



#### 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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	16MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	4KB (4K 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 6x10b
Oscillator Type	Internal
Operating Temperature	-10°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f262g1b5

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# PIN DESCRIPTION (Cont'd)

57



# Figure 4. TFBGA Package Pinout (view through package)

## 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 compatibility with the ST72C254, the ADCDRL and ADCDRH data registers are located with the LSB on the lower address (6Fh) and the MSB on the higher address (70h). As this scheme is not little Endian, the ADC data registers cannot be treated by C programs as an integer, but have to be treated as two char registers.

# FLASH PROGRAM MEMORY (Cont'd)

## 4.4 ICC interface

ICP needs a minimum of 4 and up to 7 pins to be connected to the programming tool. These pins are:

- RESET: device reset
- V<sub>SS</sub>: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input serial data pin
- ICCSEL: ICC selection (not required on devices without ICCSEL pin)
- OSC1: main clock input for external source (not required on devices without OSC1/OSC2 pins)
- V<sub>DD</sub>: application board power supply (optional, see Note 3)

#### Notes:

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 ICP session, the programming tool must control the 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 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.



Figure 6. Typical ICC Interface

# **5 CENTRAL PROCESSING UNIT**

# **5.1 INTRODUCTION**

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

## **5.2 MAIN FEATURES**

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- Low power HALT and WAIT modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

#### **5.3 CPU REGISTERS**

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

#### Accumulator (A)

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

#### Index Registers (X and Y)

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.

#### Program Counter (PC)

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



# Figure 7. CPU Registers

5/

# CENTRAL PROCESSING UNIT (Cont'd)

Condition Code Register (CC)

Read/Write

Reset Value: 111x1xxx



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

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

## Arithmetic Management Bits

Bit  $4 = \mathbf{H}$  Half carry.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

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

Bit 2 = N Negative.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It's a copy of the result  $7^{th}$  bit.

0: The result of the last operation is positive or null.

1: The result of the last operation is negative

(i.e. the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

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

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.

1: The result of the last operation is zero.

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

## Bit 0 = C Carry/borrow.

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

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

## **Interrupt Management Bits**

#### Bit 5,3 = 11, 10 Interrupt

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

Interrupt Software Priority	11	10
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

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

See the interrupt management chapter for more details.



# 6.2 MULTI-OSCILLATOR (MO)

The main clock of the ST7 can be generated by four different source types coming from the multioscillator block:

- an external source
- 5 crystal or ceramic resonator oscillators
- an internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in Table 3. Refer to the electrical characteristics section for more details.

**Caution:** The OSC1 and/or OSC2 pins must not be left unconnected. For the purposes of Failure Mode and Effects Analysis, it should be noted that if the OSC1 and/or OSC2 pins are left unconnected, the ST7 main oscillator may start and, in this configuration, could generate an  $f_{OSC}$  clock frequency in excess of the allowed maximum (>16MHz.), putting the ST7 in an unsafe/undefined state. The product behaviour must therefore be considered undefined when the OSC pins are left unconnected.

## **External Clock Source**

57/

In this external clock mode, a clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC1 pin while the OSC2 pin is tied to ground.

## **Crystal/Ceramic Oscillators**

This family of oscillators has the advantage of producing a very accurate rate on the main clock of the ST7. The selection within a list of 5 oscillators with different frequency ranges has to be done by option byte in order to reduce consumption (refer to Section 15.1 on page 162 for more details on the frequency ranges). In this mode of the multioscillator, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

These oscillators are not stopped during the RESET phase to avoid losing time in the oscillator start-up phase.

#### Internal RC Oscillator

This oscillator allows a low cost solution for the main clock of the ST7 using only an internal resistor and capacitor. Internal RC oscillator mode has the drawback of a lower frequency accuracy and should not be used in applications that require accurate timing.

In this mode, the two oscillator pins have to be tied to ground.

#### **Related documentation**

AN1530: Accurate timebase for low cost ST7 applications with internal RC.

#### Table 3. ST7 Clock Sources



## I/O PORTS (Cont'd)

## Analog alternate function

Configure the I/O as floating input to use an ADC input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail, connected to the ADC input.

#### Analog Recommendations

Do not change the voltage level or loading on any I/O while conversion is in progress. Do not have clocking pins located close to a selected analog pin.

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

## 9.3 I/O PORT IMPLEMENTATION

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

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 28. Other transitions are potentially risky and should be avoided, since they may present unwanted side-effects such as spurious interrupt generation.

## Figure 28. Interrupt I/O Port State Transitions



#### 9.4 UNUSED I/O PINS

Unused I/O pins must be connected to fixed voltage levels. Refer to Section 13.8.

#### 9.5 LOW POWER MODES

Mode	Description
WAIT	No effect on I/O ports. External interrupts cause the device to exit from WAIT mode.
HALT	No effect on I/O ports. External interrupts cause the device to exit from HALT mode.

#### 9.6 INTERRUPTS

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and if the I bit in the CC register is cleared (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt	
External interrupt on selected external event	-	DDRx ORx	Yes	Yes	

## Related Documentation

AN 970: SPI Communication between ST7 and EEPROM

AN1045: S/W implementation of I2C bus master AN1048: Software LCD driver

# I/O PORTS (Cont'd)

## 9.8 I/O PORT REGISTER DESCRIPTION

## DATA REGISTER (DR)

Port x Data Register PxDR with x = A, B or C. Read/Write Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

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

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

## DATA DIRECTION REGISTER (DDR)

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

## Read/Write

Reset Value: 0000 0000 (00h)

7							
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

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

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

0: Input mode 1: Output mode

57

#### **OPTION REGISTER (OR)**

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

#### Read/Write

Reset Value: 0000 0000 (00h)

7							0
07	O6	O5	O4	O3	O2	O1	00

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

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

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

Each bit is set and cleared by software.

Input mode:

0: Floating input

1: Pull-up input with or without interrupt

Output mode:

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

# I/O PORTS (Cont'd)

# Table 10. I/O Port Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Rese of all I/O p	et Value ort registers	0	0	0	0	0	0	0	0
0000h	PCDR								
0001h	PCDDR	MSB							LSB
0002h	PCOR								
0004h	PBDR								
0005h	PBDDR	MSB							LSB
0006h	PBOR								
0008h	PADR								
0009h	PADDR	MSB							LSB
000Ah	PAOR								



# WATCHDOG TIMER (Cont'd)

**\$7** 

## 11.1.4 How to Program the Watchdog Timeout

Figure 32 shows the linear relationship between the 6-bit value to be loaded in the Watchdog Counter (CNT) and the resulting timeout duration in milliseconds. This can be used for a quick calculation without taking the timing variations into account. If

Figure 32. Approximate Timeout Duration

more precision is needed, use the formulae in Figure 33.

**Caution:** When writing to the WDGCR register, always write 1 in the T6 bit to avoid generating an immediate reset.



# WATCHDOG TIMER (Cont'd)

## Figure 33. Exact Timeout Duration (t<sub>min</sub> and t<sub>max</sub>)

## WHERE:

 $t_{min0} = (LSB + 128) \times 64 \times t_{OSC2}$  $t_{max0} = 16384 \times t_{OSC2}$  $t_{OSC2} = 125ns \text{ if } f_{OSC2} = 8 \text{ MHz}$ 

CNT = Value of T[5:0] bits in the WDGCR register (6 bits) MSB and LSB are values from the table below depending on the timebase selected by the TB[1:0] bits in the MCCSR register

TB1 Bit (MCCSR Reg.)	TB0 Bit (MCCSR Reg.)	Selected MCCSR Timebase	MSB	LSB
0	0	2ms	4	59
0	1	4ms	8	53
1	0	10ms	20	35
1	1	25ms	49	54

To calculate the minimum Watchdog Timeout (t<sub>min</sub>):

**IF** CNT <  $\left[\frac{MSB}{4}\right]$  **THEN**  $t_{min} = t_{min0} + 16384 \times CNT \times t_{osc2}$ 

**ELSE** 
$$t_{min} = t_{min0} + \left[ 16384 \times \left( CNT - \left[ \frac{4CNT}{MSB} \right] \right) + (192 + LSB) \times 64 \times \left[ \frac{4CNT}{MSB} \right] \right] \times t_{osc2}$$

To calculate the maximum Watchdog Timeout (t<sub>max</sub>):

$$\begin{aligned} \text{IF CNT} \leq & \left[\frac{\text{MSB}}{4}\right] & \text{THEN } t_{\text{max}} = t_{\text{max0}} + 16384 \times \text{CNT} \times t_{\text{osc2}} \\ & \text{ELSE } t_{\text{max}} = t_{\text{max0}} + \left[16384 \times \left(\text{CNT} - \left[\frac{4\text{CNT}}{\text{MSB}}\right]\right) + (192 + \text{LSB}) \times 64 \times \left[\frac{4\text{CNT}}{\text{MSB}}\right]\right] \times t_{\text{osc2}} \end{aligned}$$

**Note:** In the above formulae, division results must be rounded down to the next integer value. **Example:** 

With 2ms timeout selected in MCCSR register

Min. Watchdog Timeout (ms) <sup>t</sup> <sub>min</sub>	Max. Watchdog Timeout (ms) t <sub>max</sub>		
1.496	2.048		
128	128.552		
	Min. Watchdog Timeout (ms) t <sub>min</sub> 1.496 128		

# SERIAL COMMUNICATIONS INTERFACE (Cont'd)

# Figure 53. SCI Block Diagram



۸Ţ/

## SERIAL COMMUNICATIONS INTERFACE (Cont'd)

#### Figure 55. SCI Baud Rate and Extended Prescaler Block Diagram



## SERIAL COMMUNICATIONS INTERFACE (Cont'd)

## **Framing Error**

A framing error is detected when:

- The stop bit is not recognized on reception at the expected time, following either a de-synchronization or excessive noise.
- A break is received.

When the framing error is detected:

- the FE bit is set by hardware
- Data is transferred from the Shift register to the SCIDR register.
- No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The FE bit is reset by a SCISR register read operation followed by a SCIDR register read operation.

#### 11.5.4.4 Conventional Baud Rate Generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits) TR = 1, 2, 4, 8, 16, 32, 64,128 (see SCT[2:0] bits) RR = 1, 2, 4, 8, 16, 32, 64,128 (see SCR[2:0] bits)

All these bits are in the SCIBRR register.

**Example:** If  $f_{CPU}$  is 8 MHz (normal mode) and if PR=13 and TR=RR=1, the transmit and receive baud rates are 38400 baud.

**Note:** the baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

## 11.5.4.5 Extended Baud Rate Generation

The extended prescaler option gives a very fine tuning on the baud rate, using a 255 value prescaler, whereas the conventional Baud Rate Generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in the Figure 55.

The output clock rate sent to the transmitter or to the receiver will be the output from the 16 divider divided by a factor ranging from 1 to 255 set in the SCIERPR or the SCIETPR register. **Note:** the extended prescaler is activated by setting the SCIETPR or SCIERPR register to a value other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16 \cdot ETPR^{*}(PR^{*}TR)} Rx = \frac{f_{CPU}}{16 \cdot ERPR^{*}(PR^{*}RR)}$$

with:

ETPR = 1,..,255 (see SCIETPR register)

ERPR = 1,.. 255 (see SCIERPR register)

#### 11.5.4.6 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 awakened by one of the following two ways:

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

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

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

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.

**Caution**: In Mute mode, do not write to the SCICR2 register. If the SCI is in Mute mode during the read operation (RWU=1) and a address mark wake up event occurs (RWU is reset) before the write operation, the RWU bit will be set again by this write operation. Consequently the address byte is lost and the SCI is not woken up from Mute mode.

57

# SERIAL COMMUNICATIONS INTERFACE (Cont'd) CONTROL REGISTER 2 (SCICR2)

## Read/Write

Reset Value: 0000 0000 (00h)

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

Bit 7 = **TIE** *Transmitter interrupt enable.* This bit is set and cleared by software. 0: Interrupt is inhibited

1: An SCI interrupt is generated whenever TDRE=1 in the SCISR register

Bit 6 = TCIE Transmission complete interrupt enable

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An SCI interrupt is generated whenever TC=1 in the SCISR register

## Bit 5 = **RIE** Receiver interrupt enable.

This bit is set and cleared by software.

- 0: Interrupt is inhibited
- 1: An SCI interrupt is generated whenever OR=1 or RDRF=1 in the SCISR register

## Bit 4 = **ILIE** *Idle line interrupt enable.*

This bit is set and cleared by software.

0: Interrupt is inhibited

1: An SCI interrupt is generated whenever IDLE=1 in the SCISR register.

## Bit 3 = **TE** *Transmitter enable.*

This bit enables the transmitter. It is set and cleared by software. 0: Transmitter is disabled

1: Transmitter is enabled

#### Notes:

- During transmission, a "0" pulse on the TE bit ("0" followed by "1") sends a preamble (idle line) after the current word.
- When TE is set there is a 1 bit-time delay before the transmission starts.

**Caution:** The TDO pin is free for general purpose I/O only when the TE and RE bits are both cleared (or if TE is never set).

## Bit 2 = **RE** Receiver enable.

This bit enables the receiver. It is set and cleared by software.

- 0: Receiver is disabled
- 1: Receiver is enabled and begins searching for a start bit

## Bit 1 = **RWU** *Receiver wake-up.*

This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake-up sequence is recognized.

- 0: Receiver in Active mode
- 1: Receiver in Mute mode

**Note:** Before selecting Mute mode (setting the RWU bit), the SCI must receive some data first, otherwise it cannot function in Mute mode with wakeup by idle line detection.

## Bit 0 = SBK Send break.

This bit set is used to send break characters. It is set and cleared by software.

0: No break character is transmitted

1: Break characters are transmitted

**Note:** If the SBK bit is set to "1" and then to "0", the transmitter will send a BREAK word at the end of the current word.

# I<sup>2</sup>C BUS INTERFACE (Cont'd) I<sup>2</sup>C STATUS REGISTER 1 (SR1)

Read Only

Reset Value: 0000 0000 (00h)

7							0
EVF	ADD10	TRA	BUSY	BTF	ADSL	M/SL	SB

#### Bit 7 = **EVF** Event flag.

This bit is set by hardware as soon as an event occurs. It is cleared by software reading SR2 register in case of error event or as described in Figure 59. It is also cleared by hardware when the interface is disabled (PE=0).

#### 0: No event

- 1: One of the following events has occurred:
  - BTF=1 (Byte received or transmitted)
  - ADSL=1 (Address matched in Slave mode while ACK=1)
  - SB=1 (Start condition generated in Master mode)
  - AF=1 (No acknowledge received after byte transmission)
  - STOPF=1 (Stop condition detected in Slave mode)
  - ARLO=1 (Arbitration lost in Master mode)
  - BERR=1 (Bus error, misplaced Start or Stop condition detected)
  - ADD10=1 (Master has sent header byte)
  - Address byte successfully transmitted in Master mode.

Bit 6 = **ADD10** *10-bit addressing in Master mode.* This bit is set by hardware when the master has sent the first byte in 10-bit address mode. It is cleared by software reading SR2 register followed by a write in the DR register of the second address byte. It is also cleared by hardware when the peripheral is disabled (PE=0).

0: No ADD10 event occurred.

1: Master has sent first address byte (header)

## Bit 5 = TRA Transmitter/Receiver.

When BTF is set, TRA=1 if a data byte has been transmitted. It is cleared automatically when BTF is cleared. It is also cleared by hardware after detection of Stop condition (STOPF=1), loss of bus arbitration (ARLO=1) or when the interface is disabled (PE=0).

0: Data byte received (if BTF=1)

## 1: Data byte transmitted

#### Bit 4 = **BUSY** Bus busy.

This bit is set by hardware on detection of a Start condition and cleared by hardware on detection of a Stop condition. It indicates a communication in progress on the bus. The BUSY flag of the I2CSR1 register is cleared if a Bus Error occurs.

0: No communication on the bus

1: Communication ongoing on the bus Note:

- The BUSY flag is NOT updated when the interface is disabled (PE=0). This can have consequences when operating in Multimaster mode; i.e. a second active I<sup>2</sup>C master commencing a transfer with an unset BUSY bit can cause a conflict resulting in lost data. A software workaround consists of checking that the I<sup>2</sup>C is not busy before enabling the I<sup>2</sup>C Multimaster cell.

#### Bit 3 = **BTF** Byte transfer finished.

This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

- Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV6 event (See Figure 59). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.
- Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

- 0: Byte transfer not done
- 1: Byte transfer succeeded

Bit 2 = **ADSL** Address matched (Slave mode). This bit is set by hardware as soon as the received slave address matched with the OAR register content or a general call is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

- 0: Address mismatched or not received
- 1: Received address matched

# I<sup>2</sup>C BUS INTERFACE (Cont'd)

## Bit 1 = M/SL Master/Slave.

This bit is set by hardware as soon as the interface is in Master mode (writing START=1). It is cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1). It is also cleared when the interface is disabled (PE=0). 0: Slave mode

1: Master mode

## Bit 0 = SB Start bit (Master mode).

This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).

0: No Start condition

1: Start condition generated

# I<sup>2</sup>C STATUS REGISTER 2 (SR2)

# Read Only

Reset Value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL

Bit 7:5 = Reserved. Forced to 0 by hardware.

## Bit 4 = **AF** Acknowledge failure.

This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while AF=1 but by other flags (SB or BTF) that are set at the same time.

- 0: No acknowledge failure
- 1: Acknowledge failure

Note:

 When an AF event occurs, the SCL line is not held low; however, the SDA line can remain low if the last bits transmitted are all 0. It is then necessary to release both lines by software.

## Bit 3 = **STOPF** *Stop detection (Slave mode).*

This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK=1). An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0). The SCL line is not held low while STOPF=1.

0: No Stop condition detected

1: Stop condition detected

## Bit 2 = **ARLO** Arbitration lost.

This bit is set by hardware when the interface loses the arbitration of the bus to another master. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

After an ARLO event the interface switches back automatically to Slave mode (M/SL=0).

The SCL line is not held low while ARLO=1.

0: No arbitration lost detected

1: Arbitration lost detected

Note:

- In a Multimaster environment, when the interface is configured in Master Receive mode it does not perform arbitration during the reception of the Acknowledge Bit. Mishandling of the ARLO bit from the I2CSR2 register may occur when a second master simultaneously requests the same data from the same slave and the I<sup>2</sup>C master does not acknowledge the data. The ARLO bit is then left at 0 instead of being set.

## Bit 1 = BERR Bus error.

This bit is set by hardware when the interface detects a misplaced Start or Stop condition. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while BERR=1.

0: No misplaced Start or Stop condition 1: Misplaced Start or Stop condition Note:

 If a Bus Error occurs, a Stop or a repeated Start condition should be generated by the Master to re-synchronize communication, get the transmission acknowledged and the bus released for further communication

Bit 0 = GCAL General Call (Slave mode).

This bit is set by hardware when a general call address is detected on the bus while ENGC=1. It is cleared by hardware detecting a Stop condition (STOPF=1) or when the interface is disabled (PE=0).

0: No general call address detected on bus 1: general call address detected on bus





57

## I/O PORT PIN CHARACTERISTICS (Cont'd)

# I/O PORT PIN CHARACTERISTICS (Cont'd)

# 13.8.2 Output Driving Current

 $T_A = -40$  to  $+85^{\circ}C$  unless otherwise specified

Symbol	Parameter		Conditions	Min	Max	Unit
V <sub>OL</sub> <sup>1)</sup>	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time		I <sub>IO</sub> =+5mA		1.2	V
			I <sub>IO</sub> =+2mA		0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time		I <sub>IO</sub> =+20mA,		1.3	
			I <sub>IO</sub> =+8mA		0.75	
V <sub>OH</sub> <sup>2)</sup>	Output high level voltage for an I/O pin when 4 pins are sourced at same time		I <sub>IO</sub> =-5mA,	V <sub>DD</sub> -1.6		
			I <sub>IO</sub> =-2mA	V <sub>DD</sub> -0.8		
V <sub>OL</sub> <sup>1)3)</sup>	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time		I <sub>IO</sub> =+2mA		0.6	
			I <sub>IO</sub> =+8mA		0.6	
V <sub>OH</sub> <sup>2)3)</sup>	Output high level voltage for an I/O pin when 4 pins are sourced at same time		I <sub>IO</sub> =-2mA T <sub>A</sub> ≤85°C	V <sub>DD</sub> -0.8		
V <sub>OL</sub> <sup>1)3)</sup>	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time		I <sub>IO</sub> =+2mA		0.7	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time		I <sub>IO</sub> =+8mA		0.7	
V <sub>OH</sub> <sup>2)3)</sup>	Output high level voltage for an I/O pin when 4 pins are sourced at same time		I <sub>IO</sub> =-2mA	V <sub>DD</sub> -0.9		

#### Notes:

1. The  $I_{\rm IO}$  current sunk must always respect the absolute maximum rating specified in Section 13.2.2 and the sum of  $I_{\rm IO}$  (I/O ports and control pins) must not exceed  $I_{\rm VSS}$ .

2. The I<sub>IO</sub> current sourced must always respect the absolute maximum rating specified in Section 13.2.2 and the sum of I<sub>IO</sub> (I/O ports and control pins) must not exceed I<sub>VDD</sub>. True open drain I/O pins does not have V<sub>OH</sub>.

3. Not tested in production, based on characterization results.

Notes:

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics.

All other names are the property of their respective owners © 2005 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia – Belgium - Brazil - Canada - China – Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

47/