Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

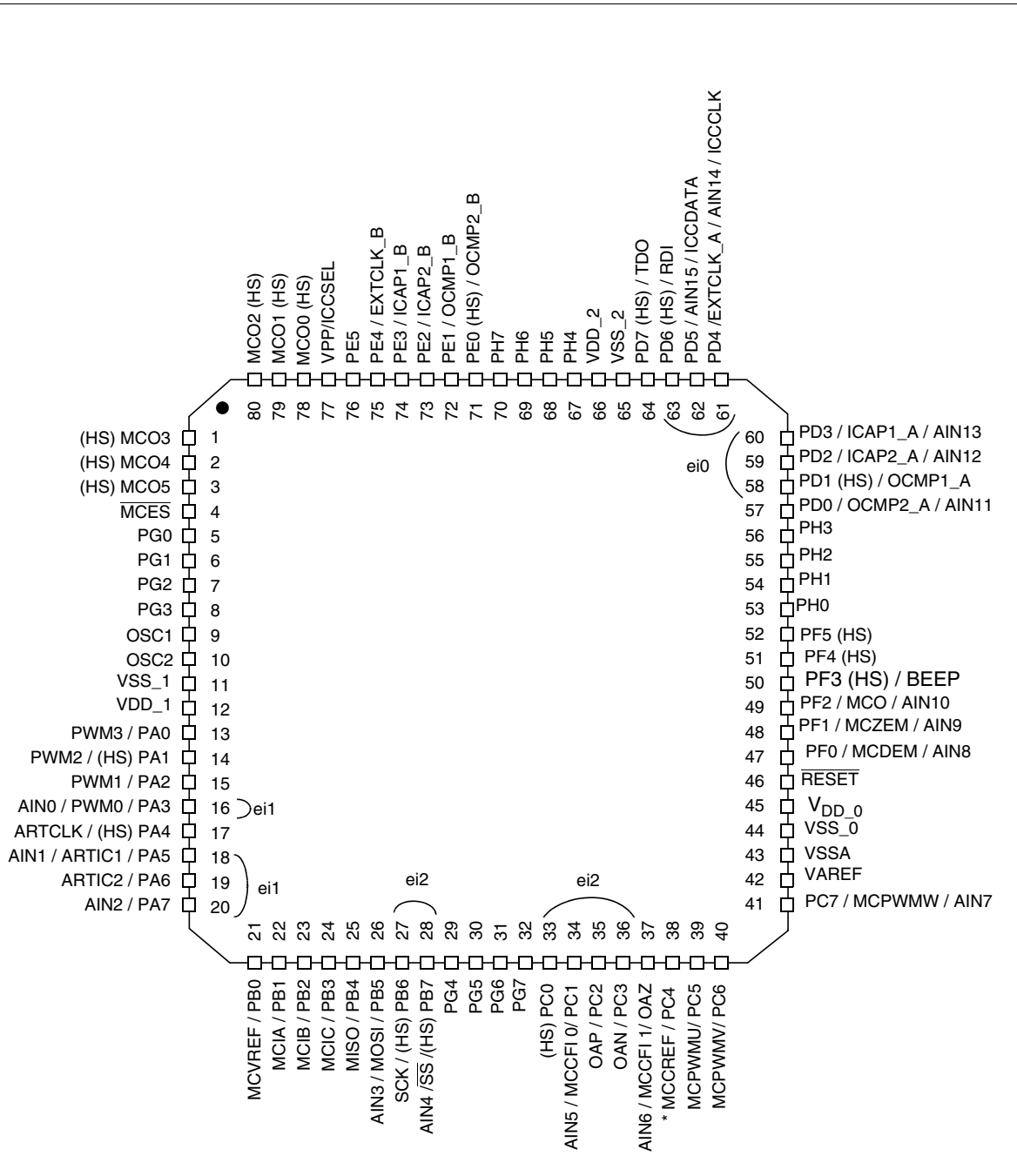
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbusSCI, SPI
Peripherals	LVD, Motor Control PWM, POR, PWM, WDT
Number of I/O	26
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc2s6t3

2 PIN DESCRIPTION

Figure 2. 80-Pin LQFP 14x14 Package Pinout



(HS) 20mA high sink capability
eix associated external interrupt vector
* Once the MTC peripheral is ON, the pin PC4 is configured to an alternate function. PC4 is no longer usable as a digital I/O

CENTRAL PROCESSING UNIT (Cont'd)

Condition Code Register (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	I1	H	I0	N	Z	C

The 8-bit Condition Code register contains the interrupt masks 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.

Arithmetic Management Bits

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 instructions. 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 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's a copy of the result 7^{th} bit.
0: The result of the last operation is positive or null.
1: The result of the last operation is negative (that is, 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.

Interrupt Management Bits

Bit 5,3 = **I1, I0** *Interrupt*

The combination of the I1 and I0 bits gives the current interrupt software priority.

Interrupt Software Priority	I1	I0
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.

INTERRUPTS (Cont'd)

7.7 EXTERNAL INTERRUPT CONTROL REGISTER (EICR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
IS11	IS10	IPB	IS21	IS20	IS31	IS30	IPA

Bit 7:6 = **IS1[1:0]** *ei2 sensitivity*
The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the following external interrupts:
- ei2 (port C3..1)

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

- ei2 (port C0, B7..6)

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

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 5 = **IPB** *Interrupt polarity for port C*

This bit is used to invert the sensitivity of the port C[3:1] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3).
0: No sensitivity inversion
1: Sensitivity inversion

Bit 4:3= **IS2[1:0]** *ei1sensitivity*
The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the following external interrupts:
- ei1 (port A3, A5...A7)

IS21	IS20	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 2:1= **IS3[1:0]** *ei0sensitivity*
The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the following external interrupts:

16-BIT TIMER (Cont'd)

10.3.3.4 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 16 Clock Control Bits).

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 16 Clock Control Bits)

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).

(Cont'd)

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	0	0

Bit 7 =
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 =
0: No match (reset value).
1: The content of the free running counter has matched 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 =
0: No timer overflow (reset value).
1: The free running counter 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.

Reading or writing the ACLR register does not clear TOF.

Bit 4 =
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 =
0: No match (reset value).
1: The content of the free running counter has matched 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 =
This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed while it is disabled.
0: Timer enabled
1: Timer prescaler, counter and outputs disabled

Bits 1:0 = Reserved, must be kept cleared.

LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

10.5.5.2 Transmitter

The transmitter can send data words of either 8 or 9 bits depending on the M bit status. When the M bit is set, word length is 9 bits and the 9th bit (the MSB) has to be stored in the T8 bit in the SCICR1 register.

Character Transmission

During an SCI transmission, data shifts out least significant bit first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see Figure 62).

Procedure

– Select the M bit to define the word length.

– Select the desired baud rate using the SCIBRR and the SCIETPR registers.

– Set the TE bit to send a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones (Idle Line) as first transmission.

– Access the SCISR register and write the data to send in the SCIDR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:
1. An access to the SCISR register
2. A write to the SCIDR register

The TDRE bit is set by hardware and it indicates:

– The TDR register is empty.

– The data transfer is beginning.

– The next data can be written in the SCIDR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I[1:0] bits are cleared in the CCR register.

When a transmission is taking place, a write instruction to the SCIDR register stores the data in the TDR register and which is copied in the shift register at the end of the current transmission.

When no transmission is taking place, a write instruction to the SCIDR register places the data directly in the shift register, the data transmission starts, and the TDRE bit is immediately set.

When a character transmission is complete (after the stop bit) the TC bit is set and an interrupt is generated if the TCIE is set and the I[1:0] bits are cleared in the CCR register.

Clearing the TC bit is performed by the following software sequence:
1. An access to the SCISR register
2. A write to the SCIDR register

Note: The TDRE and TC bits are cleared by the same software sequence.

Break Characters

Setting the SBK bit loads the shift register with a break character. The break character length depends on the M bit (see Figure 63).

As long as the SBK bit is set, the SCI sends break characters to the TDO pin. After clearing this bit by software, the SCI inserts a logic 1 bit at the end of the last break character to guarantee the recognition of the start bit of the next character.

Idle Line

Setting the TE bit drives the SCI to send a preamble of 10 (M = 0) or 11 (M = 1) consecutive '1's (idle line) before the first character.

In this case, clearing and then setting the TE bit during a transmission sends a preamble (idle line) after the current word. Note that the preamble duration (10 or 11 consecutive '1's depending on the M bit) does not take into account the stop bit of the previous character.

Note: Resetting and setting the TE bit causes the data in the TDR register to be lost. Therefore the best time to toggle the TE bit is when the TDRE bit is set, that is, before writing the next byte in the SCIDR.

LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd)

CONTROL REGISTER 1 (SCICR1)

Read/Write
Reset Value: x000 0000 (x0h)

7							0
R8	T8	SCID	M	WAKE	PCE[1)]	PS	PIE

[1)]This bit has a different function in LIN mode, please refer to the LIN mode register description.

Bit 7 = **R8** *Receive data bit 8*
This bit is used to store the 9th bit of the received word when M = 1.

Bit 6 = **T8** *Transmit data bit 8*
This bit is used to store the 9th bit of the transmitted word when M = 1.

Bit 5 = **SCID** *Disabled for low power consumption*
When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption.This bit is set and cleared by software.
0: SCI enabled
1: SCI prescaler and outputs disabled

Bit 4 = **M** *Word length*
This bit determines the word length. It is set or cleared by software.
0: 1 Start bit, 8 Data bits, 1 Stop bit
1: 1 Start bit, 9 Data bits, 1 Stop bit

Note: The M bit must not be modified during a data transfer (both transmission and reception).

Bit 3 = **WAKE** *Wake-Up method*
This bit determines the SCI Wake-Up method, it is set or cleared by software.
0: Idle Line
1: Address Mark

Note: If the LINE bit is set, the WAKE bit is deactivated and replaced by the LHDM bit.

Bit 2 = **PCE** *Parity control enable*
This bit is set and cleared by software. It selects the hardware parity control (generation and detection for byte parity, detection only for LIN parity).
0: Parity control disabled
1: Parity control enabled

Bit 1 = **PS** *Parity selection*
This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte.
0: Even parity
1: Odd parity

Bit 0 = **PIE** *Parity interrupt enable*
This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). The parity error involved can be a byte parity error (if bit PCE is set and bit LPE is reset) or a LIN parity error (if bit PCE is set and bit LPE is set).
0: Parity error interrupt disabled
1: Parity error interrupt enabled

MOTOR CONTROLLER (Cont'd)

10.6.6.5 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 30.

The D event filter can be used to select the number of consecutive D events needed to generate the D_H event.

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 simulated 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.

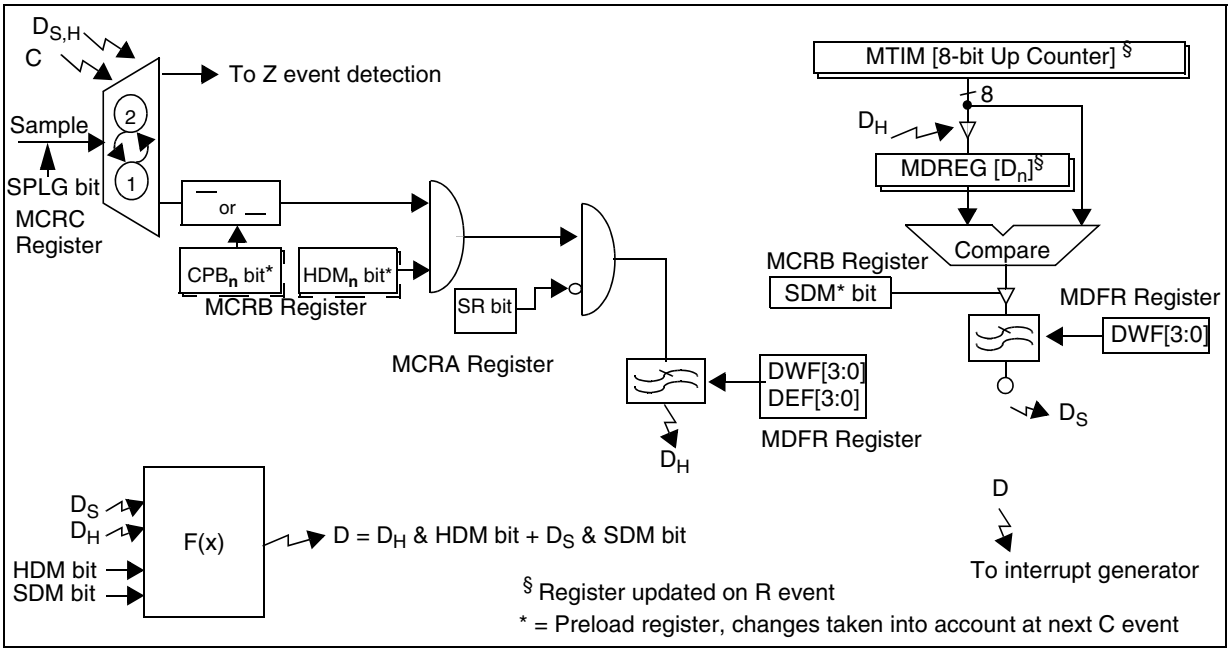
Simulated demagnetization can also be always used if the HDM bit is reset and the SDM bit is set. This mode works as a programmable masking time between the C_H and Z events. To drive the motor securely, the masking time must be always greater than the real demagnetization time in order to avoid a spurious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

Caution 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.

Caution 2: Due to the event generation protection in the MZREG, MCOMP and MDREG registers for Soft Event generation (See "Built-in Checks and Controls for simulated events" on page 175.), the value written in the MDREG register in soft demagnetisation mode (SDM=1) is checked by hardware after the C event. If this value is less than or equal to the MTIM counter value at this moment, the Software demagnetisation event is generated immediately and the MTIM current value overwrites the value in the MDREG register to be able to reuse the right demagnetisation time for another simulated event generation.

Figure 81. D Event Generation Mechanism



MOTOR CONTROLLER (Cont'd)

10.6.6.6 Z Event Generation (BEMF Zero Crossing)

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 87). After Z event, the PWM can also be switched to another group of outputs before the next C event.

A BEMF voltage is present on the non-powered terminal but referred to the 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, this means if the PWM is applied on the high side driver, free-wheeling will be done through the low 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. When the BEMF signal crosses the threshold voltage close to zero, it is called a hardware zero-crossing event Z_H. A filter can be implemented on the Z_H event detection (see Figure 83).

The Z event filter register (MZFR) is used to select the number of consecutive Z events needed to generate the Z_H event. Alternatively, the PZ bit can be used to enable protection as described in Figure 83. on page 157

For this reason the MTC outputs can be split in two groups called LOW and HIGH and the BEMF reading will be done only when PWM is applied on one of these two groups. The REO bit in the MPOL register is used to select the group to be used for BEMF sensing (high side group). It has to be configured whatever the sampling mode.

When enabled by the HZ bit in MCRC register, the current value of the MTIM timer is captured in register MZREG when this event occurs in order to be able to compute the real delay in the delay manager part for hardware commutation but also to be able to simulate zero-crossing events for other steps.

When enabled by the SZ bit set in the MCRC register, a zero-crossing event can also be simulated by comparing the MTIM timer value with the MZREG register. This kind of zero-crossing event is called simulated zero-crossing Z_S.

If both HZ and SZ bits are set in MCRC register, the first event that occurs, triggers a zero-crossing event.

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 (either Z_H or Z_S) sets the ZI bit in the MISR register and generates an interrupt if the ZIM bit is set in the MIMR register.

Caution 1: Due to the alternate automatic capture and compare of the MTIM timer with MZREG register by Z_H and Z_S events, the MZREG register should be manipulated with special care.

Caution 2: Due to the event generation protection in the MZREG, MCOMP and MDREG registers for Soft Event generation, the value written in the MZREG register in simuated zero-crossing mode (SZ=1) is checked by hardware after the D (either D_H or D_S) event. If this value is less than or equal to the MTIM counter value at this moment, the simulated zero-crossing event is generated immediately and the MTIM current value overwrites the value in the MZREG register. See "Built-in Checks and Controls for simulated events" on page 175.

The Z event also triggers some timer/multiplier operations, for more details see Section 10.6.7

MOTOR CONTROLLER (Cont'd)

In conclusion, there are 4 sampling types that are available for Z event detection in sensorless mode.

1. Sampling at the end of the OFF time of the PWM signal at the PWM frequency
2. Sampling, at a programmable frequency independent of the PWM state (during ON time or OFF time of the signal). Sampling is done at f_{SCF}, see Table 82.
3. Sampling during the ON time of the PWM signal by adding a delay at PWM frequency
4. Sampling, at a programmable frequency during the ON time (addition of a programmable delay) of the PWM signal. Sampling is done at f_{SCF}, see Table 82.

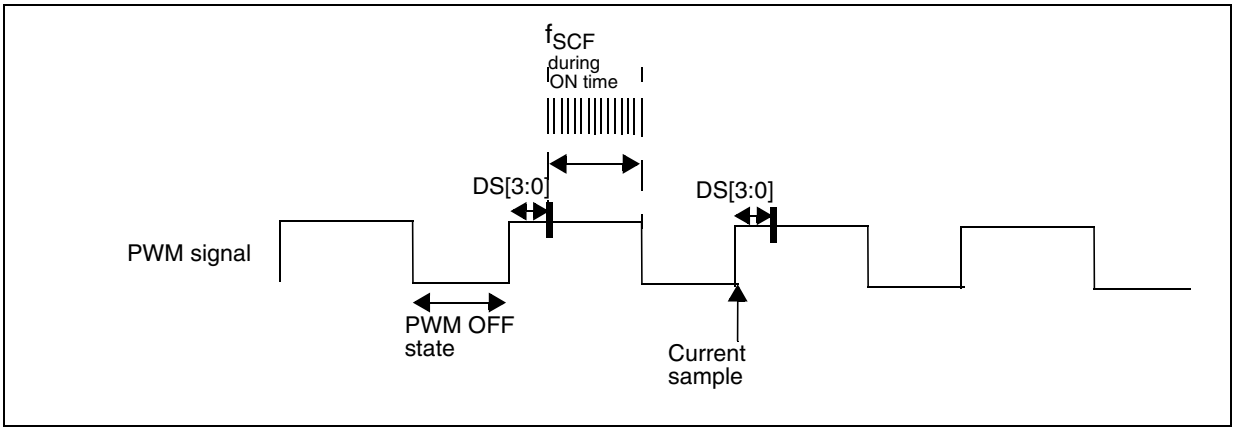
Note 1: The sampling type is applied only for Z event detection after the D event has occured. Whatever the sampling type for Z event detection, the sampling of the signal for D event detection is always done at the selected f_{SCF} frequency (see Table 82), independently of the PWM signal (either during ON or OFF time). Table 34 explains the different sampling types in sensorless and in sensor mode.

Note 2: When the MOE bit in the MCRA register is reset (MCOx outputs in reset state), and the SR bit in the MCRA register is reset (sensorless mode) and the SPLG bit in the MCRC register is reset (sampling at PWM frequency) then, depending on the state of the ZSV bit in the MSCR register, Z event sampling can run or be stopped (and D event is sampled).

Note 3: When BEMF sampling is performed at the end of the PWM signal off-time, the inputs in OFF-state are grounded or put in HiZ as selected by the DISS bit in the MSCR register.

Note 4: The ZEF[3:0] event counter in the MZFR register is active in all configurations.

Figure 86. Sampling during ON time at f_{SCF}



MOTOR CONTROLLER (Cont'd)

Table 40. Step Frequency/Period Range (4MHz)

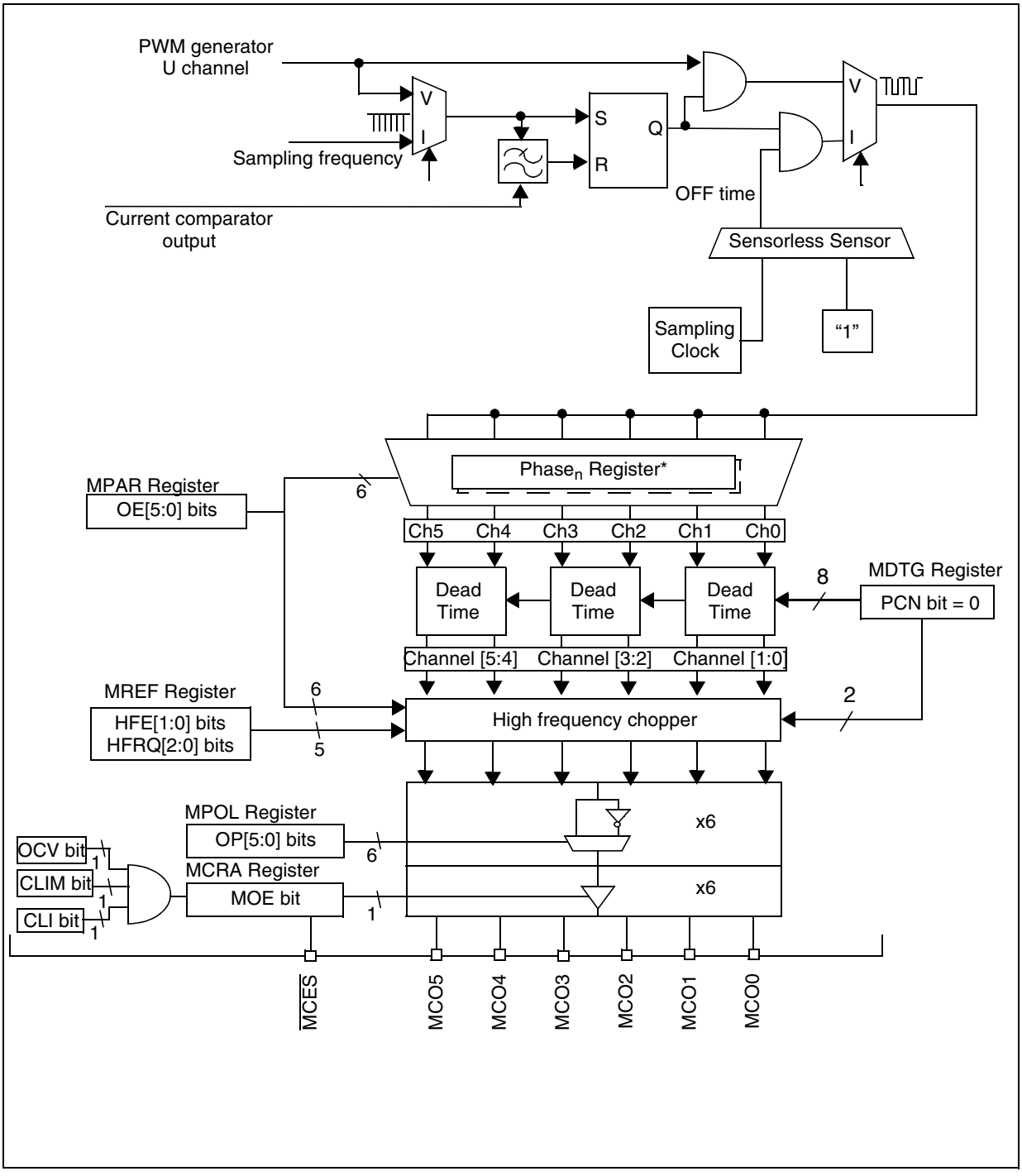
Step Ratio Bits ST[3:0] in MPRSR Register	Maximum Step Frequency	Minimum Step Frequency	Minimum Step Period	Maximum Step Period
0000	23.5 kHz	7.85 kHz	42.5 µs	127.5 µs
0001	11.7 kHz	3.93 kHz	85 µs	255 µs
0010	5.88 kHz	1.96 kHz	170 µs	510 µs
0011	2.94 kHz	980 Hz	340 µs	1.02 ms
0100	1.47 kHz	490 Hz	680 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 41. Modes of Accessing MTIM Timer-Related Registers

State of MCRA / MCRB / MPAR Register Bits					Access to MTIM Timer Related Registers	
RST bit	TES[1:0]	SWA bit	CKE bit	Mode	Read Only Access	Read / Write Access
0	xx	x	0	Configuration Mode		MTIM, MTIML, MZPRV, MZREG, MCOMP, MDREG, ST[3:0]
0	00	0	1	Switched Mode	MTIM, ST[3:0]	MCOMP, MDREG, MZREG, MZPRV 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	00	1	1	Autoswitched Mode	MTIM, ST[3:0]	MDREG, MCOMP, MZREG, MZPRV, RMI, RPI bit of MISR: Set by hardware, (increment ST[3:0]) Cleared by software
0	01 10 11	x	1	Speed Sensor Mode	MTIM, MTIML, ST[3:0]	MDREG,MZREG, MZPRV, RMI, RPI bit of MISR, : Set by hardware, (increment or decrement ST[3:0]), cleared by software.

MOTOR CONTROLLER (Cont'd)

Figure 115. Channel Manager Output Block Diagram with PWM generator delivering 1 PWM signal



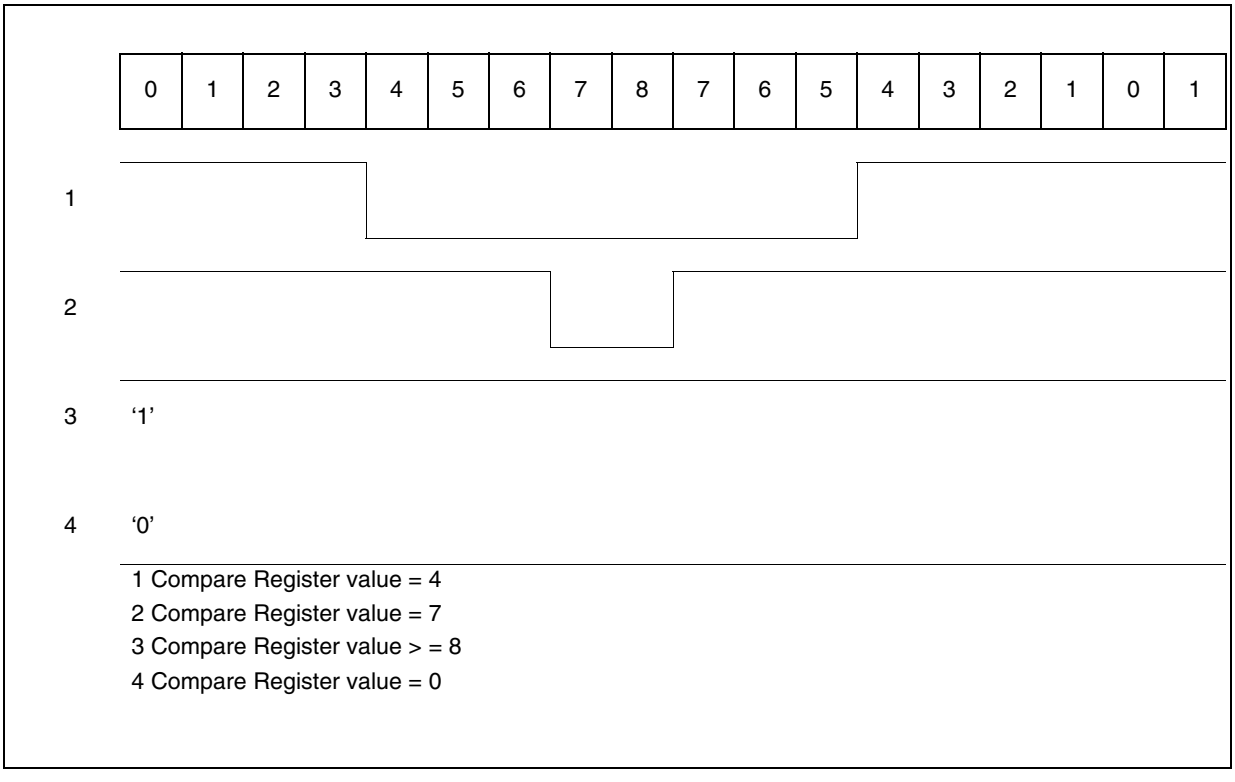
MOTOR CONTROLLER (Cont'd)

If the 13-bit Compare register value is greater than the extended Compare 0 Register (the 13th bit is set to '0'), the corresponding PWM output signal is held at '1'.

If the 13-bit Compare register value is 0, the corresponding PWM output signal is held at '0'.

Figure 119 shows some center-aligned PWM waveforms in an example where the Compare 0 register value = 8.

Figure 119. Center-aligned PWM Waveforms (Compare 0 Register = 8)



MOTOR CONTROLLER (Cont'd)

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

7	6	5	4	3	2	1	0
MOE	CKE	SR	DAC	V0C1	SWA	PZ	DCB

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

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

Notes:

– The reset state is either high impedance, high or low state depending on the corresponding option bit.

– When the MOE bit in the MCRA register is reset (MCOx outputs in reset state), and the SR bit in the MCRA register is reset (sensorless mode) and the SPLG bit in the MCRC register is reset (sampling at PWM frequency) then, depending on the state of the ZSV bit in the MSCR register, Z event sampling can run or be stopped (and D event is sampled).

Bit 6 = **CKE**: *Clock Enable Bit.*
0: Motor Control peripheral Clocks disabled
1: Motor Control peripheral Clocks enabled

Note: Clocks disabled means that all peripheral internal clocks (Delay manager, internal sampling clock, PWM generator) are disabled. Therefore, the peripheral can no longer detect events and the preload registers do not operate.

When Clocks are disabled, write accesses are allowed, so for example, MTIM counter register can be reset by software.

Table 56. Output configuration summary

CKE bit	MOE bit	DAC bit	Peripheral Clock	Effect on MCOx Output
0	0	x	Disabled	Reset state
0	1	0	Disabled	Peripheral frozen (see note 1 below)
0	1	1	Disabled	Direct access via MPHST (only logical level)
1	0	x	Enabled	Reset state
1	1	0	Enabled	Standard running mode.
1	1	1	Enabled	Direct access via MPHST (PWM can be applied)

Note 1: "Peripheral frozen" configuration is not recommended, as the peripheral may be stopped in a unknown state (depending on PWM generator outputs,etc.). It is better practice to exit from run mode by first setting output state (by toggling either MOE or DAC bits) and then to disabling the clock if needed.

Note 2: In Direct Access Mode (DAC=1), when CKE=0 (Peripheral Clock disabled) only logical level can be applied on the MCOx outputs when they are enabled whereas when CKE=1 (Peripheral Clock enabled), a PWM signal can be applied on them. Refer to Table 74, "DeadTime generator set-up," on page 221

Note 3: When clocks are disabled (CKE bit reset) while outputs are enabled (MOE bit set), the effects on the MCOx outputs where PWM signal is applied depend on the running mode selected:

– in voltage mode (VOC1 bit=0), the MCOx outputs where PWM signal is applied stay at level 1.

– in current mode (VOC1 bit=1), the MCOx outputs where PWM signal is applied are put to level 0.

In all cases, MCOx outputs where a level 1 was applied before disabling the clocks stay at level 1. That is why it is recommended to disable the MCOx outputs (reset MOE bit) before disabling the clocks. This will put all the MCOx outputs under reset state defined by the corresponding option bit.

OP-AMP MODULE (Cont'd)

10.7.6 Low power modes

Note: The Op-Amp can be disabled by resetting the OAON bit. This feature allows reduced power consumption when the amplifier is not used.

Mode	Description
Wait	No effect on Op-Amp
Halt	Op-Amp disabled After wake-up from Halt mode, the Op-Amp requires a stabilization time (see Electrical characteristics) (to be defined)

10.7.7 Interrupts

None.

10.7.8 Register Description

CONTROL/STATUS REGISTER (OACSR)

Read/Write (except bit 7 read only)

Reset Value: 0000 0000(00h)

7	6	5	4	3	2	1	0
CMP OVR	OFF CMP	AVG CMP	OAO N	HIGH GAIN	0	0	0

Bit 7 = **CMPOVR** *Compensation Completed*
This read-only bit contains the offset compensation status.
0: No offset compensation if OFFCMP = 0, or Offset compensation cycle not completed if OFFCMP = 1
1: Offset compensation completed if OFFCMP = 1

Bit 6 = **OFFCMP** *Offset Compensation*
0: Reset offset compensation values
1: Request to start offset compensation

Bit 5 = **AVGCMP** *Average Compensation*
0: One-shot offset compensation
1: Average offset compensation over 16 times

Bit 4 = **OAON** *Amplifier On*
0: Op-Amp powered off
1: Op-Amp on

Bit 3 = **HIGHGAIN** *Gain range selection*
This bit must be programmed depending on the application. It can be used to ensure 35dB open loop gain when high, it must be low when the closed loop gain is below 20dB for stability reasons.
0: Closed loop gain up to 20dB
1: Closed loop gain more than 20dB

Bits 2:0 = Reserved, must be kept cleared.

INSTRUCTION SET OVERVIEW (Cont'd)

11.1.1 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 (level 3)
RIM	Reset Interrupt Mask (level 0)
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

11.1.2 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

11.1.3 Direct

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

The direct addressing mode consists of two submodes:

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.

11.1.4 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.

11.1.5 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.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

12.4.5 On-Chip Peripherals

Symbol	Parameter	Conditions		Typ	Unit
$I_{DD(TIM)}$	16-bit Timer supply current [1]	f_{CPU}=8MHz	V_{DD}=5.0V	50	µA
$I_{DD(ART)}$	ART PWM supply current [2]	f_{CPU}=8MHz	V_{DD}=5.0V	75	
$I_{DD(SPI)}$	SPI supply current [3]	f_{CPU}=8MHz	V_{DD}=5.0V	400	
$I_{DD(SCI)}$	SCI supply current [4]	f_{CPU}=8MHz	V_{DD}=5.0V	400	
$I_{DD(MTC)}$	MTC supply current [5]	f_{CPU}=8MHz	V_{DD}=5.0V	500	
$I_{DD(ADC)}$	ADC supply current when converting [6]	f_{ADC}=4MHz	V_{DD}=5.0V	400	
$I_{DD(OPAMP)}$	OPAMP supply current [7]	f_{CPU}=8MHz	V_{DD}=5.0V	1500	

Notes:

1. Data based on a differential I_{DD} measurement between reset configuration (timer counter running at $f_{CPU}/4$) and timer counter stopped (only TIMD bit set). Data valid for one timer.

2. Data based on a differential I_{DD} measurement betwwen reset configuration (timer stopped) and timer counter enable (only TCE bit set)

3. Data based on a differential I_{DD} measurement between reset configuration (SPI disabled) and a permanent SPI master communication at maximum speed (data sent equal to 55h). This measurement includes the pad toggling consumption.

4. Data based on a differential I_{DD} measurement between SCI low power state (SCID=1) and a permanent SCI data transmit sequence.

5. Data based on a differential I_{DD} measurement between reset configuration (motor control disabled) and the whole motor control cell enable in speed measurement mode. MCO outputs are not validated.

6. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

7. Data based on a differential measurement between reset configuration (OPAMP disabled) and amplification of a sinewave (no load, A_{VCL}=1, V_{DD}=5V).

I/O PORT PIN CHARACTERISTICS (Cont'd)

12.8.2 Output Driving Current

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} 1)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 141)	V_{DD}=5V	I_{IO}=+5mA		1.2	V
			I_{IO}=+2mA		0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see Figure 142)		I_{IO}=+20mA, $T_A \leq 85°C$ $T_A \geq 85°C$		1.3 1.5	
			I_{IO}=+8mA		0.6	
V_{OH} 2)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see Figure 143)		I_{IO}=-5mA, $T_A \leq 85°C$ $T_A \geq 85°C$	V_{DD}-1.4 V_{DD}-1.6		
			I_{IO}=-2mA	V_{DD}-0.7		

Figure 141. Typical V_{OL} at V_{DD}=5V (standard)

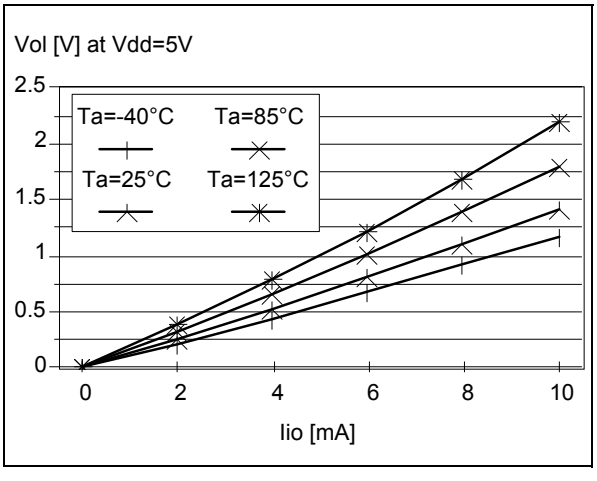


Figure 142. Typical V_{OL} at V_{DD}=5V (high-sink)

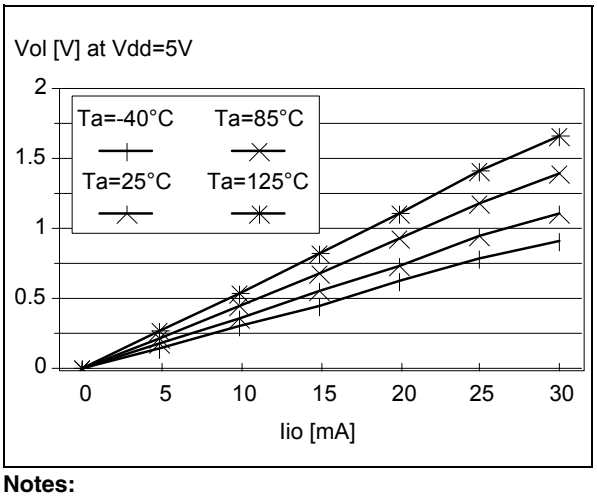
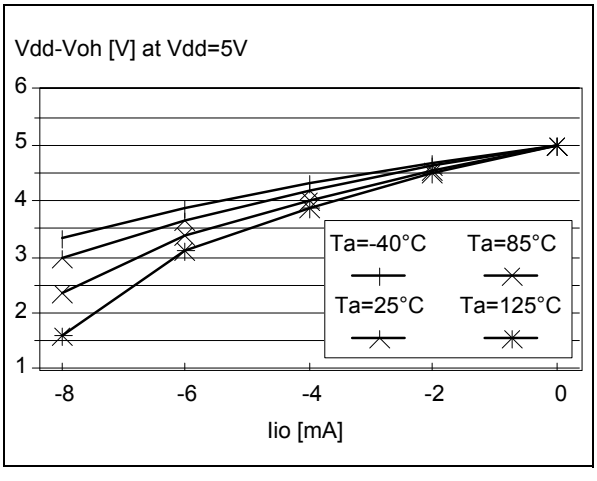


Figure 143. Typical V_{DD}-V_{OH} at V_{DD}=5V



Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
2. The I_{IO} current sourced must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}.

ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

ST7MC MICROCONTROLLER OPTION LIST
(Last update: September 2008)

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference/ROM or FASTROM Code* : . . . . . .
*The ROM or FASTROM code name is assigned by STMicroelectronics.
ROM or FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device type/memory size/package (check only one option):

ROM	8K	16K	32K	48K	60K
LQFP32:	[ ]				
LQFP44:		[ ]			

FASTROM	8K	16K	32K	48K	60K
LQFP32:	[ ]	[ ]			
LQFP44:		[ ]	[ ]		
LQFP64:			[ ]	[ ]	
LQFP80:					[ ]

Conditioning (check only one option):

[ ] Tape & Reel [ ] Tray

Special Marking [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ " (10 char. max)

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Temperature range [ ] - 40°C to + 85°C [ ] - 40°C to + 125°C

MCO (Motor Control Output state under reset)	[ ] Hiz	[ ] Low	[ ] High
DIV2	[ ] Disabled		[ ] Enabled
CKSEL	[ ] Oscillator clock		[ ] PLL clock
Watchdog Selection	[ ] Software Activation		[ ] Hardware Activation
Halt when Watchdog on	[ ] Reset		[ ] No reset
Readout Protection	[ ] Disabled		[ ] Enabled
LVD Reset	[ ] Disabled		[ ] Enabled
AVD Interrupt (if LVD enabled)	[ ] Disabled		[ ] Enabled
Reset Delay	[ ] 256 Cycles		[ ] 4096 Cycles

Supply Operating Range in the application: . . . . . .

Notes . . . . . .

Date . . . . . .

Signature . . . . . .

Please download the latest version of this option list from:
http://www.st.com/mcu > downloads > ST7 microcontrollers > Option list

ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.3 DEVELOPMENT TOOLS

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

14.3.1 Starter kits

ST offers complete, affordable **starter kits** and full-featured that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete hardware/software tool packages that include features and samples to to help you quickly start developing your application.

14.3.2 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16K of code.

The range of hardware tools includes full-featured **ST7-EMU2B series emulators** and the low-cost **RLink** in-circuit debugger/programmer. These tools are supported by the **ST7 Toolset** from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

14.3.3 Programming tools

During the development cycle, the **ST7-EMU3 series emulators** and the **RLink** provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides a low-cost dedicated in-circuit programmer, the **ST7-STICK**, as well as **ST7 Socket Boards** which provide all the sockets required for programming any of the devices in a specific ST7 sub-family on a platform that can be used with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

14.3.4 Order codes for ST7MC development tools

Table 91. Development tool order codes for the ST7MC family

MCU	Starter kit	Emulator	Programming tool
ST7MC1 ST7MC2	ST7MC-KIT/BLDC	ST7MDT50-EMU3	ST7-STICK[1)2)] STX-RLINK[3)]

1. Add suffix /EU, /UK or /US for the power supply for your region

2. Parallel port connection to PC

3. RLink with ST7 tool set

For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.