremented, the MTC continues working with the same prescaler value, i.e. with a lower accuracy. No RMI interrrupt can be generated.

– If the prescaler contents reach the value 15, it can no longer be automatically incremented. When the timer reaches the value FFh, the prescaler and all the relevant registers remain unchanged and no interrupt is generated, the timer clock is disabled, and its contents stay at FFh The PWM is still generated and the D and Z detection circuitry still work, enabling the capture of the maximum timer value.

The automatically updated registers are: MTIM, MZREG, MZPRV, MCOMP and MDREG. Access to these registers is summarised in Table 21.

MOTOR CONTROLLER (Cont'd)

Table 19. MTIM Timer-related Registers

Name	Reset Value	Contents
MTIM	00h	Timer Value
MZPRV	00h	Capture Zn-1
MZREG	00h	Capture Zn
MCOMP	00h	Compare Cn+1
MDREG	00h	Demagnetization Dn

Note on using the auto-updated MTIM timer: The auto-updated MTIM timer works accurately within its operating range but some care has to be taken when processing timer-dependent data such as the step duration for regulation or demagnetization.

For example if an overflow occurs when calculating a software end of demagnetization (MCOMP+demagnetisation_time>FFh), the value that stored in MDREG will be:
7Fh+(MCOMP+demagnetization_time-FFh)/2.

Note on commutation interrupts: It is good practice to modify the configuration for the next step as soon as possible, i.e within the commutation interrupt routine.

All registers that need to be changed at each step have a preload register that enables the modifications for a complete new configuration to be performed at the same time (at C event in normal mode or when writing the MPHST register in direct access mode).

These configuration bits are:

CPB, HDM, SDM and OS2 in the MCRB register and IS[1:0], OO[5:0] in the MPHST register.

Note on initializing the MTC: As shown in Table 21 all the MTIM timer registers are in read-write mode until the MTC clock is enabled (with the MOE and DAC bits). This allows the timer, prescaler and compare registers to be properly initialized for start-up.

In sensorless mode, the motor has to be started in switched mode until a BEMF voltage is present on the inputs. This means the prescaler ST[3:0] bits and MCOMP register have to be modified by software. When running the ST[3:0] bits can only be incremented/decremented, so the initial value is very important.

When starting directly in autoswitched mode (in sensor mode for example), write an appropriate value in the MZREG and MZPRV register to perform a step calculation as soon as the clock is enabled.

MOTOR CONTROLLER (Cont'd)

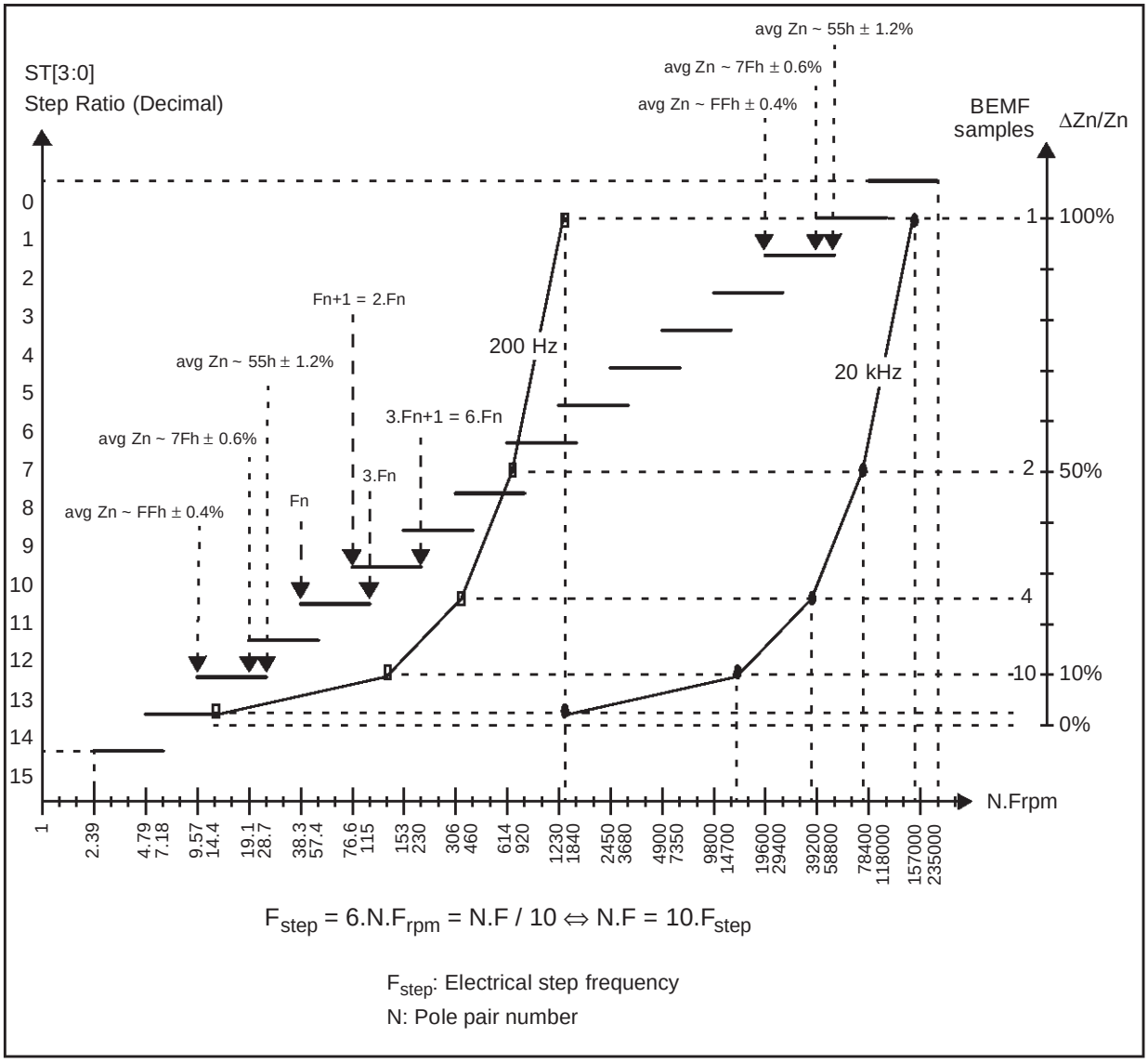
The Figure 36 gives the step ratio register value (left axis) and the number of BEMF sampling during one electrical step with the corresponding accuracy on the measure (right axis) as a function of the mechanical frequency.

For a given prescaler value (step ratio register) the mechanical frequency can vary between two fixed values shown on the graph as the segment ends. In autoswitched mode, this register is automatically incremented/decremented when the step frequency goes out of this segment.

At f_{cpu}=4MHz, the range covered by the Step Ratio mechanism goes from 2.39 to 235000 (pole pair x rpm) with a minimum accuracy of 1.2% on the step period.

To read the number of samples for Zn within one step (right Y axis), select the mechanical frequency on the X axis and the sampling frequency curve used for BEMF detection (PWM frequency or measurment window frequency). For example, for N.Frpm = 15,000 and a sampling frequency of 20kHz, there are approximately 10 samples in one step and there is a 10% error rate on the measurement.

Figure 36. Step Ratio Bits decoding and accuracy results and BEMF Sampling Rate



$$F_{step} = 6.N.F_{rpm} = N.F / 10 \Leftrightarrow N.F = 10.F_{step}$$

F_{step}: Electrical step frequency

N: Pole pair number

MOTOR CONTROLLER (Cont'd)

Table 20. Step Frequency/Period Range

Step Ratio Bits ST[3:0]	Maximum Step Frequency	Minimum Step Frequency	Minimum Step Period	Maximum Step Period
0000	23.5 kHz	7.85 kHz	42.5 µs	127.5 µs
0001	11.7 kHz	3.93 kHz	85 µs	255 µs
0010	5.88 kHz	1.96 kHz	170 µs	510 µs
0011	2.94 kHz	980 Hz	340 µs	1.02 ms
0100	1.47 kHz	490 Hz	680 µs	2.04 ms
0101	735 Hz	245 Hz	1.36 ms	4.08 ms
0110	367 Hz	123 Hz	2.72 ms	8.16 ms
0111	183 Hz	61.3 Hz	5.44 ms	16.32 ms
1000	91.9 Hz	30.7 Hz	10.9 ms	32.6 ms
1001	45.9 Hz	15.4 Hz	21.8 ms	65.2 ms
1010	22.9 Hz	7.66 Hz	43.6 ms	130 ms
1011	11.4 Hz	3.83 Hz	87 ms	261 ms
1100	5.74 Hz	1.92 Hz	174 ms	522 ms
1101	2.87 Hz	0.958 Hz	349 ms	1.04 s
1110	1.43 Hz	0.479 Hz	697 ms	2.08 s
1111	0.718 Hz	0.240 Hz	1.40 s	4.17 s

Table 21. Modes of Accessing MTIM Timer-Related Registers

State of MCRA Register Bits				Access to MTIM Timer Related Registers	
RST bit	SWA bit	MOE bit	Mode	Read Only Access	Read / Write Access
0	0	0	Configuration Mode		MTIM, MZPRV, MZREG, MCOMP, MDREG, ST[3:0]
0	0	1	Switched Mode	MTIM, MZPRV, MZREG, ST[3:0]	MCOMP, MDREG, RMI bit of MISR: 0: No action 1: Decrement ST[3:0] RPI bit of MISR: 0: No action 1: Increment ST[3:0]
0	1	0	Emergency Stop		MTIM, MZPRV, MZREG, MCOMP, MDREG, ST[3:0]
0	1	1	Autoswitched Mode	MTIM, MZPRV, MZREG, MCOMP, ST[3:0]	MDREG,RMI, RPI bit of MISR: Set by hardware, (increment ST[3:0]) Cleared by software

MOTOR CONTROLLER (Cont'd)

8.1.4.3 PWM Manager

The PWM manager controls the motor via the six output channels in voltage mode or current mode depending on the V0C1 bit in the MCRA register. A block diagram of this part is given in Figure 37.

Voltage Mode

In Voltage mode (V0C1 bit = "0"), the PWM is generated by the 16-bit A Timer.

Its duty cycle is programmed by software (refer to the chapter on the 16-bit Timer) as required by the application (speed regulation for example).

The current comparator is used for safety purposes as a current limitation. For this feature, the detected current must be present on the MCCFI pin and the current limitation must be present on pin OCP1A. This current limitation is fixed by a voltage reference depending on the maximum current acceptable for the motor. This current limitation is generated with the V_{DD} voltage by means of an external divider but can also be adjusted with an external reference voltage (≤ 5 V). The external components are adjusted by the user depending on the application needs. In Voltage mode, it is mandatory to set a current limitation.

In sensorless mode the BEMF zero crossing is done during the PWM off time.

The PWM signal is directed to the channel manager that connects it to the programmed outputs (See Figure 39).

Current Mode

In current mode, the PWM output signal is generated by a combination of the output of the measurement window generator (see Figure 38) and the output of the current comparator, and is directed to the output channel manager as well (Figure 39).

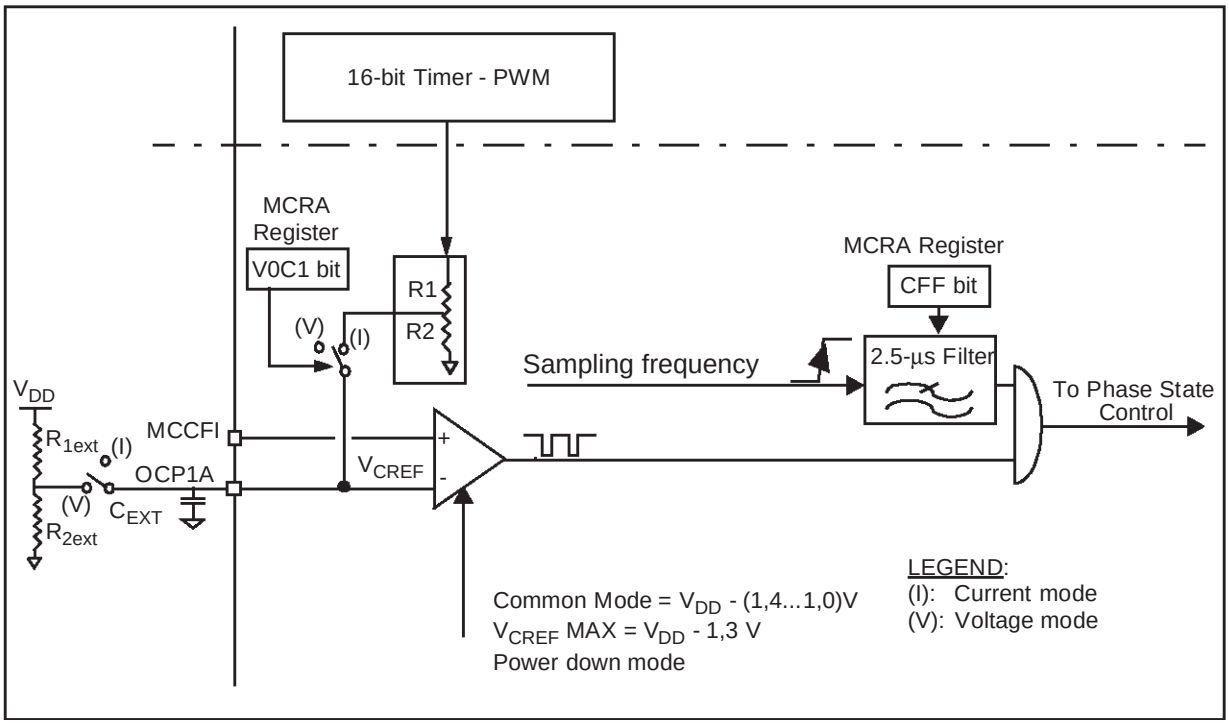
The current reference is provided to the comparator by the PWM output of the 16-bit Timer (0.25% accuracy), filtered through a RC filter (external capacitor on pin OCP1A and an internal voltage divider 30K and 70K).

The detected current input must be present on the MCCFI pin.

To avoid spurious commutations due to parasitic noise after switching on the PWM, a 2.5-µs filter can be applied on the comparator output by setting the CFF bit in the MCRB register.

The On state of the resulting PWM starts at the end of the measurement window (rising edge), and ends either at the beginning of the next measurement window (falling edge), or when the current level is reached.

Figure 37. Current Feedback



MOTOR CONTROLLER (Cont'd)

The measurement window frequency can be programmed between 195Hz and 25KHz by the means of the SA[3:0] bits in the MPRSR register. In sensorless mode this measurement window can be used to detect either End of Demagnetization or BEMF zero crossing events. Its width can be defined between 5µs and 30µs in sensorless mode by the OT[1:0] bits in the MPOL register. In sensor mode (SR=1) this off time is fixed at 1.25µs.

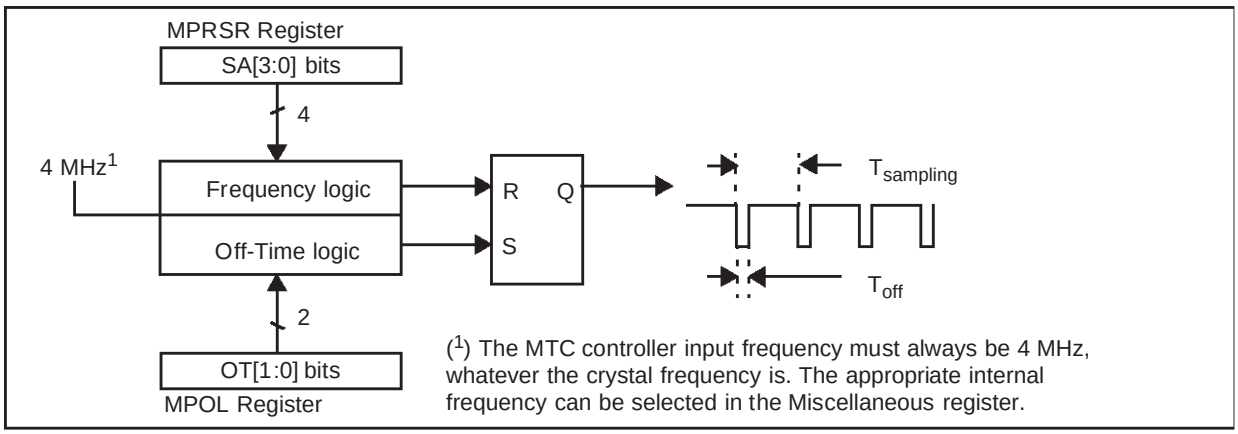
Table 22. Off-Time Table

OT1 bit	OT0 bit	Off-Time Sensorless Mode (SR bit=0)	Off-Time Sensor Mode (SR bit =1)
0	0	5 µs	1.25 µs
0	1	10 µs	
1	0	15 µs	
1	1	30 µs	

Table 23. Sampling Frequency Selection

SA3	SA2	SA1	SA0	Sampling Frequency
0	0	0	0	25.0 KHz
0	0	0	1	20.0 KHz
0	0	1	0	18.1 KHz
0	0	1	1	15.4 KHz
0	1	0	0	12.5 KHz
0	1	0	1	10.0 KHz
0	1	1	0	6.25 KHz
0	1	1	1	3.13 KHz
1	0	0	0	1.56 KHz
1	0	0	1	1.25 KHz
1	0	1	0	1.14 KHz
1	0	1	1	961 Hz
1	1	0	0	781 Hz
1	1	0	1	625 Hz
1	1	1	0	390 Hz
1	1	1	1	195 Hz

Figure 38. Sampling clock generation block



MOTOR CONTROLLER (Cont'd)

8.1.4.4 Channel Manager

The channel manager consists of:

– A Phase State register with preload and polarity function

– A multiplexer to direct the PWM to the odd and/or even channel group

– A tristate buffer asynchronously driven by an emergency input.

The block diagram is shown in Figure 39.

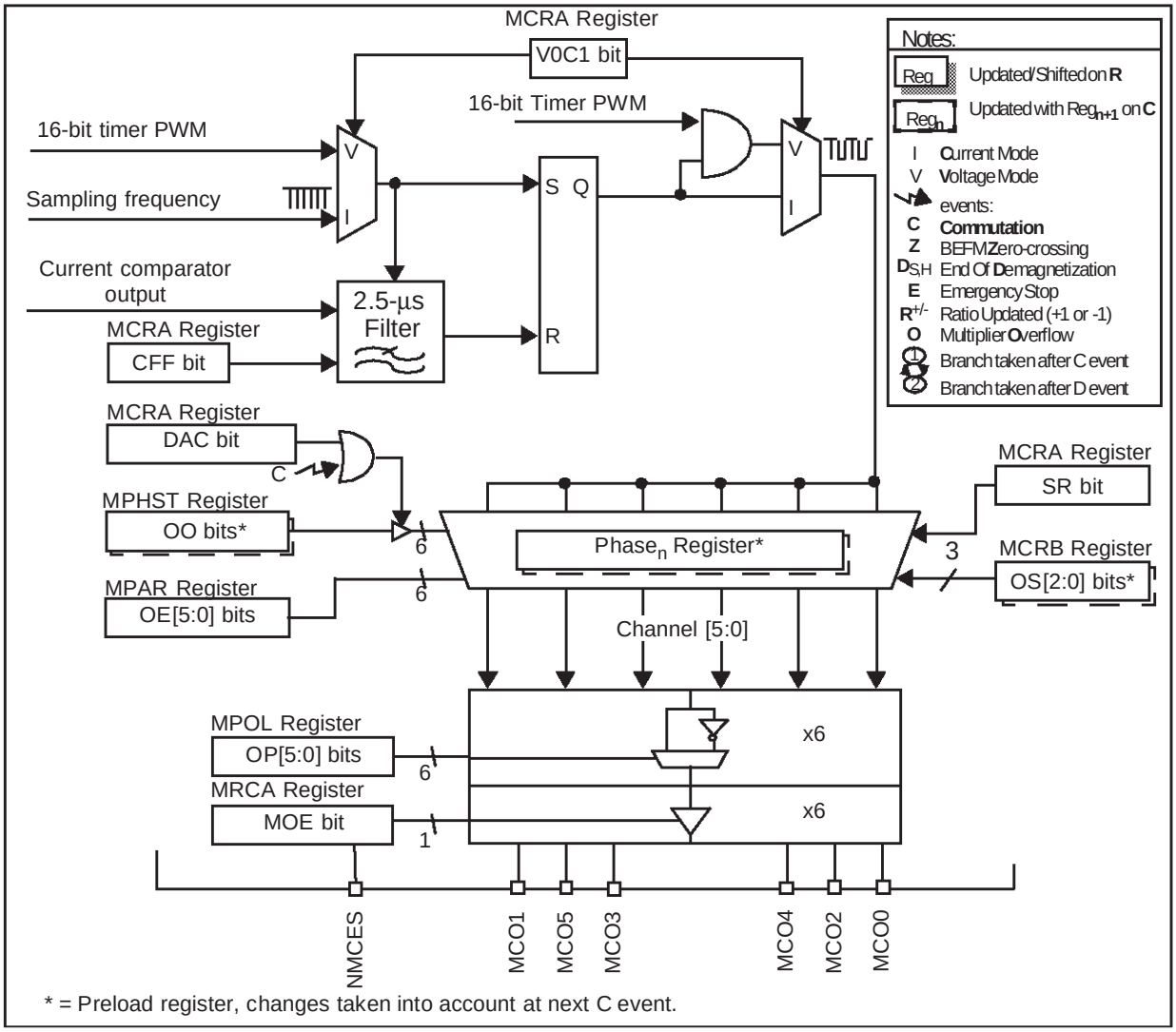
MPHST Phase State Register

A preload register enables software to asynchronously update (during the previous commutation interrupt routine for example) the channel configuration for the next step: the OO[5:0] bits in the MPHST register are copied to the Phase register on a C event.

Table 24. Output State

OP[5:0] bit	OO[5:0] bit	MCO[5:0] Pin
0	0	1 (OFF)
0	1	0-(PWM allowed)
1	0	0 (OFF)
1	1	1-(PWM allowed)

Figure 39. Channel Manager Block Diagram



MOTOR CONTROLLER (Cont'd)

Direct access to the phase register is also possible when the DAC bit in the MCRA register is set.

Table 25. DAC and MOE Bit Meaning

MOE bit	DAC bit	Effect on Output	Effect on MTIM Timer
0	x	High Z	Clock disabled
1	0	Standard running mode	Standard running mode
1	1	MPHST value same as MPOL value	Clock disabled

The polarity register is used to match the polarity of the power drivers keeping the same control logic and software. If one of the OPx bits in the MPOL register is set, this means the switch x is ON when MCOx is V_{DD}.

Each output status depends also on the momentary state of the PWM, its group (odd or even), and the peripheral state.

PWM Features

The outputs can be split in two PWM groups in order to differentiate the high side and the low side switches. This output property can be programmed using the OE[5:0] bits in the MPAR register

Table 26. Meaning of the OE[5:0] Bits

OE[5:0]	Channel group
0	Even channel
1	Odd channel

The multiplexer directs the PWM to the upper channel, the lower channel or both of them alternatively or simultaneously according to the peripheral state.

This means that the PWM can affect any of the upper or lower channels allowing the selection of the most appropriate reference potential when freewheeling the motor in order to:

– Improve system efficiency
– Speed up the demagnetization phase
– Enable Back EMF zero crossing detection.

The OS[2:0] bits in the MCRB register allow the PWM configuration to be configured for each case as shown in Figure 41, Figure 42 and Figure 40. This configuration depends also on the current/voltage mode (V0C1 bit in the MCRA register) because the OS[2:0] have not the same meaning in voltage mode and in current mode.

During demagnetization, the OS2 bit is used to control PWM mode, and it is latched in a preload register so it can be modified when a commutation event occurs.

The OS[1:0] bits are used to control the PWM between the D and C events.

Warning: In Voltage Mode the OS[2:0] bits have a special configuration value: OS[2:0] = 010.

In this mode, there is NO current limitation and NO PWM applied to active outputs. The active outputs are always at 100% whether in demagnetization, or normal mode.

Note about demagnetization speed-up: during demagnetization the voltage on the winding has to be as high as possible in order to reduce the demagnetization time. Software can apply a different PWM configuration on the outputs between the C and D events, to force the free wheeling on the appropriate diodes to maximize the demagnetization voltage.

Emergency Feature

When the NMCES pin goes low

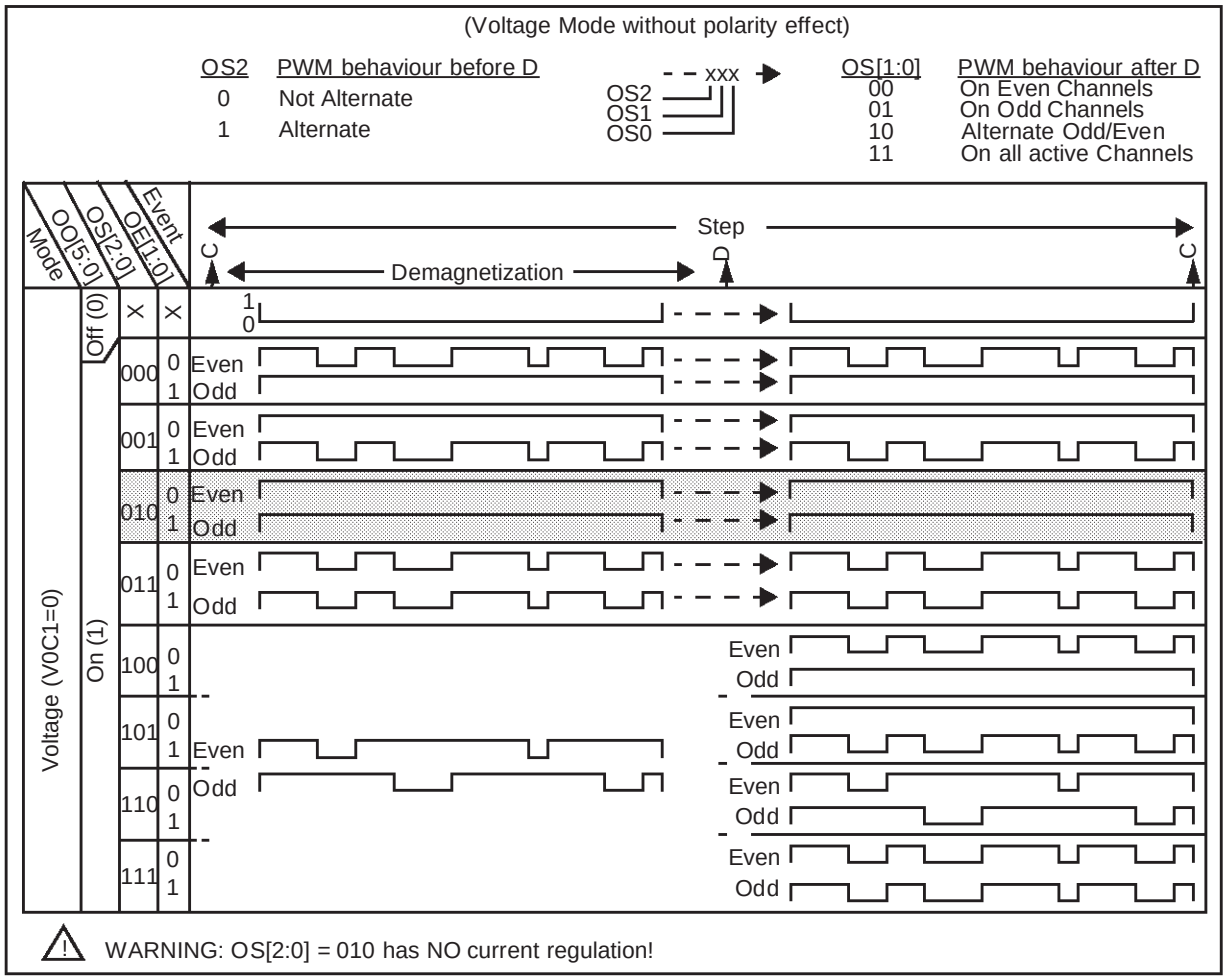
– The tristate output buffer is put in HiZ asynchronously
– The MOE bit in the MCRA register is reset
– An interrupt request is sent to the CPU if the EIM bit in the MIMR register is set

This bit can be connected to an alarm signal from the drivers, thermal sensor or any other security component.

This feature functions even if the MCU oscillator is off.

MOTOR CONTROLLER (Cont'd)

Figure 40. Step Behaviour of one Output Channel MCO[n] in Voltage Mode



MOTOR CONTROLLER (Cont'd)

Figure 41. Step Behaviour of one Output Channel MCO[n] in Current / Sensorless Mode

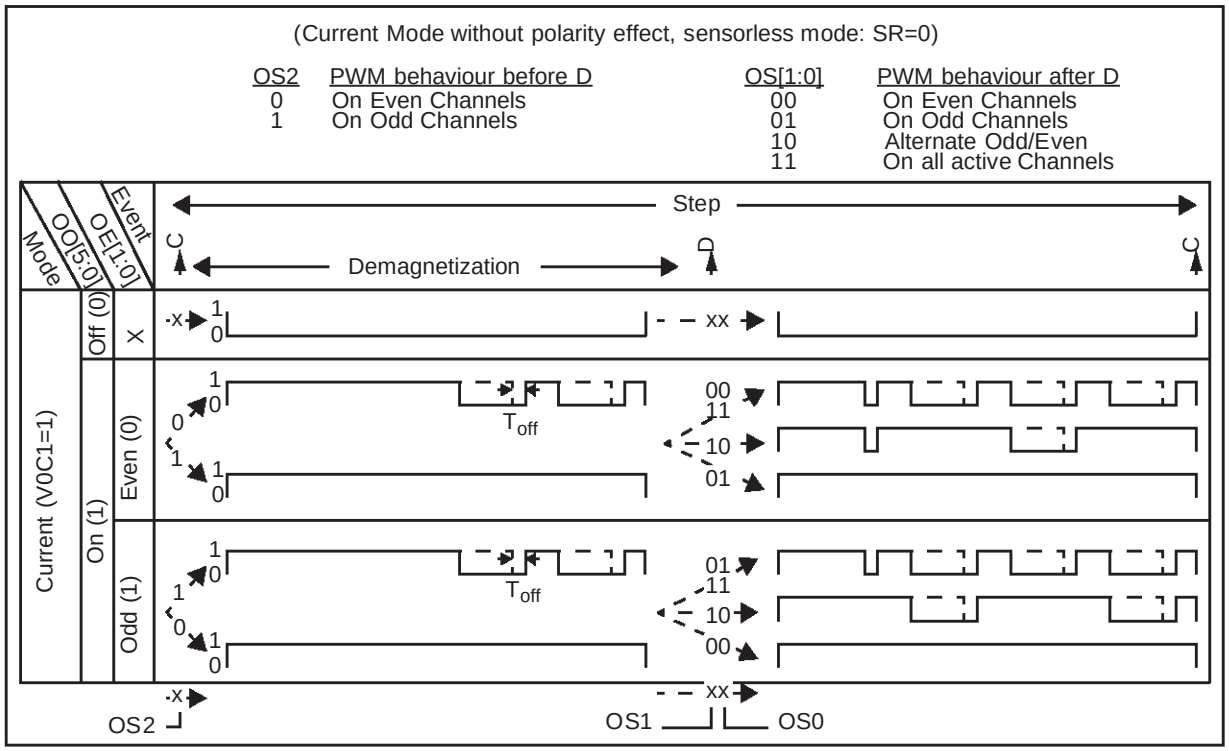
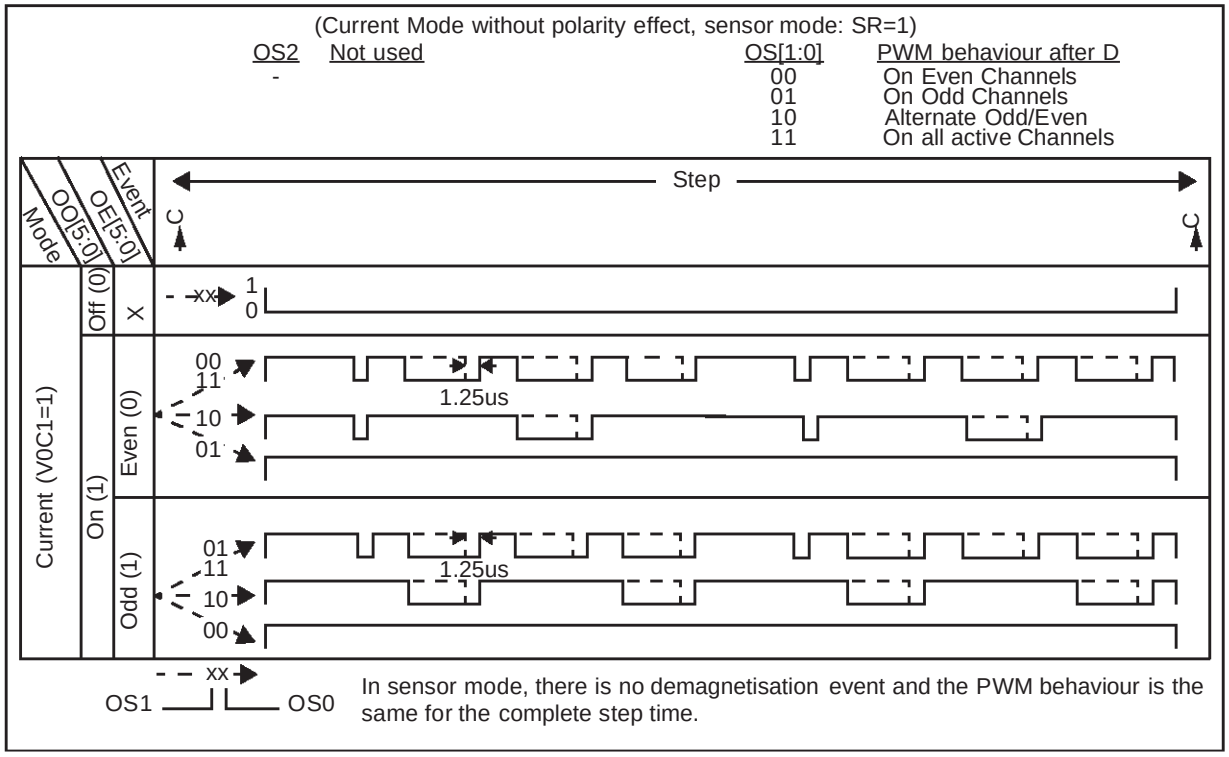


Figure 42. Step Behaviour of one Output Channel MCO[n] in Current / Sensor Mode



MOTOR CONTROLLER (Cont'd)

8.1.5 Low Power Modes

Before executing a HALT or WFI instruction, software must stop the motor, and may choose to put the outputs in high impedance.

Mode	Description
WAIT	No effect on MTC interface. MTC interrupts exit from Wait mode.
HALT	MTC registers are frozen. In Halt mode, the MTC interface is inactive. The MTC interface becomes operational again when the MCU is woken up by an interrupt with "exit from Halt mode" capability.

8.1.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Ratio increment	RPI	RIM	Yes	No
Ratio decrement	RMI		Yes	No
Multiplier overflow	OI	OIM	Yes	No
Emergency Stop	EI	EIM	Yes	No
BEMF Zero-Crossing	ZI	ZIM	Yes	No
End of Demagnetization	DI	DIM	Yes	No
Commutation	CI	CIM	Yes	No

The MTC interrupt events are connected to the three interrupt vectors (see Interrupts chapter).

They generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

MOTOR CONTROLLER (Cont'd)

8.1.7 Register Description

TIMER COUNTER REGISTER (MTIM)
Read Only
Reset Value: 0000 0000 (00h)

7							0
T7	T6	T5	T4	T3	T2	T1	T0

Bit 7:0 = **T[7:0]**: *MTIM Counter Value.*
These bits contain the current value of the 8-bit up counter.

CAPTURE Z_{n-1} REGISTER (MZPRV)
Read Only
Reset Value: 0000 0000 (00h)

7							0
ZP7	ZP6	ZP5	ZP4	ZP3	ZP2	ZP1	ZP0

Bit 7:0 = **ZP[7:0]**: *Previous Z Value.*
These bits contain the previous captured BEMF value (Z_{N-1}).

CAPTURE Z_n REGISTER (MZREG)
Read Only
Reset Value: 0000 0000 (00h)

7							0
ZC7	ZC6	ZC5	ZC4	ZC3	ZC2	ZC1	ZC0

Bit 7:0 = **ZC[7:0]**: *Current Z Value.*
These bits contain the current captured BEMF value (Z_N).

COMPARE C_{n+1} REGISTER (MCOMP)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

Bit 7:0 = **DC[7:0]**: *Next Compare Value.*
These bits contain the compare value for the next commutation (C_{N+1}).

DEMAGNETIZATION REGISTER (MDREG)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
DN7	DN6	DN5	DN4	DN3	DN2	DN1	DN0

Bit 7:0 = **DN[7:0]**: *D Value.*
These bits contain the compare value for software demagnetization (D_N) and the captured value for hardware demagnetization (D_H).

A_N WEIGHT REGISTER (MWGHT)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

Bit 7:0 = **AN[7:0]**: *A Weight Value.*
These bits contain the A_N weight value for the multiplier. In autoswitched mode the MCOMP register is automatically loaded with:

$$\frac{Z_n \times MWGHT}{32(d)} \quad \text{or} \quad \frac{Z_{N-1} \times MWGHT}{32(d)} \qquad (*)$$

when a Z event occurs.
(*) depending on the DCB bit in the MCRA register.

PRESCALER & SAMPLING REGISTER (MPRSR)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
SA3	SA2	SA1	SA0	ST3	ST2	ST1	ST0

Bit 7:4 = **SA[3:0]**: *Sampling Ratio.*
These bits contain the sampling ratio value for current mode. Refer to Table 23.

Bit 3:0 = **ST[3:0]**: *Step Ratio.*
These bits contain the step ratio value. It acts as a prescaler for the MTIM timer and is auto incremented/decremented with each R+ or R- event. Refer to Table 20.

MOTOR CONTROLLER (Cont'd)

INTERRUPT MASK REGISTER (MIMR)
Read/Write (except bits 7:6)
Reset Value: 0000 0000 (00h)

7							0
HST	CL	RIM	OIM	EIM	ZIM	DIM	CIM

Bit 7 = **HST**: *Hysteresis Comparator Value.*
This read only bit contains the hysteresis comparator output.
0: Demagnetisation/BEMF comparator is under V_{REF}
1: Demagnetisation/BEMF comparator is above V_{REF}

Bit 6 = **CL**: *Current Loop Comparator Value.*
This read only bit contains the current loop comparator output value.
0: Current detect voltage is under V_{CREF}
1: Current detect voltage is above V_{CREF}

Bit 5 = **RIM**: *Ratio update Interrupt Mask bit.*
0: Ratio update interrupts (R+ and R-) disabled
1: Ratio update interrupts (R+ and R-) enabled

Bit 4 = **OIM**: *Multiplier Overflow Interrupt Mask bit.*
0: Multiplier Overflow interrupt disabled
1: Multiplier Overflow interrupt enabled

Bit 3 = **EIM**: *Emergency stop Interrupt Mask bit.*
0: Emergency stop interrupt disabled
1: Emergency stop interrupt enabled

Bit 2 = **ZIM**: *Back EMF Zero-crossing Interrupt Mask bit.*
0: BEMF Zero-crossing Interrupt disabled
1: BEMF Zero-crossing Interrupt enabled

Bit 1 = **DIM**: *End of Demagnetization Interrupt Mask bit.*
0: End of Demagnetization interrupt disabled
1: End of Demagnetization interrupt enabled if the HDM or SDM bit in the MCRB register is set

Bit 0 = **CIM**: *Commutation Interrupt Mask bit*
0: Commutation Interrupt disabled
1: Commutation Interrupt enabled

INTERRUPT STATUS REGISTER (MISR)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
0	RPI	RMI	OI	EI	ZI	DI	CI

Bit 7 = Reserved. Forced by hardware to 0.

Bit 6 = **RPI**: *Ratio Increment interrupt flag.*
Autoswitched mode (SWA bit =0)**:**
0: No R+ interrupt pending
1: R+ Interrupt pending

Switched mode (SWA bit =1)**:**
0: No R+ action
1: The hardware will increment the ST[3:0] bits when the next commutation occurs and shift all timer registers right.

Bit 5 = **RMI**: *Ratio Decrement interrupt flag.*
Autoswitched mode (SWA bit =0)**:**
0: No R- interrupt pending
1: R- Interrupt pending
Switched mode (SWA bit =1)**:**
0: No R- action
1: The hardware will decrement the ST[3:0] bits when the next commutation occurs and shift all timer registers left.

Bit 4 = **OI**: *Multiplier Overflow interrupt flag.*
0: No Multiplier Overflow interrupt pending
1: Multiplier Overflow interrupt pending

Bit 3 = **EI**: *Emergency stop Interrupt flag.*
0: No Emergency stop interrupt pending
1: Emergency stop interrupt pending

Bit 2 = **ZI**: *BEMF Zero-crossing interrupt flag.*
0: No BEMF Zero-crossing Interrupt pending
1: BEMF Zero-crossing Interrupt pending

Bit 1 = **DI**: *End of Demagnetization interrupt flag.*
0: No End of Demagnetization interrupt pending
1: End of Demagnetization interrupt pending

Bit 0 = **CI**: *Commutation interrupt flag*
0: No Commutation Interrupt pending
1: Commutation Interrupt pending

MOTOR CONTROLLER (Cont'd)

Table 27. Step Ratio Update

MOE bit	SWA bit	Clock State	Read	Ratio Increment (Slow Down)	Ratio Decrement (Speed-Up)
0	x	Disabled	Always possible	Write the ST[3:0] value directly in the MPRSR register	
1	0	Enabled		Set RPI bit in the MISR register **till** next commutation	Set RMI bit in the MISR register **till** next commutation
1	1	Enabled		Updated automatically according to MZREG value	

CONTROL REGISTER A (MCRA)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
MOE	RST	SR	DAC	V0C1	SWA	CFF	DCB

Bit 7 = **MOE:** *Output Enable bit.*
0: Outputs and Clocks disabled
1: Outputs and Clocks enabled

MOE bit	MCO[5:0] Output pin State
0	Tristate
1	Output enabled

Bit 6 = **RST**: *Reset MTC registers.*
Software can set this bit to reset all MTC registers without resetting the ST7.
0: No MTC register reset
1: Reset all MTC registers

Bit 5 = **SR**: *Sensor ON/OFF.*
0: Sensorless mode
1: Sensor mode

Table 28. Sensor Mode Selection

SR bit	Mode	OS2 bit enable	Behaviour of the output PWM
0	Sensors not used	OS2 enabled	"Before D" behaviour & "after D" behaviour
1	Sensors used	OS2 disabled	Only "after D" behaviour

Bit 4 = **DAC**: *Direct Access to phase state register.*
0: No Direct Access (reset value). In this mode all the registers with a preload register are taken into account at the C event.
1: Direct Access enabled. In this mode, write a value in the MPHST register to access the outputs directly. All other registers with a preload register are taken into account at the same time.

Table 29. DAC Bit Meaning

MOE bit	DAC bit	Effect on Output	Effect on MTIM Timer
0	x	High Z	Clock disabled
1	0	Standard running mode	Standard running mode
1	1	MPHST register value (depending on MPOL register value)	Clock disabled

Bit 3 = **V0C1**: *Voltage/Current Mode*
0: Voltage Mode
1: Current Mode

Bit 2 = **SWA**: *Switched/Autoswitched Mode*
0: Switched Mode
1: Autoswitched Mode

Table 30. Switched and Autoswitched Modes

SWA bit	Commutation Type	MCOMP Register access
0	Switched mode	Read/Write
1	Autoswitched mode	Read only

Bit 1 = **CFF**: *Current Feedback Filter*
0: Current Feedback Filter disabled
1: Current Feedback Filter enabled

Bit 0 = **DCB**: *Data Capture bit*
0: Use MZPRV (Z_N-1) for multiplication
1: Use MZREG (Z_N) for multiplication

Table 31. Multiplier Result

DCB bit	Commutation Delay
0	MCOMP = MWGHT x MZPRV / 32
1	MCOMP = MWGHT x MZREG / 32

MOTOR CONTROLLER (Cont'd)

CONTROL REGISTER B (MCRB)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
VR1	VR0	CPB*	HDM*	SDM*	OS2*	OS1	OS0

Bit 7:6 = **VR[1:0]**: *BEMF/demagnetization Reference threshold*

These bits select the V_{REF} value as shown in the following table.

VR1	VR0	V_{REF} Voltage threshold
0	0	0.2V
0	1	0.6V
1	0	1.2V
1	1	2.5V

Bit 5 = **CPB***: *Compare Bit for Zero-crossing detection.*
0: Zero crossing detection on falling edge
1: Zero crossing detection on rising edge

Bit 4 = **HDM***: *Hardware Demagnetization event Mask bit*
0: Hardware Demagnetization disabled
1: Hardware Demagnetization enabled

Bit 3 = **SDM***: *Software Demagnetization event Mask bit*
0: Software Demagnetization disabled
1: Software Demagnetization enabled

Bit 2:0 = **OS2*,OS[1:0]**: *Operating output mode Selection bits*

Refer to the Step behaviour diagrams (Figure 40, Figure 41, Figure 42) and Table 32.

These bits are used to configure the various PWM output configurations.

Note: The OS2 bit is the only one with a preload register.

Table 32. Step Behaviour Summary

Mode		OS2 bit	PWM after C and before D	OS [1:0] bits	PWM after D and before C
Voltage mode(V0C1=0)	Sensorless (SR=0)	0	Not Alternate	00	On even channels
				01	On odd channels
				10	Alternate odd/even
				11	All active channels
		1	Alternate	00	On even channels
				01	On odd channels
				10	Alternate odd/even
				11	All active channels
	Sensor (SR=1)	x	Unused	00	On even channels
				01	On odd channels
				10	Alternate odd/even
				11	All active channels
Current mode (V0C1=1)	Sensorless (SR=0)	0	On even Channels	00	On even channels
				01	On odd channels
				10	Alternate odd/even
				11	All active channels
		1	On odd channels	00	On even channels
				01	On odd channels
				10	Alternate odd/even
				11	All active channels
	Sensor (SR=1)	x	Unused	00	On even channels
				01	On odd channels
				10	Alternate odd/even
				11	All active channels

Note: For more details, see Step behaviour diagrams (Figure 40, Figure 41, and Figure 42).

* Preload bits, new value taken into account at next C event.

MOTOR CONTROLLER (Cont'd)

PHASE STATE REGISTER (MPHST)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
IS1*	IS0*	OO5*	OO4*	OO3*	OO2*	OO1*	OO0*

Bit 7:6 = **IS[1:0]***: *Input Selection bits*

These bits select the input to connect to comparator as shown in the following table:

Table 33. Input Channel Selection

IS1	IS0	Channel selected
0	0	MCIA
0	1	MCIB
1	0	MCIC
1	1	Not Used

Bit 5:0 =**OO[5:0]***: Channel On/Off bits

These bits are used to switch channels on/off at the next C event if the DAC bit =0 or directly if DAC=1
0: Channel Off, the relevant switch is OFF, no PWM possible
1: Channel On the relevant switch is ON, PWM is possible.

Table 34. OO[5:0] Bit Meaning

OO[5:0]	Output Channel State
0	Inactive
1	Active

* Preload bits, new value taken into account at next C event.

PARITY REGISTER (MPAR)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
ZVD	REO	OE5	OE4	OE3	OE2	OE1	OE0

Bit 7 = **ZVD**: *Z vs D edge polarity.*
0: Zero-crossing and End of Demagnetisation have opposite edges
1: Zero-crossing and End of Demagnetisation have same edge

Bit 6 = **REO**: *Read on Even or Odd channel bit*
0: Read the BEMF signal during the off time on even channels
1: Read on odd channels

Bit 5:0 = **OE[5:0]**: *Output Parity Mode.*
0: Output channel is Even
1: Output channel Odd

POLARITY REGISTER (MPOL)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
OT1	OT0	OP5	OP4	OP3	OP2	OP1	OP0

Bit 7:6 = **OT[1:0]**: *Off Time selection.*
These bits are used to select the off time in sensorless mode as shown in the following table.

Table 35. Off-Time bit Meaning

OT1	OT0	Off-Time Sensorless Mode (SR=0)	Off-Time Sensor Mode (SR=1)
0	0	5 µs	1.25 µs
0	1	10 µs	
1	0	15 µs	
1	1	30 µs	

Bit 5:0 = **OP[5:0]**: *Output channel polarity.*
These bits are used together with the OO**[5:0]** bits in the MPHST register to control the output channels.
0: Output channel is Active Low
1: Output channel is Active High.

Table 36. Output Channel State Control

OP[5:0] bit	OO[5:0] bit	MCO[5:0] pin
0	0	1 (Off)
0	1	0 (PWM possible)
1	0	0 (Off)
1	1	1 (PWM possible)

MOTOR CONTROLLER (Cont'd)

Note: The CPB, HDM, SDM, OS2 bits in the MCRB and the bits OE[5:0] are marked with *. It means that these bits are taken into account at the following commutation event (in normal mode) or when a value is written in the MPHST register when in direct access mode. For more details, refer to the description of the DAC bit in the MCRA register. The use of a Preload register allows all the registers to be updated at the same time.

Warning: Access to Preload registers

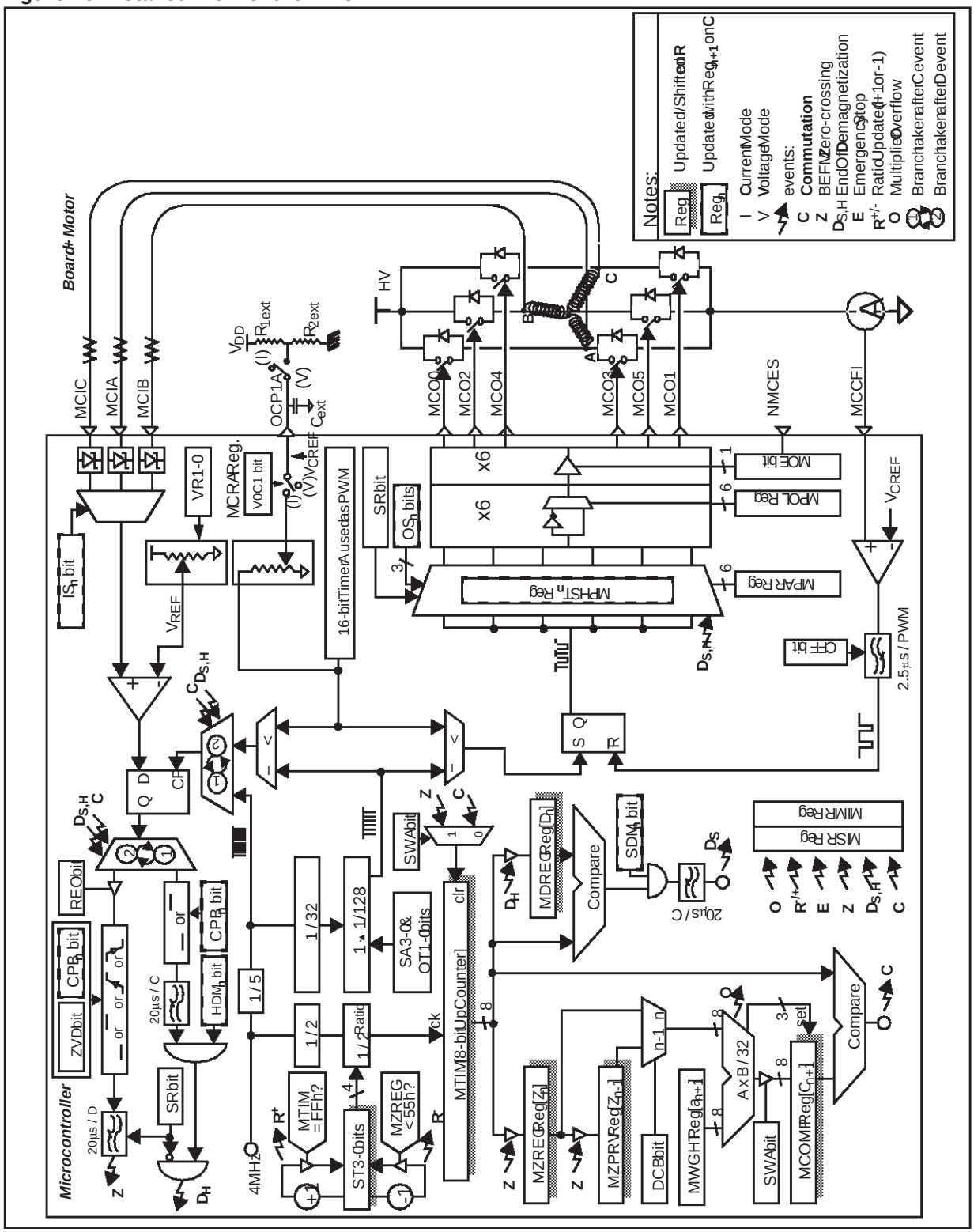
Special care has to be taken with Preload registers, especially when using the ST7 BSET and BRES instructions on MTC registers.

For instance, while writing to the MPHST register, you will write the value in the preload register. However, while reading at the same address, you will get the current value in the register and not the value of the preload register.

All preload registers are loaded in the real registers at the same time. In normal mode this is done automatically when a C event occurs, however in direct access mode (DAC bit=1) the preload registers are loaded as soon as a value is written in the MPHST register.

MOTOR CONTROLLER (Cont'd)

Figure 43. Detailed view of the MTC



MOTOR CONTROLLER (Cont'd)

Table 37. MTC Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
0060h	**MTIM**	T7	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	0	0	0	0	0	0	0
0061h	**MZPRV**	ZP7	ZP6	ZP5	ZP4	ZP3	ZP2	ZP1	ZP0
	Reset Value	0	0	0	0	0	0	0	0
0062h	**MZREG**	ZC7	ZC6	ZC5	ZC4	ZC3	ZC2	ZC1	ZC0
	Reset Value	0	0	0	0	0	0	0	0
0063h	**MCOMP**	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
	Reset Value	0	0	0	0	0	0	0	0
0064h	**MDREG**	DN7	DN6	DN5	DN4	DN3	DN2	DN1	DN0
	Reset Value	0	0	0	0	0	0	0	0
0065h	**MWGHT**	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
	Reset Value	0	0	0	0	0	0	0	0
0066h	**MPRSR**	SA3	SA2	SA1	SA0	ST3	ST2	ST1	ST0
	Reset Value	0	0	0	0	0	0	0	0
0067h	**MIMR**	HST	CL	RIM	OIM	EIM	ZIM	DIM	CIM
	Reset Value	0	0	0	0	0	0	0	0
0068h	**MISR**		RPI	RMI	OI	EI	ZI	DI	CI
	Reset Value	0	0	0	0	0	0	0	0
0069h	**MCRA**	MOE	RST	SR	DAC	V0C1	SWA	CFF	DCB
	Reset Value	0	0	0	0	0	0	0	0
006Ah	**MCRB**	VR1	VR0	CPB	HDM	SDM	OS2	OS1	OS0
	Reset Value	0	0	0	0	0	0	0	0
006Bh	**MPHST**	IS1	IS0	OO5	OO4	OO3	OO2	OO1	OO0
	Reset Value	0	0	0	0	0	0	0	
006Ch	**MPAR**	ZVD	REO	OE5	OE4	OE3	OE2	OE1	OE0
	Reset Value	0	0	0	0	0	0	0	0
006Dh	**MPOL**	OT1	OT0	OP5	OP4	OP3	OP2	OP1	OP0
	Reset Value	0	0	0	0	0	0	0	0

8.2 WATCHDOG TIMER (WDG)

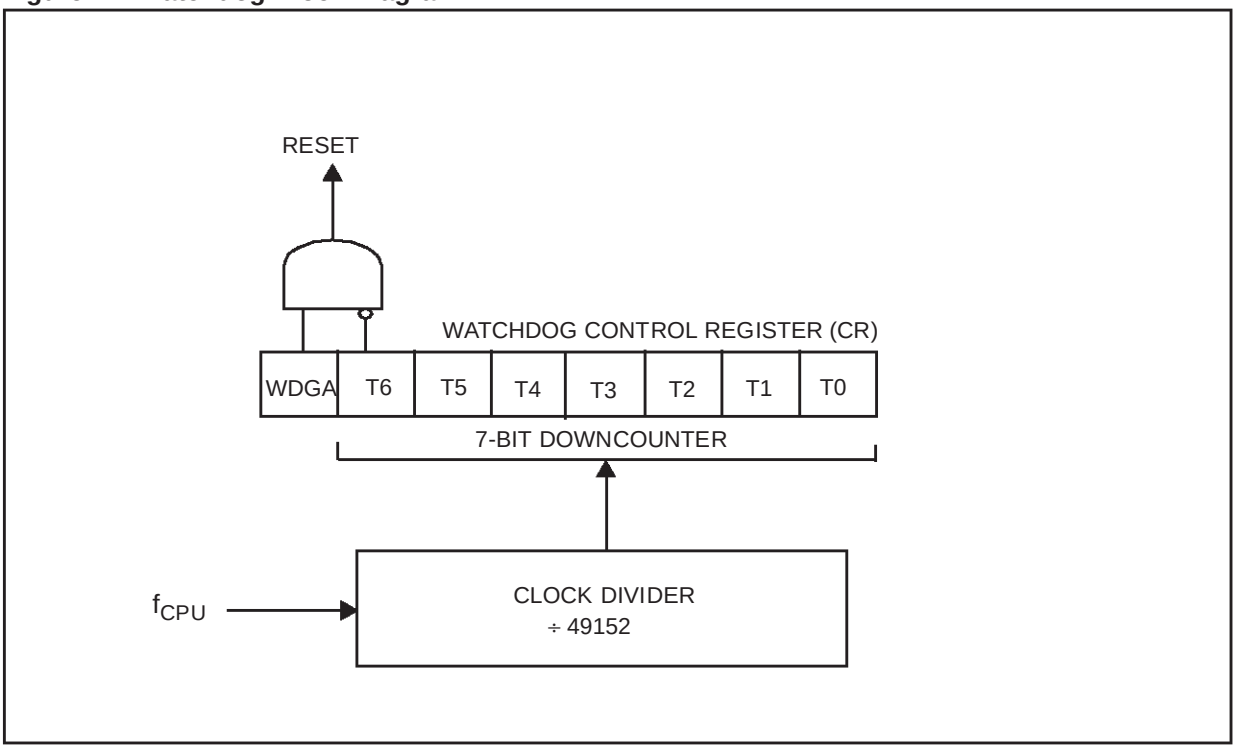
8.2.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

8.2.2 Main Features

- Programmable timer (64 increments of 49,152 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) after a HALT instruction or when the T6 bit reaches zero
- Watchdog Reset indicated by status flag

Figure 44. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)

8.2.3 Functional Description

The counter value stored in the CR register (bits T6:T0), is decremented every 49,152 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T6:T0) rolls over from 40h to 3Fh (T6 become cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 38 . Watchdog Timing (fCPU = 8 MHz)):

– The WDGA bit is set (watchdog enabled)

– The T6 bit is set to prevent generating an immediate reset

– The T5:T0 bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Table 38. Watchdog Timing (f_{CPU} = 8 MHz)

	CR Register initial value	WDG timeout period (ms)
Max	FFh	393.216
Min	C0h	6.144

Notes: Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

8.2.4 Low Power Modes

Mode	Description
WAIT	No effect on Watchdog.
HALT	Immediate reset generation as soon as the HALT instruction is executed if the Watchdog is activated (WDGA bit is set).

8.2.5 Interrupts

None.

8.2.6 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7= **WDGA** *Activation bit.*
This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.
0: Watchdog disabled
1: Watchdog enabled

Bit 6:0 = **T[6:0]** *7-bit timer (MSB to LSB).*
These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared) if WDGA=1.

STATUS REGISTER (SR)

Read/Write

Reset Value*: xxxx xxxx0

7							0
-	-	-	-	-	-	-	WDOGF

Bit 0 = **WDOGF** *Watchdog flag.*
This bit is set by a watchdog reset and cleared by software or a power on/off reset. This bit is useful for distinguishing power/on off or external reset and watchdog reset.
0: No Watchdog reset occurred
1: Watchdog reset occurred

* Only by software and power on/off reset

WATCHDOG TIMER (Cond't)

Table 39. Watchdog Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0024h	**WDGCR**	WDGA	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	1	1	1	1	1	1	1
0025h	**WDGSR**	-	-	-	-	-	-	-	WDOGF
	Reset Value	0	0	0	0	0	0	0	0

8.3 16-BIT TIMER

8.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including measuring the pulse lengths of up to two input signals (*input capture*) or generating up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

Some ST7 devices have two on-chip 16-bit timers. They are completely independent, and do not share any resources. They are synchronized after a MCU reset as long as the timer clock frequencies are not modified.

This description covers one or two 16-bit timers. In ST7 devices with two timers, register names are prefixed with TA (Timer A) or TB (Timer B).

8.3.2 Main Features

- Programmable prescaler: f_{CPU} divided by 2, 4 or 8.
- Overflow status flag and maskable interrupt
- External clock input (must be at least 4 times slower than the CPU clock speed) with the choice of active edge
- Output compare functions with:
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Input capture functions with:
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse Width Modulation mode (PWM)
- One Pulse mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)*

The Block Diagram is shown in Figure 1.

***Note:** Some timer pins may not be available (not bonded) in some ST7 devices. Refer to the device pin out description.
When reading an input signal on a non-bonded pin, the value will always be '1'.

8.3.3 Functional Description

8.3.3.1 Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high & low.

Counter Register (CR):

- Counter High Register (CHR) is the most significant byte (MS Byte).
- Counter Low Register (CLR) is the least significant byte (LS Byte).

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte).
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte).

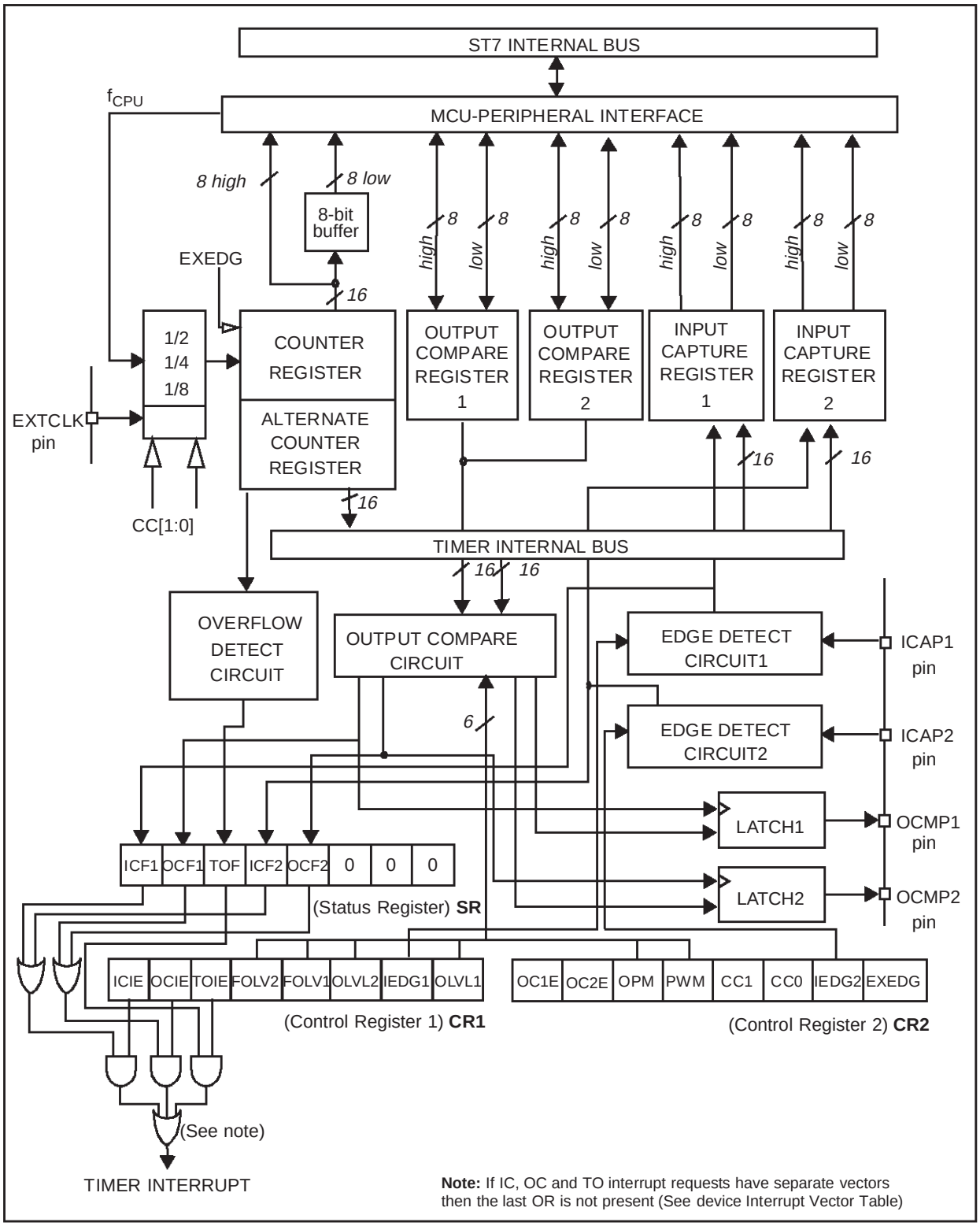
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register (SR). (See note at the end of paragraph titled 16-bit read sequence).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.
Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in Table 1. The value in the counter register repeats every 131.072, 262.144 or 524.288 CPU clock cycles depending on the CC[1:0] bits.
The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

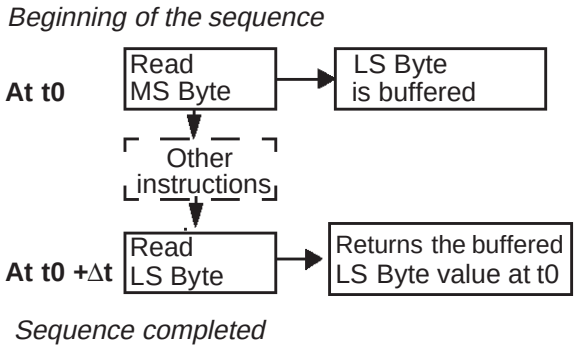
16-BIT TIMER (Cont'd)

Figure 45. Timer Block Diagram



16-BIT TIMER (Cont'd)

16-bit Read Sequence: (from either the Counter Register or the Alternate Counter Register).



The user must read the MS Byte first, then the LS Byte value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MS Byte several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LS Byte of the count value at the time of the read.

Whatever the timer mode used (input capture, output compare, One Pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h then:

– The TOF bit of the SR register is set.

– A timer interrupt is generated if:

  – TOIE bit of the CR1 register is set and

  – I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Note: The TOF bit is not cleared by accessing the ACLR register. The advantage of accessing the ACLR register rather than the CLR register is that it allows simultaneous use of the overflow function and reading the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

In HALT mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).

8.3.3.2 External Clock

The external clock (where available) is selected if CC0=1 and CC1=1 in the CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronised with the falling edge of the internal CPU clock.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

16-BIT TIMER (Cont'd)

Figure 46. Counter Timing Diagram, internal clock divided by 2

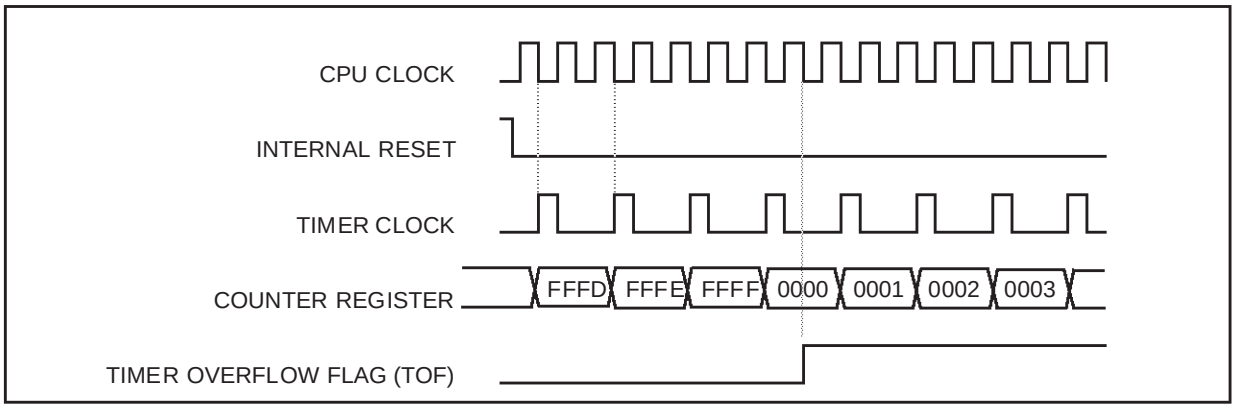


Figure 47. Counter Timing Diagram, internal clock divided by 4

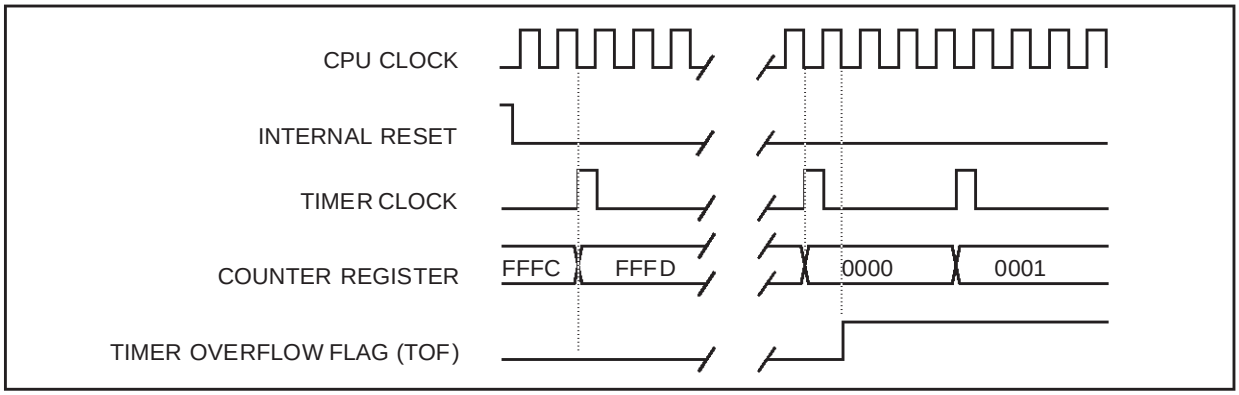
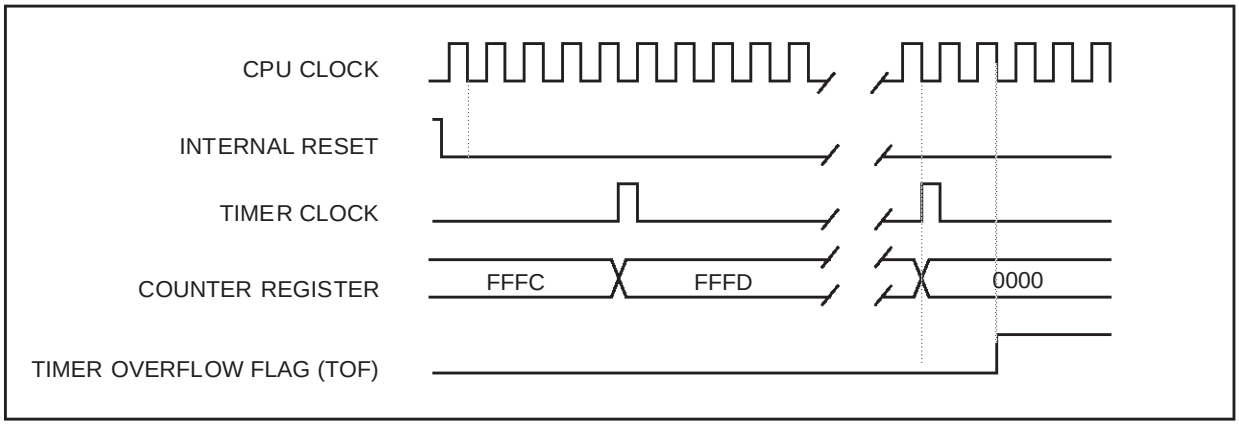


Figure 48. Counter Timing Diagram, internal clock divided by 8



Note: The MCU is in reset state when the internal reset signal is high. When it is low, the MCU is running.

16-BIT TIMER (Cont'd)

8.3.3.3 Input Capture

In this section, the index, *i*, may be 1 or 2 because there are 2 input capture functions in the 16-bit timer.

The two input capture 16-bit registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition is detected by the ICAP*i* pin (see figure 5).

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR

The IC*i*R register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of Control Registers (CR*i*).

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure:

To use the input capture function, select the following in the CR2 register:

– Select the timer clock (CC[1:0]) (see Table 1).

– Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as a floating input).

And select the following in the CR1 register:

– Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as a floating input).

When an input capture occurs:

– The ICF*i* bit is set.

– The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see Figure 6).

– A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request (i.e. clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

Notes:

1. After reading the IC*i*HR register, the transfer of input capture data is inhibited and ICF*i* will never be set until the IC*i*LR register is also read.
2. The IC*i*R register contains the free running counter value which corresponds to the most recent input capture.
3. The 2 input capture functions can be used together even if the timer also uses the 2 output compare functions.
4. In One Pulse mode and PWM mode only the input capture 2 function can be used.
5. The alternate inputs (ICAP1 & ICAP2) are always directly connected to the timer. So any transitions on these pins activate the input capture function.
   Moreover if one of the ICAP*i* pin is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.
   This can be avoided if the input capture function *i* is disabled by reading the IC*i*HR (see note 1).
6. The TOF bit can be used with an interrupt in order to measure events that exceed the timer range (FFFFh).

16-BIT TIMER (Cont'd)

Figure 49. Input Capture Block Diagram

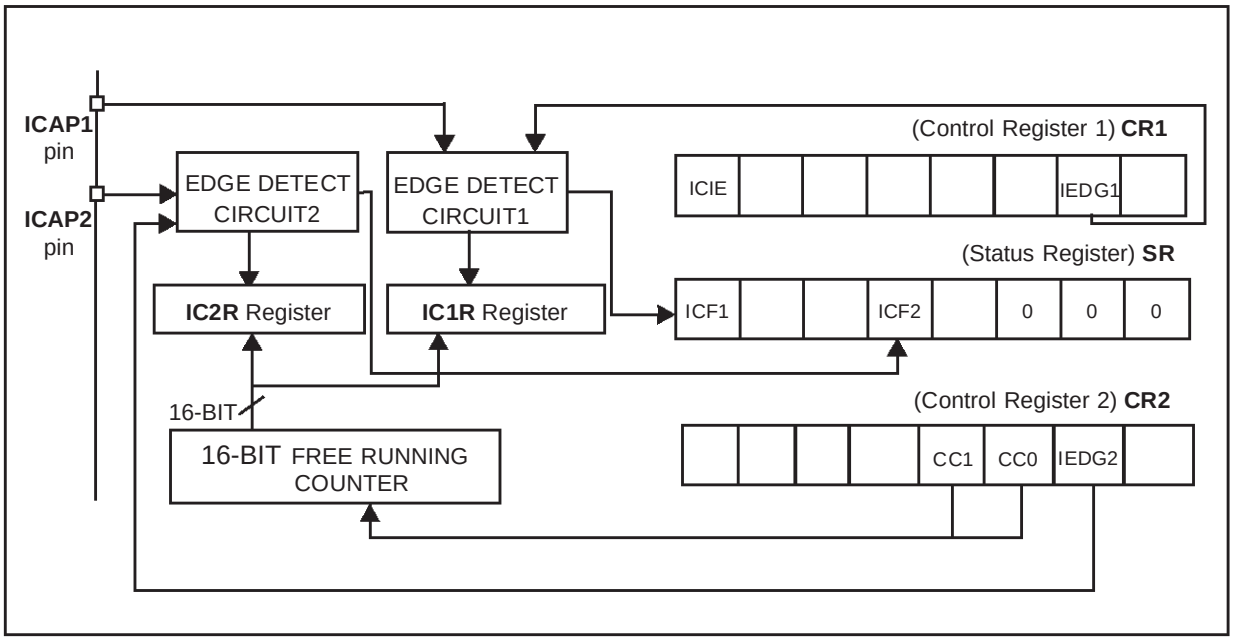
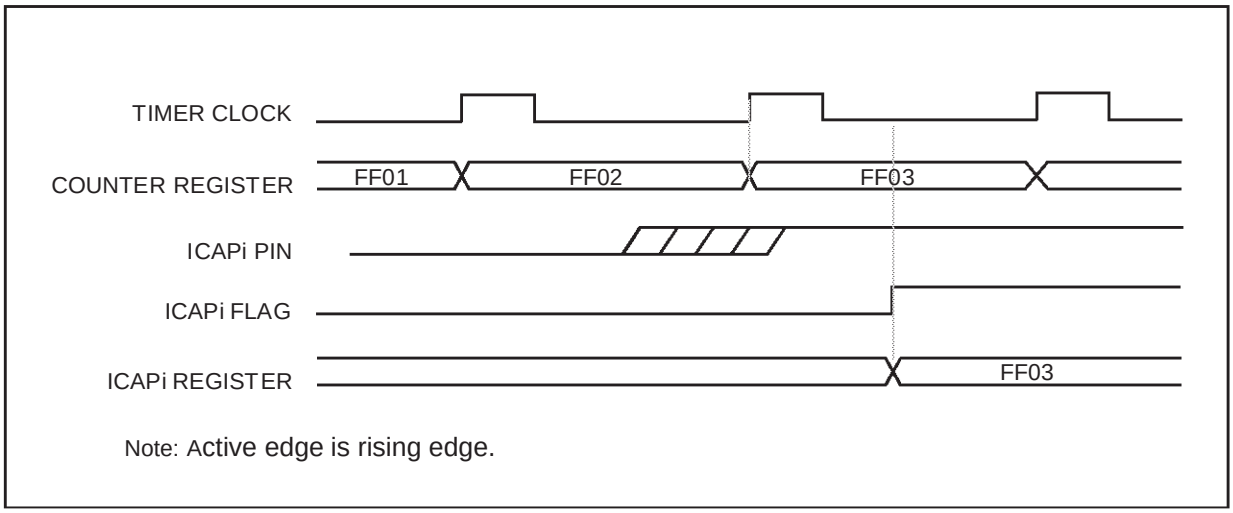


Figure 50. Input Capture Timing Diagram



16-BIT TIMER (Cont'd)

8.3.3.4 Output Compare

In this section, the index, *i*, may be 1 or 2 because there are 2 output compare functions in the 16-bit timer.

This function can be used to control an output waveform or indicate when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OCIE bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OC1R) and Output Compare Register 2 (OC2R) contain the value to be compared to the counter register each timer clock cycle.

	MS Byte	LS Byte
OC*i*R	OC*i*HR	OC*i*LR

These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

Timing resolution is one count of the free running counter: ($f_{CPU/CC[1:0]}$).

Procedure:

To use the output compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* signal.
- Select the timer clock (CC[1:0]) (see Table 1).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When a match is found between OCRi register and CR register:

- OCF*i* bit is set.
- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset).
- A timer interrupt is generated if the OCIE bit is set in the CR2 register and the I bit is cleared in the CC register (CC).

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{ OC}i\text{R} = \frac{\Delta t * f_{CPU}}{\text{PRESC}}$$

Where:

Δt = Output compare period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see Table 1)

If the timer clock is an external clock, the formula is:

$$\Delta \text{ OC}i\text{R} = \Delta t * f_{EXT}$$

Where:

Δt = Output compare period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (i.e. clearing the OCF*i* bit) is done by:

1. Reading the SR register while the OCF*i* bit is set.
2. An access (read or write) to the OC*i*LR register.

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

16-BIT TIMER (Cont'd)

Notes:

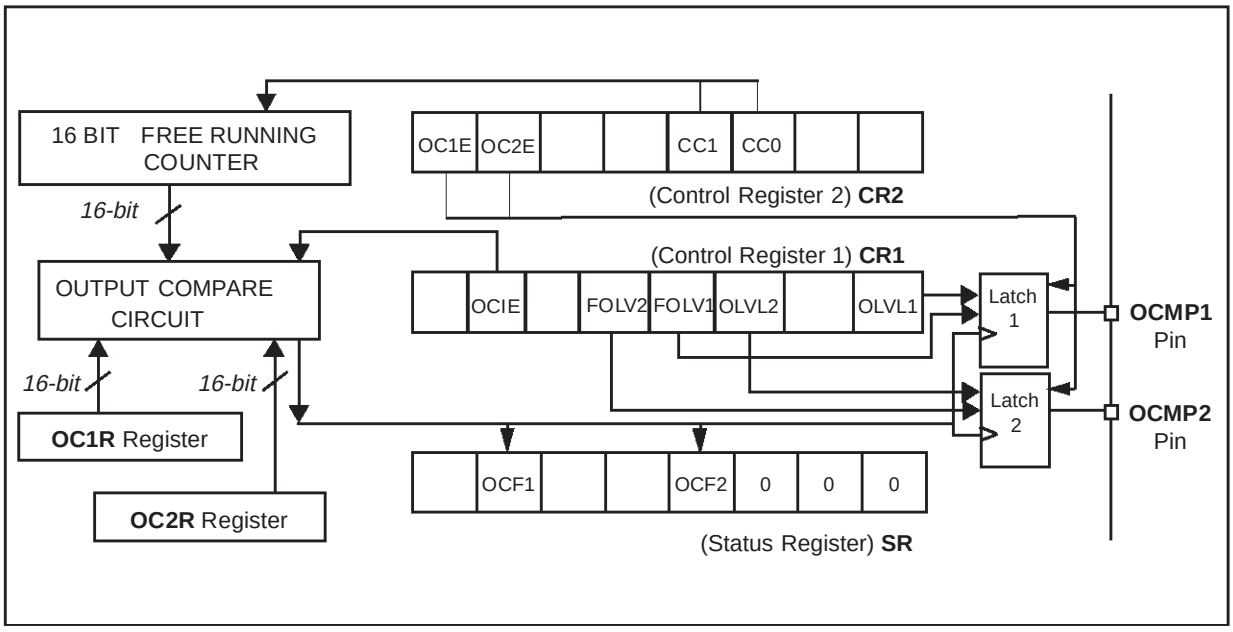
1. After a processor write cycle to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
2. If the OC*i*E bit is not set, the OCMP*i* pin is a general I/O port and the OLVL*i* bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.
3. When the timer clock is $f_{CPU}/2$, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value (see Figure 8). This behaviour is the same in OPM or PWM mode. When the timer clock is $f_{CPU}/4$, $f_{CPU}/8$ or in external clock mode, OCF*i* and OCMP*i* are set while the counter value equals the OC*i*R register value plus 1 (see Figure 9).
4. The output compare functions can be used both for generating external events on the OCMP*i* pins even if the input capture mode is also used.
5. The value in the 16-bit OC*i*R register and the OLV*i* bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

Forced Compare Output capability

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit=1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

FOLVL*i* bits have no effect in either One-Pulse mode or PWM mode.

Figure 51. Output Compare Block Diagram



16-BIT TIMER (Cont'd)

Figure 52. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/2$

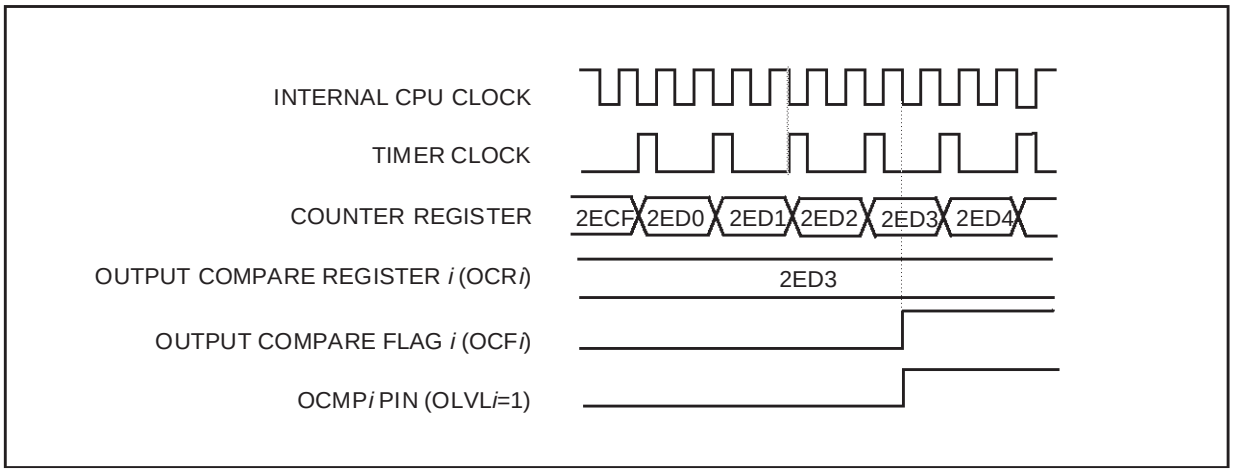
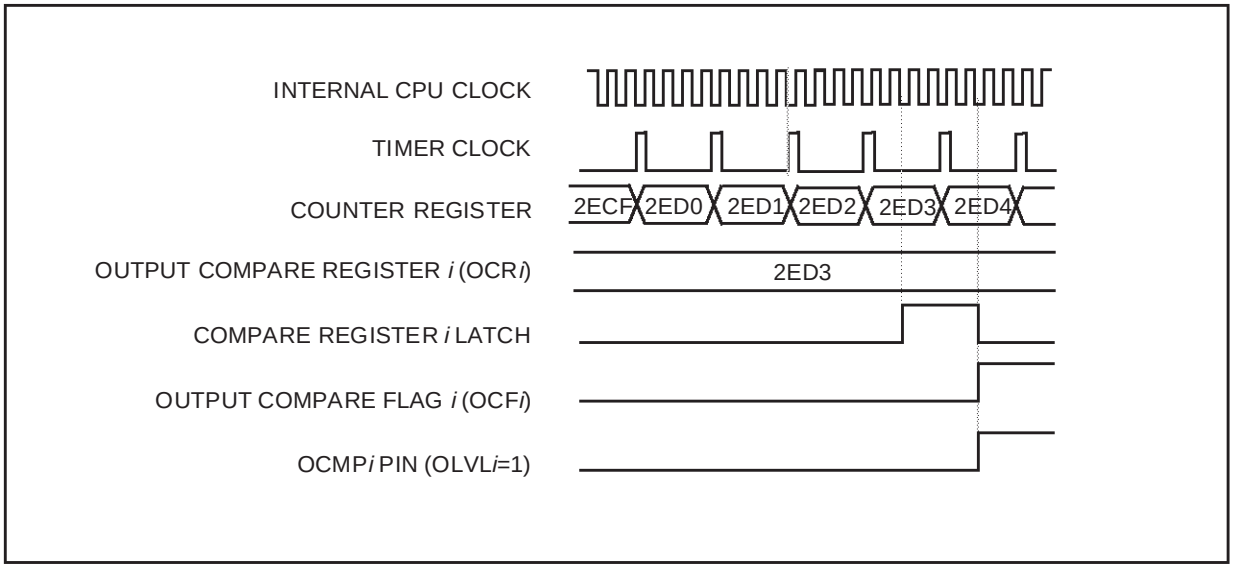


Figure 53. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/4$



16-BIT TIMER (Cont'd)

8.3.3.5 One Pulse Mode

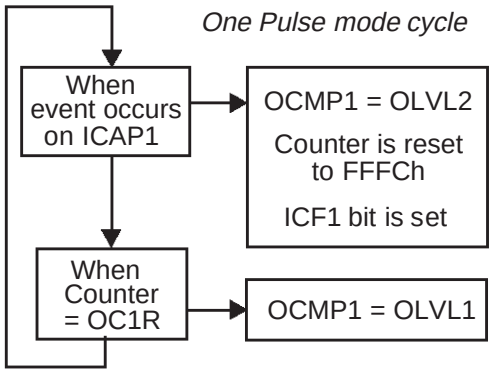
One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

The One Pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure:

To use One Pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in the opposite column).
2. Select the following in the CR1 register:
   – Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
   – Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
   – Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
   – Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
   – Set the OPM bit.
   – Select the timer clock CC[1:0] (see Table 1).



Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and the OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (i.e. clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

The OC1R register value required for a specific timing application can be calculated using the following formula:

$$OCiR\ Value = \frac{t * f_{CPU}}{PRESC} - 5$$

Where:

t = Pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits, see Table 1)

If the timer clock is an external clock the formula is:

$$OCiR = t * f_{EXT} - 5$$

Where:

t = Pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

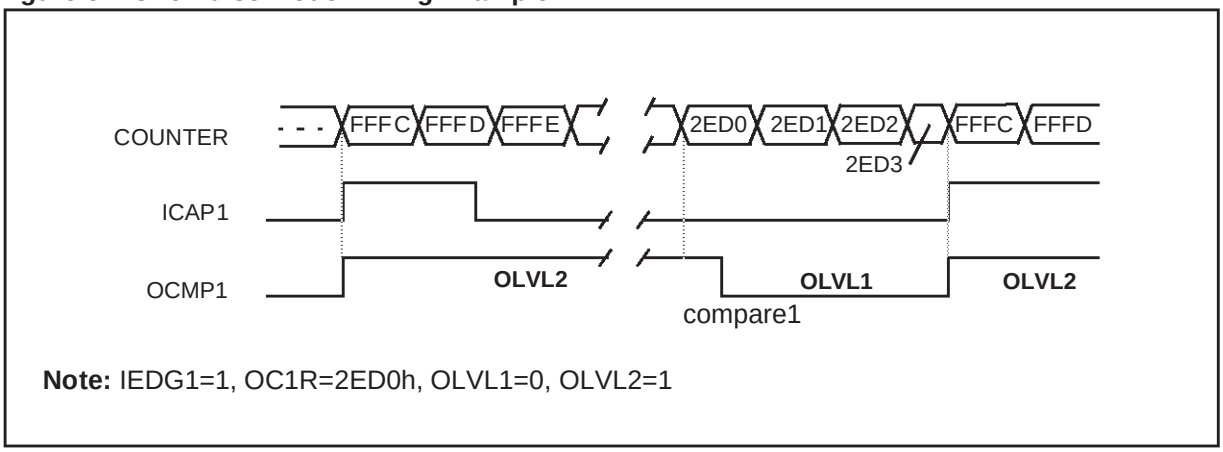
When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see Figure 10).

Notes:

1. The OCF1 bit cannot be set by hardware in One Pulse mode but the OCF2 bit can generate an Output Compare interrupt.
2. When the Pulse Width Modulation (PWM) and One Pulse mode (OPM) bits are both set, the PWM mode is the only active one.
3. If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.
4. The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.
5. When One Pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate that a period of time has elapsed but cannot generate an output waveform because the OLVL2 level is dedicated to One Pulse mode.

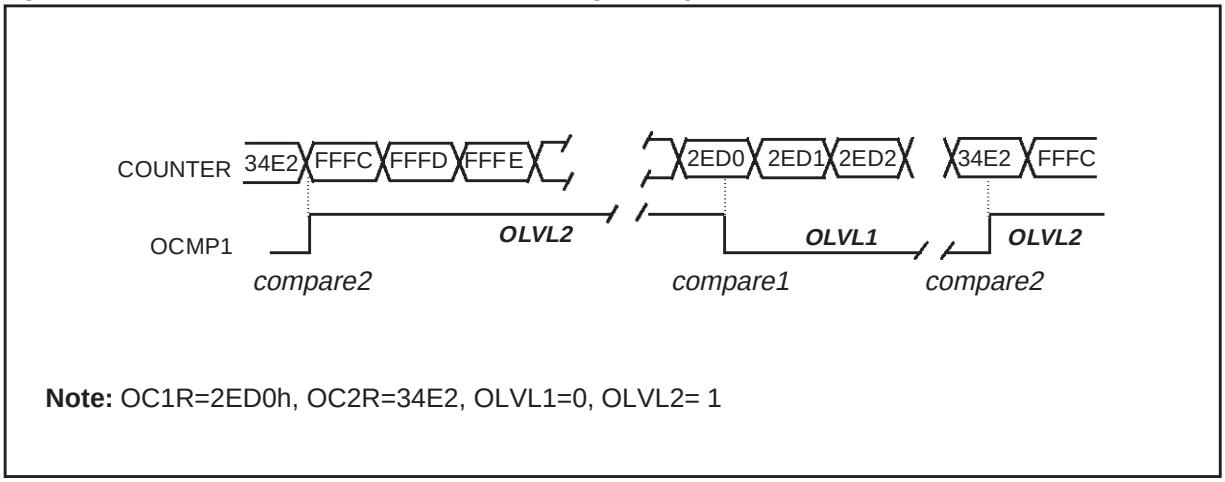
16-BIT TIMER (Cont'd)

Figure 54. One Pulse Mode Timing Example



Note: IEDG1=1, OC1R=2ED0h, OLVL1=0, OLVL2=1

Figure 55. Pulse Width Modulation Mode Timing Example



Note: OC1R=2ED0h, OC2R=34E2, OLVL1=0, OLVL2= 1

16-BIT TIMER (Cont'd)

8.3.3.6 Pulse Width Modulation Mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

The Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so these functions cannot be used when the PWM mode is activated.

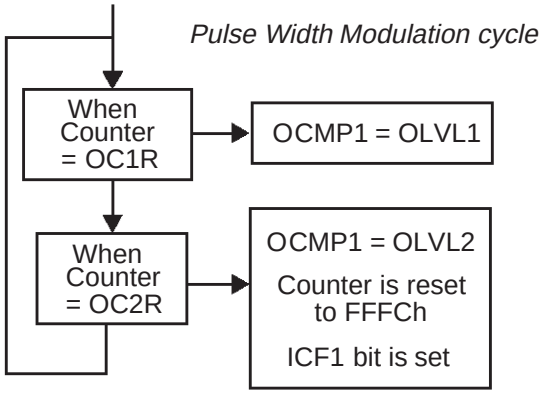
Procedure

To use Pulse Width Modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the formula in the opposite column.
2. Load the OC1R register with the value corresponding to the period of the pulse if OLVL1=0 and OLVL2=1, using the formula in the opposite column.
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see Table 1).

If OLVL1=1 and OLVL2=0, the length of the positive pulse is the difference between the OC2R and OC1R registers.

If OLVL1=OLVL2 a continuous signal will be seen on the OCMP1 pin.



The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$OCiR\ Value = \frac{t * f_{CPU}}{PRESC} - 5$$

Where:

t = Signal or pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see Table 1)

If the timer clock is an external clock the formula is:

$$OCiR = t * f_{EXT} - 5$$

Where:

t = Signal or pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 11)

Notes:

1. After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
2. The OCF1 and OCF2 bits cannot be set by hardware in PWM mode, therefore the Output Compare interrupt is inhibited.
3. The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.
4. In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected from the timer. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset after each period and ICF1 can also generate an interrupt if ICIE is set.
5. When the Pulse Width Modulation (PWM) and One Pulse mode (OPM) bits are both set, the PWM mode is the only active one.

16-BIT TIMER (Cont'd)

8.3.4 Low Power Modes

Mode	Description
WAIT	No effect on 16-bit Timer. Timer interrupts cause the device to exit from WAIT mode.
HALT	16-bit Timer registers are frozen. In HALT mode, the counter stops counting until Halt mode is exited. Counting resumes from the previous count when the MCU is woken up by an interrupt with "exit from HALT mode" capability or from the counter reset value when the MCU is woken up by a RESET. If an input capture event occurs on the ICAP*i* pin, the input capture detection circuitry is armed. Consequently, when the MCU is woken up by an interrupt with "exit from HALT mode" capability, the ICF*i* bit is set, and the counter value present when exiting from HALT mode is captured into the IC*i*R register.

8.3.5 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Input Capture 1 event/Counter reset in PWM mode	ICF1	ICIE	Yes	No
Input Capture 2 event	ICF2		Yes	No
Output Compare 1 event (not available in PWM mode)	OCF1	OCIE	Yes	No
Output Compare 2 event (not available in PWM mode)	OCF2		Yes	No
Timer Overflow event	TOF	TOIE	Yes	No

Note: The 16-bit Timer interrupt events are connected to the same interrupt vector (see Interrupts chapter). These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

8.3.6 Summary of Timer modes

MODES	AVAILABLE RESOURCES			
	Input Capture 1	Input Capture 2	Output Compare 1	Output Compare 2
Input Capture (1 and/or 2)	Yes	Yes	Yes	Yes
Output Compare (1 and/or 2)	Yes	Yes	Yes	Yes
One Pulse mode	No	Not Recommended [1)]	No	Partially [2)]
PWM Mode	No	Not Recommended [3)]	No	No

1) See note 4 in Section 0.1.3.5 One Pulse Mode

2) See note 5 in Section 0.1.3.5 One Pulse Mode

3) See note 4 in Section 0.1.3.6 Pulse Width Modulation Mode

16-BIT TIMER (Cont'd)

8.3.7 Register Description

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

CONTROL REGISTER 1 (CR1)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = **ICIE** *Input Capture Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.

Bit 6 = **OCIE** *Output Compare Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.

Bit 5 = **TOIE** *Timer Overflow Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Bit 4 = **FOLV2** *Forced Output Compare 2.*
This bit is set and cleared by software.
0: No effect on the OCMP2 pin.
1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison.

Bit 3 = **FOLV1** *Forced Output Compare 1.*
This bit is set and cleared by software.
0: No effect on the OCMP1 pin.
1: Forces OLVL1 to be copied to the OCMP1 pin, if the OC1E bit is set and even if there is no successful comparison.

Bit 2 = **OLVL2** *Output Level 2.*
This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse mode and Pulse Width Modulation mode.

Bit 1 = **IEDG1** *Input Edge 1.*
This bit determines which type of level transition on the ICAP1 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **OLVL1** *Output Level 1.*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

16-BIT TIMER (Cont'd)

CONTROL REGISTER 2 (CR2)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and one-pulse mode). Whatever the value of the OC1E bit, the internal Output Compare 1 function of the timer remains active.
0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP1 pin alternate function enabled.

Bit 6 = **OC2E** *Output Compare 2 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the internal Output Compare 2 function of the timer remains active.
0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O).
1: OCMP2 pin alternate function enabled.

Bit 5 = **OPM** *One Pulse mode.*
0: One Pulse mode is not active.
1: One Pulse mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

Bit 4 = **PWM** *Pulse Width Modulation.*
0: PWM mode is not active.
1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

Bits 3:2 = **CC[1:0]** *Clock Control.*

The timer clock mode depends on these bits:

Table 40. Clock Control Bits

Timer Clock	CC1	CC0
f_{CPU} / 4	0	0
f_{CPU} / 2	0	1
f_{CPU} / 8	1	0
External Clock (where available)	1	1

Note: If the external clock pin is not available, programming the external clock configuration stops the counter.

Bit 1 = **IEDG2** *Input Edge 2.*
This bit determines which type of level transition on the ICAP2 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = **EXEDG** *External Clock Edge.*
This bit determines which type of level transition on the external clock pin (EXTCLK) will trigger the counter register.
0: A falling edge triggers the counter register.
1: A rising edge triggers the counter register.

16-BIT TIMER (Cont'd)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2	0	0	0

Bit 7 = **ICF1** *Input Capture Flag 1.*
0: No input capture (reset value).
1: An input capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 = **OCF1** *Output Compare Flag 1.*
0: No match (reset value).
1: The content of the free running counter matches the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 = **TOF** *Timer Overflow Flag.*
0: No timer overflow (reset value).
1: The free running counter has rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Note: Reading or writing the ACLR register does not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2.*
0: No input capture (reset value).
1: An input capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 = **OCF2** *Output Compare Flag 2.*
0: No match (reset value).
1: The content of the free running counter matches the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2-0 = Reserved, forced by hardware to 0.

INPUT CAPTURE 1 HIGH REGISTER (IC1HR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

INPUT CAPTURE 1 LOW REGISTER (IC1LR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

OUTPUT COMPARE 1 HIGH REGISTER (OC1HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

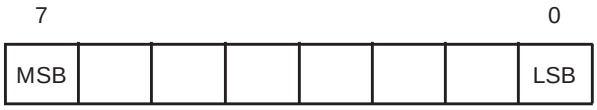
7							0
MSB							LSB

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

COUNTER HIGH REGISTER (CHR)

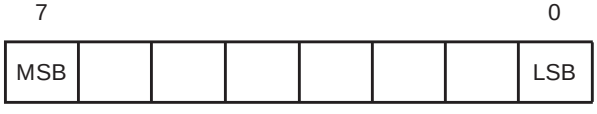
Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

COUNTER LOW REGISTER (CLR)

Read Only
Reset Value: 1111 1100 (FCh)

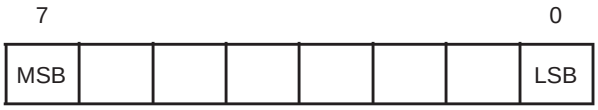
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the SR register clears the TOF bit.



ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

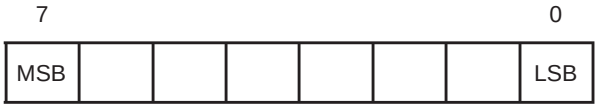
This is an 8-bit register that contains the high part of the counter value.



ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only
Reset Value: 1111 1100 (FCh)

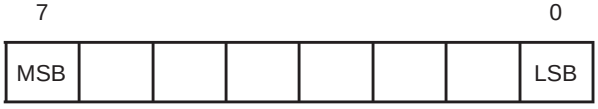
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to SR register does not clear the TOF bit in SR register.



INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

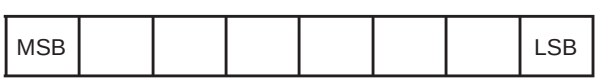
This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).



INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).



16-BIT TIMER (Cont'd)

Table 41. 16-Bit Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Timer A: 32 Timer B: 42	**CR1** Reset Value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
Timer A: 31 Timer B: 41	**CR2** Reset Value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
Timer A: 33 Timer B: 43	**SR** Reset Value	ICF1 0	OCF1 0	TOF 0	ICF2 0	OCF2 0	- 0	- 0	- 0
Timer A: 34 Timer B: 44	**ICHR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 35 Timer B: 45	**ICLR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 36 Timer B: 46	**OCHR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 37 Timer B: 47	**OCLR1** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3E Timer B: 4E	**OCHR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3F Timer B: 4F	**OCLR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 38 Timer B: 48	**CHR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 39 Timer B: 49	**CLR** Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3A Timer B: 4A	**ACHR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 3B Timer B: 4B	**ACLR** Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3C Timer B: 4C	**ICHR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer A: 3D Timer B: 4D	**ICLR2** Reset Value	MSB -	-	-	-	-	-	-	LSB -

8.4 SERIAL PERIPHERAL INTERFACE (SPI)

8.4.1 Introduction

The Serial Peripheral Interface (SPI) allows full-duplex, synchronous, serial communication with external devices. An SPI system may consist of a master and one or more slaves or a system in which devices may be either masters or slaves.

The SPI is normally used for communication between the microcontroller and external peripherals or another microcontroller.

Refer to the Pin Description chapter for the device-specific pin-out.

8.4.2 Main Features

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- Four master mode frequencies
- Maximum slave mode frequency = fCPU/2.
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision flag protection
- Master mode fault protection capability.

8.4.3 General description

The SPI is connected to external devices through 4 alternate pins:

- MISO: Master In Slave Out pin
- MOSI: Master Out Slave In pin
- SCK: Serial Clock pin
- $\overline{SS}$: Slave select pin

A basic example of interconnections between a single master and a single slave is illustrated on Figure 56.

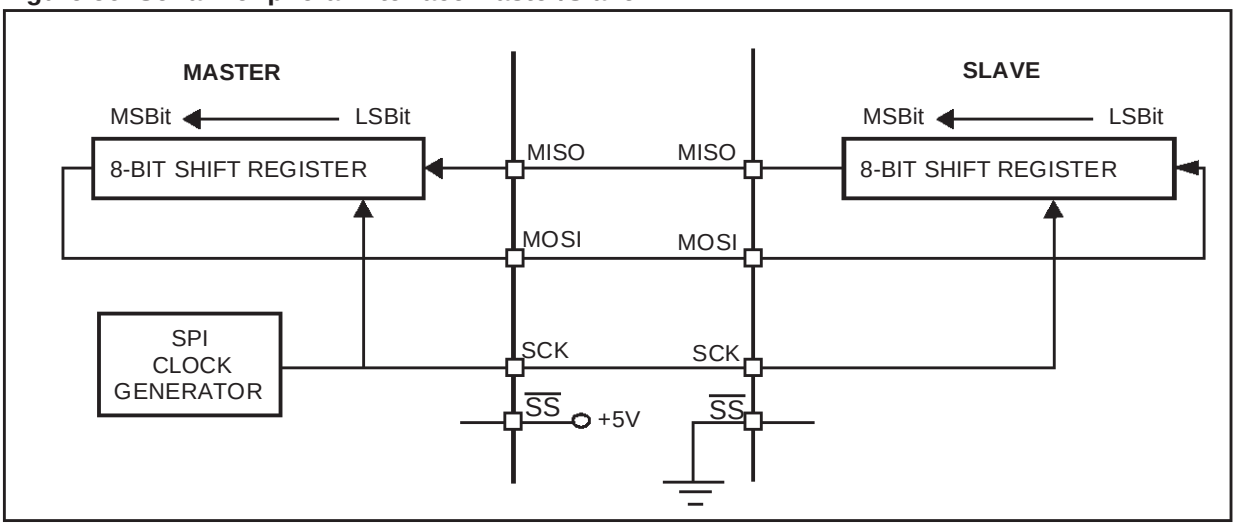
The MOSI pins are connected together as are MISO pins. In this way data is transferred serially between master and slave (most significant bit first).

When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full bits. A status flag is used to indicate that the I/O operation is complete.

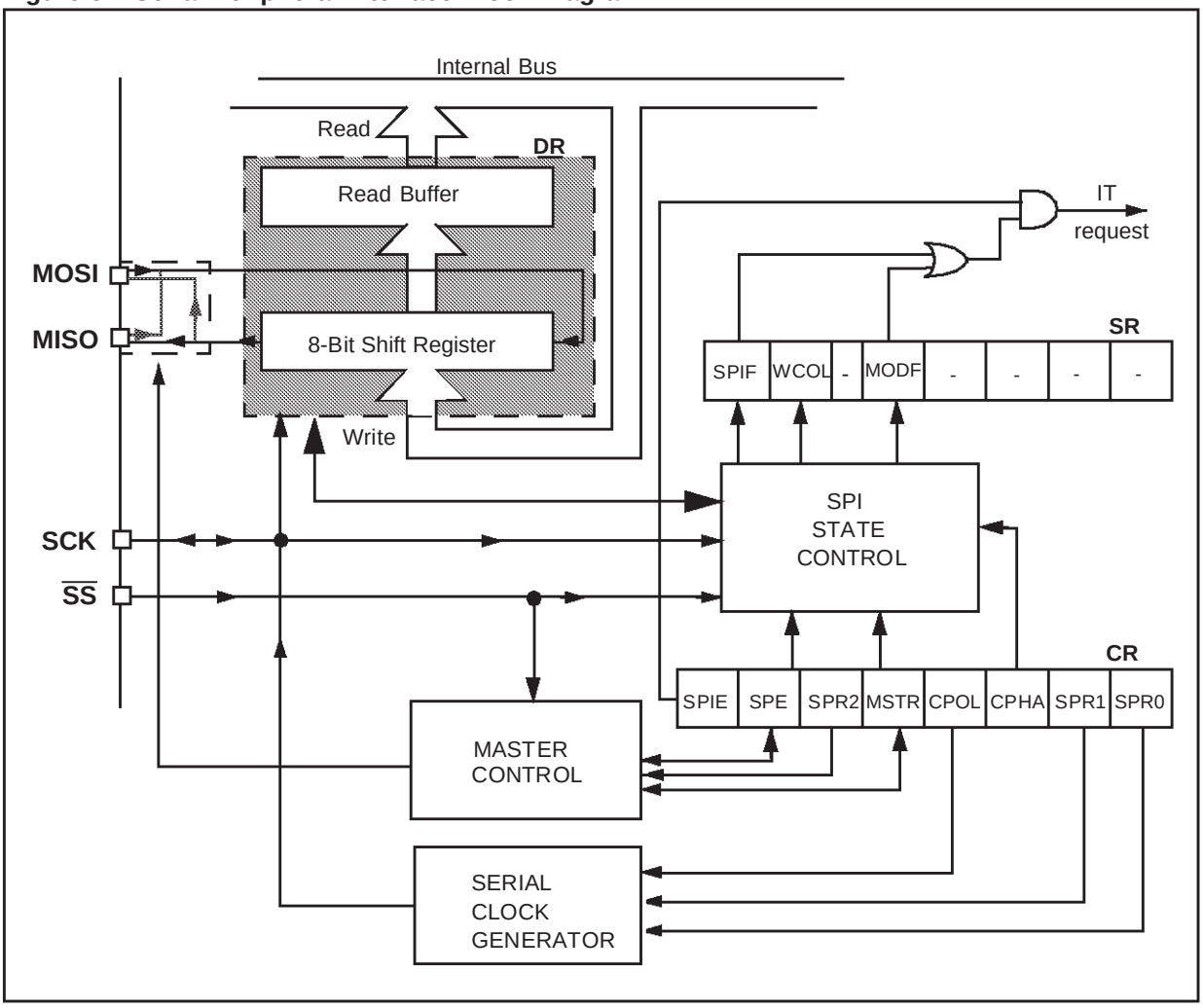
Four possible data/clock timing relationships may be chosen (see Figure 59) but master and slave must be programmed with the same timing mode.

Figure 56. Serial Peripheral Interface Master/Slave



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 57. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.4 Functional Description

Figure 56 shows the serial peripheral interface (SPI) block diagram.

This interface contains 3 dedicated registers:

- A Control Register (CR)
- A Status Register (SR)
- A Data Register (DR)

Refer to the CR, SR and DR registers in Section 8.4.7for the bit definitions.

8.4.4.1 Master Configuration

In a master configuration, the serial clock is generated on the SCK pin.

Procedure

- Select the SPR0 & SPR1 bits to define the serial clock baud rate (see CR register).
- Select the CPOL and CPHA bits to define one of the four relationships between the data transfer and the serial clock (see Figure 59).
- The $\overline{SS}$ pin must be connected to a high level signal during the complete byte transmit sequence.
- The MSTR and SPE bits must be set (they remain set only if the $\overline{SS}$ pin is connected to a high level signal).

In this configuration the MOSI pin is a data output and to the MISO pin is a data input.

Transmit sequence

The transmit sequence begins when a byte is written the DR register.

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt is generated if the SPIE bit is set and the I bit in the CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SR register while the SPIF bit is set
2. A read to the DR register.

Note: While the SPIF bit is set, all writes to the DR register are inhibited until the SR register is read.

SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.4.2 Slave Configuration

In slave configuration, the serial clock is received on the SCK pin from the master device.

The value of the SPR0 & SPR1 bits is not used for the data transfer.

Procedure

- For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 59.
- The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence.
- Clear the MSTR bit and set the SPE bit to assign the pins to alternate function.

In this configuration the MOSI pin is a data input and the MISO pin is a data output.

Transmit Sequence

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt is generated if SPIE bit is set and I bit in CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SR register while the SPIF bit is set.
2. A read to the DR register.

Notes: While the SPIF bit is set, all writes to the DR register are inhibited until the SR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an overrun condition (see Section 8.4.4.6).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the DR register between each data byte transfer to avoid a write collision (see Section 8.4.4.4).

SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.4.3 Data Transfer Format

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The serial clock is used to synchronize the data transfer during a sequence of eight clock pulses.

The $\overline{SS}$ pin allows individual selection of a slave device; the other slave devices that are not selected do not interfere with the SPI transfer.

Clock Phase and Clock Polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits.

The CPOL (clock polarity) bit controls the steady state value of the clock when no data is being transferred. This bit affects both master and slave modes.

The combination between the CPOL and CPHA (clock phase) bits selects the data capture clock edge.

Figure 59, shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin, the MOSI pin are directly connected between the master and the slave device.

The $\overline{SS}$ pin is the slave device select input and can be driven by the master device.

The master device applies data to its MOSI pin-clock edge before the capture clock edge.

CPHA bit is set

The second edge on the SCK pin (falling edge if the CPOL bit is reset, rising edge if the CPOL bit is set) is the MSBit capture strobe. Data is latched on the occurrence of the second clock transition.

No write collision should occur even if the $\overline{SS}$ pin stays low during a transfer of several bytes (see Figure 58).

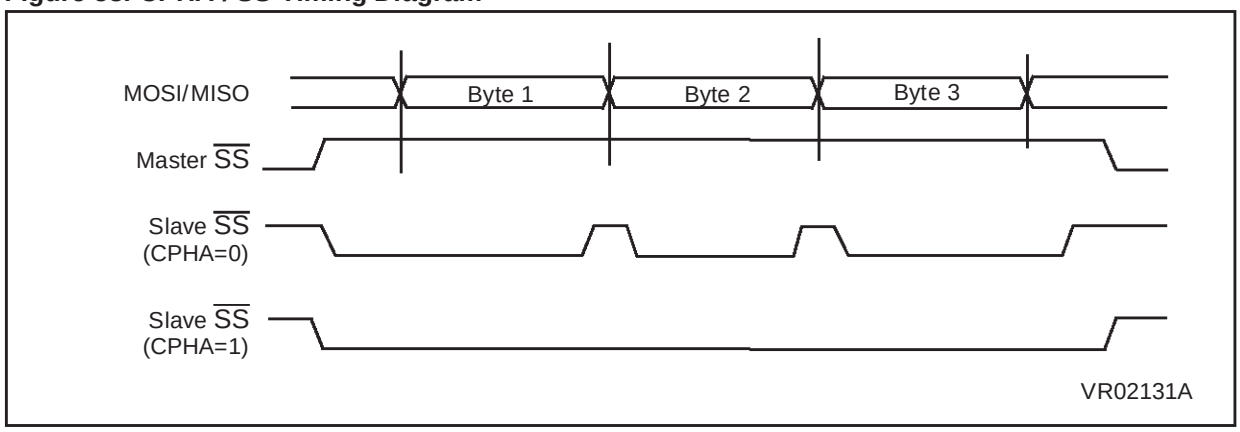
CPHA bit is reset

The first edge on the SCK pin (falling edge if CPOL bit is set, rising edge if CPOL bit is reset) is the MSBit capture strobe. Data is latched on the occurrence of the first clock transition.

The $\overline{SS}$ pin must be toggled high and low between each byte transmitted (see Figure 58).

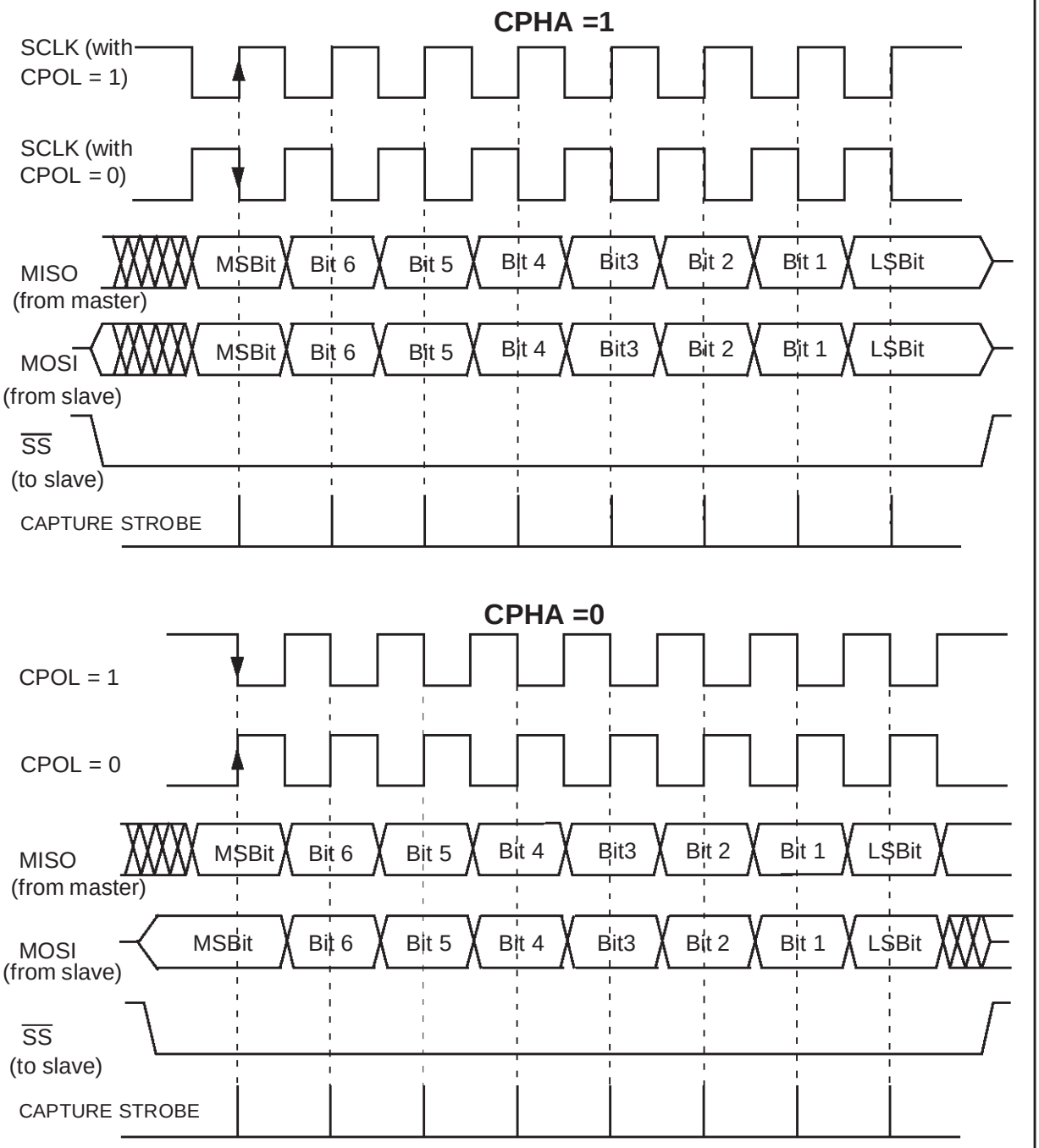
To protect the transmission from a write collision a low value on the $\overline{SS}$ pin of a slave device freezes the data in its DR register and does not allow it to be altered. Therefore the $\overline{SS}$ pin must be high to write a new data byte in the DR without producing a write collision.

Figure 58. CPHA / $\overline{SS}$ Timing Diagram



SERIAL PERIPHERAL INTERFACE (Cont'd)

Figure 59. Data Clock Timing Diagram



Note: This figure should not be used as a replacement for parametric information. Refer to the Electrical Characteristics chapter.

VR02131B

SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.4.4 Write Collision Error

A write collision occurs when the software tries to write to the DR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted; and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode.

Note: a "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation.

In Slave mode

When the CPHA bit is set:

The slave device will receive a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device DR register and output the MSBit on to the external MISO pin of the slave device.

The $\overline{SS}$ pin low state enables the slave device but the output of the MSBit onto the MISO pin does not take place until the first data transfer clock edge.

When the CPHA bit is reset:

Data is latched on the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when software attempts to write the DR register after its $\overline{SS}$ pin has been pulled low.

For this reason, the $\overline{SS}$ pin must be high, between each data byte transfer, to allow the CPU to write in the DR register without generating a write collision.

In Master mode

Collision in the master device is defined as a write of the DR register while the internal serial clock (SCK) is in the process of transfer.

The $\overline{SS}$ pin signal must be always high on the master device.

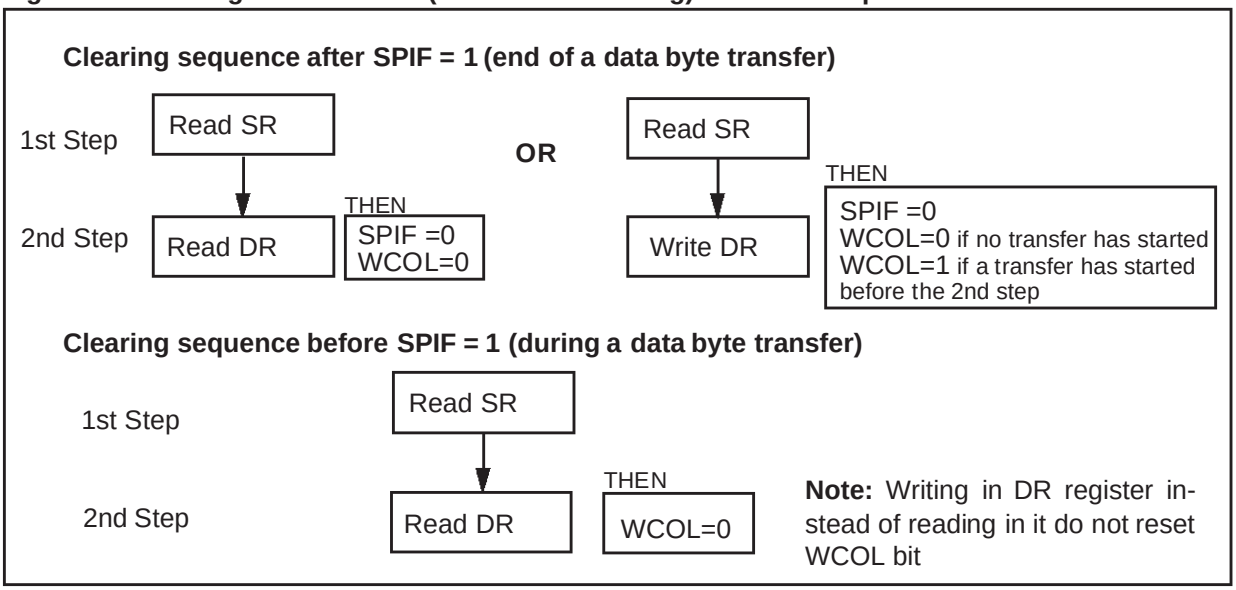
WCOL bit

The WCOL bit in the SR register is set if a write collision occurs.

No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see Figure 60).

Figure 60. Clearing the WCOL bit (Write Collision Flag) Software Sequence



SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.4.5 Master Mode Fault

Master mode fault occurs when the master device has its $\overline{SS}$ pin pulled low, then the MODF bit is set.

Master mode fault affects the SPI peripheral in the following ways:

- The MODF bit is set and an SPI interrupt is generated if the SPIE bit is set.
- The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.
- The MSTR bit is reset, thus forcing the device into slave mode.

Clearing the MODF bit is done through a software sequence:

1. A read or write access to the SR register while the MODF bit is set.
2. A write to the CR register.

Notes: To avoid any multiple slave conflicts in the case of a system comprising several MCUs, the $\overline{SS}$ pin must be pulled high during the clearing sequence of the MODF bit. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

In a slave device the MODF bit can not be set, but in a multi master configuration the device can be in slave mode with this MODF bit set.

The MODF bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state using an interrupt routine.

8.4.4.6 Overrun Condition

An overrun condition occurs when the master device has sent several data bytes and the slave device has not cleared the SPIF bit issuing from the previous data byte transmitted.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the DR register returns this byte. All other bytes are lost.

This condition is not detected by the SPI peripheral.

SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.4.7 Single Master and Multimaster Configurations

There are two types of SPI systems:

– Single Master System

– Multimaster System

Single Master System

A typical single master system may be configured, using an MCU as the master and four MCUs as slaves (see Figure 61).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line the master allows only one active slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written its DR register.

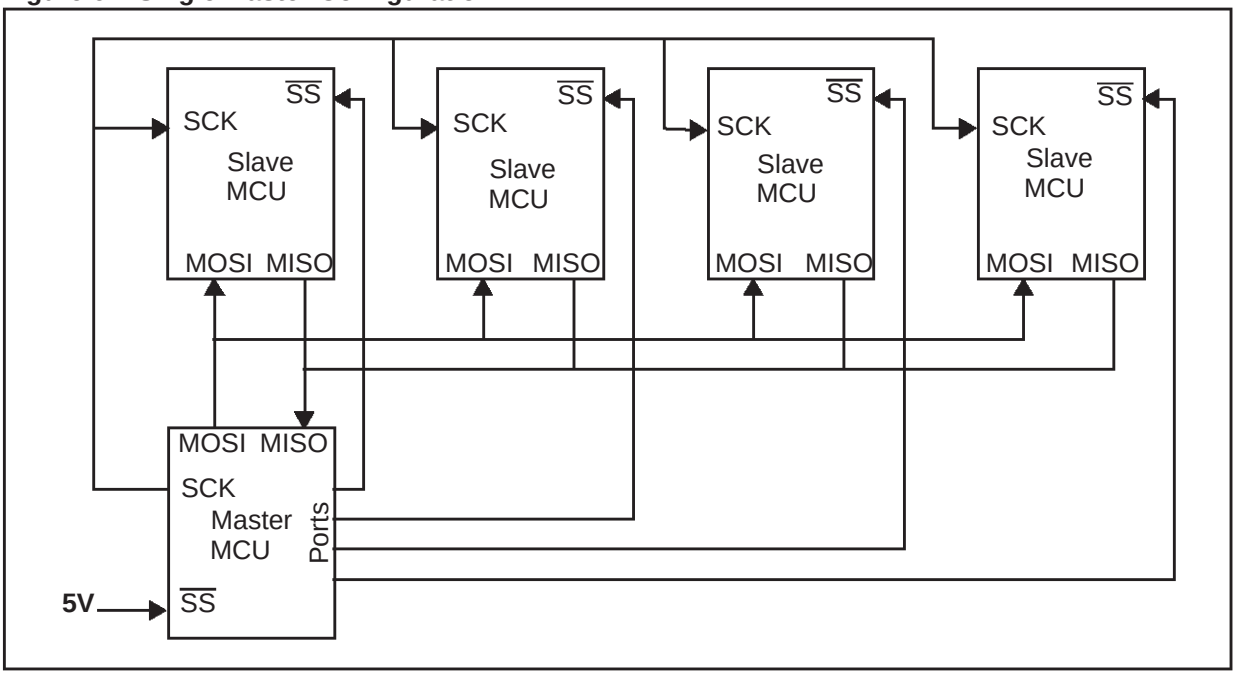
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Multi-master System

A multi-master system may also be configured by the user. Transfer of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system.

The multi-master system is principally handled by the MSTR bit in the CR register and the MODF bit in the SR register.

Figure 61. Single Master Configuration



SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.5 Low Power Modes

Mode	Description
WAIT	No effect on SPI. SPI interrupt events cause the device to exit from WAIT mode.
HALT	SPI registers are frozen. In HALT mode, the SPI is inactive. SPI operation resumes when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

8.4.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
SPI End of Transfer Event	SPIF	SPIE	Yes	No
Master Mode Fault Event	MODF		Yes	No

Note: The SPI interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

SERIAL PERIPHERAL INTERFACE (Cont'd)

8.4.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0000xxxx (0xh)

7							0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0

Bit 7 = **SPIE** *Serial peripheral interrupt enable.*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SPI interrupt is generated whenever SPIF=1 or MODF=1 in the SR register

Bit 6 = **SPE** *Serial peripheral output enable.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 8.4.4.5 Master Mode Fault).
0: I/O port connected to pins
1: SPI alternate functions connected to pins

The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins.

Bit 5 = **SPR2** *Divider Enable.*

this bit is set and cleared by software and it is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to Table 42.
0: Divider by 2 enabled
1: Divider by 2 disabled

Bit 4 = **MSTR** *Master.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 8.4.4.5 Master Mode Fault).
0: Slave mode is selected
1: Master mode is selected, the function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

Bit 3 = **CPOL** *Clock polarity.*
This bit is set and cleared by software. This bit determines the steady state of the serial Clock. The CPOL bit affects both the master and slave modes.
0: The steady state is a low value at the SCK pin.
1: The steady state is a high value at the SCK pin.

Bit 2 = **CPHA** *Clock phase.*
This bit is set and cleared by software.
0: The first clock transition is the first data capture edge.
1: The second clock transition is the first capture edge.

Bit 1:0 = **SPR[1:0]** *Serial peripheral rate.*
These bits are set and cleared by software.Used with the SPR2 bit, they select one of six baud rates to be used as the serial clock when the device is a master.

These 2 bits have no effect in slave mode.

Table 42. Serial Peripheral Baud Rate

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

SERIAL PERIPHERAL INTERFACE (Cont'd)

STATUS REGISTER (SR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	-	MODF	-	-	-	-

Bit 7 = **SPIF** *Serial Peripheral data transfer flag.*
This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE=1 in the CR register. It is cleared by a software sequence (an access to the SR register followed by a read or write to the DR register).
0: Data transfer is in progress or has been approved by a clearing sequence.
1: Data transfer between the device and an external device has been completed.

Note: While the SPIF bit is set, all writes to the DR register are inhibited.

Bit 6 = **WCOL** *Write Collision status.*

This bit is set by hardware when a write to the DR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 60).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = Unused.

Bit 4 = **MODF** *Mode Fault flag.*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 8.4.4.5 Master Mode Fault). An SPI interrupt can be generated if SPIE=1 in the CR register. This bit is cleared by a software sequence (An access to the SR register while MODF=1 followed by a write to the CR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bits 3-0 = Unused.

DATA I/O REGISTER (DR)

Read/Write

Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The DR register is used to transmit and receive data on the serial bus. In the master device only a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

Warning:

A write to the DR register places data directly into the shift register for transmission.

A write to the the DR register returns the value located in the buffer and not the contents of the shift register (See Figure 57).

SERIAL PERIPHERAL INTERFACE (Cont'd)

Table 43. SPI Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0021h	**SPIDR** Reset Value	MSB x	x	x	x	x	x	x	LSB x
0022h	**SPICR** Reset Value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0023h	**SPISR** Reset Value	SPIF 0	WCOL 0	0	MODF 0	0	0	0	0

8.5 8-BIT A/D CONVERTER (ADC)

8.5.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 8 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 8 different sources.

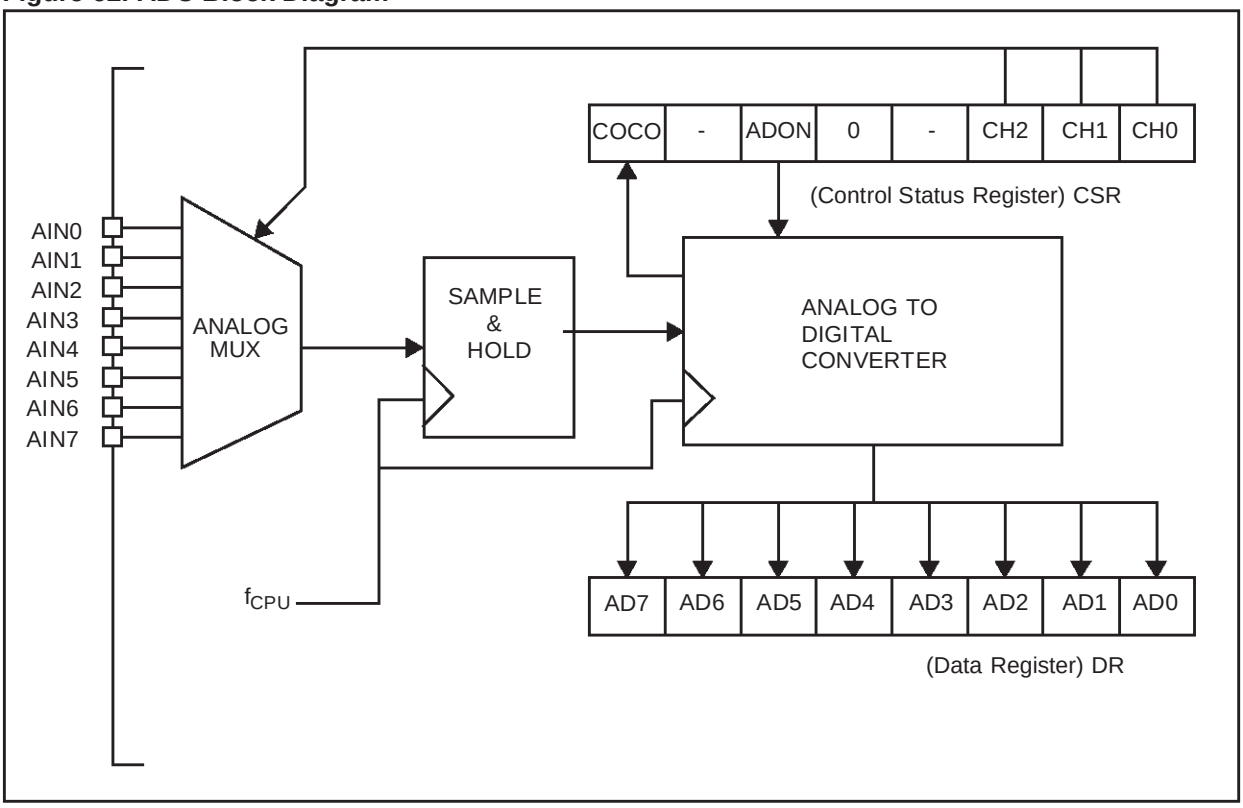
The result of the conversion is stored in a 8-bit Data Register. The A/D converter is controlled through a Control/Status Register.

8.5.2 Main Features

- 8-bit conversion
- Up to 8 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/Off bit (to reduce consumption)

The block diagram is shown in Figure 62.

Figure 62. ADC Block Diagram



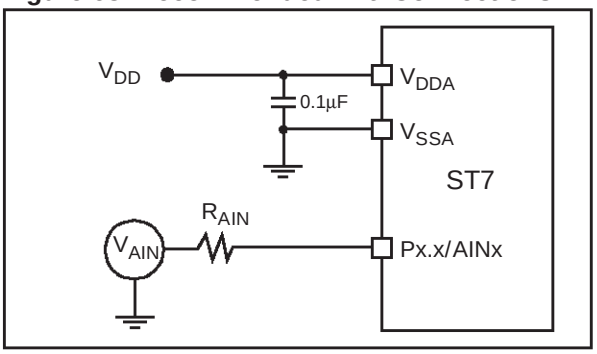
8-BIT A/D CONVERTER (ADC) (Cont'd)

8.5.3 Functional Description

The high level reference voltage V_{DDA} must be connected externally to the V_{DD} pin. The low level reference voltage V_{SSA} must be connected externally to the V_{SS} pin. In some devices (refer to device pin out description) high and low level reference voltages are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be degraded by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

Figure 63. Recommended Ext. Connections



Characteristics:

The conversion is monotonic, meaning the result never decreases if the analog input does not and never increases if the analog input does not.

If input voltage is greater than or equal to V_{DD} (voltage reference high) then results = FFh (full scale) without overflow indication.

If input voltage $\leq V_{SS}$ (voltage reference low) then the results = 00h.

The conversion time is 64 CPU clock cycles including a sampling time of 31.5 CPU clock cycles.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

The A/D converter is linear and the digital result of the conversion is given by the formula:

$$\text{Digital result} = \frac{255 \times \text{Input Voltage}}{\text{Reference Voltage}}$$

Where Reference Voltage is $V_{DD} - V_{SS}$.

The accuracy of the conversion is described in the Electrical Characteristics Section.

Procedure:

Refer to the CSR and DR register description section for the bit definitions.

Each analog input pin must be configured as input, no pull-up, no interrupt. Refer to the "I/O Ports" chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the CSR register:

– Select the CH2 to CH0 bits to assign the analog channel to convert. Refer to Table 44 Channel Selection.

– Set the ADON bit. Then the A/D converter is enabled after a stabilization time (typically 30 µs). It then performs a continuous conversion of the selected channel.

When a conversion is complete

– The COCO bit is set by hardware.

– No interrupt is generated.

– The result is in the DR register.

A write to the CSR register aborts the current conversion, resets the COCO bit and starts a new conversion.

8.5.4 Low Power Modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed.

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilisation time before accurate conversions can be performed.

8.5.5 Interrupts

None.

8-BIT A/D CONVERTER (ADC) (Cont'd)

8.5.6 Register Description

CONTROL/STATUS REGISTER (CSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
COCO	-	ADON	0	-	CH2	CH1	CH0

Bit 7 = **COCO** *Conversion Complete*

This bit is set by hardware. It is cleared by software reading the result in the DR register or writing to the CSR register.
0: Conversion is not complete.
1: Conversion can be read from the DR register.

Bit 6 = **Reserved**. Must always be cleared.

Bit 5 = **ADON** *A/D converter On*

This bit is set and cleared by software.
0: A/D converter is switched off.
1: A/D converter is switched on.

Note: A typical 30 µs delay time is necessary for the ADC to stabilize when the ADON bit is set.

Bit 4 = **Reserved**. Forced by hardware to 0.

Bit 3 = **Reserved**. Must always be cleared.

Bits 2:0: **CH[2:0]** *Channel Selection*

These bits are set and cleared by software. They select the analog input to convert.

Table 44. Channel Selection

Pin*	CH2	CH1	CH0
AIN0	0	0	0
AIN1	0	0	1
AIN2	0	1	0
AIN3	0	1	1
AIN4	1	0	0
AIN5	1	0	1
AIN6	1	1	0
AIN7	1	1	1

***IMPORTANT NOTE:** The number of pins AND the channel selection vary according to the device. REFER TO THE DEVICE PINOUT).

DATA REGISTER (DR)

Read Only

Reset Value: 0000 0000 (00h)

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Bit 7:0 = **AD[7:0]** *Analog Converted Value*

This register contains the converted analog value in the range 00h to FFh.

Reading this register resets the COCO flag.

Table 45. ADC Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0070h	**ADCDR** Reset Value	IS11 0	IS10 0	MCO 0	IS21 0	IS20 0	CP1 0	CP0 0	SMS 0
0071h	**ADCCSR Standard** Reset Value	COCO 0	0	ADON 0	0	0	CH2 0	CH1 0	CH0 0

9 INSTRUCTION SET

9.1 ST7 ADDRESSING MODES

The ST7 Core features 17 different addressing modes which can be classified in 7 main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two sub-modes called long and short:

– Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.

– Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 46. ST7 Addressing Mode Overview

Mode			Syntax	Destination/ Source	Pointer Address (Hex.)	Pointer Size (Hex.)	Length (Bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0 (with X register) + 1 (with Y register)
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC-128/PC+127[1)]			+ 1
Relative	Indirect		jrne [$10]	PC-128/PC+127[1)]	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

Note 1. At the time the instruction is executed, the Program Counter (PC) points to the instruction following JRxx.

ST7 ADDRESSING MODES (Cont'd)

9.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

9.1.2 Immediate

Immediate instructions have two bytes, the first byte contains the opcode, the second byte contains the operand value.

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

9.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two sub-modes:

Direct (short)

The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

9.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three sub-modes:

Indexed (No Offset)

There is no offset, (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (Short)

The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

9.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two sub-modes:

Indirect (short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

ST7 ADDRESSING MODES (Cont'd)

9.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 47. Instructions Supporting Direct, Indexed, Indirect and Indirect Indexed Addressing Modes

Long and Short Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
BCP	Bit Compare

Short Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

9.1.7 Relative Mode (Direct, Indirect)

This addressing mode is used to modify the PC register value by adding an 8-bit signed offset to it.

Available Relative Direct/ Indirect Instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

The offset is defined in memory, of which the address follows the opcode.

9.2 INSTRUCTION GROUPS

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Code Condition Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four bytes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

- PC-2 End of previous instruction
- PC-1 Prebyte
- PC Opcode
- PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

- PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.
- PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
  It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.
- PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

INSTRUCTION GROUPS (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg	reg	M					
		pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

10 ELECTRICAL CHARACTERISTICS

10.1 ABSOLUTE MAXIMUM RATINGS

This product contains devices for protecting the inputs against damage due to high static voltages, however it is advisable to take normal precautions to avoid applying any voltage higher than the specified maximum rated voltages.

For proper operation it is recommended that V_I and V_O be higher than V_{SS} and lower than V_{DD}. Reliability is enhanced if unused inputs are connected to an appropriate logic voltage level (V_{DD} or V_{SS}).

Power Considerations. The average chip-junction temperature, T_J, in Celsius can be obtained from:

T_J = TA + PD x RthJA

Where: T_A = Ambient Temperature.
RthJA = Package thermal resistance (junction-to ambient).
P_D = P_{INT} + P_{PORT}.
P_{INT} = I_{DD} x V_{DD} (chip internal power).
P_{PORT} = Port power dissipation determined by the user)

Symbol	Ratings	Value	Unit
V_{DD} - V_{SS}	Supply voltage	6.5	V
V_{IN}	Input voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
V_{OUT}	Output voltage	V_{SS} - 0.3 to V_{DD} + 0.3	V
ESD	ESD susceptibility	2000	V
I_{VDD_i}	Total current into V_{DD_i} (source)	80	mA
I_{VSS_i}	Total current out of V_{SS_i} (sink)	80	

Note:

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

General Warning:

Direct connection to V_{DD} or V_{SS} of the RESET and I/O pins could damage the device in case of unintentional internal reset generation or program counter corruption (due to unwanted change of the I/O configuration). To guarantee safe conditions, this connection has to be done through a 10KΩ typical pull-up or pull-down resistor.

Thermal Characteristics

Symbol	Ratings		Value	Unit
R_{thJA}	Package thermal resistance	SO34 SDIP32	75 60	°C/W
T_{Jmax}	Max. junction temperature		150	°C
T_{STG}	Storage temperature range		-65 to +150	°C
PD	Power dissipation		500	mW

10.2 RECOMMENDED OPERATING CONDITIONS

GENERAL						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
V_{DD}	Supply voltage		4.0		5.5	V
f_{OSC}	Resonator oscillator frequency		8 or 16 [2]			MHz
	External clock source					
T_A	Ambient temperature range	1 Suffix Version	0		70	°C
		6 Suffix Version	-40		85	

10.3 DC ELECTRICAL CHARACTERISTICS

Recommended operating conditions with T_A=-40 to +85°C, V_{DD}-V_{SS}=5V unless otherwise specified.

Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
I_{DD}	Supply current in RUN mode [3]	f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz		5 7	8 12	mA
	Supply current in SLOW mode [3]	f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz		0.7 1	1.1 1.5	
	Supply current in WAIT mode [4]	f_{OSC} = 8MHz, f_{CPU} = 4 MHz f_{OSC} = 16MHz, f_{CPU} = 8 MHz		2 3.3	3 5	
	Supply current in SLOW WAIT mode [4]	f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz		0.65 0.8	1 1.2	
	Supply current in HALT mode [5]	I_{LOAD} = 0mA (current on IOs)			200	µA
V_{RM}	Data retention mode [6]	HALT mode	2			V

10.4 GENERAL TIMING CHARACTERISTICS

Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
t_{INST}	Instruction time		2		12	t_{CPU}
t_{IRT}	Interrupt reaction time	$t_{IRT} = \Delta t_{INST} + 10$ [7]	10		22	t_{CPU}

Notes:

1) Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

2) Fixed frequencies required to obtain 4MHz for the motor control peripheral.

3) CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS}, all peripherals switched off; clock input (OSC2) driven by external square wave.

4) All I/O pins in input mode with a static value at V_{DD} or V_{SS}, all peripherals switched off; clock input (OSC2) driven by external square wave.

5) All I/O pins in input mode with a static value at V_{DD} or V_{SS}.

6) Data based on characterization results, not tested in production.

7) Δt_{INST} is the number of t_{CPU} to finish the current instruction execution.

10.5 I/O PORT CHARACTERISTICS

Recommended operating conditions
with T_A=-40 to +85°C and 4.5V<V_{DD}-V_{SS}<5.5V unless otherwise specified.

I/O PORT PINS						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
V_{IL}	Input low level voltage [2]				0.3xV_{DD}	V
V_{IH}	Input high level voltage [2]		0.7xV_{DD}			
V_{HYS}	Schmitt trigger voltage hysteresis [3]			400		mV
V_{OL}	Output low level voltage for standard I/O port pins	I=-5mA			1.3	V
		I=-2mA			0.5	
	Output low level voltage for high sink I/O port pins	I=-20mA			1.3	
		I=-8mA			0.5	
V_{OH}	Output high level voltage	I=-5mA	V_{DD}-2.0			
		I=-2mA	V_{DD}-0.8			
R_{PU}	Pull-up equivalent resistor	V_{IN} > V_{IH} V_{IN} < V_{IL}	20 60	40 120	80 240	kΩ
I_L	Input leakage current	V_{SS}<V_{PIN}<V_{DD}			1	µA
I_{SV}	Static current consumption [2]	Floating input mode			200	
I_{PINJ}	Single pin injected current	Positive [5]: V_{EXT}>V_{DD}			5	mA
		Negative [6]: V_{EXT}<V_{SS}			-5	
I_{INJ}	Total injected current [7] (sum of all I/O and control pins)	Positive: V_{EXT}>V_{DD}			20	
		Negative: V_{EXT}<V_{SS}			20	
t_{OHL}	Output high to low level fall time	C_l=50pF	14.8 [4]	25	45.6 [4]	ns
t_{OLH}	Output low to high rise time		14.4 [4]	25	45.9 [4]	
t_{ITEXT}	External interrupt pulse time [8]		1			t_{CPU}

Notes:

1) Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

2) Data based on design simulations and/or technology characteristics, not tested in production.

3) Hysteresis voltage between Schmitt trigger switching levels. Based on characterisation results, not tested.

4) Data based on characterization results, not tested in production.

5) Positive injection (I_{INJ+})
The I_{INJ+} is performed through protection diodes insulated from the substrate of the die.
The true open-drain pins do not accept positive injection. In this case the maximum voltage rating must be respected.

6) ADC accuracy reduced by negative injection (I_{INJ-})
The I_{INJ-} is performed through protection diodes NOT INSULATED from the substrate of the die. The drawback is a small leakage (a few µA) induced inside the die when a negative injection is performed. This leakage is tolerated by the digital structure, but it acts on the analog line depending on the impedance versus a leakage current of a few µA (if the MCU has an AD converter). The effect depends on the pin which is submitted to the injection. Of course, external digital signals applied to the component must have a maximum impedance close to 50KΩ.
Location of the negative current injection:
- Pins with analog input capability are the most sensitive. I_{INJ-} maximum is 0.8 mA (assuming that the impedance of the analog voltage is lower than 25KΩ)
- Pure digital pins can tolerate 1.6mA. In addition, the best choice is to inject the current as far as possible from the analog input pins.

7) When several inputs are submitted to a current injection, the maximum I_{INJ} is the sum of the positive (or negative) currents (instantaneous values). These results are based on characterisation with I_{INJ} maximum current injection on four I/O port pins of the device.

8) To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

10.6 SUPPLY, RESET AND CLOCK CHARACTERISTICS

10.6.1 Supply Manager

Recommended operating conditions
with T_A=-40 to +85°C and voltage are referred to V_{SS} unless otherwise specified.

LOW VOLTAGE DETECTOR (LVD)						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
V_{LVDr}	Reset release threshold	V_{DD} rise		4.05	4.30	V
V_{LVDf}	Reset generation threshold	V_{DD} fall	3.5	3.75	4.0	
$V_{LVDhyst}$	V_{LVD} Hysteresis [2]	V_{LVDr} - V_{LVDf}		250		mV
I_{DD}	LVD Supply Current	HALT mode		100	200 [3]	µA

10.6.2 RESET Sequence Manager

Recommended operating conditions
with T_A=-40...+85°C and 4.5V<V_{DD}-V_{SS}<5.5V unless otherwise specified.

RESET SEQUENCE MANAGER (RSM)						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [4]	**Max**	**Unit**
R_{ON}	Reset weak pull-up resistance	$V_{IN} > V_{IH}$ $V_{IN} \geq V_{SS}$	5 20	10 80	20 160	kΩ
$t_{DELAYmin}$	Reset delay for external and watchdog reset sources			6 30		$1/f_{SFOSC}$ µs
t_{PULSE}	External $\overline{RESET}$ pin Pulse time		20			µs

10.6.3 Clock System

Recommended operating conditions
with T_A=-40 to +85°C and voltage are referred to V_{SS} unless otherwise specified.

EXTERNAL CLOCK SOURCE						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
V_{OSC2h}	OSC2 input pin high level voltage	Square wave signal with ~50% Duty Cycle	$0.7xV_{DD}$		V_{DD}	V
V_{OSC2l}	OSC2 input pin low level voltage		V_{SS}		$0.3xV_{DD}$	

CRYSTAL AND CERAMIC RESONATOR OSCILLATORS						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [4]	**Max**	**Unit**
f_{OSC}	Oscillator Frequency [5]		8		16	MHz
C_{Li}	Load Capacitor	R_{Smax}=100Ω [6]	15 [7]	18	21 [4]	pF
I_{DD}	Supply Current			700	1100 [3]	µA
t_{START}	Oscillator start-up time	Depends on resonator quality. A typical value is 10ms				

Notes:

1) LVD typical data are based on T_A=25°C. They are given only as design guidelines and are not tested.

2) The $V_{LVDhyst}$ hysteresis is constant.

3) Data based on characterization results, not tested in production.

4) Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

5) These data are based on typical R_{Smax}. The oscillator selection can be optimized in terms of supply current with a high quality resonator.

6) R_{Smax} is the equivalent serial resistor of the crystal or ceramic resonator.

7) Data based on design simulations and/or technology characteristics, not tested in production.

10.7 MEMORY AND PERIPHERAL CHARACTERISTICS

EPROM						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
W_{ERASE}	UV lamp	Lamp wavelength 2537Å		15		W-sec/cm^2
t_{ERASE}	Erase Time	UV lamp placed 1 inch from the device window without any interposed filters	15		20	min

Recommended operating conditions.

WATCHDOG						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
t_{DOG}	Watchdog time-out	f_{CPU} = 4MHz f_{CPU} = 8MHz	49152 6.144		3145728 393.216	t_{CPU} ms
t_{WDGRST}	Watchdog RESET pulse width			500		ns

Recommended operating conditions with T_A=-40 to +85°C and V_{DD}-V_{SS}=5V unless otherwise specified.

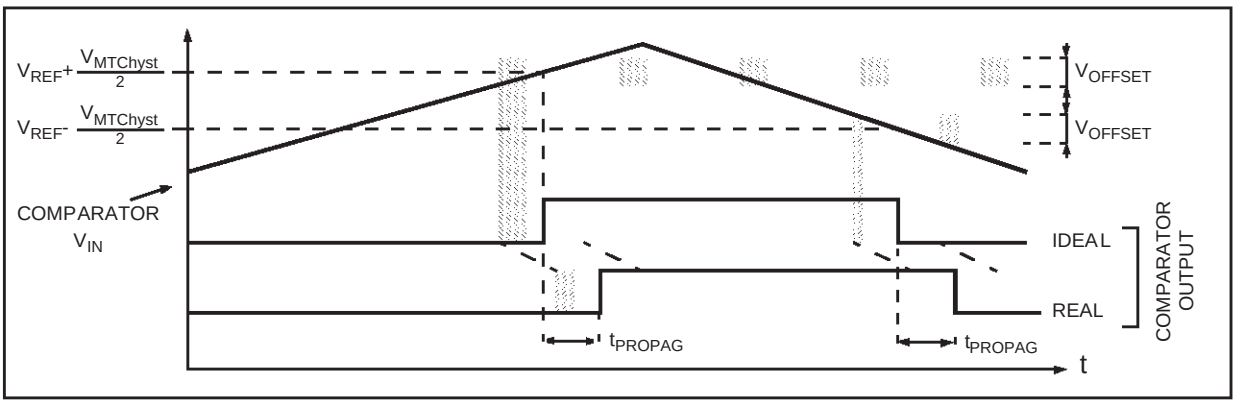
MOTOR CONTROL						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
V_{OFFSET}	Comparator offset error				100	mV
$V_{MTChyst}$	MCIA/B/C comparator hysteresis [2]			100		mV
t_{PROPAG}	Comparator propagation delay				1	µs
$\frac{\Delta V_{REF}}{V_{REF}}$	Reference voltage tolerance				5	%
R1	V_{CREF} resistance bridge			30		kΩ
R2				70		
$\alpha=\frac{R2}{R1+R2}$				0.7		
$\Delta\alpha/\alpha$	α tolerance				5	%

Note:

1) Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

2) The $V_{MTChyst}$ hysteresis is constant.

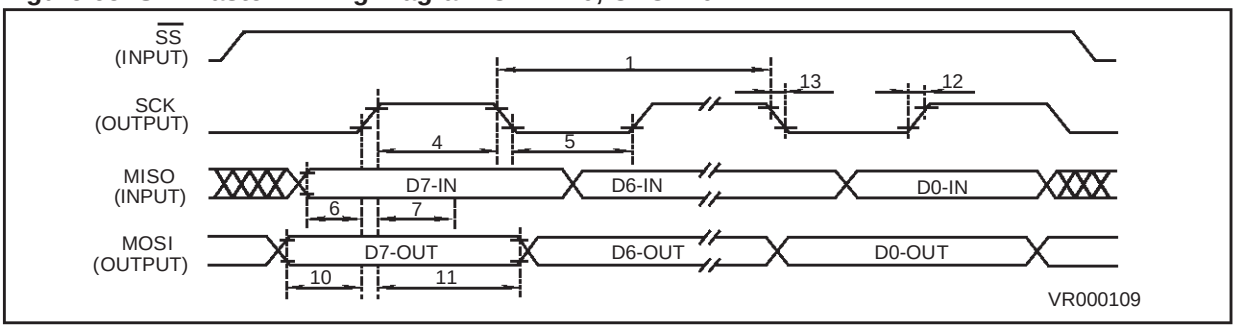
Figure 64. Motor Control Comparator Characteristics



MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

SPI Serial Peripheral Interface						
Ref.	Symbol	Parameter	Condition	Value [1]		Unit
				Min	Max	
	f_{SPI}	SPI frequency	Master Slave	1/128 dc	1/4 1/2	f_{CPU}
1	t_{SPI}	SPI clock period	Master Slave	4 2		t_{CPU}
2	t_{Lead}	Enable lead time	Slave	120		ns
3	t_{Lag}	Enable lag time	Slave	120		ns
4	t_{SPI_H}	Clock (SCK) high time	Master Slave	100 90		ns
5	t_{SPI_L}	Clock (SCK) low time	Master Slave	100 90		ns
6	t_{SU}	Data set-up time	Master Slave	100 100		ns
7	t_H	Data hold time (inputs)	Master Slave	100 100		ns
8	t_A	Access time (time to data active from high impedance state)	Slave	0	120	ns
9	t_{Dis}	Disable time (hold time to high impedance state)			240	ns
10	t_V	Data valid	Master (before capture edge) Slave (after enable edge)	0.25	 120	t_{CPU} ns
11	t_{Hold}	Data hold time (outputs)	Master (before capture edge) Slave (after enable edge)	0.25 0		t_{CPU} ns
12	t_{Rise}	Rise time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)	Outputs: SCK,MOSI,MISO Inputs: SCK,MOSI,MISO,$\overline{SS}$		100 100	ns µs
13	t_{Fall}	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200pF)	Outputs: SCK,MOSI,MISO Inputs: SCK,MOSI,MISO,$\overline{SS}$		100 100	ns µs

Figure 65. SPI Master Timing Diagram CPHA=0, CPOL=0 [2]



Notes:

1) Data based on characterization results, not tested in production.

2) Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI timing diagram

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

Figure 66. SPI Master Timing Diagram CPHA=0, CPOL=1 1)

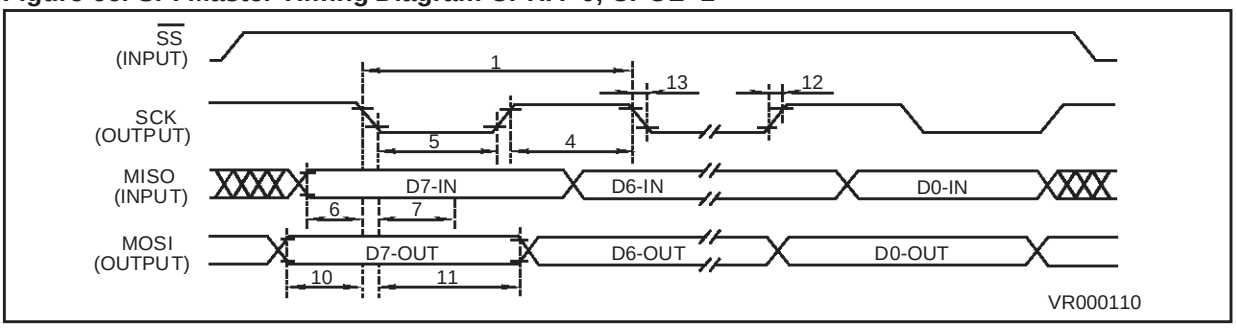


Figure 67. SPI Master Timing Diagram CPHA=1, CPOL=0 1)

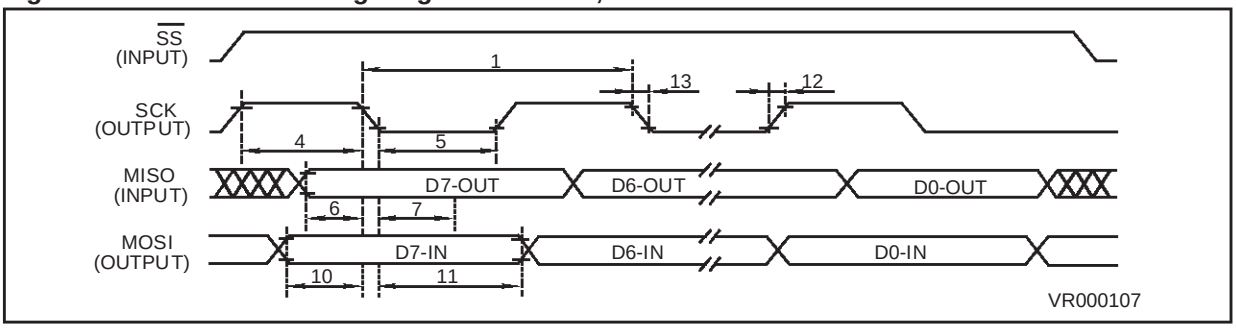
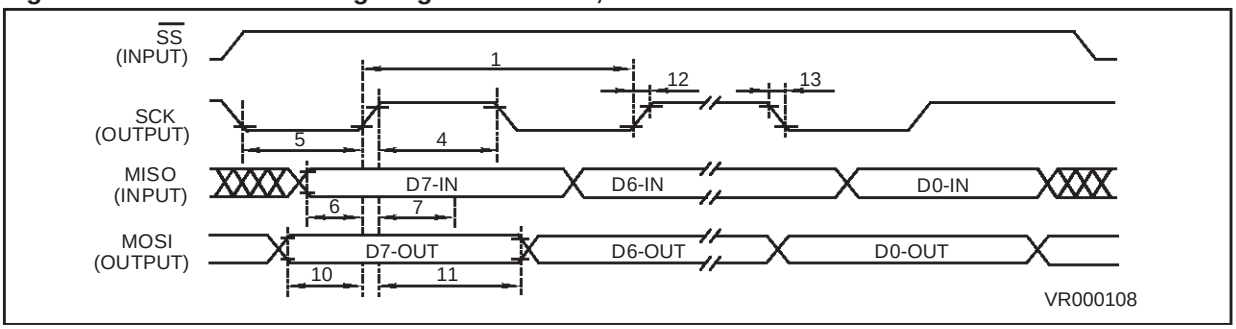


Figure 68. SPI Master Timing Diagram CPHA=1, CPOL=1 1)



Note:
1) Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI timing diagram

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI Timing Diagram

Figure 69. SPI Slave Timing Diagram CPHA=0, CPOL=0 1)

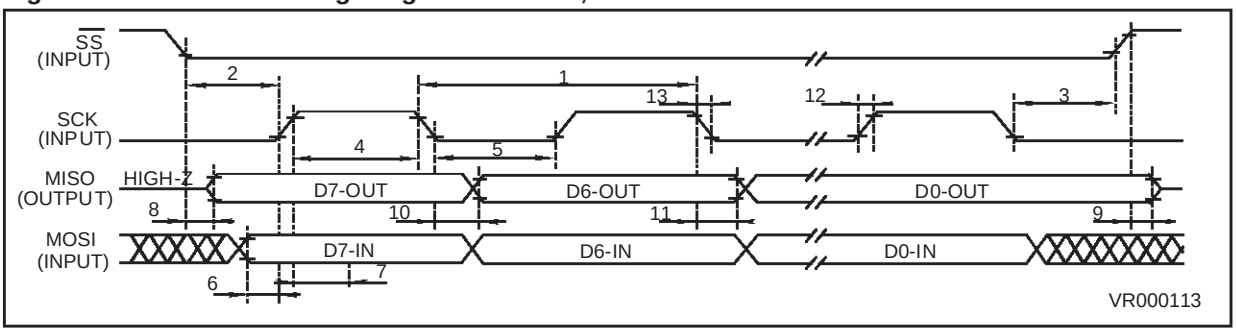


Figure 70. SPI Slave Timing Diagram CPHA=0, CPOL=1 1)

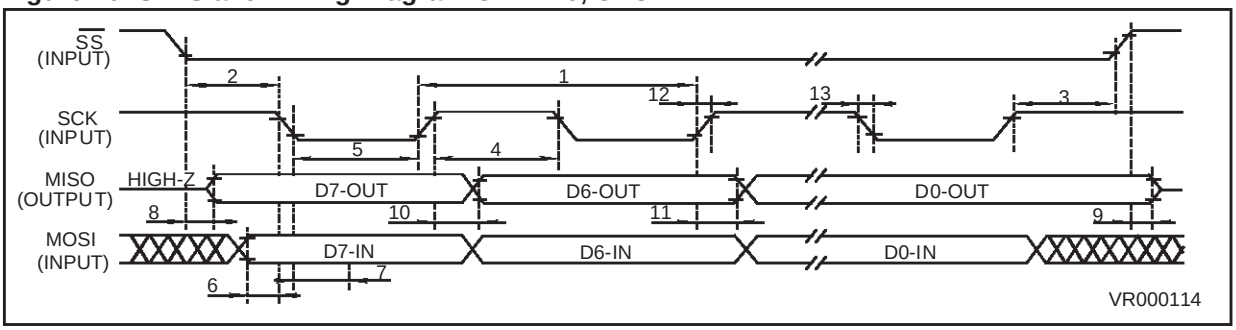


Figure 71. SPI Slave Timing Diagram CPHA=1, CPOL=0 1)

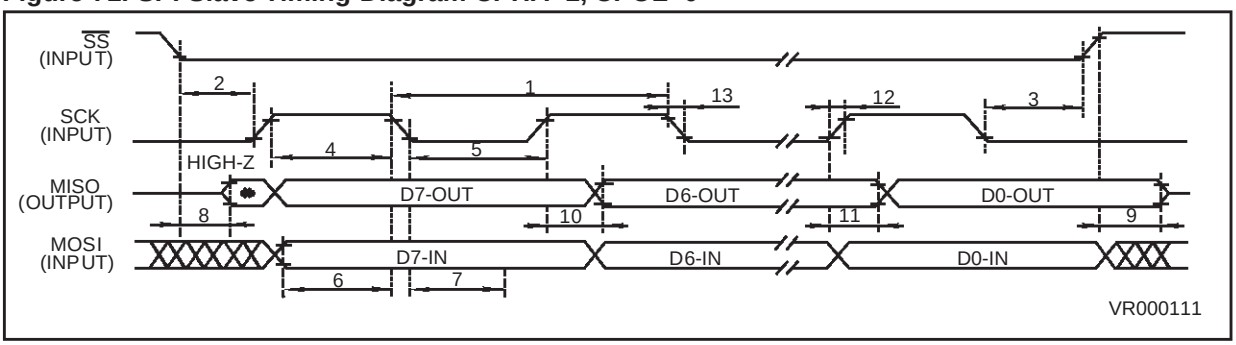
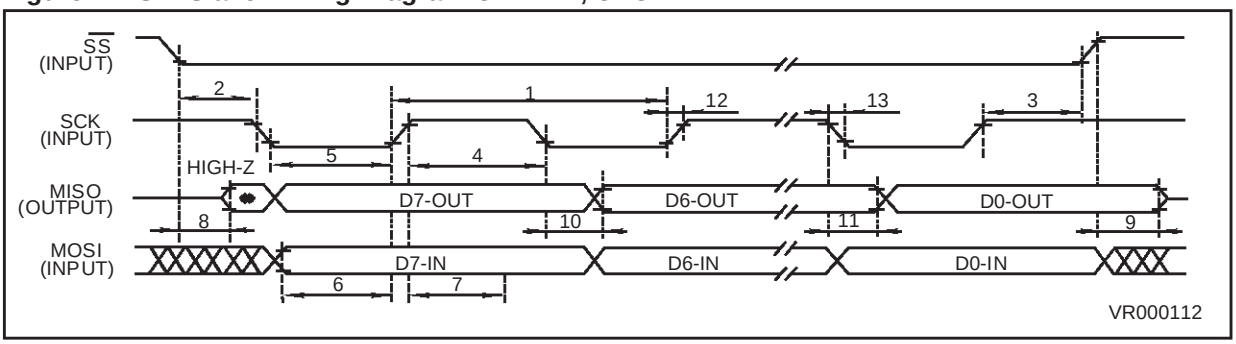


Figure 72. SPI Slave Timing Diagram CPHA=1, CPOL=1 1)



Note:

1) Measurement points are V_{OL}, V_{OH}, V_{IL} and V_{IH} in the SPI timing diagram

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

ADC Analog to Digital Converter (8-bit)						
Symbol	**Parameter**	**Conditions**	**Min**	**Typ** [1]	**Max**	**Unit**
|TUE|	Total unadjusted error [3]	T_A=25°C, V_{DD}=V_{DDA}=5V, [2] f_{CPU}=8MHz			2	LSB
OE	Offset error [3]		-1		1	
GE	Gain Error [3]		-2		2	
|DLE|	Differential linearity error [3]				1	
|ILE|	Integral linearity error [3]				2	
V_{AIN}	Conversion range voltage		V_{SSA}		V_{DDA}	V
I_{ADC}	A/D conversion supply current	f_{ADC}=f_{CPU}=4MHz V_{DD}=V_{DDA}=5V		1		mA
t_{STAB}	Stabilization time after ADC enable				30	µs
t_{LOAD}	Sample capacitor loading time		8 32			µs $1/f_{ADC}$
t_{CONV}	Hold conversion time		8 32			µs $1/f_{ADC}$
R_{AIN}	External input resistor				20 [4]	kΩ
R_{ADC}	Internal input resistor				18	kΩ
C_{SAMPLE}	Sample capacitor				22	pF

Notes:

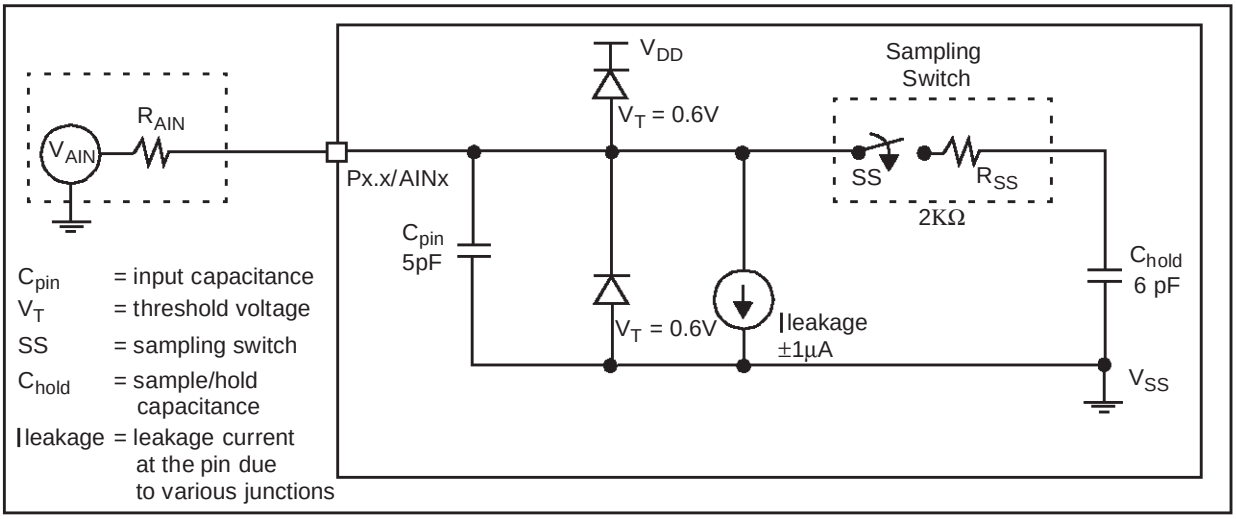
1) Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only for design guidelines and are not tested.

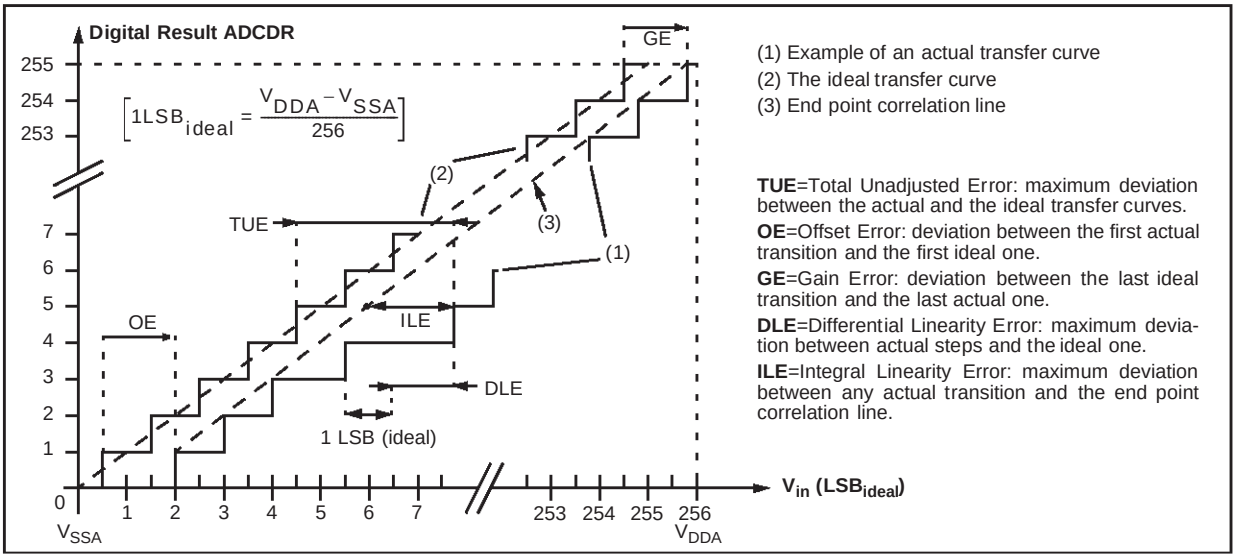
2) Tested in production at T_A=25°C, characterized over all temperature range.

3) ADC Accuracy vs. Negative Injection Current:
For I_{INJ-}=0.8mA, the typical leakage induced inside the die is 1.6µA and the effect on the ADC accuracy is a loss of 1 LSB by 10KΩ increase of the external analog source impedance.
These measurement results and recommendations have been done under worst conditions for injection:
- negative injection
- injection to an Input with analog capability, adjacent to the enabled Analog Input
- at 5V V_{DD} supply, and worst case temperature.

4) Data based on characterization results, not tested in production.

MEMORY AND PERIPHERAL CHARACTERISTICS (Cont'd)

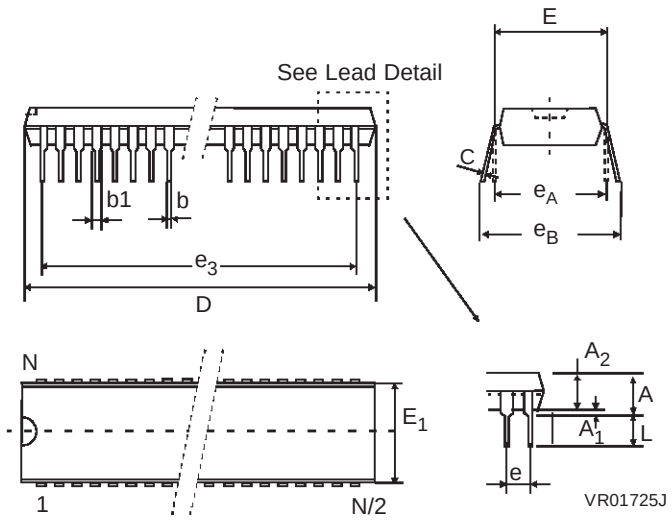




11 GENERAL INFORMATION

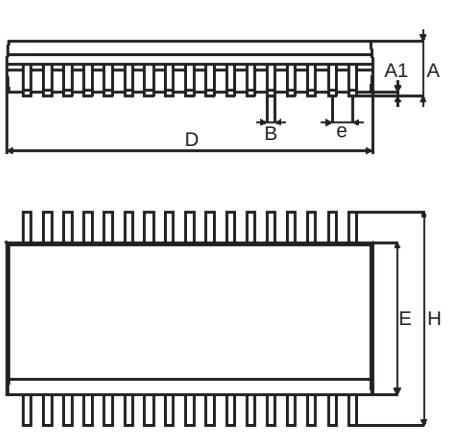
11.1 PACKAGE MECHANICAL DATA

Figure 73. 32-Pin Shrink Plastic Dual In Line Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	3.56	3.76	5.08	0.140	0.148	0.200
A1	0.51			0.020		
A2	3.05	3.56	4.57	0.120	0.140	0.180
b	0.36	0.46	0.58	0.014	0.018	0.023
b1	0.76	1.02	1.40	0.030	0.040	0.055
C	0.20	0.25	0.36	0.008	0.010	0.014
D	27.43	27.94	28.45	1.080	1.100	1.120
E	9.91	10.41	11.05	0.390	0.410	0.435
E1	7.62	8.89	9.40	0.300	0.350	0.370
e		1.78			0.070	
eA		10.16			0.400	
eB			12.70			0.500
L	2.54	3.05	3.81	0.100	0.120	0.150
	Number of Pins					
N	32					

Figure 74. 34-Pin Plastic Small Outline Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.46		2.64	0.097		0.104
A1	0.13		0.29	0.005		0.0115
B	0.36		0.48	0.014		0.019
C	0.23		0.32	0.0091		0.0125
D	17.73		18.06	0.698		0.711
E	7.42		7.59	0.292		0.299
e		1.02			0.040	
H	10.16		10.41	0.400		0.410
h	0.64		0.74	0.025		0.029
K				0°		8°
L	0.61		1.02	0.024		0.040
	Number of Pins					
N	34					

11.2 ORDERING INFORMATION

Transfer Of Customer Code

Customer code is made up of the ROM contents and the list of the selected options (if any). The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Figure 75. ROM Factory Coded Device Types

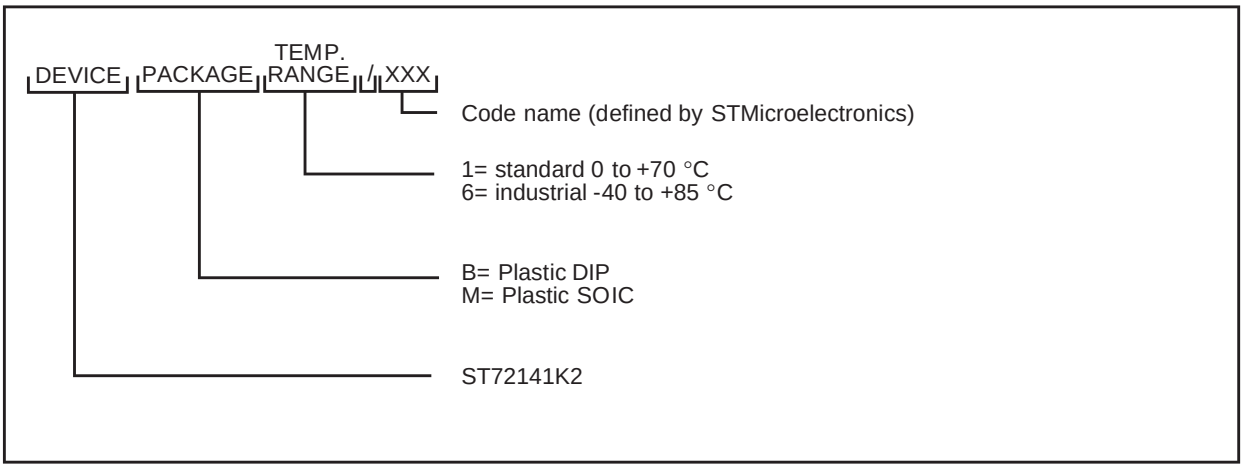
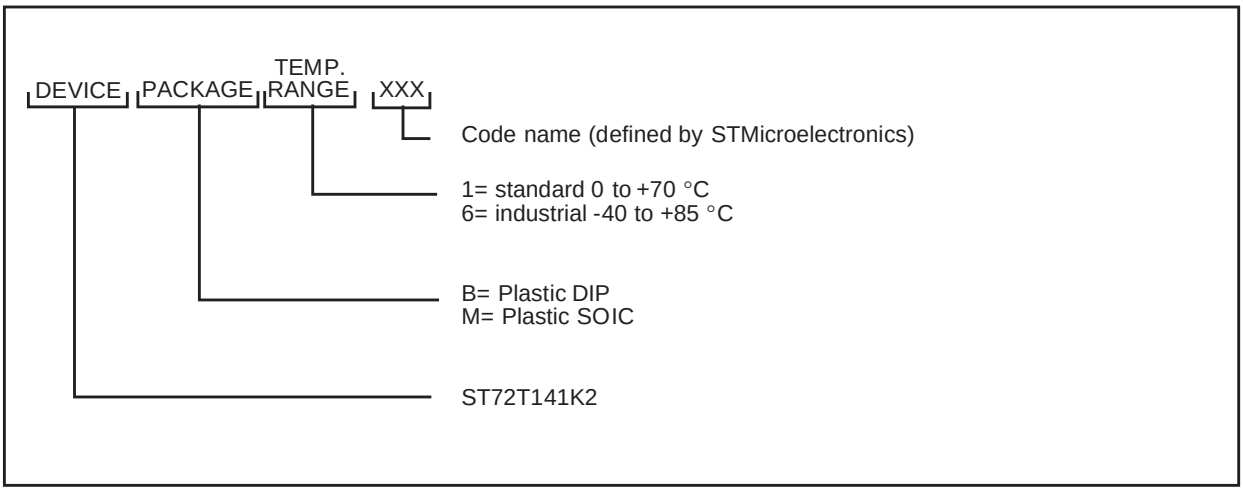


Figure 76. OTP User Programmable Device Types



MICROCONTROLLER OPTION LIST

Customer . . . . . .
Address . . . . . .
. . . . . .
Contact . . . . . .
Phone No . . . . . .
Reference . . . . . .

STMicroelectronics references

Device:	[ ] ST72141K2	[ ] ST72T141K2
Package:	[ ] SO34	[ ] SDIP32
Temperature Range:	[ ] 0°C to + 70°C	[ ] - 40°C to + 85°C

Comments :
Supply Operating Range in the application:
Notes . . . . . .
Signature . . . . . .
Date . . . . . .

12 SUMMARY OF CHANGES

Description of the changes between the current release of the specification and the previous one.

Rev.	Main Changes	Date
1.6	Update of Motor Controller macrocell	Sept 00

Notes: