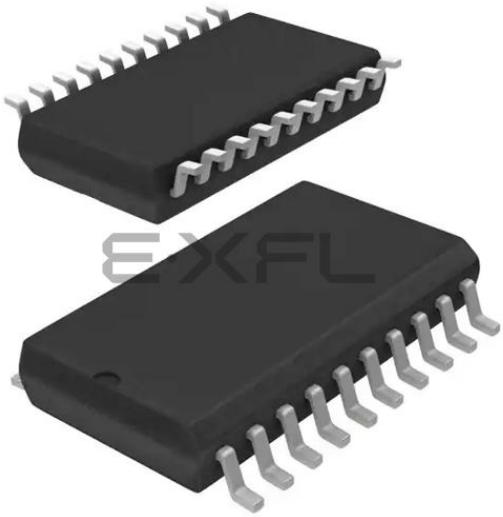


Welcome to E-XFL.COM

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	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flit10bf0m6

Table of Contents

PIN DESCRIPTION (Cont'd)

Legend / Abbreviations for Table 1**:**

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

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

Output level: HS = 20mA high sink (on N-buffer only)

Port and control configuration:

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

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

The RESET configuration of each pin is shown in bold which is valid as long as the device is in reset state.

Table 1. Device Pin Description

Pin No.			Pin Name	Type	Level		Port / Control						Main Function (after reset)	Alternate Function
							Input				Output			
SO20/DPI20	QFN20	SO16/DIP16			Input	Output	float	wpu	int	ana	OD	PP		
1	19	1	V_{SS} 1)	S									Ground	
2	20	2	V_{DD} 1)	S									Main power supply	
3	1	3	$\overline{RESET}$	I/O	C_T			**X**			X		Top priority non maskable interrupt (active low)	
4	2	4	PB0/COMPIN+/ AIN0/$\overline{SS}$	I/O	C_T		**X**	ei3		X	X	X	**Port B0**	ADC Analog Input 0 2) or SPI Slave Select (active low) or Analog Comparator Input **Caution:** No negative current injection allowed on this pin.
5	3	5	PB1/AIN1/SCK	I/O	C_T		**X**			X	X	X	**Port B1**	ADC Analog Input 1 2) or SPI Serial Clock
6	4	6	PB2/AIN2/MISO	I/O	C_T		**X**			X	X	X	**Port B2**	ADC Analog Input 2 2) or SPI Master In/ Slave Out Data
7	5	7	PB3/AIN3/MOSI	I/O	C_T		**X**	ei2		X	X	X	**Port B3**	ADC Analog Input 3 2) or SPI Master Out / Slave In Data
8	6	8	PB4/AIN4/CLKIN/ COMPIN-	I/O	C_T		**X**			X	X	X	**Port B4**	ADC Analog Input 4 2) or External clock input or Analog Comparator External Reference Input
9	7	-	PB5/AIN5	I/O	C_T		**X**			X	X	X	**Port B5**	ADC Analog Input 5 2)
10	8	-	PB6/AIN6	I/O	C_T		**X**			X	X	X	**Port B6**	ADC Analog Input 6 2)
11	9	9	PA7/COMPOUT	I/O	C_T	HS	**X**	ei1			X	X	**Port A7**	Analog Comparator Output

Address	Block	Register Label	Register Name	Reset Status	Remarks
0002Fh	FLASH	FCSR	Flash Control/Status Register	00h	R/W
00030h	EEPROM	EECSR	Data EEPROM Control/Status Register	00h	R/W
0031h 0032h 0033h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control Status Register	xxh 0xh 00h	R/W R/W R/W
0034h 0035h 0036h	ADC	ADCCSR ADCDRH ADCDRL	A/D Control Status Register A/D Data Register High A/D Amplifier Control/Data Low Register	00h xxh 0xh	R/W Read Only R/W
0037h	ITC	EICR	External Interrupt Control Register	00h	R/W
0038h	MCC	MCCSR	Main Clock Control/Status Register	00h	R/W
0039h 003Ah	Clock and Reset	RCCR SICSR	RC oscillator Control Register System Integrity Control/Status Register	FFh 0110 0xx0b	R/W R/W
003Bh	PLL clock select	PLLTST	PLL test register	00h	R/W
003Ch	ITC	EISR	External Interrupt Selection Register	0Ch	R/W
003Dh to 0048h	Reserved area (12 bytes)				
0049h 004Ah	AWU	AWUPR AWUCSR	AWU Prescaler Register AWU Control/Status Register	FFh 00h	R/W R/W
004Bh 004Ch 004Dh 004Eh 004Fh 0050h 0051h	DM[3]	DMCR DMSR DMBK1H DMBK1L DMBK2H DMBK2L DMCR2	DM Control Register DM Status Register DM Breakpoint Register 1 High DM Breakpoint Register 1 Low DM Breakpoint Register 2 High DM Breakpoint Register 2 Low DM Control Register 2	00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W
0052h to 007Fh	Reserved area (46 bytes)				

Legend: x=undefined, R/W=read/write

Notes:

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.

2. The bits associated with unavailable pins must always keep their reset value.

3. For a description of the Debug Module registers, see ICC protocol reference manual.

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 interruptible 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 (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.

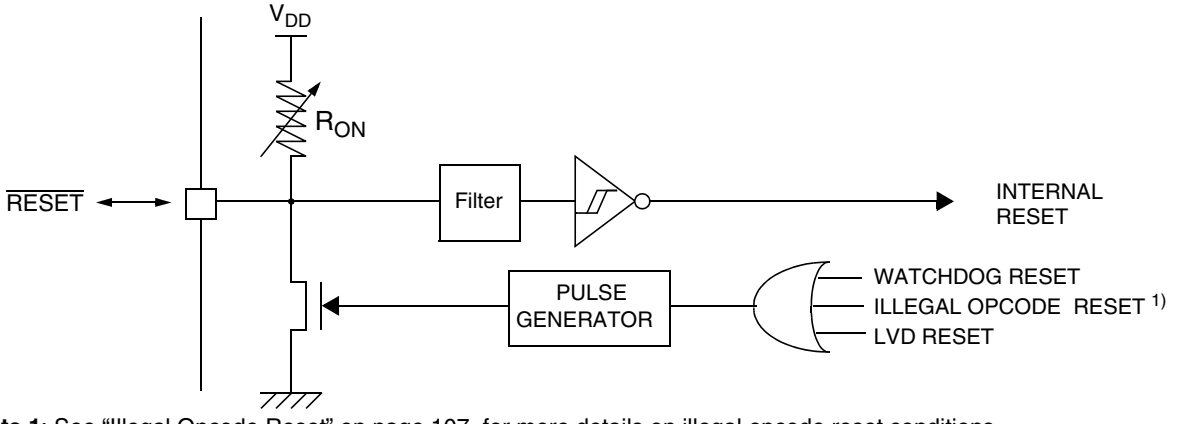
This bit is accessed by the JREQ and JRNE test instructions.

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.

Figure 16. Reset Block Diagram



Note 1: See “Illegal Opcode Reset” on page 107. for more details on illegal opcode reset conditions.

SYSTEM INTEGRITY MANAGEMENT (Cont'd)

7.6.4 Register Description

SYSTEM INTEGRITY (SI) CONTROL/STATUS REGISTER (SICSR)

Read/Write

Reset Value: 0110 0xx0 (6xh)

7							0
LOCK 32	CR1	CR0	WDG RF	LOCKED	LVDRF	AVDF	AVDIE

Bit 7 = **LOCK32** *PLL 32Mhz Locked Flag*

This bit is set and cleared by hardware. It is set automatically when the PLL 32Mhz reaches its operating frequency

0: PLL32 not locked

1: PLL32 locked

Bits 6:5 = **CR[1:0]** *RC Oscillator Frequency Adjustment bits*

These bits, as well as CR[9:2] bits in the RCCR register must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. Refer to section 7.3 on page 25.

Bit 4 = **WDGRF** *Watchdog Reset flag*
This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (reading the SICSR register or writing 0 to this bit) or by an LVD Reset (to ensure a stable cleared state of the WDGRF flag when the CPU starts).
Combined with the LVDRF flag information, the flag description is given by the following table.

RESET Sources	LVDRF	WDGRF
External $\overline{\text{RESET}}$ pin	0	0
Watchdog	0	1
LVD	1	X

Bit 3 = **LOCKED** *PLL Locked Flag*
This bit is set and cleared by hardware. It is set automatically when the PLL reaches its operating frequency.
0: PLL not locked
1: PLL locked

Bit 2 = **LVDRF** *LVD reset flag*
This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (by reading). When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.

Bit 1 = **AVDF** *Voltage Detector Flag*
This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit is set. Refer to Figure 20 and to Section 7.6.2.1 for additional details.
0: V_{DD} over AVD threshold
1: V_{DD} under AVD threshold

Bit 0 = **AVDIE** *Voltage Detector Interrupt Enable*
This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag is set. The pending interrupt information is automatically cleared when software enters the AVD interrupt routine.
0: AVD interrupt disabled
1: AVD interrupt enabled

Application notes

The LVDRF flag is not cleared when another RESET type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.
In this case, a watchdog reset can be detected by software while an external reset can not.

PLL TEST REGISTER (PLLTST)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
PLLdiv2	0	0	0	0	0	0	0

Bit 7 : **PLLdiv2** *PLL clock divide by 2*

This bit is read or write by software and cleared by hardware after reset. This bit will divide the PLL output clock by 2.
0 : PLL output clock
1 : Divide by 2 of PLL output clock

Refer "Clock Management Block Diagram" on page 26

Note : Write of this bit will be effective after 2 Tcpu cycles (if system clock is 8mhz) else 1 cycle (if system clock is 4mhz) i.e. effective time is 250ns.

Bit 6:0 : Reserved , Must always be cleared

INTERRUPTS (cont'd)

Figure 21. Interrupt Processing Flowchart

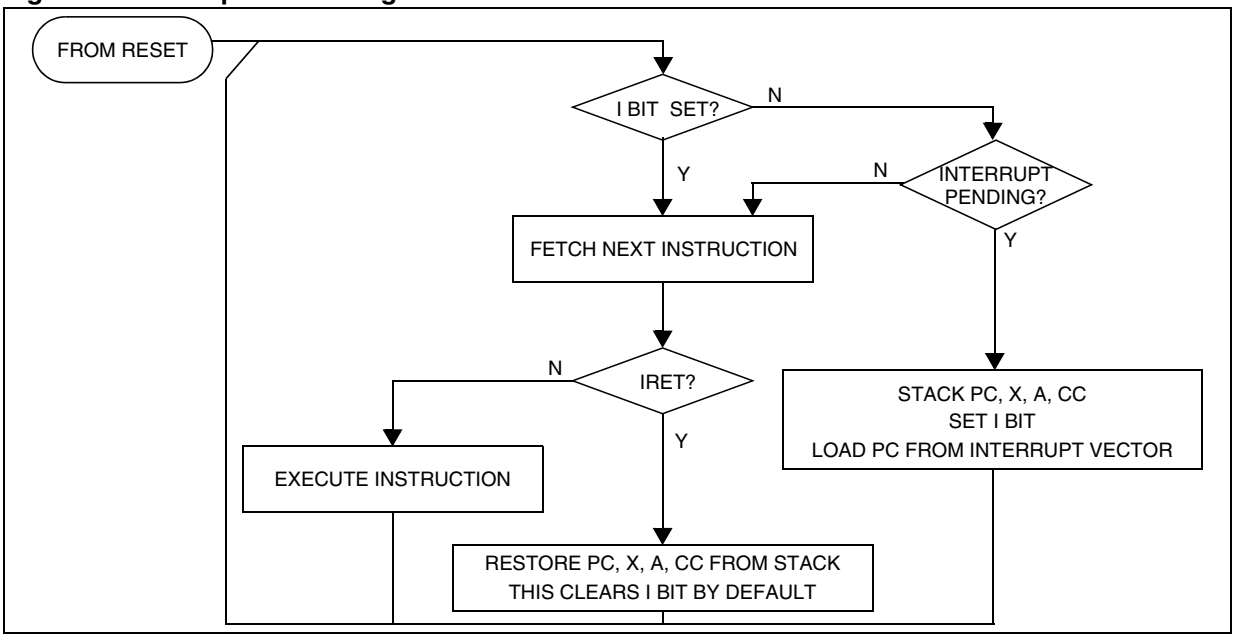


Table 5. Interrupt Mapping

N°	Source Block	Description	Register Label	Priority Order	Exit from HALT or AWUFH	Address Vector
	RESET	Reset	N/A	Highest Priority	yes	FFFEh-FFFFh
	TRAP	Software Interrupt			no	FFFCh-FFFDh
0	AWU	Auto Wake Up Interrupt	AWUCSR		yes[1)]	FFFAh-FFFBh
1	ei0	External Interrupt 0	N/A		yes	FFF8h-FFF9h
2	ei1	External Interrupt 1				FFF6h-FFF7h
3	ei2	External Interrupt 2				FFF4h-FFF5h
4	ei3	External Interrupt 3				FFF2h-FFF3h
5	LITE TIMER	LITE TIMER RTC2 interrupt	LTCSR2		no	FFF0h-FFF1h
6	Comparator	Comparator Interrupt	CMPCR		no	FFEEh-FFEFh
7	SI	AVD interrupt	SICSR		no	FFECh-FFEDh
8	AT TIMER	AT TIMER Output Compare Interrupt or Input Capture Interrupt	PWMxCSR or ATCSR		no	FFEAh-FFEBh
9		AT TIMER Overflow Interrupt	ATCSR		yes[2)]	FFE8h-FFE9h
10	LITE TIMER	LITE TIMER Input Capture Interrupt	LTCSR		no	FFE6h-FFE7h
11		LITE TIMER RTC1 Interrupt	LTCSR		yes[2)]	FFE4h-FFE5h
12	SPI	SPI Peripheral Interrupts	SPICSR	Lowest Priority	yes	FFE2h-FFE3h
13	AT TIMER	AT TIMER Overflow Interrupt	ATCSR2		no	FFE0h-FFE1h

Note 1: This interrupt exits the MCU from "Auto Wake-up from Halt" mode only.
Note 2 : These interrupts exit the MCU from "ACTIVE-HALT" mode only.

INTERRUPTS (Cont'd)

EXTERNAL INTERRUPT CONTROL REGISTER (EICR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
IS31	IS30	IS21	IS20	IS11	IS10	IS01	IS00

Bits 7:6 = **IS3[1:0]** *ei3 sensitivity*
These bits define the interrupt sensitivity for ei3 (Port B0) according to Table 6.

Bits 5:4 = **IS2[1:0]** *ei2 sensitivity*
These bits define the interrupt sensitivity for ei2 (Port B3) according to Table 6.

Bits 3:2 = **IS1[1:0]** *ei1 sensitivity*
These bits define the interrupt sensitivity for ei1 (Port A7) according to Table 6.

Bits 1:0 = **IS0[1:0]** *ei0 sensitivity*
These bits define the interrupt sensitivity for ei0 (Port A0) according to Table 6.

Notes:

1. These 8 bits can be written only when the I bit in the CC register is set.

2. Changing the sensitivity of a particular external interrupt clears this pending interrupt. This can be used to clear unwanted pending interrupts. Refer to section "External Interrupt Function" on page 48.

Table 6. Interrupt Sensitivity Bits

ISx1	ISx0	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

EXTERNAL INTERRUPT SELECTION REGISTER (EISR)

Read/Write

Reset Value: 0000 1100 (0Ch)

7							0
ei31	ei30	ei21	ei20	ei11	ei10	ei01	ei00

Bits 7:6 = **ei3[1:0]** *ei3 pin selection*
These bits are written by software. They select the Port B I/O pin used for the ei3 external interrupt according to the table below.

External Interrupt I/O pin selection

ei31	ei30	I/O Pin
0	0	PB0 [1]
0	1	PB1
1	0	PB2

Note:

1. Reset State

Bits 5:4 = **ei2[1:0]** *ei2 pin selection*
These bits are written by software. They select the Port B I/O pin used for the ei2 external interrupt according to the table below.

External Interrupt I/O pin selection

ei21	ei20	I/O Pin
0	0	PB3 [1]
0	1	PB4 [2]
1	0	PB5
1	1	PB6

Notes:

1. Reset State

2. PB4 cannot be used as an external interrupt in HALT mode.

I/O PORTS (Cont'd)

10.7 DEVICE-SPECIFIC I/O PORT CONFIGURATION

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

Standard Ports

PA7:0, PB6:0

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

PC1:0 (multiplexed with OSC1,OSC2)

MODE	DDR
floating input	0
push-pull output	1

The selection between OSC1 or PC0 and OSC2 or PC1 is done by option byte. Refer to section 15.1 on page 149. Interrupt capability is not available on PC1:0.

Note: PCOR not implemented but p-transistor always active in output mode (refer to Figure 32 on page 50)

Interrupt Ports

Ports where the external interrupt capability is selected using the EISR register

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

Table 10. Port Configuration (Standard ports)

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:0	floating	pull-up	open drain	push-pull
Port B	PB6:0	floating	pull-up	open drain	push-pull

Note: On ports where the external interrupt capability is selected using the EISR register, the configuration will be as follows:

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	PB6:0	floating	pull-up interrupt	open drain	push-pull

Table 11. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0000h	**PADR**	MSB							LSB
	Reset Value	1	1	1	1	1	1	1	1
0001h	**PADDR**	MSB							LSB
	Reset Value	0	0	0	0	0	0	0	0

DUAL 12-BIT AUTORELOAD TIMER 4 (Cont'd)

11.2.3.3 Break Function

The break function can be used to perform an emergency shutdown of the application being driven by the PWM signals.

The break function is activated by the external BREAK pin or internal comparator output. This can be selected by using the BRSEL bit in BREAKCR Register. In order to use the break function it must be previously enabled by software setting the BPEN bit in the BREAKCR register.

The Break active level can be programmed by the BREDGE bit in the BREAKCR register. When an active level is detected on the BREAK pin, the BA bit is set and the break function is activated. In this case, the PWM signals are forced to BREAK value if respective OEx bit is set in PWMCR register.

Software can set the BA bit to activate the break function without using the BREAK pin. The BREN1 and BREN2 bits in the BREAKEN Register are used to enable the break activation on the 2 counters respectively. In Dual Timer Mode, the break for PWM2 and PWM3 is enabled by the BREN2 bit. In Single Timer Mode, the BREN1 bit enables the break for all PWM channels.

When a break function is activated (BA bit =1 and BREN1/BREN2 =1):

– The break pattern (PWM[3:0] bits in the BREAKCR) is forced directly on the PWMx output pins if respective OEx is set. (after the inverter).

– The 12-bit PWM counter CNTR1 is put to its reset value, i.e. 00h (if BREN1 = 1).

– The 12-bit PWM counter CNTR2 is put to its reset value,i.e. 00h (if BREN2 = 1).

– ATR1, ATR2, Preload and Active DCRx are put to their reset values.

– Counters stop counting.

When the break function is deactivated after applying the break (BA bit goes from 1 to 0 by software), Timer takes the control of PWM ports.

Figure 41. Block Diagram of Break Function

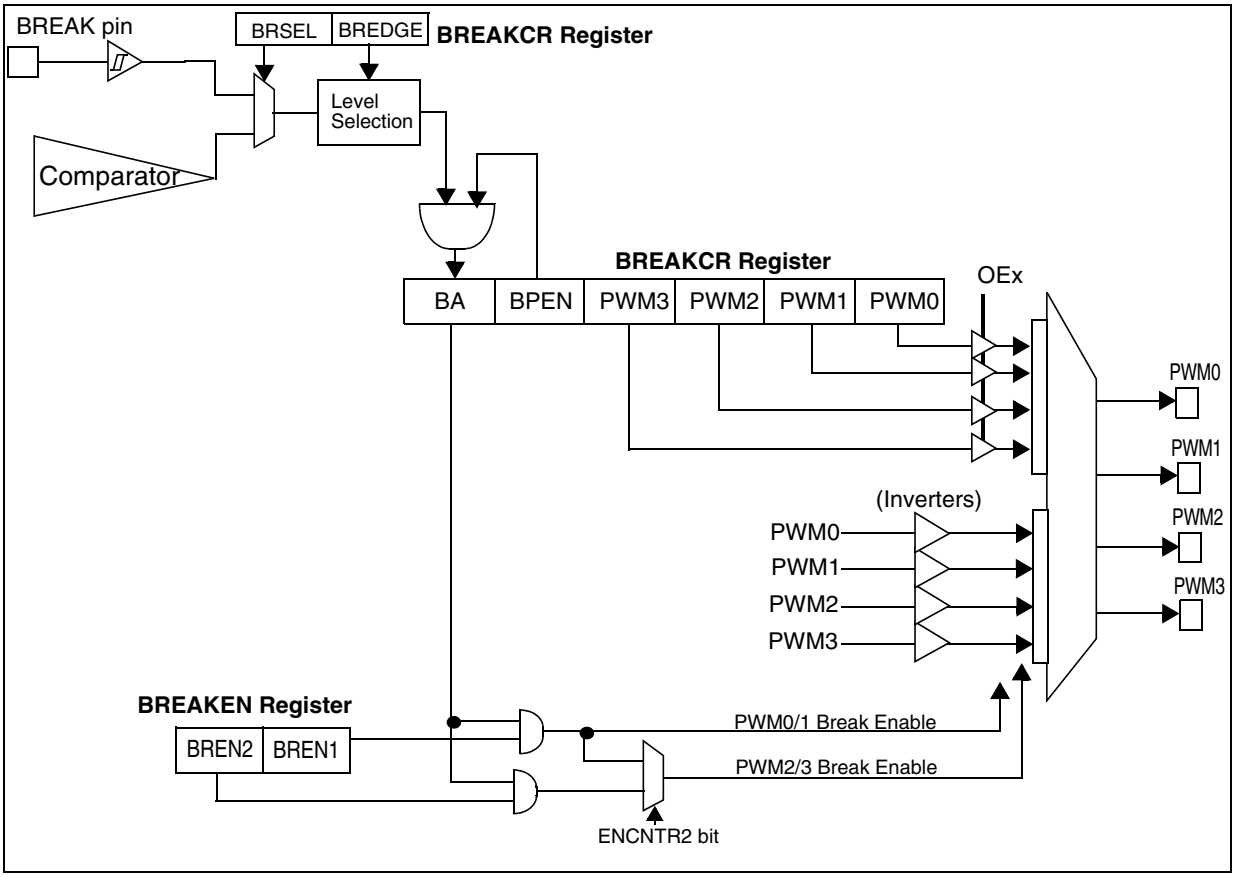
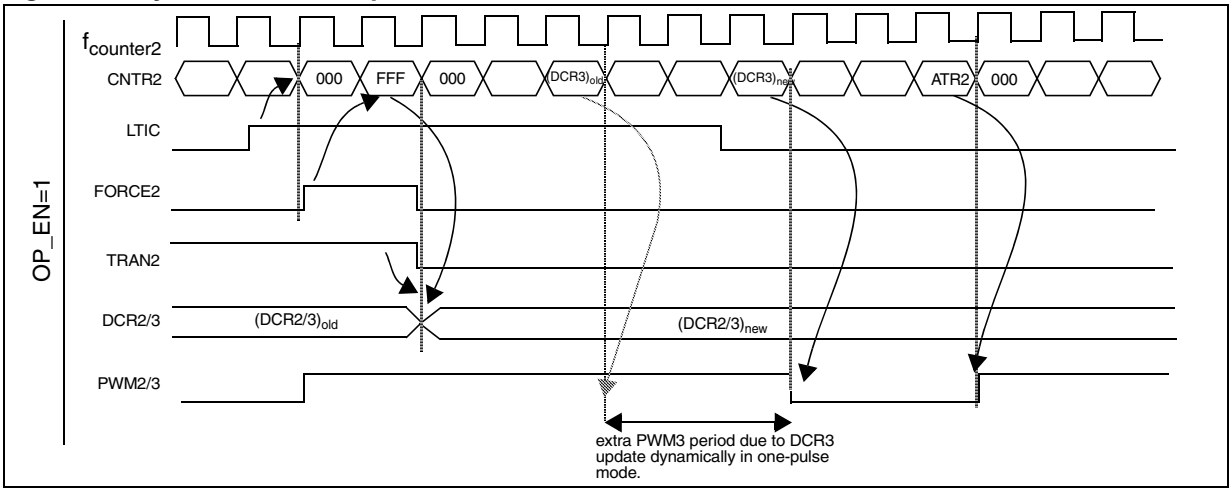


Figure 49. Dynamic DCR2/3 update in One Pulse Mode



LITE TIMER (Cont'd)

Table 15. Lite Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
08	**LTCSR2** Reset Value	0	0	0	0	0	0	TB2IE 0	TB2F 0
09	**LTARR** Reset Value	AR7 0	AR6 0	AR5 0	AR4 0	AR3 0	AR2 0	AR1 0	AR0 0
0A	**LTCNTR** Reset Value	CNT7 0	CNT6 0	CNT5 0	CNT4 0	CNT3 0	CNT2 0	CNT1 0	CNT0 0
0B	**LTCSR1** Reset Value	ICIE 0	ICF x	TB 0	TB1IE 0	TB1F 0	0	0	0
0C	**LTICR** Reset Value	ICR7 0	ICR6 0	ICR5 0	ICR4 0	ICR3 0	ICR2 0	ICR1 0	ICR0 0

SERIAL PERIPHERAL INTERFACE (cont'd)

11.4.3.2 Slave Select Management

As an alternative to using the $\overline{SS}$ pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see Figure 4).

In software management, the external $\overline{SS}$ pin is free for other application uses and the internal $\overline{SS}$ signal level is driven by writing to the SSI bit in the SPICSR register.

In Master mode:

– $\overline{SS}$ internal must be held high continuously

In Slave Mode:

There are two cases depending on the data/clock timing relationship (see Figure 3):

If CPHA = 1 (data latched on second clock edge):

– $\overline{SS}$ internal must be held low during the entire transmission. This implies that in single slave applications the $\overline{SS}$ pin either can be tied to V_{SS}, or made free for standard I/O by managing the $\overline{SS}$ function by software (SSM = 1 and SSI = 0 in the in the SPICSR register)

If CPHA = 0 (data latched on first clock edge):

– $\overline{SS}$ internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If $\overline{SS}$ is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see Section 0.1.5.3).

Figure 55. Generic $\overline{SS}$ Timing Diagram

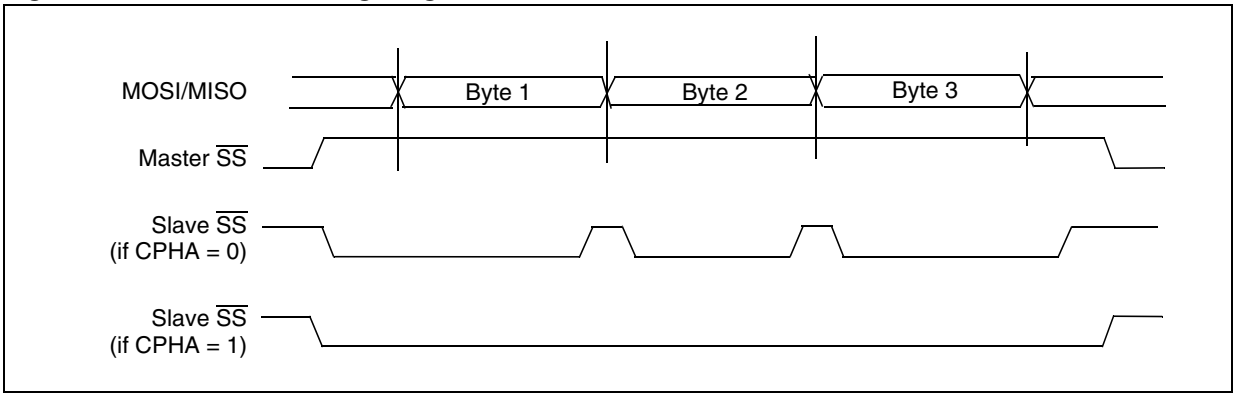
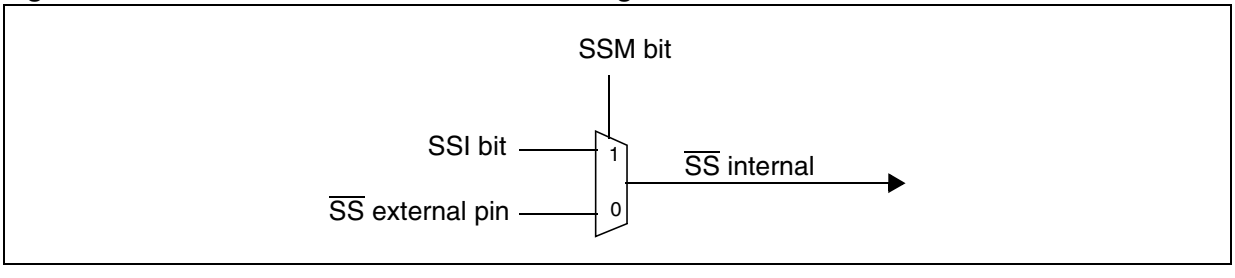


Figure 56. Hardware/Software Slave Select Management



SERIAL PERIPHERAL INTERFACE (cont'd)

11.4.6 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 device is woken up by an interrupt with "exit from HALT mode" capability. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wake-up event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the Device.

11.4.6.1 Using the SPI to wake up the device from Halt mode

In slave configuration, the SPI is able to wake up the device from HALT mode through a SPIF interrupt. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetch). If multiple data transfers have been performed before software clears the SPIF bit, then the OVR bit is set by hardware.

Note: When waking up from HALT mode, if the SPI remains in Slave mode, it is recommended to perform an extra communications cycle to bring the SPI from HALT mode state to normal state. If the SPI exits from Slave mode, it returns to normal state immediately.

Caution: The SPI can wake up the device from HALT mode only if the Slave Select signal (external $\overline{SS}$ pin or the SSI bit in the SPICSR register) is low when the device enters HALT mode. So, if Slave selection is configured as external (see Section 0.1.3.2), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters HALT mode.

11.4.7 Interrupts

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

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)

SPI CONTROL/STATUS REGISTER (SPICSR)
Read/Write (some bits Read Only)
Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** *Serial Peripheral Data Transfer Flag (Read only)*
This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).
0: Data transfer is in progress or the flag has been cleared.
1: Data transfer between the device and an external device has been completed.

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

Bit 6 = **WCOL** *Write Collision status (Read only)*
This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 6).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = **OVR** *SPI Overrun error (Read only)*
This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 0.1.5.2). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register.
0: No overrun error
1: Overrun error detected

Bit 4 = **MODF** *Mode Fault flag (Read only)*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 0.1.5.1 Master Mode Fault (MODF)). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = **SOD** *SPI Output Disable*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)
0: SPI output enabled (if SPE = 1)
1: SPI output disabled

Bit 1 = **SSM** $\overline{SS}$ *Management*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See Section 0.1.3.2 Slave Select Management.
0: Hardware management ($\overline{SS}$ managed by external pin)
1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)

Bit 0 = **SSI** $\overline{SS}$ *Internal Mode*
This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the $\overline{SS}$ slave select signal when the SSM bit is set.
0: Slave selected
1: Slave deselected

SPI DATA I/O REGISTER (SPIDR)
Read/Write
Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, 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.

While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see Figure 1).

ANALOG COMPARATOR (Cont'd)

11.6.4 Register Description

Internal Voltage Reference Register (VREFCR)

Read/Write

Reset Value : 0000 0000 (00h)

7							0
VCEXT	VCBGR	VR3	VR2	VR1	VR0	0	0

Bit 7 = **VCEXT** *External Voltage Reference for Comparator*
This bit is set or cleared by software. It is used to connect the external reference voltage to the VN comparator input.
0: External reference voltage not connected to VN
1: External reference voltage connected to VN

Bit 6 = **VCBGR** *Bandgap Voltage for Comparator*
This bit is set or cleared by software. It is used to connect the bandgap voltage of 1.2V to the VN comparator input.
0: Bandgap voltage not connected to VN
1: Bandgap voltage connected to VN

Bits 5:2 = **VR[3:0]** *Programmable Internal Voltage Reference Range Selection*
These bits are set or cleared by software. They are used to select one of 16 different voltages available from the internal voltage reference module and connect it to comparator input VN.

Refer to Table 20.

Table 20. Voltage Reference Programming

VCEXT bit	VCBGR bit	VR3 bit	VR2 bit	VR1 bit	VR0 bit	VN Voltage
1	x	x	x	x	x	VEXT
0	1	x	x	x	x	1.2 bandgap
0	0	1	1	1	1	3.2V
0	0	1	1	1	0	3V
0	0	1	1	0	1	2.8V
0	0	1	1	0	0	2.6V
0	0	1	0	1	1	2.4V
0	0	1	0	1	0	2.2V
0	0	1	0	0	1	2V
0	0	1	0	0	0	1.8V
0	0	0	1	1	1	1.6V
0	0	0	1	1	0	1.4V
0	0	0	1	0	1	1.2V
0	0	0	1	0	0	1V
0	0	0	0	1	1	0.8V
0	0	0	0	1	0	0.6V
0	0	0	0	0	1	0.4V
0	0	0	0	0	0	0.2V

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

Comparator Control Register (CMPCR)

Read/Write

Reset Value : 1000 0000 (80h)

7							0
CHY-ST	0	CINV	CMPIF	CMPIE	CMP	COUT	CMPON

Bit 7= **CHYST** *Comparator Hysteresis Enable*

This bit is set or cleared by software and set by hardware reset. When this bit is set, the comparator hysteresis is enabled.
0: Hysteresis disabled
1: Hysteresis enabled
Note: To avoid spurious toggling of the output of the comparator due to noise on the voltage reference, it is recommended to enable the hysteresis.

Bit 6 = Reserved, Must be kept cleared

Bit 5 = **CINV** *Comparator Output Inversion Select*

This bit is set or cleared by software and cleared by hardware reset. When this bit is set, the comparator output is inverted.

If interrupt enable bit CMPIE is set in the CMPCR register, the CINV bit is also used to select which type of level transition on the comparator output will generate the interrupt. When this bit is reset, interrupt will be generated at the rising edge of the comparator output change (COMP signal, refer to Figure 62 on page 101). When this bit is set, interrupt will be generated at the falling edge of comparator output change (COMP signal, refer to Figure 62 on page 101).
0: Comparator output not inverted and interrupt generated at the rising edge of COMP
1: Comparator output inverted and interrupt generated at the falling edge of COMP

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	

13.4 SUPPLY CURRENT CHARACTERISTICS

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode for which the clock is stopped).

13.4.1 Supply Current

T_A = -40 to +85°C unless otherwise specified

Symbol	Parameter	Conditions		Typ	Max	Unit
I_{DD}	Supply current in RUN mode	V_{DD}=5.5V	f_{CPU}=8MHz 1)	7	9	mA
	Supply current in WAIT mode		f_{CPU}=8MHz 2)	3	3.6	
	Supply current in SLOW mode		f_{CPU}=250kHz 3)	0.7	0.9	
	Supply current in SLOW WAIT mode7		f_{CPU}=250kHz 4)	0.5	0.8	
	Supply current in HALT mode5)		-40°C≤T_A≤+125°C	<1	6	µA
	Supply current in AWUFH mode 6)7)		-40°C≤T_A≤+125°C	20		

Notes:

1. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

2. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

3. SLOW mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

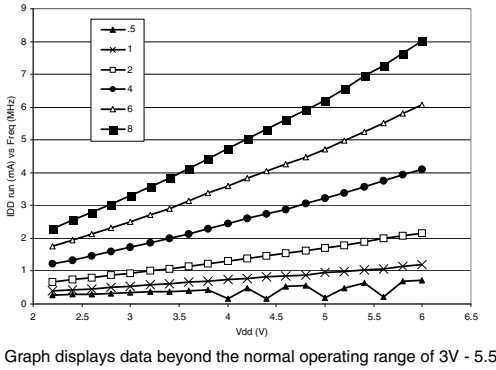
4. SLOW-WAIT mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

5. All I/O pins in output mode with a static value at V_{SS} (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max and f_{CPU} max.

6. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load). Data tested in production at V_{DD} max. and f_{CPU} max.

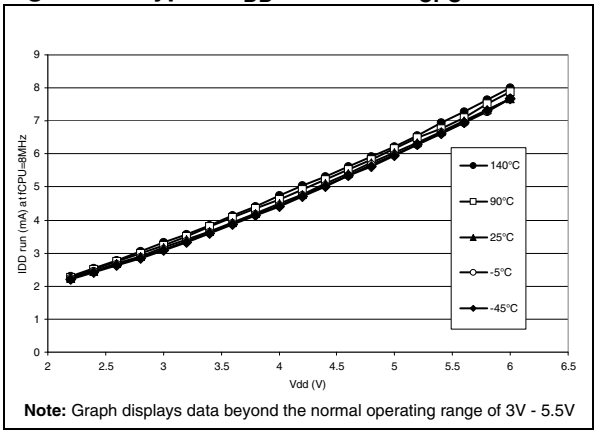
7. This consumption refers to the Halt period only and not the associated run period which is software dependent.

Figure 71. Typical I_{DD} in RUN vs. f_{CPU}



Note: Graph displays data beyond the normal operating range of 3V - 5.5V

Figure 72. Typical I_{DD} in RUN at f_{CPU} = 8MHz



Note: Graph displays data beyond the normal operating range of 3V - 5.5V

13.10 COMMUNICATION INTERFACE CHARACTERISTICS

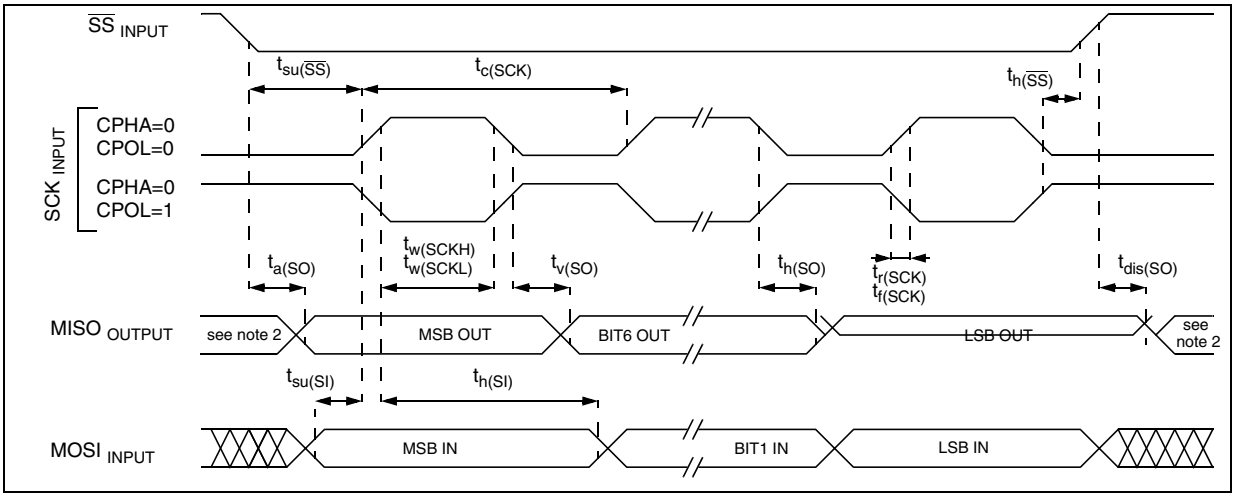
13.10.1 SPI - Serial Peripheral Interface

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU}=8MHz	$f_{CPU}/128$ 0.0625	$f_{CPU}/4$ 2	MHz
		Slave f_{CPU}=8MHz	0	$f_{CPU}/2$ 4	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$ [1]	$\overline{SS}$ setup time [4]	Slave	$(4 \times T_{CPU}) + 50$		ns
$t_{h(\overline{SS})}$ [1]	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ [1] $t_{w(SCKL)}$ [1]	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ [1] $t_{su(SI)}$ [1]	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ [1] $t_{h(SI)}$ [1]	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$ [1]	Data output access time	Slave	0	120	
$t_{dis(SO)}$ [1]	Data output disable time	Slave		240	
$t_{v(SO)}$ [1]	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$ [1]	Data output hold time		0		
$t_{v(MO)}$ [1]	Data output valid time	Master (after enable edge)		120	
$t_{h(MO)}$ [1]	Data output hold time		0		

Figure 107. SPI Slave Timing Diagram with CPHA=0 [3]



Notes:

1. Data based on design simulation, not tested in production.

2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

3. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

4. Depends on f_{CPU}. For example, if f_{CPU}=8MHz, then $T_{CPU} = 1/f_{CPU}$ =125ns and $t_{su(\overline{SS})}$=550ns

ADC CHARACTERISTICS (Cont'd)

ADC Accuracy with V_{DD}=5.0V

Symbol	Parameter	Conditions	Typ	Max	Unit
E_T	Total unadjusted error	f_{CPU}=8MHz, f_{ADC}=4MHz	4	6 [1)]	LSB
E_O	Offset error		3	5 [1)]	
E_G	Gain Error		0.5	4 [1)]	
E_D	Differential linearity error [3)]		1.5	3 [2)]	
E_L	Integral linearity error [3)]		1.5	3 [2)]	

Notes:

1. Data based on characterization results. Not tested in production.

2. Injecting negative current on any of the analog input pins significantly reduces the accuracy of any conversion being performed on any analog input.
Analog pins can be protected against negative injection by adding a Schottky diode (pin to ground). Injecting negative current on digital input pins degrades ADC accuracy especially if performed on a pin close to the analog input pins.
Any positive injection current within the limits specified for I_{INJ}(PIN) and ΣI_{INJ}(PIN) in Section 13.8 does not affect the ADC accuracy.

3. Data based on characterization results over the whole temperature range, monitored in production.

Figure 111. ADC Accuracy Characteristics with amplifier disabled