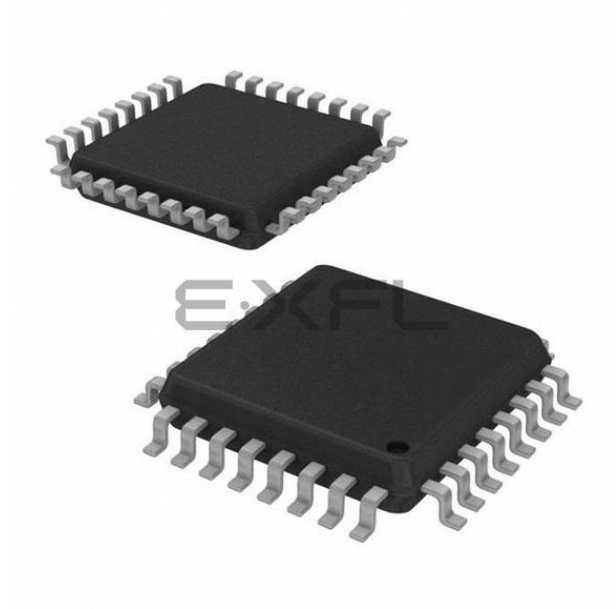


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

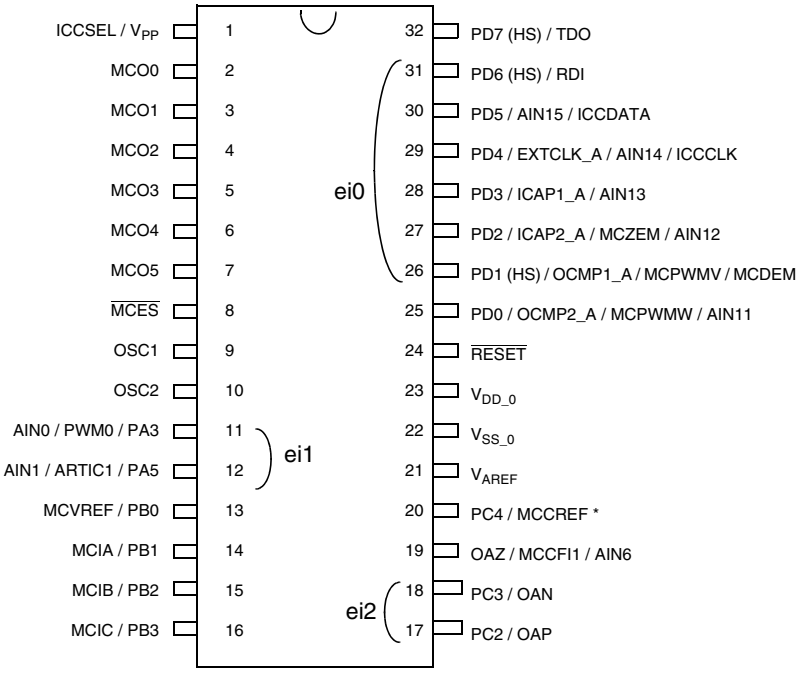
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbusSCI
Peripherals	LVD, Motor Control PWM, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc1k2t6-tr

PIN DESCRIPTION (Cont'd)

Figure 4. 32-Pin SDIP Package Pinouts



(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

6.2 RESET SEQUENCE MANAGER (RSM)

6.2.1 Introduction

The reset sequence manager includes three RESET sources as shown in Figure 15:

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

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to section 11.2.1 on page 244 for further details.

These sources act on the $\overline{\text{RESET}}$ pin and it is always kept low during the delay phase.

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

The basic RESET sequence consists of 3 phases as shown in Figure 14:

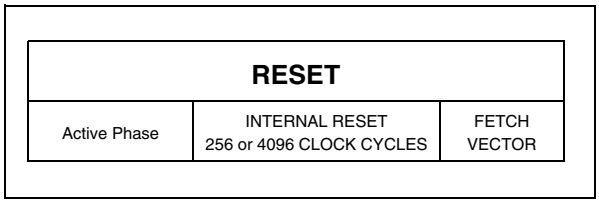
- Active Phase depending on the RESET source
- 256 or 4096 CPU clock cycle delay (selected by option byte)
- RESET vector fetch

Caution: When the ST7 is unprogrammed or fully erased, the Flash is blank and the RESET vector is not programmed. For this reason, it is recommended to keep the RESET pin in low state until programming mode is entered, in order to avoid unwanted behavior.

The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay should be selected by option byte to correspond to the stabilization time of the external oscillator used in the application.

The RESET vector fetch phase duration is 2 clock cycles.

Figure 14. RESET Sequence Phases

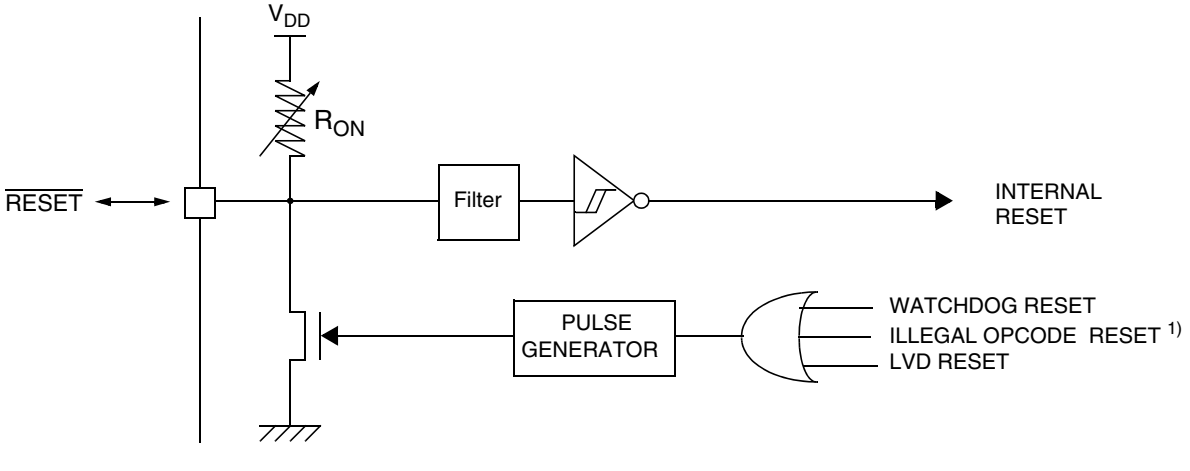


6.2.2 Asynchronous External $\overline{\text{RESET}}$ pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. 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. See Electrical Characteristic section for more details.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see Figure 16). This detection is asynchronous and therefore the MCU can enter reset state even in Halt mode.

Figure 15. Reset Block Diagram



Note 1: See “Illegal Opcode Reset” on page 244. for more details on illegal opcode reset conditions.

6.4 MAIN CLOCK CONTROLLER WITH REAL-TIME CLOCK AND BEEPER (MCC/RTC)

The Main Clock Controller consists of three different functions:

- a programmable CPU clock prescaler
- a clock-out signal to supply external devices
- a real-time clock timer with interrupt capability

Each function can be used independently and simultaneously.

6.4.1 Programmable CPU Clock Prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages Slow power saving mode (See Section 8.2 SLOW MODE for more details).

The prescaler selects the f_{CPU} main clock frequency and is controlled by three bits in the MCCSR register: CP[1:0] and SMS.

6.4.2 Clock-out Capability

The clock-out capability is an alternate function of an I/O port pin that outputs a f_{OSC2} clock to drive external devices. It is controlled by the MCO bit in the MCCSR register.

CAUTION: When selected, the clock out pin suspends the clock during Active-halt mode.

6.4.3 Real-time Clock Timer (RTC)

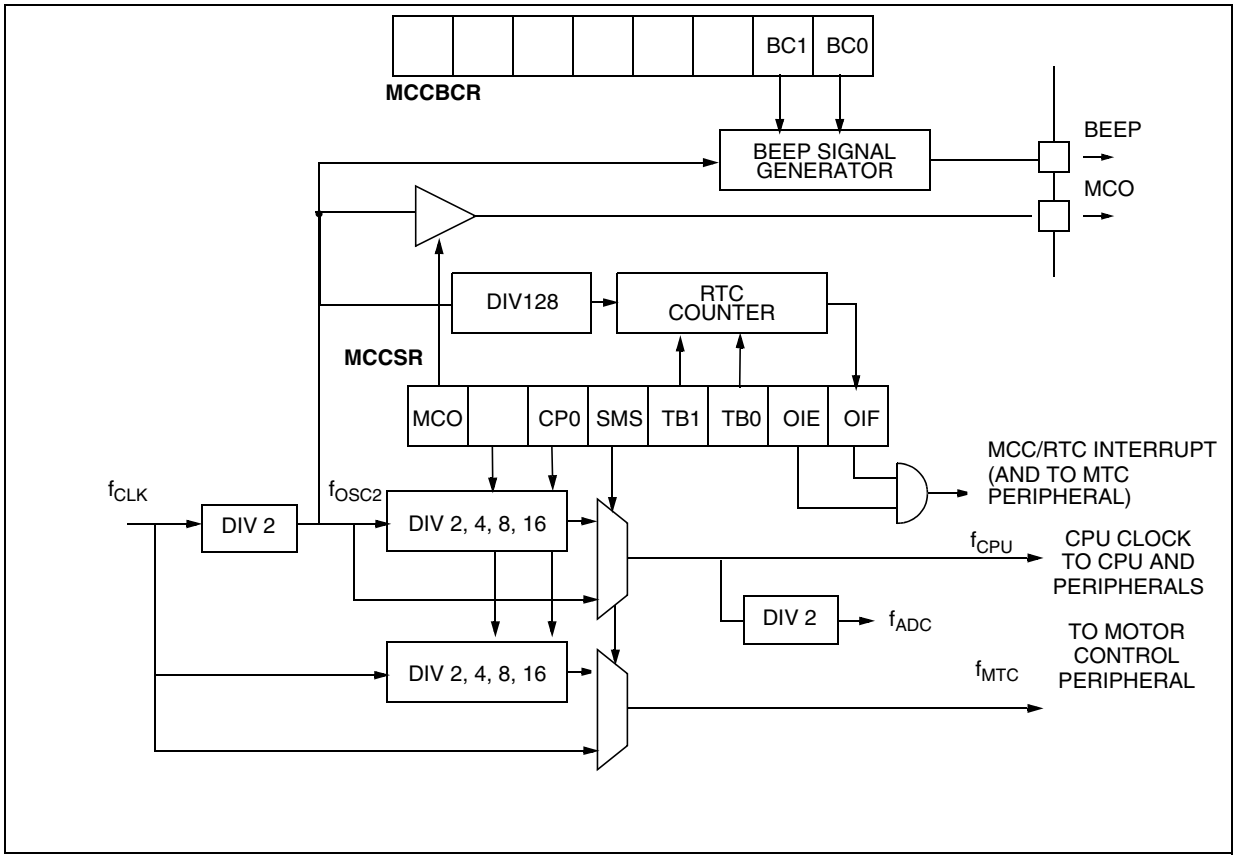
The counter of the real-time clock timer allows an interrupt to be generated based on an accurate real-time clock. Four different time bases depending directly on f_{OSC2} are available. The whole functionality is controlled by four bits of the MCCSR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters Active-halt mode when the HALT instruction is executed. See Section 8.4 ACTIVE-HALT AND HALT MODES for more details.

6.4.4 Beeper

The beep function is controlled by the MCCBCR register. It can output three selectable frequencies on the BEEP pin (I/O port alternate function).

Figure 19. Main Clock Controller (MCC/RTC) Block Diagram



I/O PORTS (Cont'd)

9.5.1 I/O Port Implementation

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

Standard Ports

PA4, PA2:0, PB5:0, PC7:4, PD7:6, PE5:0, PF5:0, PG7:0, PH7:0

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

Interrupt Ports

PA6, PA3, PB6, PC3, PC1, PD5, PD4, PD2 (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

PA7, PA5, PB7, PC2, PC0, PD6, PD3, PD1 (without pull-up)

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

Table 12. Port Configuration

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7, PA5	floating	floating interrupt	open drain	push-pull
	PA6, PA3	floating	pull-up interrupt	open drain	push-pull
	PA2:0	floating	pull-up	open drain	push-pull
Port B	PB7	floating	floating interrupt	open drain	push-pull
	PB6	floating	pull-up interrupt	open drain	push-pull
	PB5:0	floating	pull-up	open drain	push-pull
Port C	PC7:4	floating	pull-up	open drain	push-pull
	PC3, PC1	floating	pull-up interrupt	open drain	push-pull
	PC2, PC0	floating	floating interrupt	open drain	push-pull
Port D	PD7, PD0	floating	pull-up	open drain	push-pull
	PD6, PD3, PD1	floating	floating interrupt	open drain	push-pull
	PD5, PD4, PD2	floating	pull-up interrupt	open drain	push-pull
Port E	PE5:0	floating	pull-up	open drain	push-pull
Port F	PF5:0	floating	pull-up	open drain	push-pull
Port G	PG7:0	floating	pull-up	open drain	push-pull
Port H	PH7:0	floating	pull-up	open drain	push-pull

WINDOW WATCHDOG (Cont'd)

10.1.9 Interrupts

None.

10.1.10 Register Description

CONTROL REGISTER (WDGCR)

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

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

Bits 6:0 = **T[6:0]** *7-bit counter (MSB to LSB).*
These bits contain the value of the watchdog counter. It is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

WINDOW REGISTER (WDGWR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
-	W6	W5	W4	W3	W2	W1	W0

Bit 7 = Reserved

Bits 6:0 = **W[6:0]** *7-bit window value*
These bits contain the window value to be compared to the downcounter.

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

10.5.5.6 Receiver Muting and Wake-up Feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non-addressed receivers.

The non-addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

All the reception status bits can not be set.

All the receive interrupts are inhibited.

A muted receiver may be woken up in one of the following ways:

– by Idle Line detection if the WAKE bit is reset,

– by Address Mark detection if the WAKE bit is set.

Idle Line Detection

Receiver wakes up by Idle Line detection when the Receive line has recognized an Idle Line. Then the RWU bit is reset by hardware but the IDLE bit is not set.

This feature is useful in a multiprocessor system when the first characters of the message determine the address and when each message ends by an idle line: As soon as the line becomes idle, every receivers is waken up and analyse the first characters of the message which indicates the addressed receiver. The receivers which are not addressed set RWU bit to enter in mute mode. Consequently, they will not treat the next characters constituting the next part of the message. At the end of the message, an idle line is sent by the transmitter: this wakes up every receivers which are ready to analyse the addressing characters of the new message.

In such a system, the inter-characters space must be smaller than the idle time.

Address Mark Detection

Receiver wakes up by Address Mark detection when it received a "1" as the most significant bit of a word, thus indicating that the message is an address. The reception of this particular word wakes up the receiver, resets the RWU bit and sets the RDRF bit, which allows the receiver to receive this word normally and to use it as an address word.

This feature is useful in a multiprocessor system when the most significant bit of each character (except for the break character) is reserved for Address Detection. As soon as the receivers received an address character (most significant bit = '1'), the receivers are waken up. The receivers which are not addressed set RWU bit to enter in mute mode. Consequently, they will not treat the next characters constituting the next part of the message.

10.5.5.7 Parity Control

Hardware byte Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the character format defined by the M bit, the possible SCI character formats are as listed in Table 20.

Note: In case of wake-up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Table 20. Character Formats

M bit	PCE bit	Character format
0	0	ǀ SB ǀ 8 bit data ǀ STB ǀ
	1	ǀ SB ǀ 7-bit data ǀ PB ǀ STB ǀ
1	0	ǀ SB ǀ 9-bit data ǀ STB ǀ
	1	ǀ SB ǀ 8-bit data ǀ PB ǀ STB ǀ

Legend: SB = Start Bit, STB = Stop Bit, PB = Parity Bit

Even parity: The parity bit is calculated to obtain an even number of "1s" inside the character made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit will be 0 if even parity is selected (PS bit = 0).

Odd parity: The parity bit is calculated to obtain an odd number of "1s" inside the character made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit will be 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an even number of "1s" if even parity is selected (PS = 0) or an odd number of "1s" if odd parity is selected (PS = 1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PCIE is set in the SCICR1 register.

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

10.5.6 Low Power Modes

Mode	Description
Wait	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
Halt	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

10.5.7 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE		
Received Data Ready to be Read	RDRF	RIE		
Overrun Error or LIN Synch Error Detected	OR/ LHE			
Idle Line Detected	IDLE	ILIE		
Parity Error	PE	PIE		
LIN Header Detection	LHDF	LHIE		

The SCI 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).

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

LIN PRESCALER FRACTION REGISTER (LPFR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	LPFR 3	LPFR 2	LPFR 1	LPFR 0

Bits 7:4 = Reserved.

Bits 3:0 = **LPFR[3:0]** *Fraction of LDIV*

These 4 bits define the fraction of the LIN Divider (LDIV):

LPFR[3:0]	Fraction (LDIV)
0h	0
1h	1/16
...	...
Eh	14/16
Fh	15/16

1. When initializing LDIV, the LPFR register must be written first. Then, the write to the LPR register will effectively update LDIV and so the clock generation.

2. In LIN Slave mode, if the LPR[7:0] register is equal to 00h, the transceiver and receiver input clocks are switched off.

Examples of LDIV coding:

Example 1: LPR = 27d and LPFR = 12d

This leads to:

Mantissa (LDIV) = 27d

Fraction (LDIV) = 12/16 = 0.75d

Therefore LDIV = 27.75d

Example 2: LDIV = 25.62d

This leads to:

LPFR = rounded(16*0.62d)

= rounded(9.92d) = 10d = Ah

LPR = mantissa (25.620d) = 25d = 1Bh

Example 3: LDIV = 25.99d

This leads to:

LPFR = rounded(16*0.99d)

= rounded(15.84d) = 16d

MOTOR CONTROLLER (Cont'd)

10.6.6.8 Position Sensor Mode

In position sensor mode (SR=1 in MCRA register), the rotor position information is given to the peripheral by means of logical data on the three inputs MCIA, MCIB and MCIC (Hall sensors).

For each step one of these three inputs is selected (IS[1:0] bits in register MPHST) in order to detect the Z event. Be careful that the phase comparator is OFF until CKE and /or DAC bits are set in MCRA register.

In sensor mode, Demagnetization and the related features (such as the special PWM configuration, D_S or D_H management, programmable filter) are not available (see Table 32)

Table 32. Demagnetisation access

SR bit MCRA register	Demagnetisation feature availabilty
1	NO
0	YES

In sensor mode configuration the rotor detection doesn't need a particular phase configuration to perform the measurement and a Z event can be read from any detection window. The sampling is done at a selectable frequency (f_{SCF}), see Table 82. This means that Z event position sensoring is more precise than it is in sensorless mode.

There is no minimum off time required for current control PWM in sensor mode so the minimum off time is set automatically to 0µs as soon as the SR bit is set in the MCRA register and a true 100% duty cycle can be set in the PWM compare U register for the PWM generation in voltage mode.

In Sensor mode, the ZEF[3:0] bits in the MZFR register are active and can be used to define the number of consecutive Z samples needed to generate the active event.

Procedure for reading sensor inputs in Direct Access mode: In Direct Access mode, the sensors can be read either when the clock are enabled or disabled (depending on CKE it in MCRA register). To read the sensor data the following steps have to be performed:

1. Select Direct Access Mode (DAC bit in MCRA register)
2. Select the appropriate MCIx input pin by means of the IS[1:0] bits in the MPHST register
3. Read the comparator output (HST bit in the MREF register)

MOTOR CONTROLLER (Cont'd)

the MCOMP and MTIM register is enabled before a write access in the MCOMP register. This means that if the SC bit is set and no write access is done after in the MCOMP register, no C_S commutation event will occur.

In Speed Measurement mode, when using encoder or tachogenerator speed sensors (i.e. both TES[1:0] bits in the MPAR register are not reset

and the input detection block is set-up to process sensor signals), motor speed can be measured but it is not possible drive a motor in six-step mode, either sensored or sensorless.

Speed Measurement mode is useful for motors supplied with 3-phase sinewave-modulated PWM signals:

– AC induction motors,
– Permanent Magnet AC (PMAC) motors (although it needs three position sensors, they can be handled just like tachogenerator signals).

This mode uses only part of the Delay Manager's resources. For more details refer to "Speed Measurement Mode" on page 180.

Table 37. Switched and Autoswitched modes

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

10.6.7.1 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.

At this time all registers with a preload function are loaded (registers marked with (*) in Section 10.6.13). The CI bit of MISR is set and if the CIM bit in the MIMR register is set an interrupt is generated.

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 CKE 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 by hardware. See Table 38.

Only one update per step is allowed, so if both RPI and RMI bits are set together by software, this does not affect the MISR register: the write access to these two bits together is not taken into account and the previous state is kept. This means that if either RPI or RMI bit was set before the write access of both bits at the same time, this bit (RPI or RMI) is kept at 1. If none of them was set before the simultaneous write access, none of them will be set after the write access.

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.

In this mode, if an MTIM overflow occurs, it restarts counting from 0x00h and the OI overflow flag in the MCRC register is set if the TES[1:0] bits = 00.

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

Table 38. Step Update

Mode	TES[1:0]	CKE bit	SWA bit	Clock State	Read	Ratio Increment (Slow Down)	Ratio Decrement (Speed-Up)
x	xx	0	x	Disabled	Always possible	Write the ST[3:0] value directly in the MPRSR register	
Switched	00	1	0	Enabled		Set RPI bit in the MISR register till next commutation	Set RMI bit in the MISR register till next commutation
Autoswitched	00	1	1	Enabled		Automatically updated according to MZREG value	
Speed measure	01 10 11	1	x	Enabled			

MOTOR CONTROLLER (Cont'd)

When using hardware commutation C_H, the sequence of events needed is C_H then D and finally Z events and the value written in the registers are checked at different times.

If SDM bit is set, meaning simulated demagnetisation, a value must be written in the MDREG register to generate the simulated demagnetisation. This value must be written after the C (either C_s or C_H) event preceding the simulated demagnetisation.

If SZ bit is set, meaning simulated zero-crossing event, a value must be written in the MZREG register to generate the simulated zero-crossing. This value must be written after the D event (D_H or D_S) preceding the simulated zero-crossing.

When using simulated commutation (C_S), the result of the 8*8 hardware multiplication of the delay manager is not taken in account and must be overwritten if the SC bit has been set in a Z event interrupt and the sequence of events is broken meaning that several consecutive simulated commutations can be implemented.

As soon as the SC bit is set in the MCRC register, the system won't necessarily expect a D event after a C event. This can be used for an application in sensor mode with only one Hall Effect sensor for example.

Be careful that the D and Z events are not ignored by the peripheral, this means that for example if a Z event occurs, the MTIM timer is reset. In Simulated Commutation mode, the sequence D -> Z is expected, and this order must be repected.

As the sequence of events may not be the same when using simulated commutation, as soon as the SC bit is set, the capture/compare feature and protection on MCOMP register is reestablished only after a write to the MCOMP register. This means that as soon as the SC bit is set, if no write access is done to the MCOMP register, no commutation event will be generated, whatever the value of MCOMP compared to MTIM at the time SC is set. This does not depend on the running mode: switched or autoswitched mode (SWA bit). If software commutation event is used with a normal sequence of events C-->D-->Z, it is recommended to write the MCOMP register during the Z interrupt routine to avoid any spurious comparison as several consecutive C_s events can be generated.

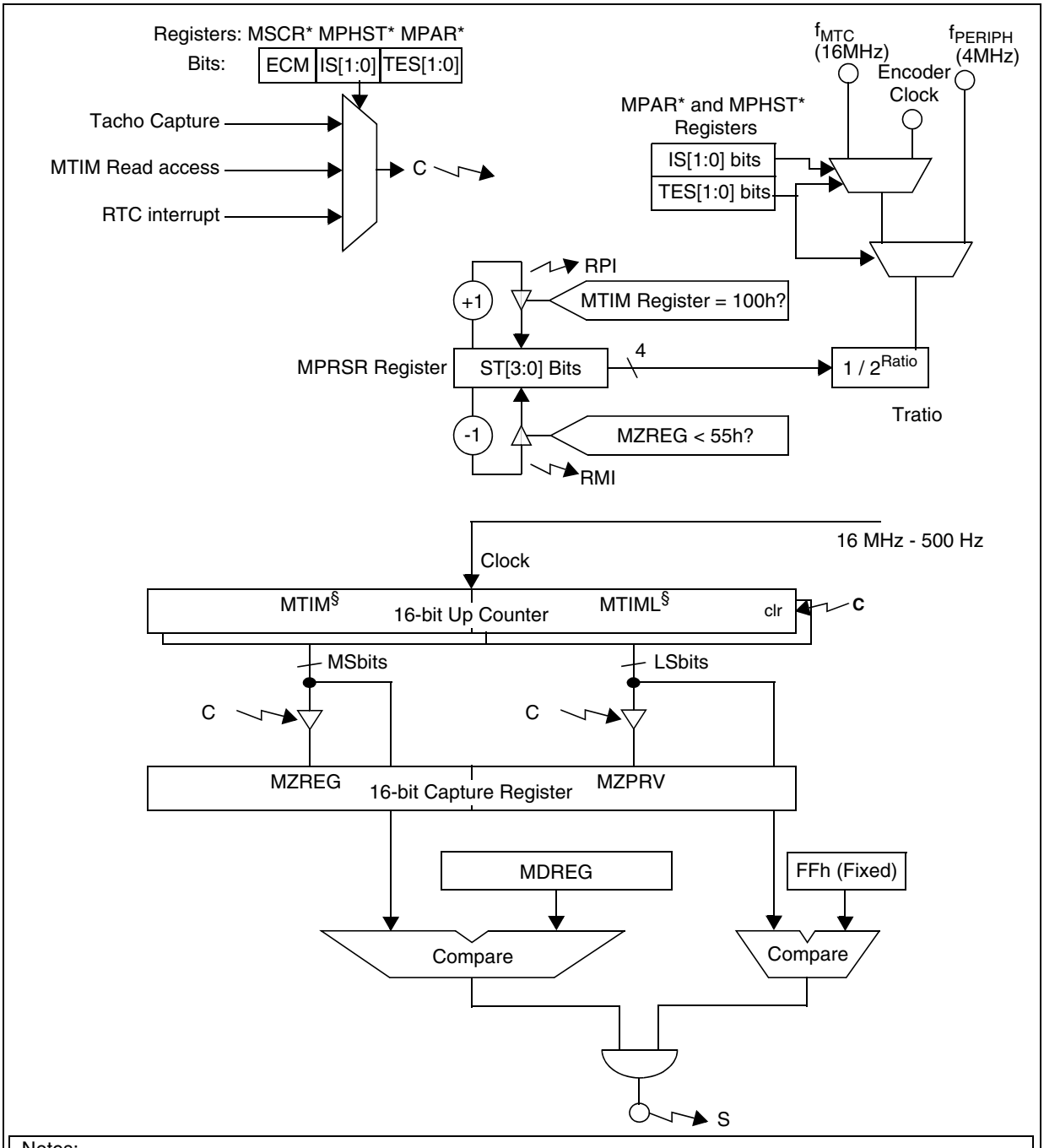
Note that two different simulated events can be used in the same step (like D_S followed by Z_S).

Note also that for more precision, it is recommended to use the value captured from the preceding hardware event to compute the value used to generate simulated events.

Figure 95, Figure 96 and Figure 97 shows details of simulated event generation.

MOTOR CONTROLLER (Cont'd)

Figure 102. Overview of MTIM Timer in Speed Measurement Mode



Notes:

§ = Register updated on R event

* = Register set-up described in Speed Sensor Mode Section

C → To interrupt generator (Capture Event)

S → To interrupt generator (Speed Error Event)

RPI → To interrupt generator (Ratio Increment Event)

RMI → To interrupt generator (Ratio Decrement Event)

MOTOR CONTROLLER (Cont'd)

10.6.10 PWM Generator Block

The PWM generator block produces three independent PWM signals based on a single carrier frequency with individually adjustable duty cycles.

Depending on the motor driving method, one or three of these signals may be redirected to the other functional blocks of the motor control peripheral, using the PCN bit in the MDTG register.

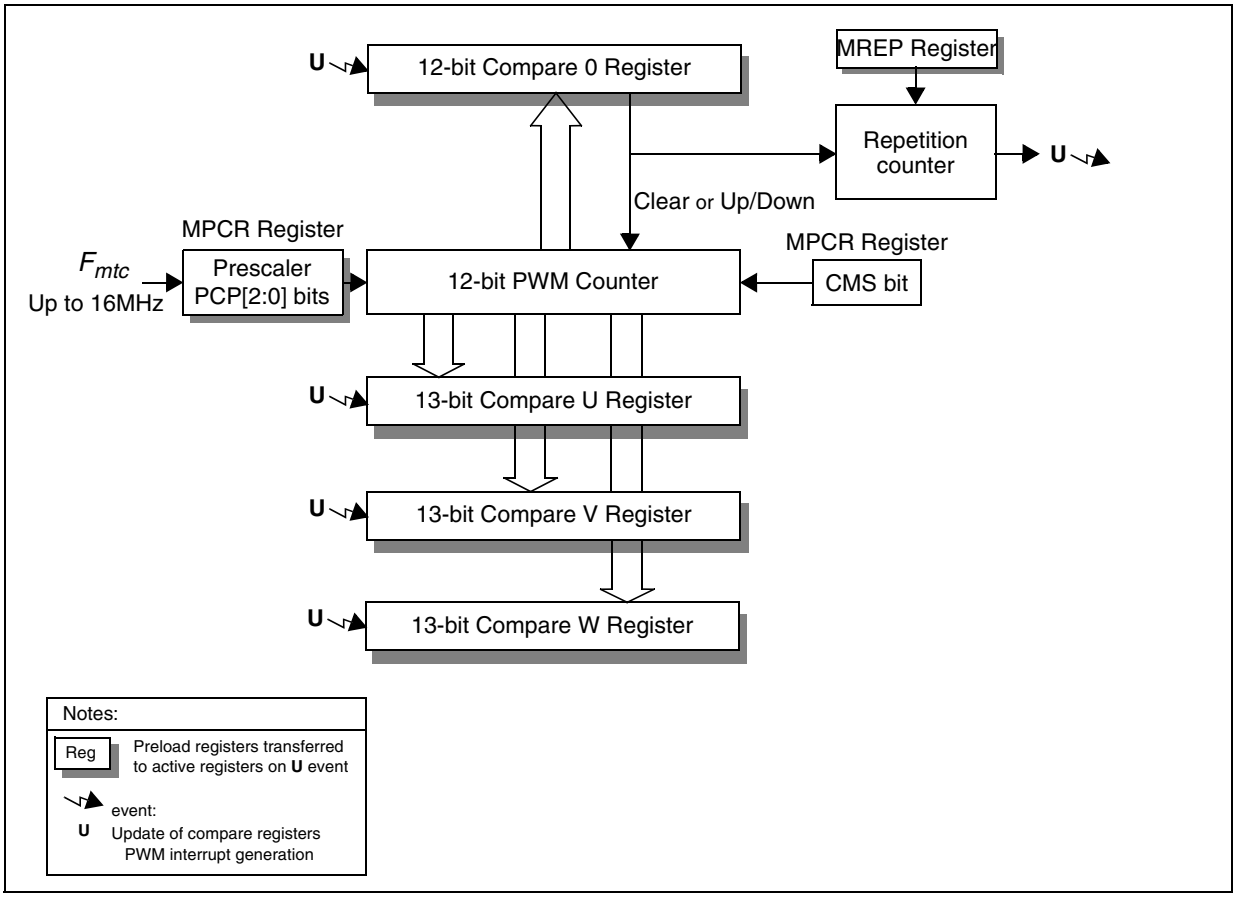
When driving PM BLDC motors in six-step mode (voltage mode only, either sensored or sensorless) a single PWM signal (Phase U) is used to supply the Input Stage, PWM and Channel Manager blocks according to the selected modes.

For other kind of motors requiring independent PWM control for each of the three phases, all PWM signals (Phases U, V and W) are directed to the channel manager, in which deadtime or a high frequency carrier may be added. This is the case of AC induction motors or PMAC motors for instance, supplied with 120° shifted sinewaves in voltage mode.

10.6.10.1 Main Features

- 12-bit PWM free-running Up/Down Counter with up to 16MHz input clock (F_{mtc}).
- Edge-aligned and center-aligned PWM operating modes
- Possibility to re-load compare registers twice per PWM period in center-aligned mode
- Full-scale PWM generation
- PWM update interrupt generation
- 8-bit repetition counter
- 8-bit PWM mode
- Timer re-synchronisation feature

Figure 117. PWM generator block diagram



MOTOR CONTROLLER (Cont'd)

A_N WEIGHT REGISTER (MWGHT)

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

7	6	5	4	3	2	1	0
AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

Bits 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:

$$\frac{Z_n \times MWGHT}{256(d)} \quad \text{or} \quad \frac{Z_{N-1} \times MWGHT}{256(d)} \quad (*)$$

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	6	5	4	3	2	1	0
SA3	SA2	SA1	SA0	ST3	ST2	ST1	ST0

Bits 7:4 = **SA[3:0]**: *Sampling Ratio.*
These bits contain the sampling ratio value for current mode. Refer to Table 47, "Sampling Frequency Selection," on page 189.

Bits 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 40, "Step Frequency/Period Range (4MHz)," on page 179 and Table 41, "Modes of Accessing MTIM Timer-Related Registers," on page 179.

INTERRUPT MASK REGISTER (MIMR)

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

7	6	5	4	3	2	1	0
PUM	SEM	RIM	CLIM	EIM	ZIM	DIM	CIM

Bit 7 = **PUM:** *PWM Update Mask bit.*
0: PWM Update interrupt disabled
1: PWM Update interrupt enabled

Bit 6 = **SEM:** Speed Error Mask bit.
0: Speed Error interrupt disabled
1: Speed Error interrupt enabled

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 = **CLIM**: *Current Limitation Interrupt Mask bit.*
0: Current Limitation interrupt disabled
1: Current Limitation interrupt enabled

This interrupt is available only in Voltage Mode (VOC1 bit=0 in MCRA register) and occurs when the Motor current feedback reaches the external current limitation value.

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 / Capture Interrupt Mask bit*
0: Commutation / Capture Interrupt disabled
1: Commutation / Capture Interrupt enabled

MOTOR CONTROLLER (Cont'd)

PWM CONTROL REGISTER (MPCR)
Read/Write
Reset Value: 0000 0000 (00h)

7							0
PMS	OVFU	OVFV	OVFW	CMS	PCP2	PCP1	PCP0

Bit 7 = **PMS:** *PWM Mode Selection*.
0: Standard mode: bit b7 in the MCPxH register represents the extension bit.
1: "8-bit" mode: bit b7 (extension bit) in the MCPxH register is located in the MPCR register (OVFx bits); the number of active bits in MCPxH and MCPxL is decreased to b15:b8 instead of b15:b3.

Bit 6 = **OVFU:** *Phase U 100% duty cycle Selection.*
0: Duty cycle defined by MCPUH:MCPUL register.
1: Duty cycle set at 100% on phase U at next update event and maintained till the next one. This bit is reset once transferred to the active register on update event.

Bit 5 = **OVFV:** *Phase V 100% duty cycle Selection.*
0: Duty cycle defined by MCPVH:MCPVL register.
1: Duty cycle set at 100% on phase V at next update event and maintained till the next one. This bit is reset once transferred to the active register on update event.

Bit 4 = **OVFW:** *Phase W 100% duty cycle Selection.*
0: Duty cycle defined by MCPWH:MCPWL register.
1: Duty cycle set at 100% on phase W at next update event and maintained till the next one. This bit is reset once transferred to the active register on update event.

Bit 3 = **CMS:** *PWM Counter Mode Selection.*
0: Edge-aligned mode
1: Center-aligned mode

Bits 2:0 = **PCP[2:0]** *PWM counter prescaler value*.

This value divides the F_{mtc} frequency by N, where N is PCP[2:0] value. Table 73 shows the resulting frequency of the PWM counter input clock.

Table 73. PWM clock prescaler

PCP2	PCP1	PCP0	PWM counter input clock
0	0	0	F_{mtc}
0	0	1	$F_{mtc}/2$
0	1	0	$F_{mtc}/3$
0	1	1	$F_{mtc}/4$
1	0	0	$F_{mtc}/5$
1	0	1	$F_{mtc}/6$
1	1	0	$F_{mtc}/7$
1	1	1	$F_{mtc}/8$

MOTOR CONTROLLER (Cont'd)

DEAD TIME GENERATOR REGISTER (MDTG)
Read/Write (except bits 5:0 write once-only)
Reset Value: 1111 1111 (FFh)

7							0
PCN	DTE	DTG5	DTG4	DTG3	DTG2	DTG1	DTG0

Bit 7 = **PCN:** *Number* of *PWM Channels* .
0: Only PWM U signal is output to the PWM manager for six-step mode motor control (e.g. PM BLDC motors)
1: The three PWM signals U, V and W are output to the channel manager (e.g. for three-phase sinewave generation)

Bit 6 = **DTE***: Dead Time Generator Enable
0: Disable the Dead Time generator
1: Enable the Dead Time generator and apply complementary PWM signal to the adjacent switch
* write once-only bit if PCN bit is set, read/write if PCN bit is reset. To clear the DTE bit if PCN=1, it is mandatory to clear the PCN bit first.

Table 74. DeadTime generator set-up

DAC	PCN bit in MDTG register	DTE bit in MDTG register	Complementary PWM applied to adjacent switch
0	0	0	NO
0	0	1	YES
0	1	1	YES
0	1	0	YES, but WITHOUT deadtime
1	0	0	NO Complementary PWM
1	0	1	YES
1	1	1	YES
1	1	0	YES, but WITHOUT deadtime

Note 1: This table is true on condition that the CKE bit is set (Peripheral clock enabled) and the MOE bit is set (MCOx outputs enabled). See Table 56, "Output configuration summary," on page 210

When the PCN bit is reset (e.g. for PM BLDC motors), in Direct Access mode (DAC=1), if the DTE bit is reset, PWM signals can be applied on the MCOx outputs but not complementary PWM. Of course, logical levels can be also applied on the outputs.

If the DTE bit is set (PCN=0 and DAC=1), channels are paired and complementary PWM signals can be output on the MCOx pins. This will follow the rules detailed in Table 53, "Dead Time generator outputs," on page 197 as the channels are grouped in pairs.

In this case, the PWM application is selected by the OS0 bit in the MCRB register.

It is also possible to add a chopper on the PWM signal output using bits HFE[1:0] and HFRQ[2:0] in the MREF register.

Caution 1: The PWM mode will be selected via the 00[5:0] bits in the MPHST register, the OE[5:0] bits in the MPAR register and the OS2 and OS0 bits in the MCRB register as shown in Table 62, "PWM mode when SR=1," on page 213.

Caution 2: When driving motors with three independent pairs of complementary PWM signals (PCN=1), disabling the deadtime generator (DTE=0) causes the deadtime to be null: high and low side signals are exactly complemented.

It is therefore recommended not to disable the deadtime generator (it may damage the power stage), unless deadtimes are inserted externally.

Bits 5:0 = **DTG[5:0]*** *Dead time generator set-up*.

These bits set-up the deadtime duration and resolution. Refer to Table 52, "Dead time programming and example," on page 195 for details.

With F_{mtc} = 16MHz dead time values range from 125ns to 16µs with steps of 125ns, 250ns and 500ns.

* Write-once bits; once write-accessed these bits cannot be re-written unless the processor is reset (See "Caution: Access to write-once bits" on page 220.).

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)

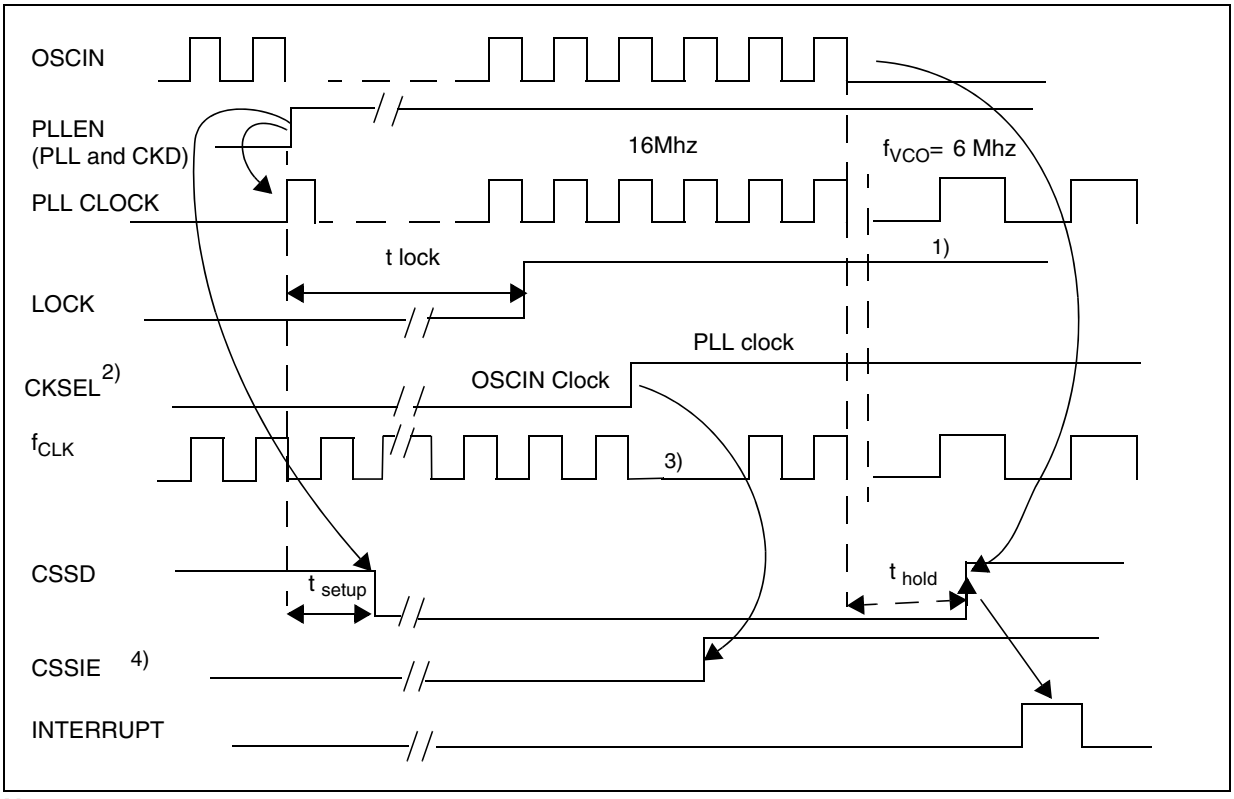
The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

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.

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

Figure 137. PLL And Clock Detector Signal Start Up Sequence



Notes:

1. Lock does not go low without resetting the PLLEN bit.

2. Before setting the CKSEL bit by software in order to switch to the PLL clock, a period of t_{lock} must have elapsed.

3. 2 clock cycles are missing after CKSEL = 1

4. CKSEL bit must be set before enabling the CSS interrupt (CSSIE=1).

12.13 OPERATIONAL AMPLIFIER CHARACTERISTICS

Subject to general operating conditions for f_{OSC}, and T_A unless otherwise specified.
(T_A = -40..+125°C, V_{DD}-V_{SSA} = 4.5..5.5V unless otherwise specified

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
R_L	Resistive Load (max 500µA @ 5V)			10			kΩ
C_L	Capacitive Load at V_{OUT} pin					150	pF
V_{CMIR}	Common Mode Input Range			V_{SSA}		V_{DD}/2	V
V_{io}	Input Offset Voltage (+ or -) 3)	After calibration, V_{IC} =1V			2.5	10 4)	mV
ΔV_{io}	Input Offset Voltage Drift from the calibrated Voltage, temperature conditions	with respect to temperature				8.5 5)	µV/°C
		with respect to common mode input				1 5)	mV/V
		with respect to supply				3.1 5)	mV/V
CMR	Common Mode Rejection Ratio	HIGHGAIN=0 @ 100kHz			74		dB
SVR	Supply Voltage Rejection Ratio	@ 100kHz		50 2)	65		dB
A_{vd}	Voltage Gain	R_L=10kΩ		(1.5) 2)	12		V/mV
V_{SAT_OH}	High Level Output Saturation Voltage (V_{DD}-V_{OUT})	R_L=10kΩ			60	90 2)	mV
V_{SAT_OL}	Low Level Output Saturation Voltage	R_L=10kΩ			30	90 2)	mV
GBP	Gain Bandwidth Product	V_{OAP} = V_{DD} / 2, measured at 0dB gain, feedback resistor ratio = 10 (20dB inverting configuration)	HIGHGAIN=0	1.3 2)	1.5	2 2)	MHz
			HIGHGAIN=1	2.6 2)	3	4 2)	
SR^+	Slew Rate while rising	HIGHGAIN=0 (A_{VCL}=1, R_L=10kΩ, C_L=150pF, V_i=1.75V to 2.75V) 1)		1 2)	2		V/µs
SR^-	Slew Rate while falling	HIGHGAIN=0 (A_{VCL}=1, R_L=10kΩ, C_L=150pF, V_i=1.75V to 2.75V) 1)		2.5 2)	7.5		V/µs
Φm	Phase Margin	HIGHGAIN=0			73		degrees
		HIGHGAIN=1			75		
T_{wakeup}	Wakeup time for the opamp from off state			0.8 6)		1.6 6)	µs

Note:

1. A_{VCL} = closed loop gain
2. Data based on characterization results, not tested in production.
3. after offset compensation has been performed.
4. The amplifier accuracy is dependent on the environment. The offset value is given for a measurement done with all digital I/Os stable. Negative injection current on the I/Os close to the inputs may reduce the accuracy. In particular care must be taken to avoid switching on I/Os close to the inputs when the opamp is in use. This phenomenon is even more critical when a big external serial resistor is added on the inputs.
5. The Data provided from simulations (not tested in production) to guide the user when re-calibration is needed.
6. The Data provided from simulations (not tested in production).

ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

OPTION BYTE 1

OPT7:5= **PKG[2:0]** *package selection*
These option bits are used to select the device package.

Selected Package	PKG2	PKG1	PKG0
LQFP32 / SDIP32	0	0	0
LQFP44	0	0	1
SDIP 56	0	1	0
LQFP64	0	1	1
LQFP80	1	x	x

OPT4:2= Reserved

OPT1:0 = **MCO** *Motor Control Output Options*
MCO port under reset.

Motor Control Output	bit 1	bit 0
HiZ	0	0
Low	0	1
High	1	0
HiZ	1	1