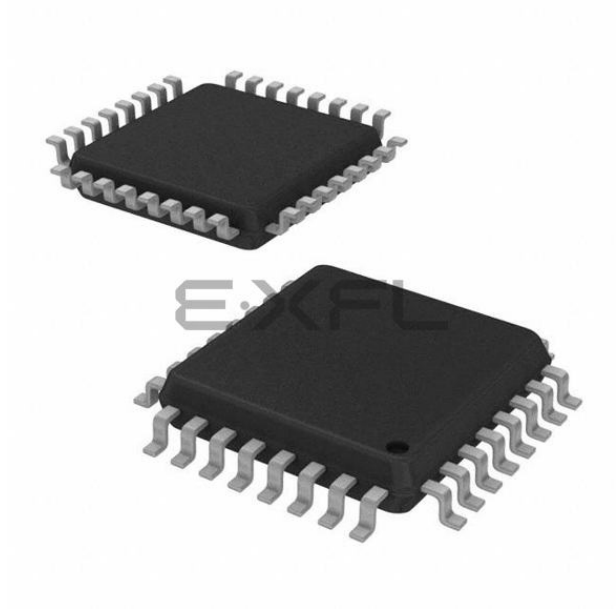


Welcome to E-XFL.COM

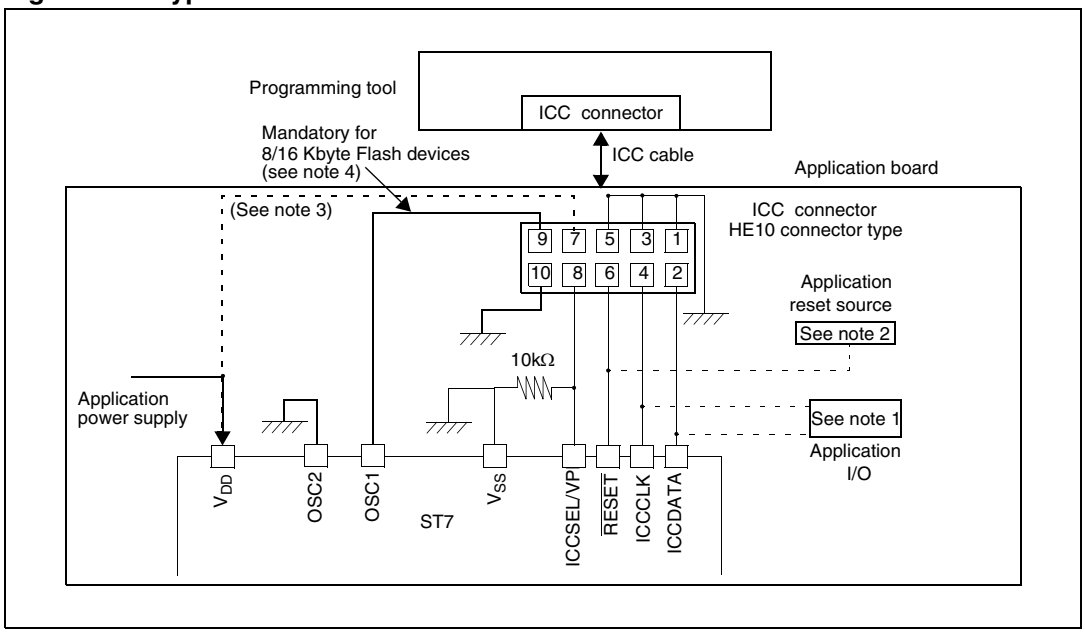
What is "Embedded - Microcontrollers"?

"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	SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 12x10b
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/st72f324bk4t6

Figure 6. Typical ICC interface



1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.
2. During the ICC session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (PUSH-pull output or pull-up resistor <1K). A schottky diode can be used to isolate the application reset circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor >1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of Pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 has to be connected to the OSC1 (OSCIN) pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.

Caution: External clock ICC entry mode is mandatory in ST72F324B 8/16 Kbyte Flash devices. In this case pin 9 must be connected to the OSC1 (OSCIN) pin of the ST7 and OSC2 must be grounded. 32 Kbyte Flash devices may use external clock or application clock ICC entry mode.

4.5 ICP (in-circuit programming)

To perform ICP the microcontroller must be switched to ICC (in-circuit communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection serial communication interface for downloading).

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see *Figure 6*). For more details on the pin locations, refer to the device pinout description.

5.3.1 Accumulator (A)

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

5.3.2 Index registers (X and Y)

These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

5.3.3 Program counter (PC)

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

5.3.4 Condition Code register (CC)

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.

CC Reset value: 111x1xxx

7	6	5	4	3	2	1	0
1	1	I1	H	I0	N	Z	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6. Arithmetic management bits

Bit	Name	Function
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.
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 result 7th 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.

Table 6. Arithmetic management bits (continued)

Bit	Name	Function
1	Z	Zero (Arithmetic Management bit) 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.
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.

Table 7. Software interrupt bits

Bit	Name	Function
5	I1	Software Interrupt Priority 1 The combination of the I1 and I0 bits determines the current interrupt software priority (see *Table 8*).
3	I0	Software Interrupt Priority 0 The combination of the I1 and I0 bits determines the current interrupt software priority (see *Table 8*).

Table 8. Interrupt software priority selection

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low ↓ High	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 (ISPRx). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See *Section 7: Interrupts on page 41* for more details.

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

- under full software control
- in static safe reset

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

During an LVD reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

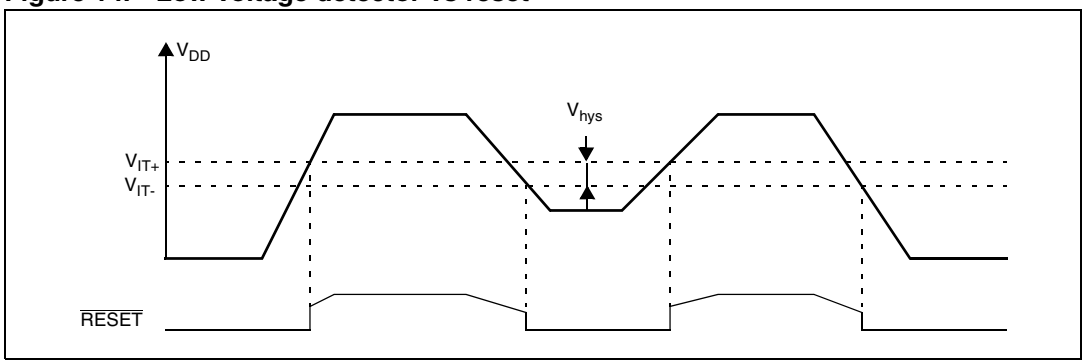
Note: 1 The LVD allows the device to be used without any external reset circuitry.

2 If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8V, device operation is not guaranteed.

3 The LVD is an optional function which can be selected by option byte.

4 It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from reset, to ensure the application functions properly.

Figure 14. Low voltage detector vs reset



6.5.2 AVD (auxiliary voltage detector)

The AVD is based on an analog comparison between a $V_{IT-(AVD)}$ and $V_{IT+(AVD)}$ reference value and the V_{DD} main supply. The V_{IT-} reference value for falling voltage is lower than the V_{IT+} reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real-time status bit (AVDF) in the SICSR register. This bit is read only.

Caution: The AVD function is active only if the LVD is enabled through the option byte (see *Section 14.1 on page 179*).

Monitoring the V_{DD} main supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see *Section 14.1 on page 179*).

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(AVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit toggles).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See *Figure 15*.

6.6 SI registers

6.6.1 System integrity (SI) control/status register (SICSR)

SICSR Reset value: 000x 000x (00h)

7	6	5	4	3	2	1	0
Res	AVDIE	AVDF	LVDRF	Reserved			WDGRF
-	R/W	RO	R/W	-			R/W

Table 12. SICSR register description

Bit	Name	Function
7	-	Reserved, must be kept cleared
6	AVDIE	Voltage Detector Interrupt Enable This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag changes (toggles). The pending interrupt information is automatically cleared when software enters the AVD interrupt routine 0: AVD interrupt disabled 1: AVD interrupt enabled
5	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 changes value. Refer to *Figure 15* and to *Section 6.5.2: AVD (auxiliary voltage detector)* for additional details. 0: V_{DD} over $V_{IT+(AVD)}$ threshold 1: V_{DD} under $V_{IT-(AVD)}$ threshold
4	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 (writing zero). See WDGRF flag description for more details. When the LVD is disabled by option byte, the LVDRF bit value is undefined.
3:1	-	Reserved, must be kept cleared
0	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 (writing zero) or an LVD reset (to ensure a stable cleared state of the WDGRF flag when CPU starts). Combined with the LVDRF information, the flag description is given in *Table 13*.

Table 13. Reset source flags

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

Table 25. Interrupt mapping

No.	Source block	Description	Register label	Priority order	Exit from Halt/Active Halt	Address vector
	Reset	Reset	N/A		yes	FFFEh-FFFFh
	TRAP	Software interrupt			no	FFFCh-FFFDh
0	Not used					FFFAh-FFFBh
1	MCC/RTC	Main clock controller time base interrupt	MCCSR	Higher priority	yes	FFF8h-FFF9h
2	ei0	External interrupt port A3..0	N/A		yes	FFF6h-FFF7h
3	ei1	External interrupt port F2..0			yes	FFF4h-FFF5h
4	ei2	External interrupt port B3..0			yes	FFF2h-FFF3h
5	ei3	External interrupt port B7..4			yes	FFF0h-FFF1h
6	Not used					FFEEh-FFEFh
7	SPI	SPI peripheral interrupts	SPICSR		yes	FFECh-FFEDh
8	Timer A	Timer A peripheral interrupts	TASR		no	FFEAh-FFEBh
9	Timer B	Timer B peripheral interrupts	TBSR		no	FFE8h-FFE9h
10	SCI	SCI peripheral interrupts	SCISR	Lower priority	no	FFE6h-FFE7h
11	AVD	Auxiliary voltage detector interrupt	SICSR		no	FFE4h-FFE5h

10.2.5 Low power modes

Table 37. Effect of low power modes on MCC/RTC

Mode	Description
Wait	No effect on MCC/RTC peripheral. MCC/RTC interrupt causes the device to exit from Wait mode.
Active Halt	No effect on MCC/RTC counter (OIE bit is set), the registers are frozen. MCC/RTC interrupt causes the device to exit from Active Halt mode.
Halt	MCC/RTC counter and registers are frozen. MCC/RTC operation resumes when the MCU is woken up by an interrupt with Exit from Halt capability.

10.2.6 Interrupts

The MCC/RTC interrupt event generates an interrupt if the OIE bit of the MCCSR register is set and the interrupt mask in the CC register is not active (RIM instruction).

Table 38. MCC/RTC interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from WAIT	Exit from HALT
Time base overflow event	OIF	OIE	Yes	No(1)

1. The MCC/RTC interrupt wakes up the MCU from Active Halt mode, not from Halt mode.

10.2.7 MCC registers

MCC control/status register (MCCSR)

MCCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MCO	CP[1:0]		SMS	TB[1:0]		OIE	OIF
R/W	R/W		R/W	R/W		R/W	R/W

Table 39. MCCSR register description

Bit	Name	Function
7	MCO	Main Clock Out selection This bit enables the MCO alternate function on the PF0 I/O port. It is set and cleared by software. 0: MCO alternate function disabled (I/O pin free for general-purpose I/O). 1: MCO alternate function enabled (f_{CPU} on I/O port). *Note: To reduce power consumption, the MCO function is not active in Active Halt mode.*

Table 39. MCCSR register description (continued)

Bit	Name	Function
6:5	CP[1:0]	CPU Clock Prescaler These bits select the CPU clock prescaler which is applied in different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software: 00: f_{CPU} in Slow mode = $f_{OSC2}/2$ 01: f_{CPU} in Slow mode = $f_{OSC2}/4$ 10: f_{CPU} in Slow mode = $f_{OSC2}/8$ 11: f_{CPU} in Slow mode = $f_{OSC2}/16$
4	SMS	Slow Mode Select This bit is set and cleared by software. 0: Normal mode. $f_{CPU} = f_{OSC2}$. 1: Slow mode. f_{CPU} is given by CP1, CP0. See *Section 8.2: Slow mode* and *Section 10.2: Main clock controller with real-time clock and beeper (MCC/RTC)* for more details.
3:2	TB[1:0]	Time Base control These bits select the programmable divider time base. They are set and cleared by software (see *Table 40*). A modification of the time base is taken into account at the end of the current period (previously set) to avoid an unwanted time shift. This allows to use this time base as a real-time clock.
1	OIE	Oscillator interrupt Enable This bit set and cleared by software. 0: Oscillator interrupt disabled 1: Oscillator interrupt enabled This interrupt can be used to exit from Active Halt mode. When this bit is set, calling the ST7 software HALT instruction enters the Active Halt power saving mode.
0	OIF	Oscillator interrupt Flag This bit is set by hardware and cleared by software reading the MCCSR register. It indicates when set that the main oscillator has reached the selected elapsed time (TB1:0). 0: Timeout not reached 1: Timeout reached **Caution**: The BRES and BSET instructions must not be used on the MCCSR register to avoid unintentionally clearing the OIF bit.

Table 40. Time base selection

Counter prescaler	Time base		TB1	TB0
	f_{OSC2} = 4 MHz	f_{OSC2} = 8 MHz		
16000	4ms	2ms	0	0
32000	8ms	4ms	0	1
80000	20ms	10ms	1	0
200000	50ms	25ms	1	1

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

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

Where:

t = Pulse period (in seconds)
f_{CPU} = CPU clock frequency (in hertz)
PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 50*)

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

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

Where:

t = Pulse period (in seconds)
f_{EXT} = External timer clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see *Figure 45*).

Note: 1 The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.

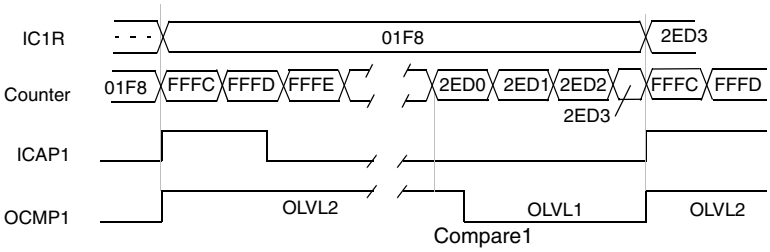
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

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

4 The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.

5 When one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 45. One Pulse mode timing example(1)



1. IEDG1 = 1, OC1R = 2ED0h, OLVL1 = 0, OLVL2 = 1

10.4.5 Error flags

Master mode fault (MODF)

Master mode fault occurs when the master device has its $\overline{SS}$ pin pulled low.

When a Master mode fault occurs:

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

Clearing the MODF bit is done through a software sequence:

1. A read access to the SPICSR register while the MODF bit is set.
2. A write to the SPICR register.

Note: To avoid any conflicts in an application with multiple slaves, the $\overline{SS}$ pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

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

Overrun condition (OVR)

An overrun condition occurs, when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previously transmitted byte.

When an Overrun occurs the OVR bit is set and an interrupt request is generated if the SPIE bit is set.

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

The OVR bit is cleared by reading the SPICSR register.

Write collision error (WCOL)

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

Write collisions can occur both in master and slave mode. See also *Slave Select management on page 99*.

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

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

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

A software sequence clears the WCOL bit (see *Figure 53*).

10.4.8 SPI registers

SPI Control Register (SPICR)

SPICR Reset value: 0000 xxxx (0xh)

7	6	5	4	3	2	1	0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Table 55. SPICR register description

Bit	Name	Function
7	SPIE	Serial Peripheral Interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited. 1: An SPI interrupt is generated whenever SPIF = 1, MODF = 1 or OVR = 1 in the SPICSR register.
6	SPE	Serial Peripheral Output Enable This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see *Master mode fault (MODF) on page 103*). The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins. 0: I/O pins free for general purpose I/O 1: SPI I/O pin alternate functions enabled
5	SPR2	Divider Enable This bit is set and cleared by software and is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to *Table 56: SPI master mode SCK frequency*. 0: Divider by 2 enabled 1: Divider by 2 disabled *Note: This bit has no effect in slave mode.*
4	MSTR	Master mode This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see *Master mode fault (MODF) on page 103*). 0: Slave mode 1: Master mode. The function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.
3	CPOL	Clock Polarity This bit is set and cleared by software. This bit determines the idle state of the serial Clock. The CPOL bit affects both the master and slave modes. 0: SCK pin has a low level idle state 1: SCK pin has a high level idle state *Note: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE bit.*

Table 57. SPICSR register description

Bit	Name	Function
7	SPIF	Serial Peripheral data transfer flag This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the 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.*
6	WCOL	Write Collision status 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 53*). 0: No write collision occurred 1: A write collision has been detected.
5	OVR	SPI Overrun error 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 *Overrun condition (OVR) on page 103*). An interrupt is generated if SPIE = 1 in SPICR register. The OVR bit is cleared by software reading the SPICSR register. 0: No overrun error 1: Overrun error detected
4	MODF	Mode Fault flag This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see *Master mode fault (MODF) on page 103*). An SPI interrupt can be generated if SPIE = 1 in the SPICSR register. This bit is cleared by a software sequence (An access to the SPICR 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.
3	-	Reserved, must be kept cleared.
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.
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 *Slave Select management on page 99*. 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).
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.

ADC Data Register Low (ADCDRL)

ADCDRL Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved						D[1:0]	
-						RO	

Table 73. ADCDRL register description

Bit	Name	Function
7:2	-	Reserved. Forced by hardware to 0.
1:0	D[1:0]	LSB of Converted Analog Value

Table 74. ADC register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0070h	ADCCSR Reset value	EOC 0	SPEED 0	ADON 0	0	CH3 0	CH2 0	CH1 0	CH0 0
0071h	ADCDRH Reset value	D9 0	D8 0	D7 0	D6 0	D5 0	D4 0	D3 0	D2 0
0072h	ADCDRL Reset value	0	0	0	0	0	0	D1 0	D0 0

12.5 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 obtain the total device consumption, the two current values must be added (except for Halt mode for which the clock is stopped).

12.5.1 ROM current consumption

Table 89. ROM current consumption

Symbol	Parameter	Conditions	32K ROM devices		16K/8K ROM devices		Unit
			Typ	Max(1)	Typ	Max(1)	
I_{DD}	Supply current in Run mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 1 MHz f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz	0.55 1.10 2.20 4.38	0.87 1.75 3.5 7.0	0.46 0.93 1.9 3.7	0.69 1.4 2.7 5.5	mA
	Supply current in Slow mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 62.5 kHz f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz	53 100 194 380	87 175 350 700	30 70 150 310	60 120 250 500	µA
	Supply current in Wait mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 1 MHz f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz	0.31 0.61 1.22 2.44	0.5 1.0 2.0 4.0	0.22 0.45 0.91 1.82	0.37 0.75 1.5 3	mA
	Supply current in Slow Wait mode(2)	f_{OSC} = 2 MHz, f_{CPU} = 62.5 kHz f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz	36 69 133 260	63 125 250 500	20 40 90 190	40 90 180 350	µA
	Supply current in Halt mode(3)	-40°C ≤ T_A ≤ +85°C	<1	10	<1	10	
		-40°C ≤ T_A ≤ +125°C	<1	50	<1	50	
	Supply current in Active Halt mode(4)	f_{OSC} = 2 MHz f_{OSC} = 4 MHz f_{OSC} = 8 MHz f_{OSC} = 16 MHz	15 28 55 107	20 38 75 200	11 22 43 85	15 30 60 150	

1. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
2. Measurements are done in the following conditions:
   - Program executed from RAM, CPU running with RAM access. The increase in consumption when executing from Flash is 50%.
   - All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load)
   - All peripherals in reset state
   - LVD disabled.
   - Clock input (OSC1) driven by external square wave
   - In Slow and Slow Wait modes, f_{CPU} is based on f_{OSC} divided by 32

   To obtain the total current consumption of the device, add the clock source (*Section 12.6.3*) and the peripheral power consumption (*Section 12.5.4*).
3. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or V_{SS} (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
4. Data based on characterization results, not tested in production. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or V_{SS} (no load); clock input (OSC1) driven by external square wave, LVD disabled. To obtain the total current consumption of the device, add the clock source consumption (*Section 12.6.3*).

Figure 70. Typical V_{OL} at V_{DD} = 5V (standard ports)

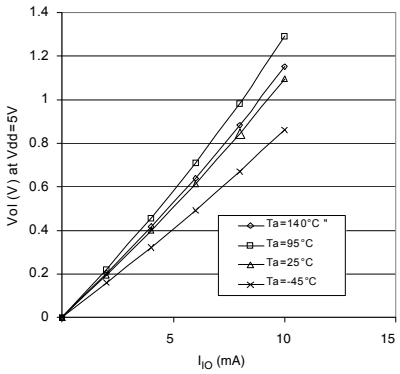


Figure 71. Typical V_{OL} at V_{DD} = 5V (high-sink ports)

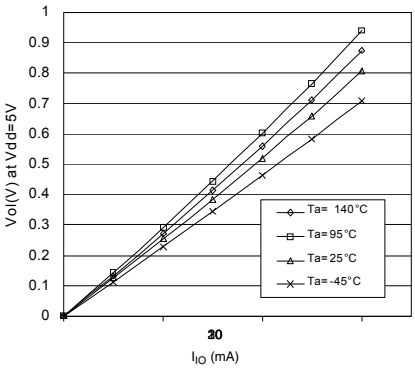
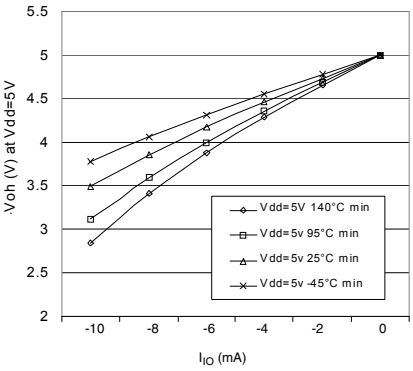


Figure 72. Typical V_{OH} at V_{DD} = 5V



12.12 Communication interface characteristics

12.12.1 Serial peripheral interface (SPI)

The following characteristics are ubject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified. The data is based on design simulation and/or characterization results, not tested in production.

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

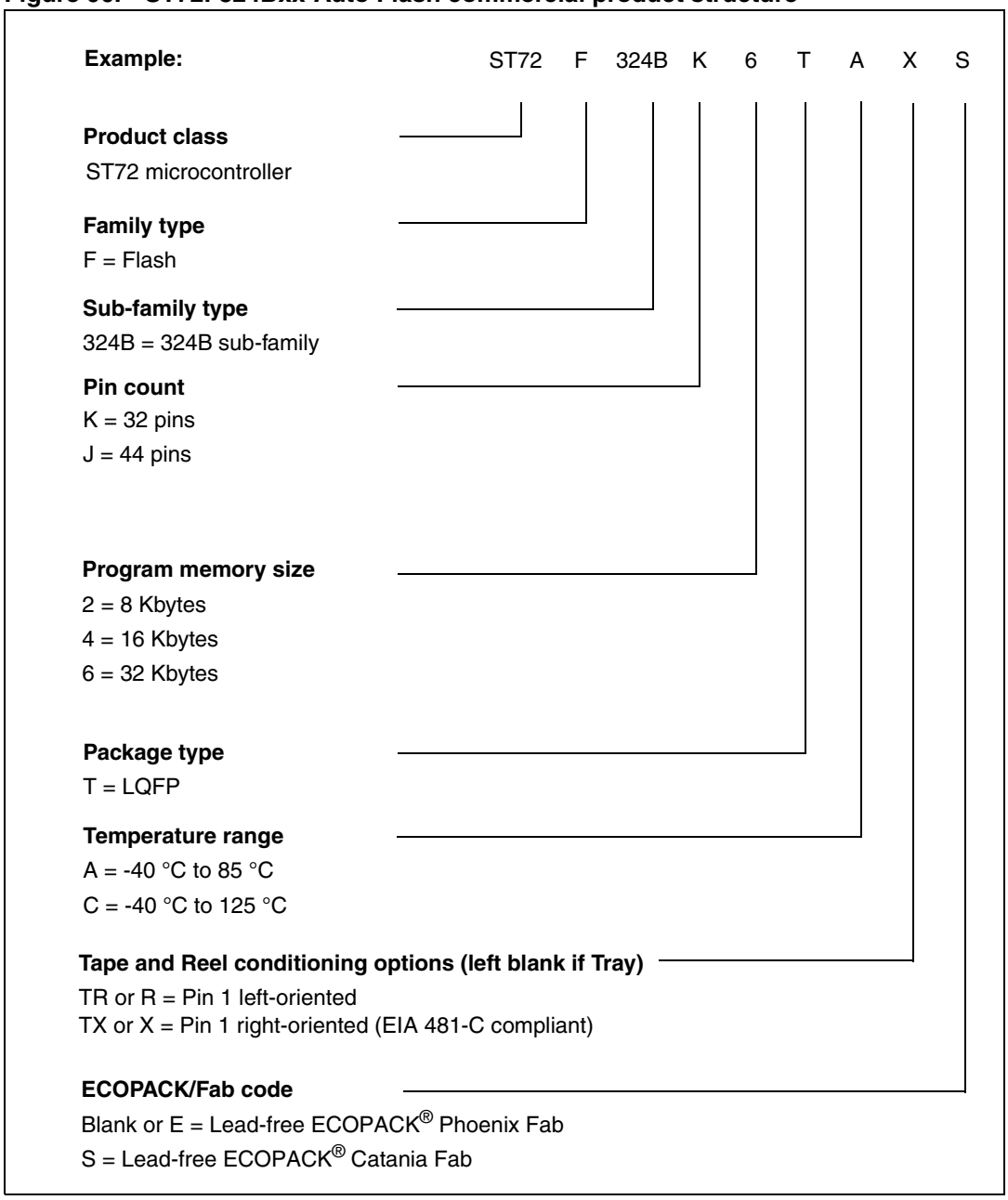
Table 110. SPI characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU} = 8 MHz	$f_{CPU}/128 = 0.0625$	$f_{CPU}/4 = 2$	MHz
		Slave f_{CPU} = 8 MHz	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(2)	Slave	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		

1. Data based on design simulation and/or characterization results, not tested in production.
2. Depends on f_{CPU}. For example, if f_{CPU} = 8 MHz, then $t_{CPU} = 1 / f_{CPU}$ = 125ns and $t_{su(SS)}$ = 175ns.

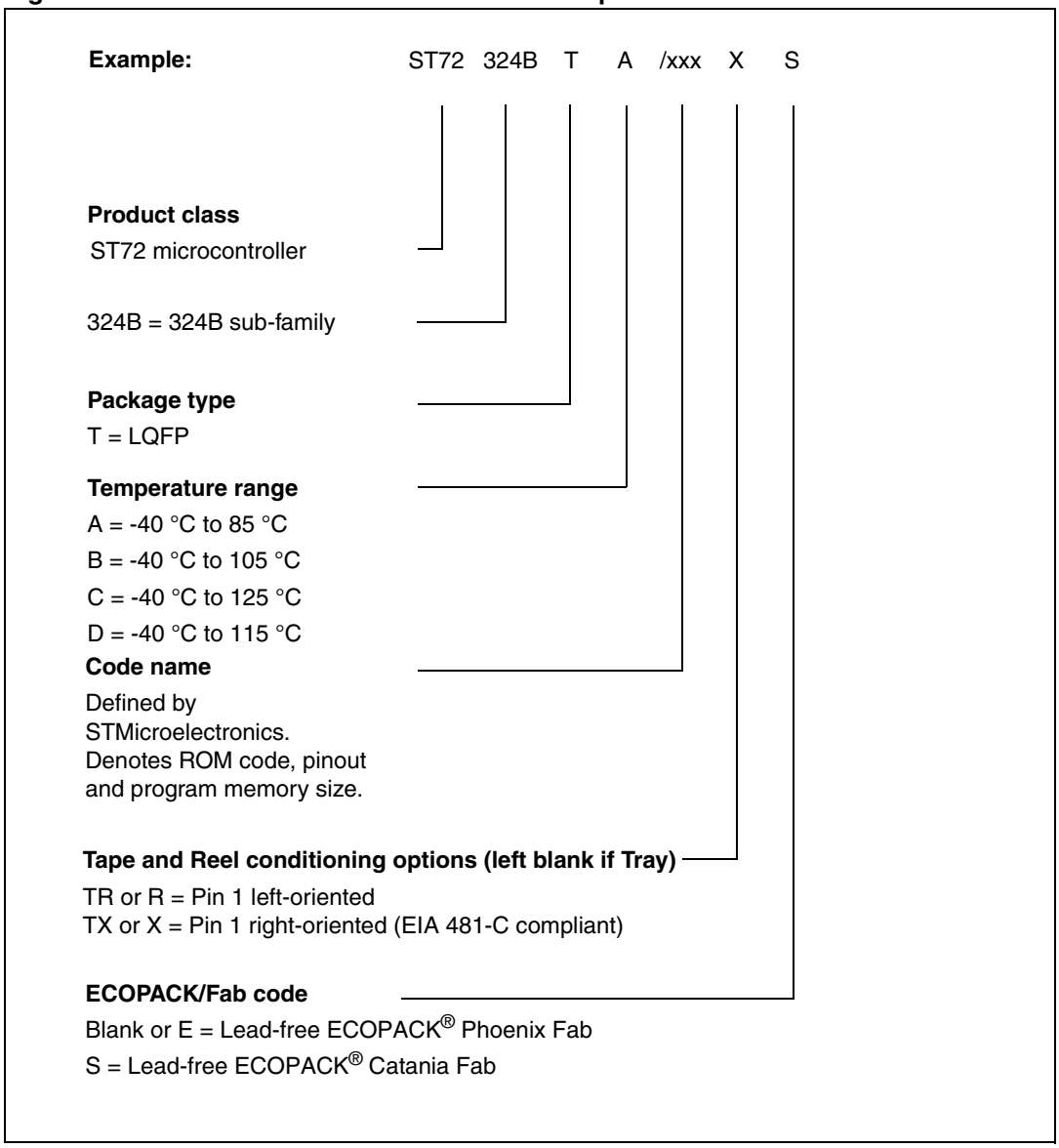
14.1.2 Flash ordering information

Figure 90. ST72F324Bxx-Auto Flash commercial product structure



1. For a list of available options (e.g. memory size, package) and orderable part numbers or for further information on any aspect of this device, please go to *www.st.com* or contact the ST Sales Office nearest to you.

Figure 92. ST72324Bxx-Auto ROM commercial product structure



Nested interrupt context

The symptom does not occur when the interrupts are handled normally, that is, when:

- The interrupt flag is cleared within its own interrupt routine
- The interrupt flag is cleared within any interrupt routine with higher or identical priority level
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

  – PUSH CC
  – SIM
  – Reset interrupt flag
  – POP CC

15.1.5 16-bit timer PWM mode

In PWM mode, the first PWM pulse is missed after writing the value FFFCh in the OC1R register (OC1HR, OC1LR). It leads to either full or no PWM during a period, depending on the OLVL1 and OLVL2 settings.

15.1.6 TIMD set simultaneously with OC interrupt

If the 16-bit timer is disabled at the same time the output compare event occurs then output compare flag gets locked and cannot be cleared before the timer is enabled again.

Impact on the application

If output compare interrupt is enabled, then the output compare flag cannot be cleared in the timer interrupt routine. Consequently the interrupt service routine is called repeatedly.

Workaround

Disable the timer interrupt before disabling the timer. Again while enabling, first enable the timer then the timer interrupts.

- Perform the following to disable the timer:
  – TACR1 or TBCR1 = 0x00h; // Disable the compare interrupt
  – TACSR I or TBCSR I = 0x40; // Disable the timer
- Perform the following to enable the timer again:
  – TACSR & or TBCSR & = ~0x40; // Enable the timer
  – TACR1 or TBCR1 = 0x40; // Enable the compare interrupt

15.1.7 SCI wrong break duration

Description

A single break character is sent by setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may have a longer duration than expected:

- 20 bits instead of 10 bits if M = 0
- 22 bits instead of 11 bits if M = 1

In the same way, as long as the SBK bit is set, break characters are sent to the TDO pin. This may lead to generate one break more than expected.

Occurrence

The occurrence of the problem is random and proportional to the baud rate. With a transmit frequency of 19200 baud (f_{CPU} = 8MHz and SCIBRR = 0xC9), the wrong break duration occurrence is around 1%.

Workaround

If this wrong duration is not compliant with the communication protocol in the application, software can request that an Idle line be generated before the break character. In this case, the break duration is always correct assuming the application is not doing anything between the idle and the break. This can be ensured by temporarily disabling interrupts.

The exact sequence is:

1. Disable interrupts
2. Reset and set TE (IDLE request)
3. Set and reset SBK (break request)
4. Re-enable interrupts

15.2 8/16 Kbyte Flash devices only

15.2.1 39-pulse ICC entry mode

ICC mode entry using ST7 application clock (39 pulses) is not supported. External clock mode must be used (36 pulses). Refer to the *ST7 Flash Programming Reference Manual*.

15.2.2 Negative current injection on pin PB0

Negative current injection on pin PB0 degrades the performance of the device and is not allowed on this pin.

15.3 8/16 Kbyte ROM devices only

15.3.1 Readout protection with LVD

Readout protection is not supported if the LVD is enabled.

15.3.2 I/O Port A and F configuration

When using an external quartz crystal or ceramic resonator, a few f_{OSC2} clock periods may be lost when the signal pattern in *Table 122* occurs. This is because this pattern causes the device to enter test mode and return to user mode after a few clock periods. User program execution and I/O status are not changed, only a few clock cycles are lost.

This happens with either one of the following configurations

- PA3 = 0, PF4 = 1, PF1 = 0 while PLL option is disabled and PF0 is toggling
- PA3 = 0, PF4 = 1, PF1 = 0, PF0 = 1 while PLL option is enabled

This is detailed in *Table 122*