Contents

1 Description 14

2 Pin description 15

3 Register and memory map 20

4 Flash program memory 23

4.1 Introduction 23

4.2 Main features 23

4.3 Structure 23

4.3.1 Readout protection 24

4.4 ICC interface 25

4.5 ICP (in-circuit programming) 26

4.6 IAP (in-application programming) 26

4.7 Related documentation 26

4.7.1 Flash Control/Status Register (FCSR) 26

5 Central processing unit (CPU) 27

5.1 Introduction 27

5.2 Main features 27

5.3 CPU registers 27

5.3.1 Accumulator (A) 28

5.3.2 Index registers (X and Y) 28

5.3.3 Program counter (PC) 28

5.3.4 Condition Code register (CC) 28

5.3.5 Stack Pointer register (SP) 30

6 Supply, reset and clock management 31

6.1 Introduction 31

6.2 PLL (phase locked loop) 31

6.3 Multi-oscillator (MO) 32

6.3.1 External clock source 32

6.3.2 Crystal/ceramic oscillators 33

Downloaded from Arrow.com.

6.3.3 Internal RC oscillator . . . 33
6.4 Reset sequence manager (RSM) . . . 34
6.4.1 Asynchronous external RESET pin . . . 34
6.5 System integrity management (SI) . . . 36
6.5.1 LVD (low voltage detector) . . . 36
6.5.2 AVD (auxiliary voltage detector) . . . 37
6.5.3 Low power modes . . . 38
6.5.4 Interrupts . . . 38
6.6 SI registers . . . 39
6.6.1 System integrity (SI) control/status register (SICSR) . . . 39

7 Interrupts . . . 41
7.1 Introduction . . . 41
7.2 Masking and processing flow . . . 41
7.2.1 Servicing pending interrupts . . . 42
7.2.2 Different interrupt vector sources . . . 43
7.2.3 Non-maskable sources . . . 43
7.2.4 Maskable sources . . . 43
7.3 Interrupts and low power modes . . . 44
7.4 Concurrent and nested management . . . 44
7.5 Interrupt registers . . . 45
7.5.1 CPU CC register interrupt bits . . . 45
7.5.2 Interrupt software priority registers (ISPRx) . . . 46
7.6 External interrupts . . . 48
7.6.1 I/O port interrupt sensitivity . . . 48
7.6.2 External interrupt control register (EICR) . . . 49

8 Power saving modes . . . 52
8.1 Introduction . . . 52
8.2 Slow mode . . . 52
8.3 Wait mode . . . 53
8.4 Active-halt and Halt modes . . . 54
8.4.1 Active-halt mode . . . 54
8.4.2 Halt mode . . . 55

9 I/O ports . . . 58

Downloaded from Arrow.com.

9.1 Introduction . . . 58
9.2 Functional description . . . 58
9.2.1 Input modes . . . 58
9.2.2 Output modes . . . 59
9.2.3 Alternate functions . . . 59
9.3 I/O port implementation . . . 62
9.4 Low power modes . . . 62
9.5 Interrupts . . . 62
9.5.1 I/O port implementation . . . 63

10 On-chip peripherals . . . 65
10.1 Watchdog timer (WDG) . . . 65
10.1.1 Introduction . . . 65
10.1.2 Main features . . . 65
10.1.3 Functional description . . . 65
10.1.4 How to program the Watchdog timeout . . . 66
10.1.5 Low power modes . . . 68
10.1.6 Hardware Watchdog option . . . 68
10.1.7 Using Halt mode with the WDG (WDGHALT option) . . . 68
10.1.8 Interrupts . . . 68
10.1.9 Control register (WDGCR) . . . 69
10.2 Main clock controller with real-time clock and beeper (MCC/RTC) . . . 69
10.2.1 Programmable CPU clock prescaler . . . 69
10.2.2 Clock-out capability . . . 70
10.2.3 Real-time clock (RTC) timer . . . 70
10.2.4 Beeper . . . 70
10.2.5 Low power modes . . . 71
10.2.6 Interrupts . . . 71
10.2.7 MCC registers . . . 71
10.3 16-bit timer . . . 74
10.3.1 Introduction . . . 74
10.3.2 Main features . . . 74
10.3.3 Functional description . . . 75
10.3.4 Low power modes . . . 88
10.3.5 Interrupts . . . 88
10.3.6 Summary of timer modes . . . 89

Downloaded from Arrow.com.

10.3.7 16-bit timer registers . . . 89
10.4 Serial peripheral interface (SPI) . . . 96
10.4.1 Introduction . . . 96
10.4.2 Main features . . . 96
10.4.3 General description . . . 97
10.4.4 Clock phase and clock polarity . . . 101
10.4.5 Error flags . . . 102
10.4.6 Low power modes . . . 104
10.4.7 Interrupts . . . 104
10.4.8 SPI registers . . . 104
10.5 Serial communications interface (SCI) . . . 108
10.5.1 Introduction . . . 108
10.5.2 Main features . . . 108
10.5.3 General description . . . 109
10.5.4 Functional description . . . 111
10.5.5 Low power modes . . . 120
10.5.6 Interrupts . . . 120
10.5.7 SCI registers . . . 120
10.6 10-bit A/D converter (ADC) . . . 128
10.6.1 Introduction . . . 128
10.6.2 Main features . . . 128
10.6.3 Functional description . . . 129
10.6.4 Low power modes . . . 130
10.6.5 Interrupts . . . 130
10.6.6 ADC registers . . . 130

11 Instruction set . . . 133
11.1 CPU addressing modes . . . 133
11.1.1 Inherent . . . 134
11.1.2 Immediate . . . 134
11.1.3 Direct . . . 135
11.1.4 Indexed (no offset, short, long) . . . 135
11.1.5 Indirect (short, long) . . . 135
11.1.6 Indirect indexed (short, long) . . . 136
11.1.7 Relative mode (direct, indirect) . . . 137
11.2 Instruction groups . . . 137

Downloaded from Arrow.com.

12 Electrical characteristics 141
12.1 Parameter conditions 141
12.1.1 Minimum and maximum values 141
12.1.2 Typical values 141
12.1.3 Typical curves 141
12.1.4 Loading capacitor 141
12.1.5 Pin input voltage 142
12.2 Absolute maximum ratings 142
12.2.1 Voltage characteristics 142
12.2.2 Current characteristics 143
12.2.3 Thermal characteristics 143
12.3 Operating conditions 144
12.4 LVD/AVD characteristics 145
12.4.1 Operating conditions with LVD 145
12.4.2 Auxiliary voltage detector (AVD) thresholds 145
12.5 Supply current characteristics 146
12.5.1 ROM current consumption 146
12.5.2 Flash current consumption 147
12.5.3 Supply and clock managers 148
12.5.4 On-chip peripherals 148
12.6 Clock and timing characteristics 149
12.6.1 General timings 149
12.6.2 External clock source 149
12.6.3 Crystal and ceramic resonator oscillators 150
12.6.4 RC oscillators 152
12.6.5 PLL characteristics 153
12.7 Memory characteristics 153
12.7.1 RAM and hardware registers 153
12.7.2 Flash memory 154
12.8 EMC characteristics 155
12.8.1 Functional electromagnetic susceptibility (EMS) 155
12.8.2 Electromagnetic interference (EMI) 156
12.8.3 Absolute maximum ratings (electrical sensitivity) 157
12.9 I/O port pin characteristics 158
12.9.1 General characteristics 158
12.9.2 Output driving current 159

Downloaded from Arrow.com.

12.10 Control pin characteristics 162
12.10.1 Asynchronous RESET pin 162
12.10.2 ICCSEL/VPP pin 165
12.11 Timer peripheral characteristics 165
12.11.1 16-bit timer 165
12.12 Communication interface characteristics 166
12.12.1 Serial peripheral interface (SPI) 166
12.13 10-bit ADC characteristics 168
12.13.1 Analog power supply and reference pins 170
12.13.2 General PCB design guidelines 170
12.13.3 ADC accuracy 171

13 Package characteristics 172
13.1 ECOPACK 172
13.2 Package mechanical data 172
13.2.1 LQFP44 package mechanical data 172
13.2.2 SDIP42 package mechanical data 173
13.2.3 LQFP32 package mechanical data 174
13.2.4 SDIP32 package mechanical data 175
13.3 Thermal characteristics 177

14 Device configuration and ordering information 178
14.1 Flash devices 179
14.1.1 Flash configuration 179
14.2 ROM devices 181
14.2.1 Transfer of customer code 181
14.3 Development tools 183
14.3.1 Introduction 183
14.3.2 Evaluation tools and starter kits 183
14.3.3 Development and debugging tools 183
14.3.4 Programming tools 183
14.3.5 Socket and emulator adapter information 184
14.4 ST7 Application notes 184

15 Known limitations 185
15.1 All Flash and ROM devices 185

Downloaded from Arrow.com.

15.1.1 Safe connection of OSC1/OSC2 pins 185
15.1.2 External interrupt missed 185
15.1.3 Unexpected reset fetch 187
15.1.4 Clearing active interrupts outside interrupt routine 187
15.1.5 16-bit timer PWM mode 188
15.1.6 TIMD set simultaneously with OC interrupt 188
15.1.7 SCI wrong break duration 188
15.2 8/16 Kbyte Flash devices only 189
15.2.1 39-pulse ICC entry mode 189
15.2.2 Negative current injection on pin PB0 189
15.3 8/16 Kbyte ROM devices only 189
15.3.1 Readout protection with LVD 189
15.3.2 I/O Port A and F configuration 189

16 Revision history 191

Downloaded from Arrow.com.

List of tables

Table 1. Device summary . . . 1
Table 2. Device pin description . . . 17
Table 3. Hardware register map . . . 20
Table 4. Sectors available in Flash devices . . . 23
Table 5. Flash control/status register address and reset value . . . 26
Table 6. Arithmetic management bits . . . 28
Table 7. Software interrupt bits . . . 29
Table 8. Interrupt software priority selection . . . 29
Table 9. ST7 clock sources . . . 33
Table 10. Effect of low power modes on SI . . . 38
Table 11. AVD interrupt control/wakeup capability . . . 38
Table 12. SICSR register description . . . 39
Table 13. Reset source flags . . . 39
Table 14. Interrupt software priority levels . . . 42
Table 15. CPU CC register interrupt bits description . . . 45
Table 16. Interrupt software priority levels . . . 45
Table 17. ISPRx interrupt vector correspondence . . . 46
Table 18. Dedicated interrupt instruction set . . . 46
Table 19. EICR register description . . . 49
Table 20. Interrupt sensitivity - ei2 . . . 49
Table 21. Interrupt sensitivity - ei3 . . . 50
Table 22. Interrupt sensitivity - ei0 . . . 50
Table 23. Interrupt sensitivity - ei1 . . . 50
Table 24. Nested interrupts register map and reset values . . . 50
Table 25. Interrupt mapping . . . 51
Table 26. MCC/RTC low power mode selection . . . 54
Table 27. DR register value and output pin status . . . 59
Table 28. I/O port mode options . . . 60
Table 29. I/O port configurations . . . 61
Table 30. Effect of low power modes on I/O ports . . . 62
Table 31. I/O port interrupt control/wakeup capability . . . 63
Table 32. Port configuration . . . 63
Table 33. I/O port register map and reset values . . . 63
Table 34. Effect of lower power modes on Watchdog . . . 68
Table 35. WDGCR register description . . . 69
Table 36. Watchdog timer register map and reset values . . . 69
Table 37. Effect of low power modes on MCC/RTC . . . 71
Table 38. MCC/RTC interrupt control/wakeup capability . . . 71
Table 39. MCCSR register description . . . 71
Table 40. Time base selection . . . 72
Table 41. MCCBCR register description . . . 73
Table 42. Beep frequency selection . . . 73
Table 43. Main clock controller register map and reset values . . . 73
Table 44. Input capture byte distribution . . . 79
Table 45. Output compare byte distribution . . . 81
Table 46. Effect of low power modes on 16-bit timer . . . 88
Table 47. 16-bit timer interrupt control/wakeup capability . . . 88
Table 48. Summary of timer modes . . . 89

Downloaded from Arrow.com.

Table 49. CR1 register description . . . 89
Table 50. CR2 register description . . . 90
Table 51. CSR register description . . . 91
Table 52. 16-bit timer register map and reset values . . . 95
Table 53. Effect of low power modes on SPI . . . 104
Table 54. SPI interrupt control/wakeup capability . . . 104
Table 55. SPICR register description . . . 105
Table 56. SPI master mode SCK frequency . . . 105
Table 57. SPICSR register description . . . 106
Table 58. SPI register map and reset values . . . 108
Table 59. Frame formats . . . 117
Table 60. Effect of low power modes on SCI . . . 120
Table 61. SCI interrupt control/wakeup capability . . . 120
Table 62. SCISR register description . . . 120
Table 63. SCICR1 register description . . . 122
Table 64. SCICR2 register description . . . 123
Table 65. SCIBRR register description . . . 125
Table 66. SCIERPR register description . . . 126
Table 67. SCIETPR register description . . . 126
Table 68. Baud rate selection . . . 126
Table 69. SCI register map and reset values . . . 127
Table 70. Effect of low power modes on ADC . . . 130
Table 71. ADCCSR register description . . . 130
Table 72. ADCDRH register description . . . 131
Table 73. ADCDRL register description . . . 132
Table 74. ADC register map and reset values . . . 132
Table 75. Addressing mode groups . . . 133
Table 76. CPU addressing mode overview . . . 133
Table 77. Inherent instructions . . . 134
Table 78. Immediate instructions . . . 135
Table 79. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes . . . 136
Table 80. Relative direct and indirect instructions and functions . . . 137
Table 81. Instruction groups . . . 137
Table 82. Instruction set overview . . . 139
Table 83. Voltage characteristics . . . 142
Table 84. Current characteristics . . . 143
Table 85. Thermal characteristics . . . 143
Table 86. Operating conditions . . . 144
Table 87. Operating conditions with LVD . . . 145
Table 88. AVD thresholds . . . 145
Table 89. ROM current consumption . . . 146
Table 90. Flash current consumption . . . 147
Table 91. Oscillators, PLL and LVD current consumption . . . 148
Table 92. On-chip peripherals current consumption . . . 148
Table 93. General timings . . . 149
Table 94. External clock source . . . 149
Table 95. Crystal and ceramic resonator oscillators (8/16 Kbyte Flash and ROM devices) . . . 150
Table 96. Crystal and ceramic resonator oscillators (32 Kbyte Flash and ROM devices) . . . 151
Table 97. OSCRANGE selection for typical resonators . . . 152
Table 98. RC oscillators . . . 152

Downloaded from Arrow.com.

Table 99. PLL characteristics . . . 153
Table 100. RAM and hardware registers . . . 153
Table 101. Dual voltage HDFlash memory . . . 154
Table 102. EMS test results . . . 156
Table 103. EMI emissions . . . 156
Table 104. Absolute maximum ratings . . . 157
Table 105. Electrical sensitivities . . . 157
Table 106. General characteristics . . . 158
Table 107. Output driving current . . . 159
Table 108. Asynchronous RESET pin . . . 162
Table 109. ICCSEL/V_{PP} pin . . . 165
Table 110. 16-bit timer . . . 165
Table 111. SPI characteristics . . . 166
Table 112. 10-bit ADC characteristics . . . 168
Table 113. ADC accuracy . . . 171
Table 114. 44-pin low profile quad flat package mechanical data . . . 172
Table 115. 42-pin dual in line package mechanical data . . . 173
Table 116. 32-pin low profile quad flat package mechanical data . . . 174
Table 117. 32-pin dual in-line package mechanical data . . . 175
Table 118. Thermal characteristics . . . 177
Table 119. Flash option bytes . . . 179
Table 120. Option byte 0 bit description . . . 179
Table 121. Option byte 1 bit description . . . 180
Table 122. Package selection (OPT7) . . . 181
Table 123. STMicroelectronics development tools . . . 184
Table 124. Suggested list of socket types . . . 184
Table 125. Port A and F configuration . . . 190
Table 126. Document revision history . . . 191

Downloaded from Arrow.com.

List of figures

Figure 1. Device block diagram . . . 14
Figure 2. 44-pin LQFP package pinout . . . 15
Figure 3. 42-pin SDIP package pinout . . . 15
Figure 4. 32-pin LQFP package pinout . . . 16
Figure 5. 32-pin SDIP package pinout . . . 16
Figure 6. Memory map . . . 20
Figure 7. Memory map and sector address . . . 24
Figure 8. Typical ICC interface . . . 25
Figure 9. CPU registers . . . 27
Figure 10. Stack manipulation example . . . 30
Figure 11. PLL block diagram . . . 31
Figure 12. Clock, reset and supply block diagram . . . 32
Figure 13. Reset sequence phases . . . 34
Figure 14. Reset block diagram . . . 35
Figure 15. RESET sequences . . . 36
Figure 16. Low voltage detector vs reset . . . 37
Figure 17. Using the AVD to monitor V_{DD} . . . 38
Figure 18. Interrupt processing flowchart . . . 42
Figure 19. Priority decision process flowchart . . . 42
Figure 20. Concurrent interrupt management . . . 44
Figure 21. Nested interrupt management . . . 45
Figure 22. External interrupt control bits . . . 48
Figure 23. Power saving mode transitions . . . 52
Figure 24. Slow mode clock transitions . . . 53
Figure 25. Wait mode flowchart . . . 53
Figure 26. Active-halt timing overview . . . 54
Figure 27. Active-halt mode flowchart . . . 55
Figure 28. Halt timing overview . . . 56
Figure 29. Halt mode flowchart . . . 56
Figure 30. I/O port general block diagram . . . 60
Figure 31. Interrupt I/O port state transitions . . . 62
Figure 32. Watchdog block diagram . . . 66
Figure 33. Approximate timeout duration . . . 66
Figure 34. Exact timeout duration (t_{min} and t_{max}) . . . 67
Figure 35. Main clock controller (MCC/RTC) block diagram . . . 70
Figure 36. Timer block diagram . . . 76
Figure 37. 16-bit read sequence . . . 77
Figure 38. Counter timing diagram, internal clock divided by 2 . . . 78
Figure 39. Counter timing diagram, internal clock divided by 4 . . . 78
Figure 40. Counter timing diagram, internal clock divided by 8 . . . 78
Figure 41. Input capture block diagram . . . 80
Figure 42. Input capture timing diagram . . . 80
Figure 43. Output compare block diagram . . . 83
Figure 44. Output compare timing diagram, $f_{TIMER} = f_{CPU}/2$ . . . 83
Figure 45. Output compare timing diagram, $f_{TIMER} = f_{CPU}/4$ . . . 83
Figure 46. One pulse mode cycle . . . 84
Figure 47. One Pulse mode timing example(1) . . . 85
Figure 48. Pulse width modulation mode timing example with two output compare functions(1)(2) . . . 86

Downloaded from Arrow.com.

Figure 49. Pulse width modulation cycle . . . 87
Figure 50. Serial peripheral interface block diagram . . . 97
Figure 51. Single master/single slave application . . . 98
Figure 52. Generic SS timing diagram . . . 99
Figure 53. Hardware/software slave select management . . . 99
Figure 54. Data clock timing diagram(1) . . . 101
Figure 55. Clearing the WCOL bit (Write collision flag) software sequence . . . 103
Figure 56. Single master/multiple slave configuration . . . 103
Figure 57. SCI block diagram . . . 110
Figure 58. Word length programming . . . 111
Figure 59. SCI baud rate and extended prescaler block diagram . . . 115
Figure 60. Bit sampling in Reception mode . . . 119
Figure 61. ADC block diagram . . . 128
Figure 62. Pin loading conditions . . . 141
Figure 63. Pin input voltage . . . 142
Figure 64. f_{CPU} max versus V_{DD} . . . 144
Figure 65. Typical application with an external clock source . . . 149
Figure 66. Typical application with a crystal or ceramic resonator (8/16 Kbyte Flash and ROM devices) . . . 151
Figure 67. Typical application with a crystal or ceramic resonator (32 Kbyte Flash and ROM devices) . . . 152
Figure 68. Typical $f_{OSC(RCINT)}$ vs T_A . . . 152
Figure 69. Integrated PLL jitter vs signal frequency(1) . . . 153
Figure 70. Unused I/O pins configured as input(1) . . . 159
Figure 71. Typical I_{PU} vs. V_{DD} with $V_{IN} = V_{SS}$ . . . 159
Figure 72. Typical V_{OL} at V_{DD} = 5 V (standard ports) . . . 160
Figure 73. Typical V_{OL} at V_{DD} = 5 V (high-sink ports) . . . 160
Figure 74. Typical V_{OH} at V_{DD} = 5 V . . . 160
Figure 75. Typical V_{OL} vs. V_{DD} (standard ports) . . . 161
Figure 76. Typical V_{OL} vs. V_{DD} (high-sink ports) . . . 161
Figure 77. Typical V_{OH} vs. V_{DD} . . . 161
Figure 78. $\overline{\text{RESET}}$ pin protection when LVD is enabled . . . 163
Figure 79. $\overline{\text{RESET}}$ pin protection when LVD is disabled . . . 164
Figure 80. Two typical applications with ICCSEL/V_{PP} pin(1) . . . 165
Figure 81. SPI slave timing diagram with CPHA = 0(1) . . . 167
Figure 82. SPI slave timing diagram with CPHA = 1(1) . . . 167
Figure 83. SPI master timing diagram(1) . . . 168
Figure 84. R_{AIN} max. vs f_{ADC} with C_{AIN} = 0 pF(1) . . . 169
Figure 85. Recommended C_{AIN} and R_{AIN} values(1) . . . 169
Figure 86. Typical A/D converter application . . . 169
Figure 87. Power supply filtering . . . 170
Figure 88. ADC accuracy characteristics . . . 171
Figure 89. 44-pin low profile quad flat package outline . . . 172
Figure 90. 42-pin plastic dual in-line package, shrink 600-mil width . . . 173
Figure 91. 32-pin low profile quad flat package outline . . . 174
Figure 92. 32-pin plastic dual in-line package, shrink 400-mil width . . . 175
Figure 93. ST72324Bxx ordering information scheme . . . 178

Downloaded from Arrow.com.

1 Description

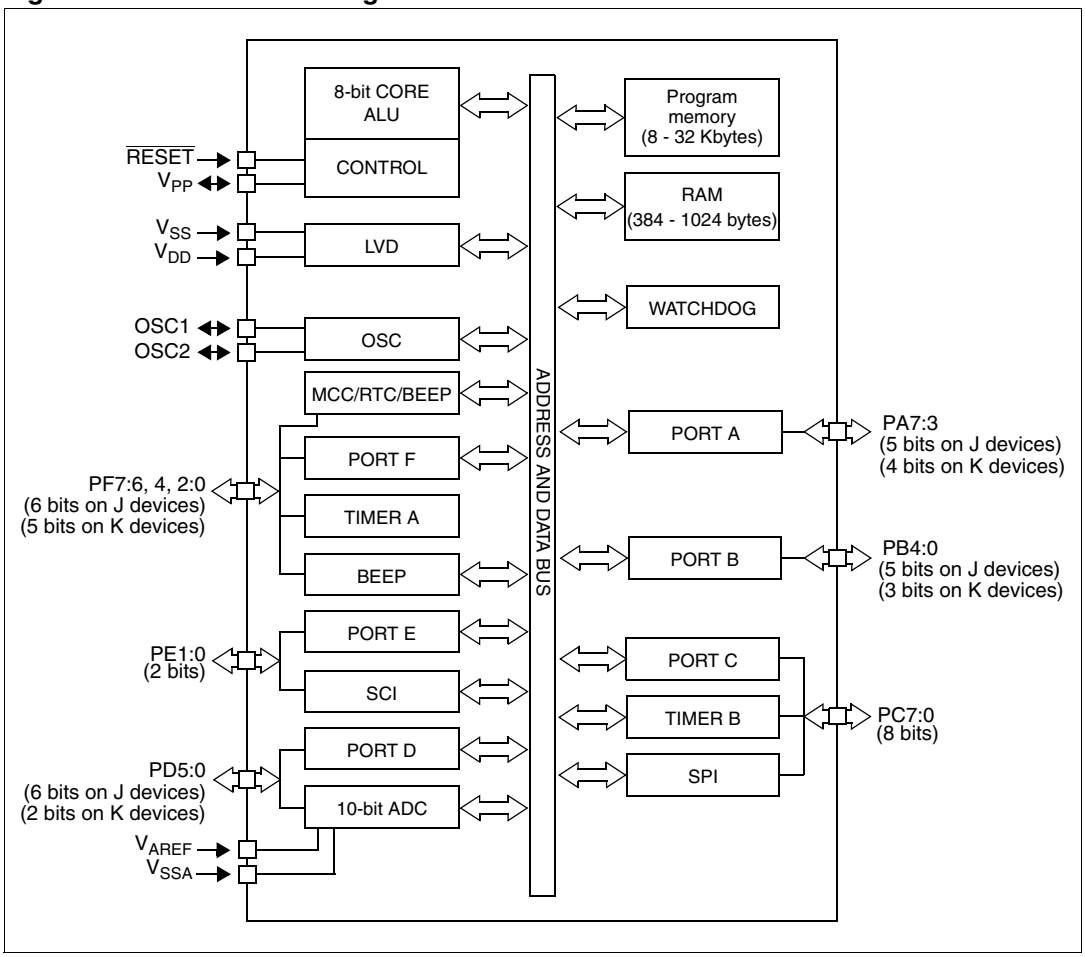
The ST72324Bxx devices are members of the ST7 microcontroller family designed for mid-range applications running from 3.8 to 5.5 V. Different package options offer up to 32 I/O pins.

All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set and are available with Flash or ROM program memory. The ST7 family architecture offers both power and flexibility to software developers, enabling the design of highly efficient and compact application code.

The on-chip peripherals include an A/D converter, two general purpose timers, an SPI interface and an SCI interface. For power economy, the microcontroller can switch dynamically into, Slow, Wait, Active-halt or Halt mode when the application is in idle or stand-by state.

Typical applications include consumer, home, office and industrial products.

Figure 1. Device block diagram



Downloaded from Arrow.com.

2 Pin description

Figure 2. 44-pin LQFP package pinout

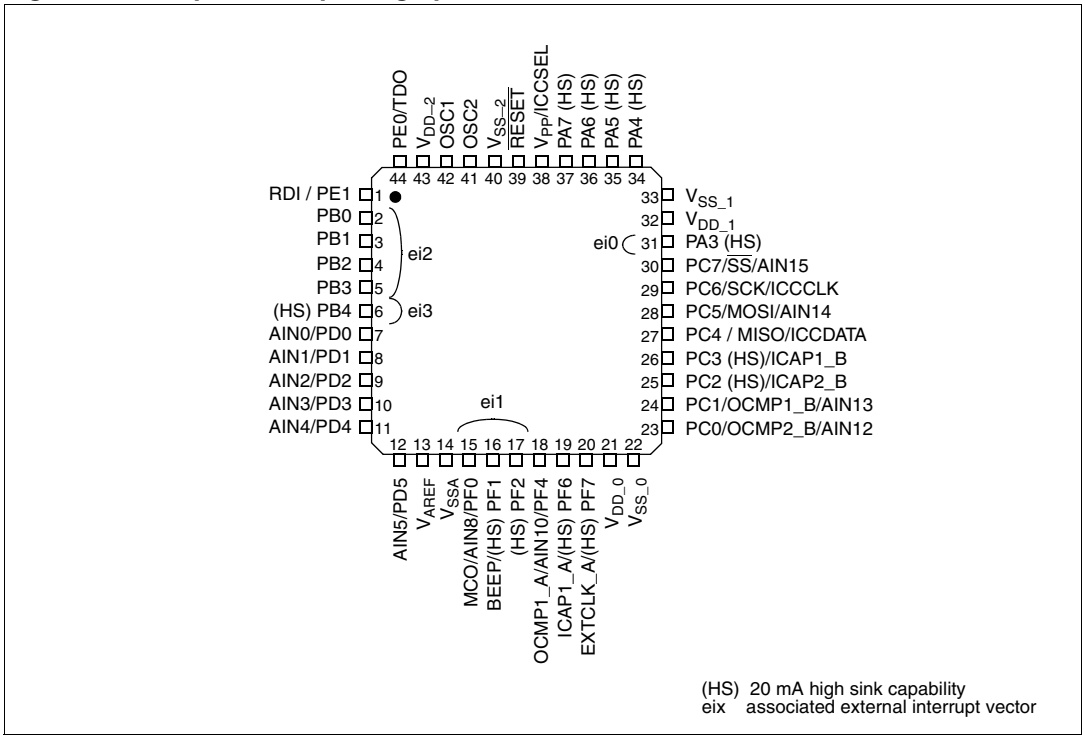
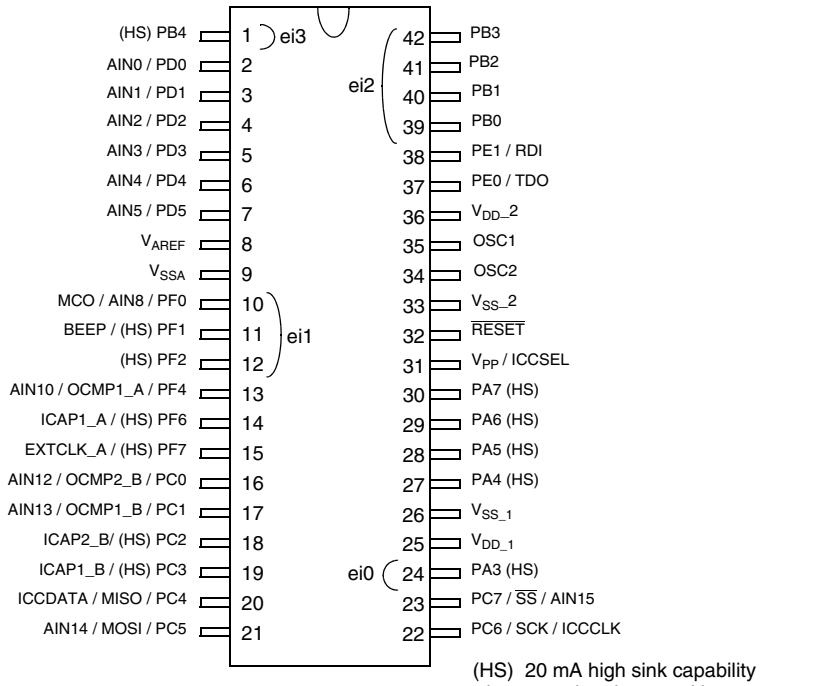


Figure 3. 42-pin SDIP package pinout



Downloaded from Arrow.com.

Figure 4. 32-pin LQFP package pinout

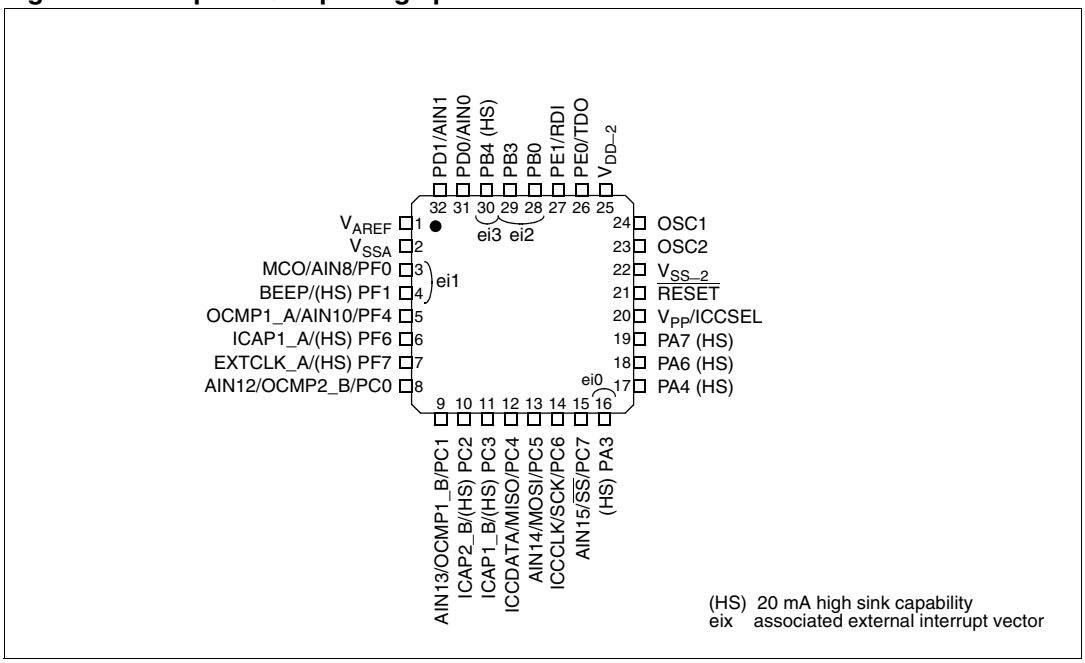
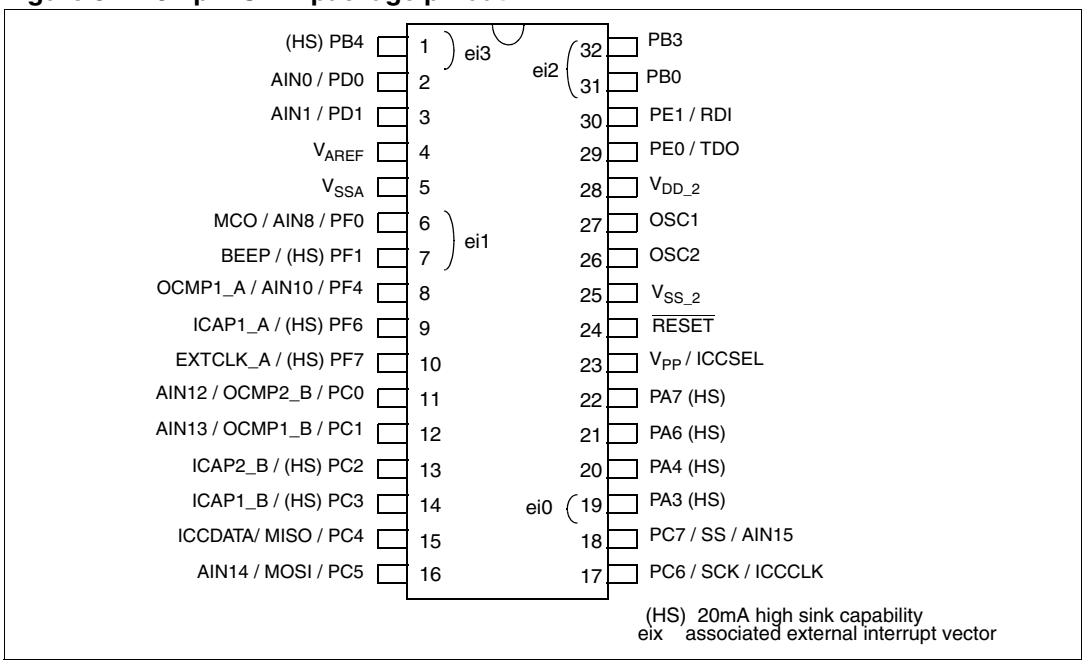


Figure 5. 32-pin SDIP package pinout



See *Section 12: Electrical characteristics on page 141* for external pin connection guidelines.

Refer to *Section 9: I/O ports on page 58* for more details on the software configuration of the I/O ports.

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

Downloaded from Arrow.com.

Legend / Abbreviations for *Table 2:*

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

Input level: A = Dedicated analog input

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

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

Port and control configuration:

Input: float = floating, wpu = weak pull-up, int = interrupt[a], ana = analog ports

Output: OD = open drain[b], PP = push-pull

Table 2. Device pin description

Pin No.				Pin Name	Type	Level		Port						Main function (after reset)	Alternate Function	
								Input				Output				
LQFP44	SDIP42	LQFP32	SDIP32			Input	Output	float	wpu	int	ana	OD	PP			
6	1	30	1	PB4 (HS)	I/O	C_T	HS	X	ei3			X	X	Port B4		
7	2	31	2	PD0/AIN0	I/O	C_T		X	X		X	X	X	Port D0	ADC Analog Input 0	
8	3	32	3	PD1/AIN1	I/O	C_T		X	X		X	X	X	Port D1	ADC Analog Input 1	
9	4			PD2/AIN2	I/O	C_T		X	X		X	X	X	Port D2	ADC Analog Input 2	
10	5			PD3/AIN3	I/O	C_T		X	X		X	X	X	Port D3	ADC Analog Input 3	
11	6			PD4/AIN4	I/O	C_T		X	X		X	X	X	Port D4	ADC Analog Input 4	
12	7			PD5/AIN5	I/O	C_T		X	X		X	X	X	Port D5	ADC Analog Input 5	
13	8	1	4	V_{AREF}[1]	S									Analog Reference Voltage for ADC		
14	9	2	5	V_{SSA}[1]	S									Analog Ground Voltage		
15	10	3	6	PF0/MCO/AIN8	I/O	C_T		X	ei1		X	X	X	Port F0	Main clock out (f_{CPU})	ADC Analog Input 8
16	11	4	7	PF1 (HS)/BEEP	I/O	C_T	HS	X	ei1			X	X	Port F1	Beep signal output	
17	12			PF2 (HS)	I/O	C_T	HS	X	ei1			X	X	Port F2		
18	13	5	8	PF4/OCMP1_A/ AIN10	I/O	C_T		X	X		X	X	X	Port F4	Timer A Output Compare 1	ADC Analog Input 10
19	14	6	9	PF6 (HS)/ICAP1_A	I/O	C_T	HS	X	X			X	X	Port F6	Timer A Input Capture 1	
20	15	7	10	PF7 (HS)/ EXTCLK_A	I/O	C_T	HS	X	X			X	X	Port F7	Timer A External Clock Source	

a. In the interrupt input column, “eiX” defines the associated external interrupt vector. If the weak pull-up column (wpu) is merged with the interrupt column (int), then the I/O configuration is pull-up interrupt input, else the configuration is floating interrupt input.

b. In the open drain output column, ‘T’ *defines a true open drain I/O (P-Buffer and protection diode to V_{DD} are not implemented). See Section 9: I/O ports and Section 12.9: I/O port pin characteristics for more details.*

Downloaded from Arrow.com.

Table 2. Device pin description (continued)

Pin No.				Pin Name	Type	Level		Port						Main function (after reset)	Alternate Function	
								Input				Output				
LQFP44	SDIP42	LQFP32	SDIP32			Input	Output	float	wpu	int	ana	OD	PP			
21				V_{DD_0} (1)	S									Digital Main Supply Voltage		
22				V_{SS_0} (1)	S									Digital Ground Voltage		
23	16	8	11	PC0/OCMP2_B /AIN12	I/O	C_T		X	X		X	X	X	Port C0	Timer B Output Compare 2	ADC Analog Input 12
24	17	9	12	PC1/OCMP1_B /AIN13	I/O	C_T		X	X		X	X	X	Port C1	Timer B Output Compare 1	ADC Analog Input 13
25	18	10	13	PC2 (HS)/ ICAP2_B	I/O	C_T	HS	X	X			X	X	Port C2	Timer B Input Capture 2	
26	19	11	14	PC3 (HS)/ ICAP1_B	I/O	C_T	HS	X	X			X	X	Port C3	Timer B Input Capture 1	
27	20	12	15	PC4/MISO/ICC DATA	I/O	C_T		X	X			X	X	Port C4	SPI Master In / Slave Out Data	ICC Data Input
28	21	13	16	PC5/MOSI/ AIN14	I/O	C_T		X	X		X	X	X	Port C5	SPI Master Out / Slave In Data	ADC Analog Input 14
29	22	14	17	PC6/SCK/ ICCCLK	I/O	C_T		X	X			X	X	Port C6	SPI Serial Clock	ICC Clock Output
30	23	15	18	PC7/$\overline{SS}$/AIN15	I/O	C_T		X	X		X	X	X	Port C7	SPI Slave Select (active low)	ADC Analog Input 15
31	24	16	19	PA3 (HS)	I/O	C_T	HS	X		ei 0		X	X	Port A3		
32	25			V_{DD_1} (1)	S									Digital Main Supply Voltage		
33	26			V_{SS_1} (1)	S									Digital Ground Voltage		
34	27	17	20	PA4 (HS)	I/O	C_T	HS	X	X			X	X	Port A4		
35	28			PA5 (HS)	I/O	C_T	HS	X	X			X	X	Port A5		
36	29	18	21	PA6 (HS)	I/O	C_T	HS	X				T		Port A6 (2)		
37	30	19	22	PA7 (HS)	I/O	C_T	HS	X				T		Port A7 (2)		
38	31	20	23	V_{PP} /ICCSEL	I									Must be tied low. In the flash programming mode, this pin acts as the programming voltage input V_{PP} See *Section 12.10.2* for more details. High voltage must not be applied to ROM devices.		
39	32	21	24	RESET	I/O	C_T								Top priority non maskable interrupt.		
40	33	22	25	V_{SS_2} (1)	S									Digital Ground Voltage		

Downloaded from Arrow.com.

Table 2. Device pin description (continued)

Pin No.				Pin Name	Type	Level		Port						Main function (after reset)	Alternate Function
								Input				Output			
LQFP44	SDIP42	LQFP32	SDIP32			Input	Output	float	wpu	int	ana	OD	PP		
41	34	23	26	OSC2(3)	O									Resonator oscillator inverter output	
42	35	24	27	OSC1(3)	I									External clock input or Resonator oscillator inverter input	
43	36	25	28	V_{DD_2}(1)	S									Digital Main Supply Voltage	
44	37	26	29	PE0/TDO	I/O	C_T		X	X			X	X	Port E0	SCI Transmit Data Out
1	38	27	30	PE1/RDI	I/O	C_T		X	X			X	X	Port E1	SCI Receive Data In
2	39	28	31	PB0	I/O	C_T		X	ei2			X	X	Port B0	**Caution:** Negative current injection not allowed on this pin(4)
3	40			PB1	I/O	C_T		X	ei2			X	X	Port B1	
4	41			PB2	I/O	C_T		X	ei2			X	X	Port B2	
5	42	29	32	PB3	I/O	C_T		X		ei2		X	X	Port B3	

1. It is mandatory to connect all available V_{DD} and V_{REF} pins to the supply voltage and all V_{SS} and V_{SSA} pins to ground.
2. On the chip, each I/O port has eight pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.
3. OSC1 and OSC2 pins connect a crystal/ceramic resonator, or an external source to the on-chip oscillator; see *Section 1: Description* and *Section 12.6: Clock and timing characteristics* for more details.
4. For details refer to *Section 12.9.1 on page 158*

Downloaded from Arrow.com.

3 Register and memory map

As shown in *Figure 6*, the MCU is capable of addressing 64 Kbytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 1024 bytes of RAM and up to 32 Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

Caution: Never access memory locations marked as 'Reserved'. Accessing a reserved area can have unpredictable effects on the device.

Figure 6. Memory map

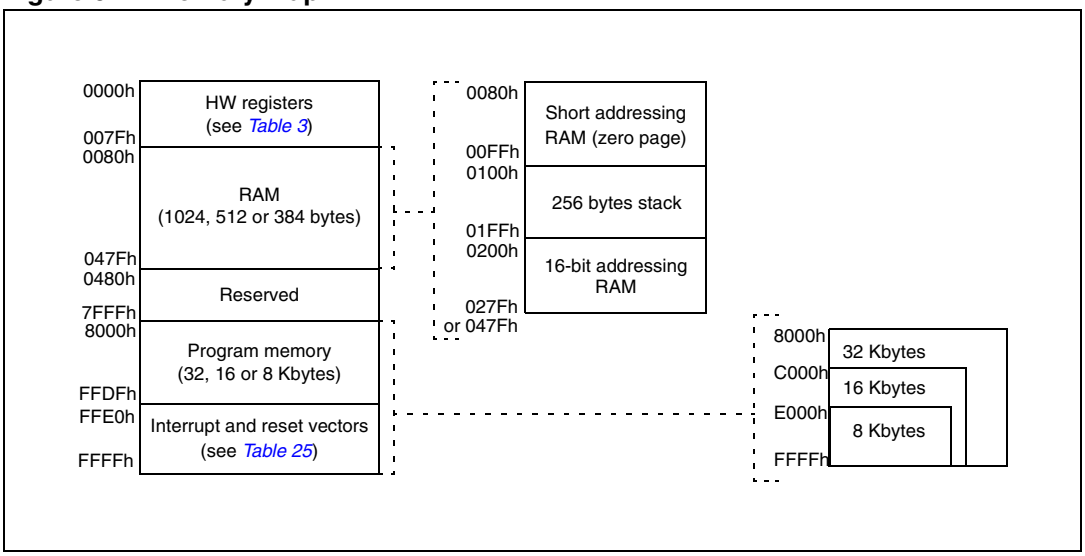


Table 3. Hardware register map

Address	Block	Register label	Register name	Reset status	Remarks
0000h	Port A(1)	PADR	Port A data register	00h(2)	R/W
0001h		PADDR	Port A data direction register	00h	R/W
0002h		PAOR	Port A option register	00h	R/W
0003h	Port B(1)	PBDR	Port B data register	00h(2)	R/W
0004h		PBDDR	Port B data direction register	00h	R/W
0005h		PBOR	Port B option register	00h	R/W
0006h	Port C	PCDR	Port C data register	00h(2)	R/W
0007h		PCDDR	Port C data direction register	00h	R/W
0008h		PCOR	Port C option register	00h	R/W
0009h	Port D(1)	PDADR	Port D data register	00h(2)	R/W
000Ah		PDDDR	Port D data direction register	00h	R/W
000Bh		PDOR	Port D option register	00h	R/W
000Ch	Port E(1)	PEDR	Port E data register	00h(2)	R/W
000Dh		PEDDR	Port E data direction register	00h	R/W(1)
000Eh		PEOR	Port E option register	00h	R/W(1)

Downloaded from Arrow.com.

Table 3. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
000Fh 0010h 0011h	Port F(1)	PFDR PFDDR PFOR	Port F data register Port F data direction register Port F option register	00h(2) 00h 00h	R/W R/W R/W
0012h to 0020h	Reserved area (15 bytes)				
0021h 0022h 0023h	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
0024h 0025h 0026h 0027h	ITC	ISPR0 ISPR1 ISPR2 ISPR3	Interrupt software priority register 0 Interrupt software priority register 1 Interrupt software priority register 2 Interrupt software priority register 3	FFh FFh FFh FFh	R/W R/W R/W R/W
0028h		EICR	External interrupt control register	00h	R/W
0029h	Flash	FCSR	Flash control/status register	00h	R/W
002Ah	Watchdog	WDGCR	Watchdog control register	7Fh	R/W
002Bh	SI	SICSR	System integrity control/status register	000x 000xb	R/W
002Ch 002Dh	MCC	MCCSR MCCBCR	Main clock control/status register Main clock controller: beep control register	00h 00h	R/W R/W
002Eh to 0030h	Reserved area (3 bytes)				
0031h 0032h 0033h 0034h 0035h 0036h 0037h 0038h 0039h 003Ah 003Bh 003Ch 003Dh 003Eh 003Fh	Timer A	TACR2 TACR1 TACSR TAIC1HR TAIC1LR TAOC1HR TAOC1LR TACHR TACLR TAACHR TAACLR TAIC2HR TAIC2LR TAOC2HR TAOC2LR	Timer A control register 2 Timer A control register 1 Timer A control/status register Timer A input capture 1 high register Timer A input capture 1 low register Timer A output compare 1 high register Timer A output compare 1 low register Timer A counter high register Timer A counter low register Timer A alternate counter high register Timer A alternate counter low register Timer A input capture 2 high register Timer A input capture 2 low register Timer A output compare 2 high register Timer A output compare 2 low register	00h 00h xxxx x0xxb xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W R/W Read only Read only R/W R/W Read only Read only Read only Read only Read only Read only R/W R/W
0040h	Reserved area (1 byte)				

Downloaded from Arrow.com.

Table 3. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0041h	Timer B	TBCR2	Timer B control register 2	00h	R/W
0042h		TBCR1	Timer B control register 1	00h	R/W
0043h		TBCSR	Timer B control/status register	xxxx x0xxb	R/W
0044h		TBIC1HR	Timer B input capture 1 high register	xxh	Read only
0045h		TBIC1LR	Timer B input capture 1 low register	xxh	Read only
0046h		TBOC1HR	Timer B output compare 1 high register	80h	R/W
0047h		TBOC1LR	Timer B output compare 1 low register	00h	R/W
0048h		TBCHR	Timer B counter high register	FFh	Read only
0049h		TBCLR	Timer B counter low register	FCh	Read only
004Ah		TBACHR	Timer B alternate counter high register	FFh	Read only
004Bh		TBACLR	Timer B alternate counter low register	FCh	Read only
004Ch		TBIC2HR	Timer B input capture 2 high register	xxh	Read only
004Dh		TBIC2LR	Timer B input capture 2 low register	xxh	Read only
004Eh		TBOC2HR	Timer B output compare 2 high register	80h	R/W
004Fh		TBOC2LR	Timer B output compare 2 low register	00h	R/W
0050h	SCI	SCISR	SCI status register	C0h	Read only
0051h		SCIDR	SCI data register	xxh	R/W
0052h		SCIBRR	SCI baud rate register	00h	R/W
0053h		SCICR1	SCI control register 1	x000 0000b	R/W
0054h		SCICR2	SCI control register 2	00h	R/W
0055h		SCIERPR	SCI extended receive prescaler register	00h	R/W
0056h			Reserved area	---	
0057h		SCIETPR	SCI extended transmit prescaler register	00h	R/W
0058h to 006Fh	Reserved area (24 bytes)				
0070h	ADC	ADCCSR	Control/status register	00h	R/W
0071h		ADCDRH	Data high register	00h	Read only
0072h		ADCDRL	Data low register	00h	Read only
0073h 007Fh	Reserved area (13 bytes)				

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

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

Downloaded from Arrow.com.

4 Flash program memory

4.1 Introduction

The ST7 dual voltage High Density Flash (HDFlash) is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a byte-by-byte basis using an external V_{PP} supply.

The HDFlash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using ICP (in-circuit programming) or IAP (in-application programming).

The array matrix organization allows each sector to be erased and reprogrammed without affecting other sectors.

4.2 Main features

- 3 Flash programming modes:
  - Insertion in a programming tool. In this mode, all sectors including option bytes can be programmed or erased.
  - ICP (in-circuit programming). In this mode, all sectors including option bytes can be programmed or erased without removing the device from the application board.
  - IAP (in-application programming). In this mode, all sectors, except Sector 0, can be programmed or erased without removing the device from the application board and while the application is running.
- ICT (in-circuit testing) for downloading and executing user application test patterns in RAM
- Readout protection
- Register Access Security System (RASS) to prevent accidental programming or erasing

4.3 Structure

The Flash memory is organized in sectors and can be used for both code and data storage.

Depending on the overall Flash memory size in the microcontroller device, there are up to three user sectors (see *Table 4*). Each of these sectors can be erased independently to avoid unnecessary erasing of the whole Flash memory when only a partial erasing is required.

The first two sectors have a fixed size of 4 Kbytes (see *Figure 7*). They are mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

Table 4. Sectors available in Flash devices

Flash size	Available sectors
4 Kbytes	Sector 0
8 Kbytes	Sectors 0, 1
>8 Kbytes	Sectors 0, 1, 2

Downloaded from Arrow.com.

4.3.1 Readout protection

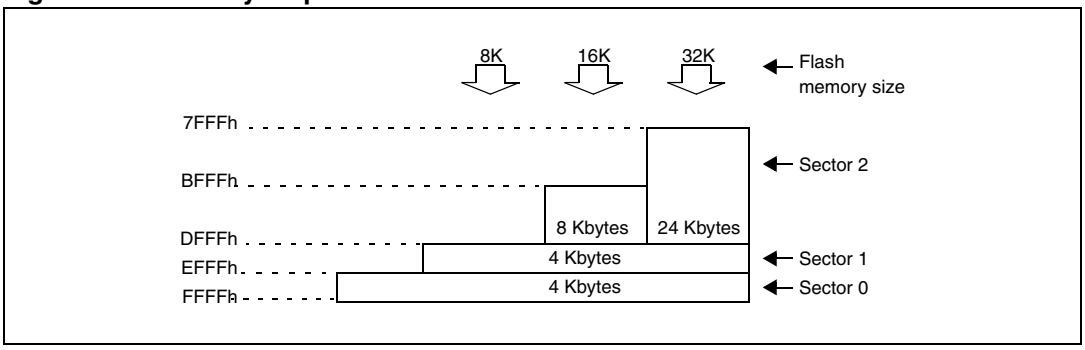
Readout protection, when selected, provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller.

In Flash devices, this protection is removed by reprogramming the option. In this case, the entire program memory is first automatically erased.

Readout protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP_R bit in the option byte.
- In ROM devices it is enabled by mask option specified in the option list.

Figure 7. Memory map and sector address



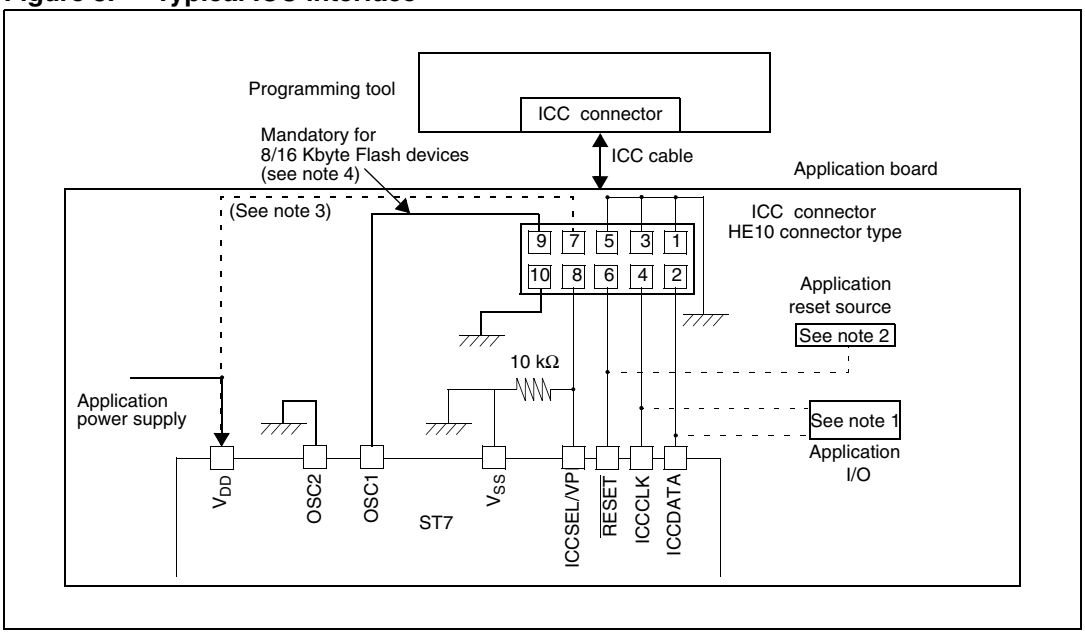
Downloaded from Arrow.com.

4.4 ICC interface

ICC needs a minimum of 4 and up to 6 pins to be connected to the programming tool (see *Figure 8*). These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input/output serial data pin
- ICCSEL/V_{PP}: programming voltage
- OSC1 (or OSCIN): main clock input for external source (optional)
- V_{DD}: application board power supply (optional, see *Figure 8*, Note 3)

Figure 8. 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 <1 kΩ). A schottky diode can be used to isolate the application reset circuit in this case. When using a classical RC network with R>1kΩ or a reset management IC with open drain output and pull-up resistor >1 kΩ, 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.

Downloaded from Arrow.com.

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 8*). For more details on the pin locations, refer to the device pinout description.

4.6 IAP (in-application programming)

This mode uses a BootLoader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (such as user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored). For example, it is possible to download code from the SPI, SCI, USB or CAN interface and program it in the Flash. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

4.7 Related documentation

For details on Flash programming and ICC protocol, refer to the *ST7 Flash Programming Reference Manual* and to the *ST7 ICC Protocol Reference Manual*.

4.7.1 Flash Control/Status Register (FCSR)

This register is reserved for use by programming tool software. It controls the Flash programming and erasing operations.

FCSR Reset value:0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 5. Flash control/status register address and reset value

Address (Hex)	Register label	7	6	5	4	3	2	1	0
0029h	FCSR reset value	0	0	0	0	0	0	0	0

Downloaded from Arrow.com.

5 Central processing unit (CPU)

5.1 Introduction

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

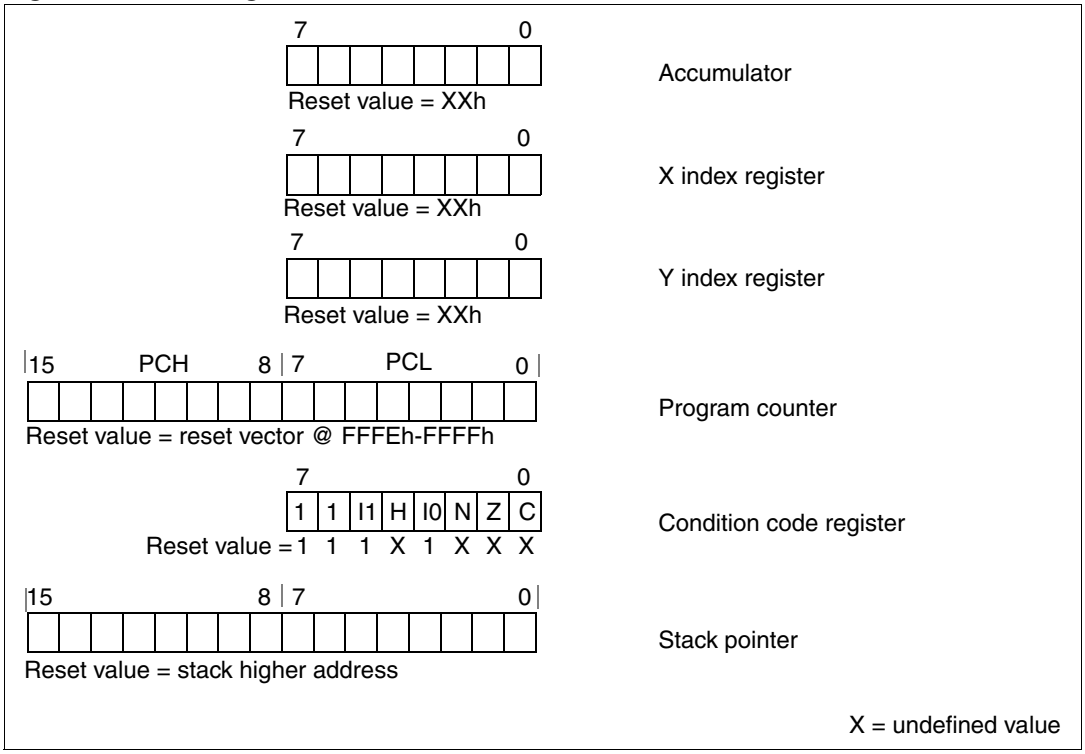
5.2 Main features

- 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 six CPU registers shown in *Figure 9* are not present in the memory mapping and are accessed by specific instructions.

Figure 9. CPU registers



Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

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.

Downloaded from Arrow.com.

5.3.5 Stack Pointer register (SP)

SP — Reset value: 01 FFh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see *Figure 10*).

Since the stack is 256 bytes deep, the 8 most significant bits are forced by hardware. Following an MCU reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by an LD instruction.

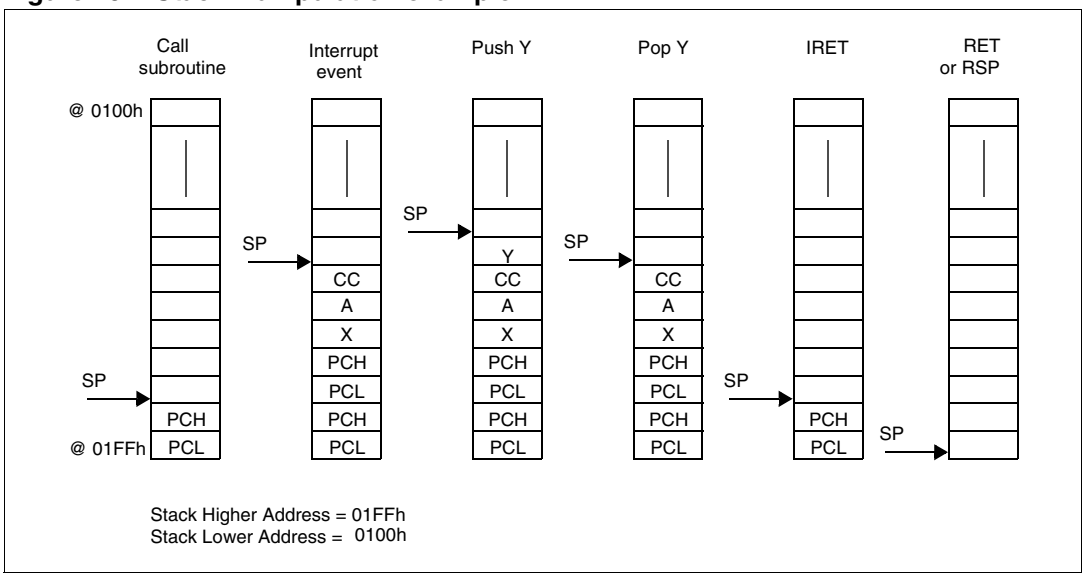
Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in *Figure 10*.

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 10. Stack manipulation example



Downloaded from Arrow.com.

6 Supply, reset and clock management

6.1 Introduction

The device includes a range of utility features for securing the application in critical situations (for example in case of a power brown-out), and reducing the number of external components. An overview is shown in *Figure 12*.

For more details, refer to dedicated parametric section.

Main features

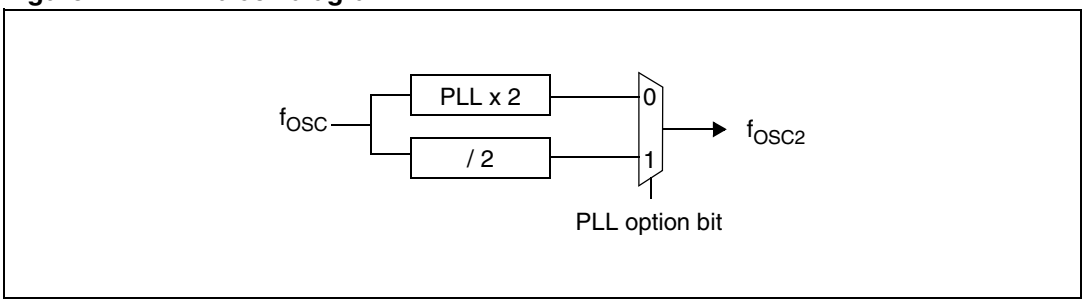
- Optional Phase Locked Loop (PLL) for multiplying the frequency by 2 (not to be used with internal RC oscillator in order to respect the max. operating frequency)
- Multi-Oscillator clock management (MO)
  - 5 crystal/ceramic resonator oscillators
  - 1 Internal RC oscillator
- Reset Sequence Manager (RSM)
- System Integrity management (SI)
  - Main supply low voltage detection (LVD)
  - Auxiliary voltage detector (AVD) with interrupt capability for monitoring the main supply

6.2 PLL (phase locked loop)

If the clock frequency input to the PLL is in the range 2 to 4 MHz, the PLL can be used to multiply the frequency by two to obtain an f_{OSC2} of 4 to 8 MHz. The PLL is enabled by option byte. If the PLL is disabled, then $f_{OSC2} = f_{OSC}/2$.

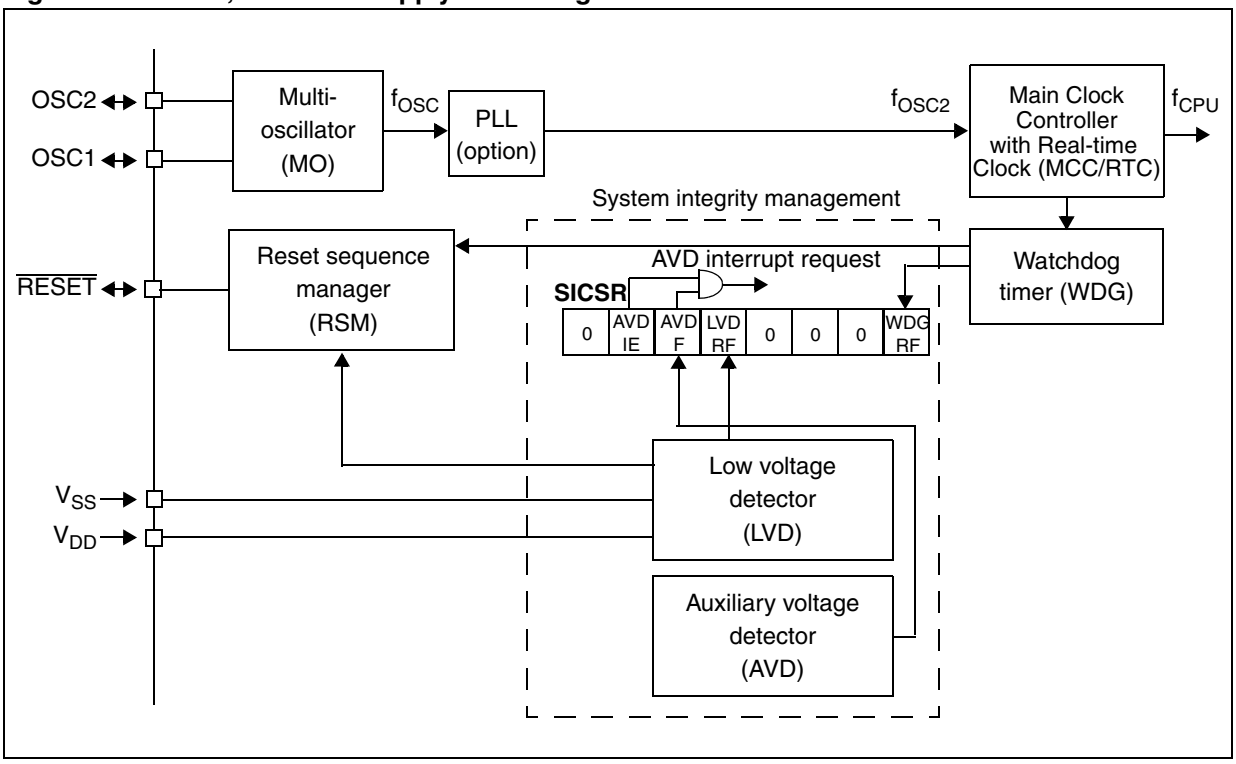
Caution: The PLL is not recommended for applications where timing accuracy is required. Furthermore, it must not be used with the internal RC oscillator.

Figure 11. PLL block diagram



Downloaded from Arrow.com.

Figure 12. Clock, reset and supply block diagram



6.3 Multi-oscillator (MO)

The main clock of the ST7 can be generated by three different source types coming from the multi-oscillator block:

- an external source
- 4 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 9*. 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 Effect 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 (> 16 MHz.), putting the ST7 in an unsafe/undefined state. The product behavior must therefore be considered undefined when the OSC pins are left unconnected.

6.3.1 External clock source

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.

Downloaded from Arrow.com.

6.3.2 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 four oscillators with different frequency ranges has to be done by option byte in order to reduce consumption (refer to *Section 14.1 on page 179* for more details on the frequency ranges). In this mode of the multi-oscillator, 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.

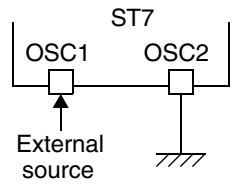
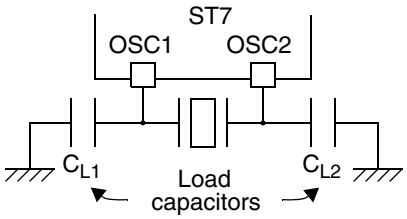
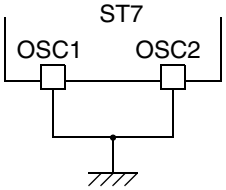
6.3.3 Internal RC oscillator

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.

In order not to exceed the maximum operating frequency, the internal RC oscillator must not be used with the PLL.

Table 9. ST7 clock sources

	Hardware configuration
External clock	
Crystal/ceramic resonators	
Internal RC oscillator	

Downloaded from Arrow.com.

6.4 Reset sequence manager (RSM)

The reset sequence manager includes three reset sources as shown in *Figure 14*:

- External reset source pulse
- Internal LVD reset
- Internal Watchdog reset

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

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

The basic reset sequence consists of three phases as shown in *Figure 13*:

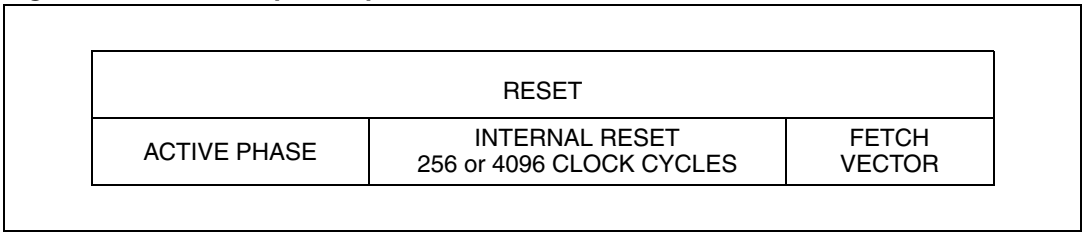
- Active Phase depending on the reset source
- 256 or 4096 CPU clock cycle delay (selected by option byte)
- Reset vector fetch

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

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

The reset vector fetch phase duration is two clock cycles.

Figure 13. Reset sequence phases



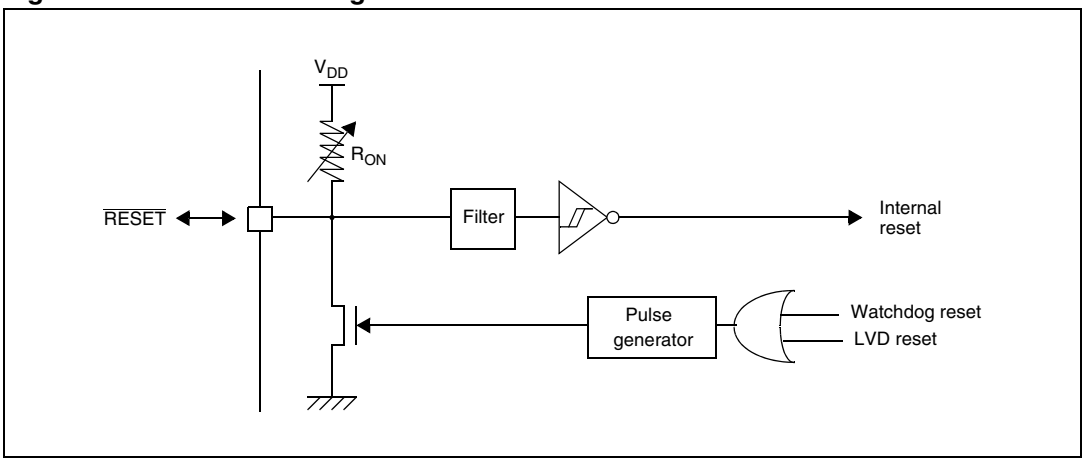
6.4.1 Asynchronous external $\overline{\text{RESET}}$ pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device. See the *Electrical characteristics* section for more details.

A reset signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see *Figure 15*). This detection is asynchronous and therefore the MCU can enter reset state even in Halt mode.

Downloaded from Arrow.com.

Figure 14. Reset block diagram



The $\overline{\text{RESET}}$ pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in the electrical characteristics section.

External power-on reset

If the LVD is disabled by option byte, to start up the microcontroller correctly, the user must ensure by means of an external reset circuit that the reset signal is held low until V_{DD} is over the minimum level specified for the selected f_{OSC} frequency.

A proper reset signal for a slow rising V_{DD} supply can generally be provided by an external RC network connected to the $\overline{\text{RESET}}$ pin.

Internal LVD reset

Two different reset sequences caused by the internal LVD circuitry can be distinguished:

- Power-On reset
- Voltage Drop reset

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD} < V_{IT+}$ (rising edge) or $V_{DD} < V_{IT-}$ (falling edge) as shown in *Figure 15*.

The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

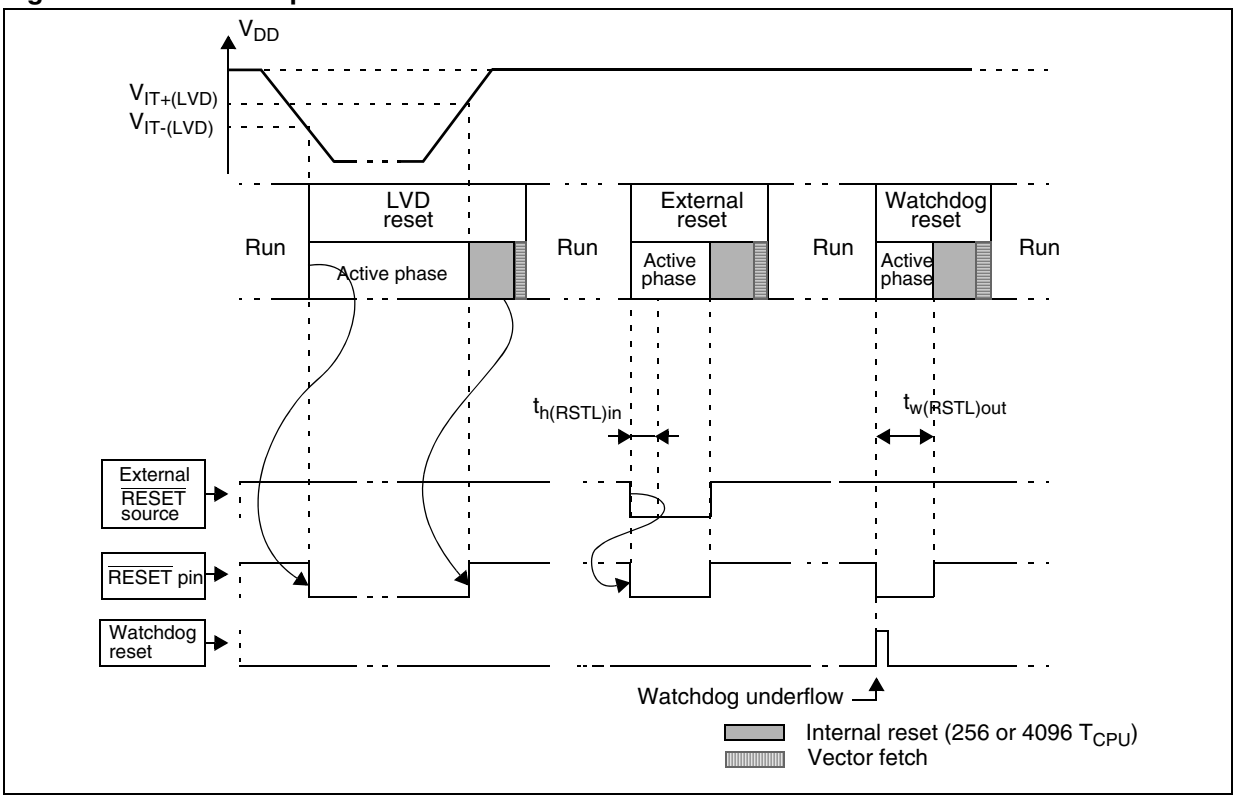
Internal Watchdog reset

The reset sequence generated by a internal Watchdog counter overflow is shown in *Figure 15*.

Starting from the Watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Downloaded from Arrow.com.

Figure 15. RESET sequences



6.5 System integrity management (SI)

The system integrity management block contains the LVD and auxiliary voltage detector (AVD) functions. It is managed by the SICSR register.

6.5.1 LVD (low voltage detector)

The LVD function generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD reset circuitry generates a reset when V_{DD} is below:

- V_{IT+} when V_{DD} is rising
- V_{IT-} when V_{DD} is falling

The LVD function is illustrated in *Figure 15*.

The voltage threshold can be configured by option byte to be low, medium or high.

Downloaded from Arrow.com.

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.

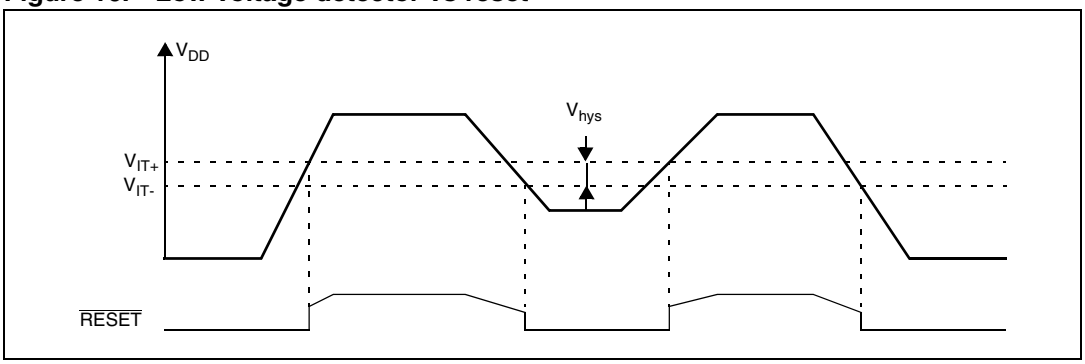
Note: 1 The LVD allows the device to be used without any external reset circuitry.

2 If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8 V, 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 16. 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 17*.

Downloaded from Arrow.com.

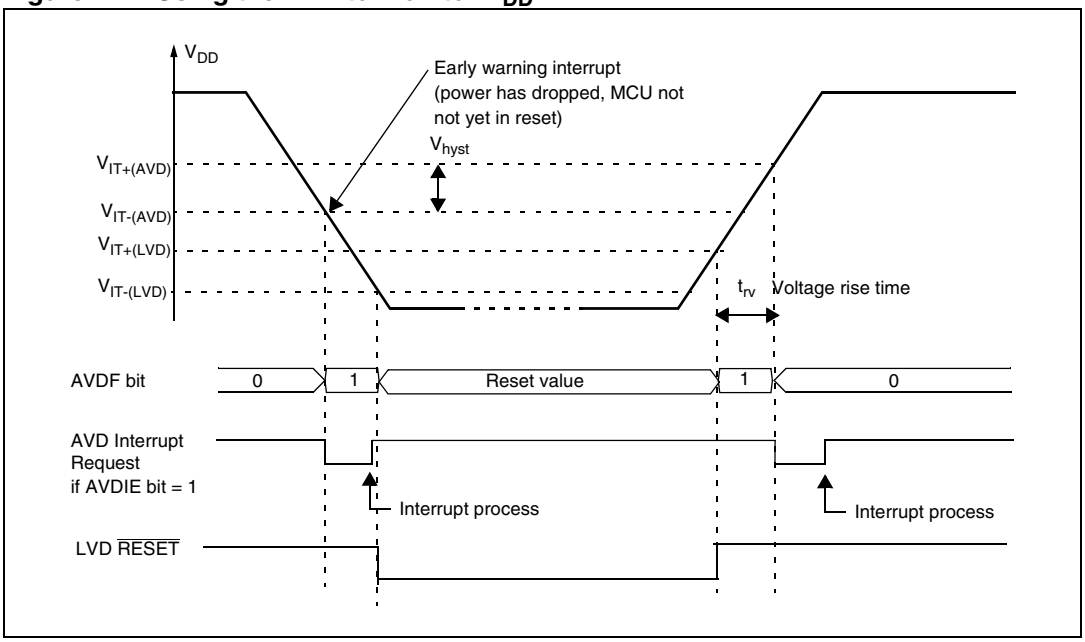
The interrupt on the rising edge is used to inform the application that the V_{DD} warning state is over.

If the voltage rise time t_{rv} is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt will be generated when $V_{IT+(AVD)}$ is reached.

If t_{rv} is greater than 256 or 4096 cycles then:

- If the AVD interrupt is enabled before the $V_{IT+(AVD)}$ threshold is reached, then 2 AVD interrupts will be received: the first when the AVDIE bit is set, and the second when the threshold is reached.
- If the AVD interrupt is enabled after the $V_{IT+(AVD)}$ threshold is reached then only one AVD interrupt will occur.

Figure 17. Using the AVD to monitor V_{DD}



6.5.3 Low power modes

Table 10. Effect of low power modes on SI

Mode	Description
Wait	No effect on SI. AVD interrupt causes the device to exit from Wait mode.
Halt	The CRSR register is frozen.

6.5.4 Interrupts

The AVD interrupt event generates an interrupt if the AVDIE bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 11. AVD interrupt control/wakeup capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
AVD event	AVDF	AVDIE	Yes	No

Downloaded from Arrow.com.

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 17* 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

Downloaded from Arrow.com.

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 cannot.

Caution: When the LVD is not activated with the associated option byte, the WDGRF flag can not be used in the application.

Downloaded from Arrow.com.

7 Interrupts

7.1 Introduction

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
  - up to 4 software programmable nesting levels
  - up to 16 interrupt vectors fixed by hardware
  - 2 non-maskable events: reset, TRAP

This interrupt management is based on:

- Bit 5 and bit 3 of the CPU CC register (I1:0)
- Interrupt software priority registers (ISPRx)
- Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

7.2 Masking and processing flow

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see *Table 14*). The processing flow is shown in *Figure 18.*

When an interrupt request has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to *Table 25: Interrupt mapping* for vector addresses).

The interrupt service routine should end with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

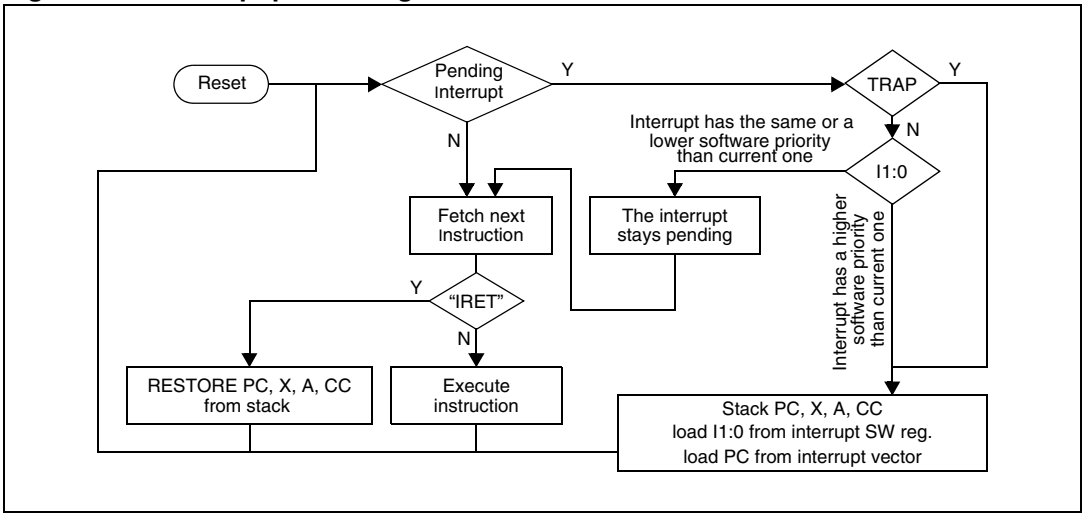
Note: As a consequence of the IRET instruction, the I1 and I0 bits will be restored from the stack and the program in the previous level will resume.

Downloaded from Arrow.com.

Table 14. Interrupt software priority levels

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1	↓	0	1
Level 2	↓	0	0
Level 3 (= interrupt disable)	High	1	1

Figure 18. Interrupt processing flowchart



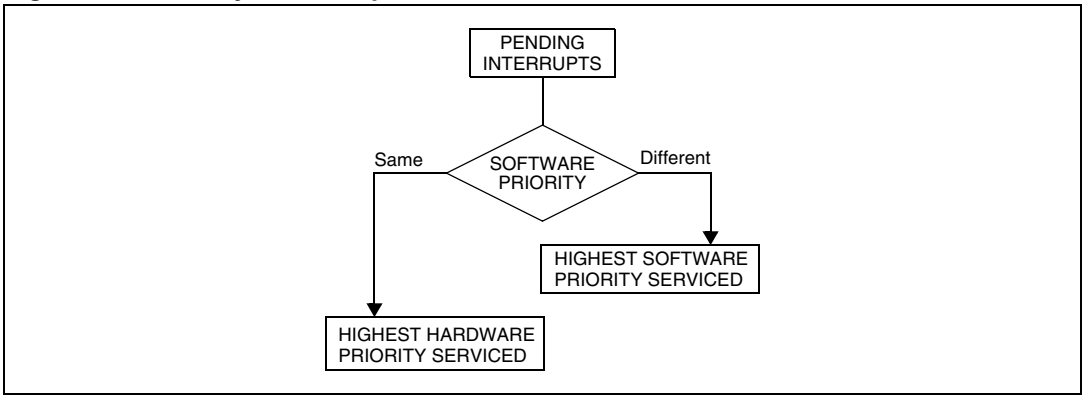
7.2.1 Servicing pending interrupts

As several interrupts can be pending at the same time, the interrupt to be taken into account is determined by the following two-step process:

- the highest software priority interrupt is serviced,
- if several interrupts have the same software priority then the interrupt with the highest hardware priority is serviced first.

Figure 19 describes this decision process.

Figure 19. Priority decision process flowchart



Downloaded from Arrow.com.

When an interrupt request is not serviced immediately, it is latched and then processed when its software priority combined with the hardware priority becomes the highest one.

Note: 1 The hardware priority is exclusive while the software one is not. This allows the previous process to succeed with only one interrupt.

2 Reset and TRAP can be considered as having the highest software priority in the decision process.

7.2.2 Different interrupt vector sources

Two interrupt source types are managed by the ST7 interrupt controller: the non-maskable type (reset, TRAP) and the maskable type (external or from internal peripherals).

7.2.3 Non-maskable sources

These sources are processed regardless of the state of the I1 and I0 bits of the CC register (see *Figure 18*). After stacking the PC, X, A and CC registers (except for reset), the corresponding vector is loaded in the PC register and the I1 and I0 bits of the CC are set to disable interrupts (level 3). These sources allow the processor to exit Halt mode.

TRAP (non-maskable software interrupt)

This software interrupt is serviced when the TRAP instruction is executed. It will be serviced according to the flowchart in *Figure 18*.

Reset

The reset source has the highest priority in the ST7. This means that the first current routine has the highest software priority (level 3) and the highest hardware priority.

See the reset chapter for more details.

7.2.4 Maskable sources

Maskable interrupt vector sources can be serviced if the corresponding interrupt is enabled and if its own interrupt software priority (in ISPRx registers) is higher than the one currently being serviced (I1 and I0 in CC register). If any of these two conditions is false, the interrupt is latched and thus remains pending.

External interrupts

External interrupts allow the processor to Exit from Halt low power mode. External interrupt sensitivity is software selectable through the External Interrupt Control register (EICR).

External interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.

If several input pins of a group connected to the same interrupt line are selected simultaneously, these will be logically ORed.

Peripheral interrupts

Usually the peripheral interrupts cause the MCU to Exit from Halt mode except those mentioned in *Table 25: Interrupt mapping*. A peripheral interrupt occurs when a specific flag is set in the peripheral status registers and if the corresponding enable bit is set in the

Downloaded from Arrow.com.

peripheral control register. The general sequence for clearing an interrupt is based on an access to the status register followed by a read or write to an associated register.

Note: The clearing sequence resets the internal latch. A pending interrupt (that is, waiting to be serviced) is therefore lost if the clear sequence is executed.

7.3 Interrupts and low power modes

All interrupts allow the processor to exit the Wait low power mode. On the contrary, only external and other specified interrupts allow the processor to exit from the Halt modes (see column Exit from Halt in *Table 25: Interrupt mapping*). When several pending interrupts are present while exiting Halt mode, the first one serviced can only be an interrupt with Exit from Halt mode capability and it is selected through the same decision process shown in *Figure 19*.

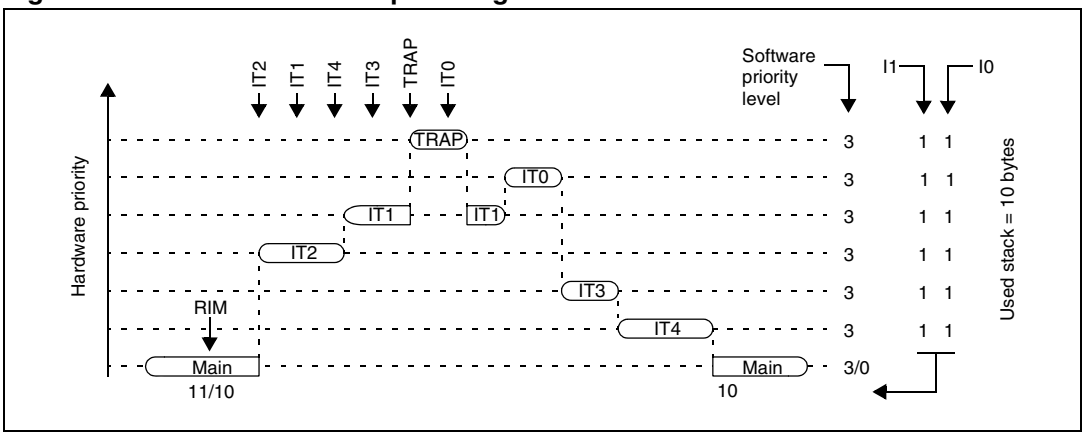
Note: If an interrupt, that is not able to exit from Halt mode, is pending with the highest priority when exiting Halt mode, this interrupt is serviced after the first one serviced.

7.4 Concurrent and nested management

Figure 20 and *Figure 21* show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in *Figure 21*. The interrupt hardware priority is given in order from the lowest to the highest as follows: MAIN, IT4, IT3, IT2, IT1, IT0. Software priority is given for each interrupt.

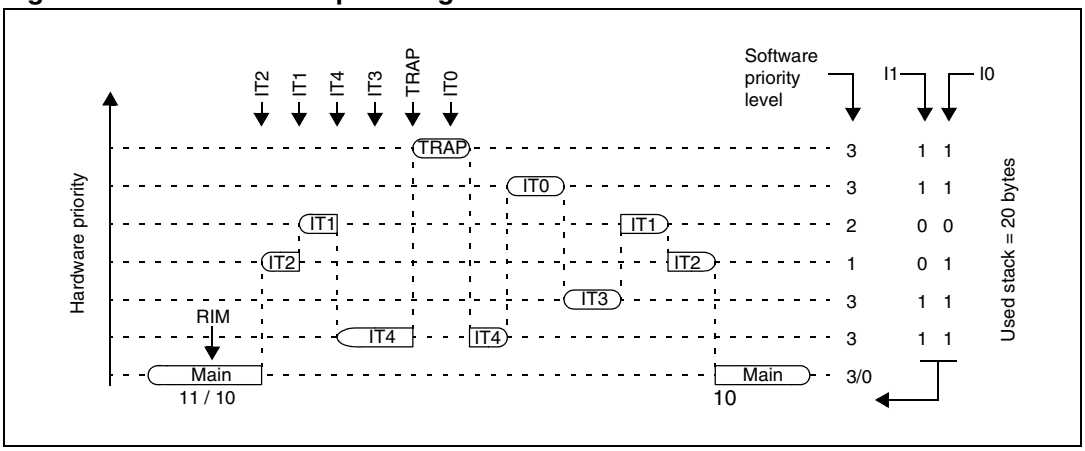
Warning: A stack overflow may occur without notifying the software of the failure.

Figure 20. Concurrent interrupt management



Downloaded from Arrow.com.

Figure 21. Nested interrupt management



7.5 Interrupt registers

7.5.1 CPU CC register interrupt bits

CPU CC Reset value: 111x 1010(xAh)

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 15. CPU CC register interrupt bits description

Bit	Name	Function
5	I1	Software Interrupt Priority 1
3	I0	Software Interrupt Priority 0

Table 16. Interrupt software priority levels

Interrupt software priority	Level	I1	I0
Level 0 (main)	Low	1	0
Level 1	↓	0	1
Level 2	↓	0	0
Level 3 (= interrupt disable)(1)	High	1	1

1. TRAP and RESET events can interrupt a level 3 program.

These two bits indicate the current interrupt software priority (see *Table 16*) and 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, HALT, WFI, IRET and PUSH/POP instructions (see *Table 18: Dedicated interrupt instruction set*).

Downloaded from Arrow.com.

7.5.2 Interrupt software priority registers (ISPRx)

ISPRx Reset value: 1111 1111 (FFh)

	7	6	5	4	3	2	1	0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ISPR3	1	1	1	1	I1_13	I0_13	I1_12	I0_12
	RO	RO	RO	RO	R/W	R/W	R/W	R/W

These four registers contain the interrupt software priority of each interrupt vector.

- Each interrupt vector (except reset and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondence is shown in the following *Table 17*.

Table 17. ISPRx interrupt vector correspondence

Vector address	ISPRx bits
FFFBh-FFFAh	I1_0 and I0_0 bits
FFF9h-FFF8h	I1_1 and I0_1 bits
...	...
FFE1h-FFE0h	I1_13 and I0_13 bits

- Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.
- Level 0 cannot be written (I1_x = 1, I0_x = 0). In this case, the previously stored value is kept (for example, previous value = CFh, write = 64h, result = 44h).

The reset, and TRAP vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

Caution: If the I1_x and I0_x bits are modified while the interrupt x is executed the following behavior has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

Table 18. Dedicated interrupt instruction set(1)

Instruction	New description	Function/example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	POP CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0=11 (level 3)	I1:0=11 ?						
JRNM	Jump if I1:0<>11	I1:0<>11 ?						

Downloaded from Arrow.com.

Table 18. Dedicated interrupt instruction set(1) (continued)

Instruction	New description	Function/example	I1	H	I0	N	Z	C
POP CC	POP CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load 10 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load 11 in I1:0 of CC	1		1			
TRAP	Software TRAP	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

1. During the execution of an interrupt routine, the HALT, POP CC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

Downloaded from Arrow.com.

7.6 External interrupts

7.6.1 I/O port interrupt sensitivity

The external interrupt sensitivity is controlled by the IPA, IPB and ISxx bits of the EICR register (*Figure 22*). This control allows up to four fully independent external interrupt source sensitivities.

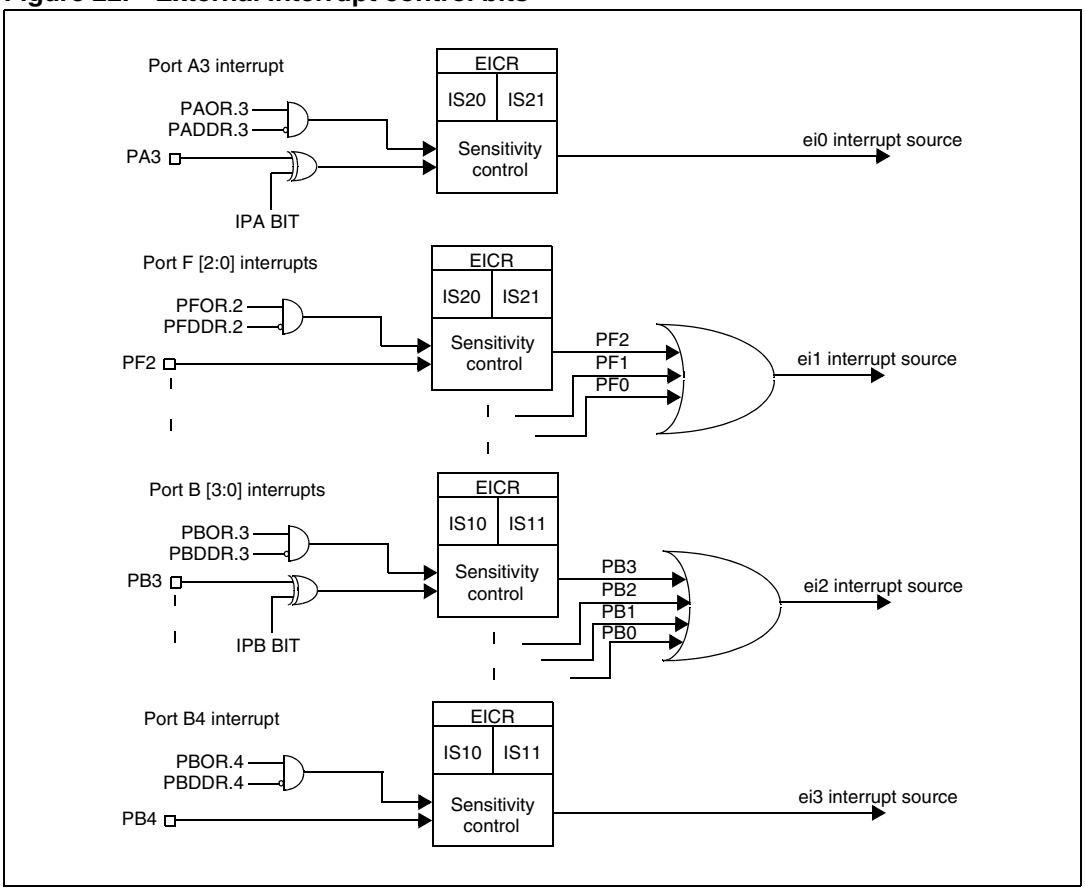
Each external interrupt source can be generated on four (or five) different events on the pin:

- Falling edge
- Rising edge
- Falling and rising edge
- Falling edge and low level
- Rising edge and high level (only for ei0 and ei2)

To guarantee correct functionality, the sensitivity bits in the EICR register can be modified only when the I1 and I0 bits of the CC register are both set to 1 (level 3). This means that interrupts must be disabled before changing sensitivity.

The pending interrupts are cleared by writing a different value in the ISx[1:0], IPA or IPB bits of the EICR.

Figure 22. External interrupt control bits



Downloaded from Arrow.com.

7.6.2 External interrupt control register (EICR)

EICR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
IS11	IS10	IPB	IS21	IS20	IPA	Reserved	
R/W	R/W	R/W	R/W	R/W	R/W	-	

Table 19. EICR register description

Bit	Name	Function
7:6	IS1[1:0]	ei2 and ei3 sensitivity The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the following external interrupts: - ei2 for port B [3:0] (see *Table 20*) - ei3 for port B4 (see *Table 21*) Bits 7 and 6 can only be written when I1 and I0 of the CC register are both set to 1 (level 3).
5	IPB	Interrupt Polarity (for port B) This bit is used to invert the sensitivity of port B [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3). 0: No sensitivity inversion 1: Sensitivity inversion
4:3	IS2[1:0]	ei0 and ei1 sensitivity The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the following external interrupts: - ei0 for port A[3:0] (see *Table 22*) - ei1 for port F[2:0] (see *Table 23*) Bits 4 and 3 can only be written when I1 and I0 of the CC register are both set to 1 (level 3).
2	IPA	Interrupt Polarity (for port A) This bit is used to invert the sensitivity of port A [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3). 0: No sensitivity inversion. 1: Sensitivity inversion.
1:0	-	Reserved, must always be kept cleared

Table 20. Interrupt sensitivity - ei2

IS11	IS10	External interrupt sensitivity	
		IPB bit = 0	IPB bit = 1
0	0	Falling edge and low level	Rising edge and high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

Downloaded from Arrow.com.

Table 21. Interrupt sensitivity - ei3

IS11	IS10	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 22. Interrupt sensitivity - ei0

IS21	IS20	External interrupt sensitivity	
		IPA bit = 0	IPA bit = 1
0	0	Falling edge and low level	Rising edge and high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

Table 23. Interrupt sensitivity - ei1

IS21	IS20	External interrupt sensitivity
0	0	Falling edge and low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

Table 24. Nested interrupts register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0024h	ISPR0	ei1		ei0		MCC + SI			
	reset value	I1_3 1	I0_3 1	I1_2 1	I0_2 1	I1_1 1	I0_1 1	1	1
0025h	ISPR1	SPI				ei3		ei2	
	reset value	I1_7 1	I0_7 1	I1_6 1	I0_6 1	I1_5 1	I0_5 1	I1_4 1	I0_4 1
0026h	ISPR2	AVD		SCI		Timer B		Timer A	
	reset value	I1_11 1	I0_11 1	I1_10 1	I0_10 1	I1_9 1	I0_9 1	I1_8 1	I0_8 1
0027h	ISPR3 reset value	1	1	1	1	I1_13 1	I0_13 1	I1_12 1	I0_12 1
0028h	EICR reset value	IS11 0	IS10 0	IPB 0	IS21 0	IS20 0	IPA 0	0	0

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

8 Power saving modes

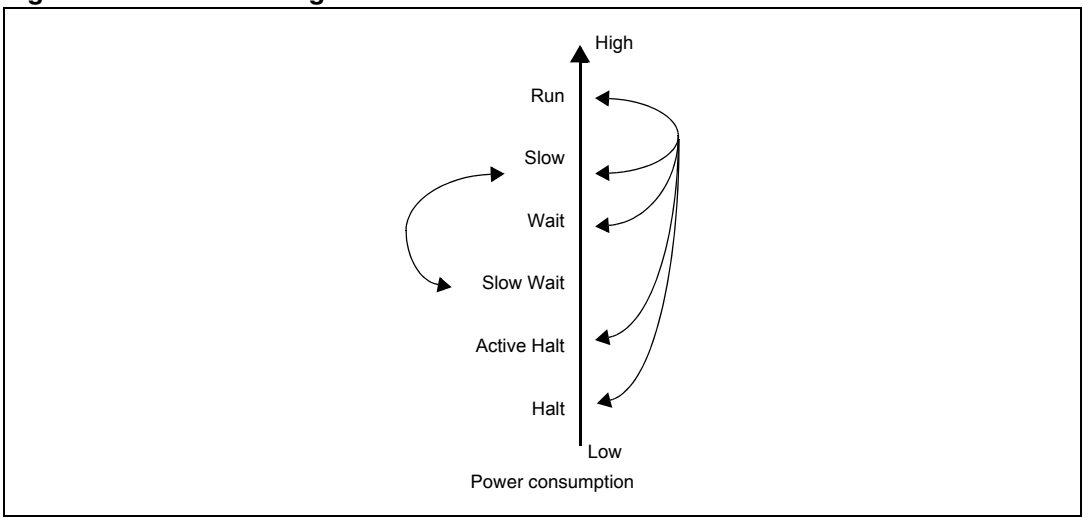
8.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see *Figure 23*): Slow, Wait (Slow Wait), Active-halt and Halt.

After a reset the normal operating mode is selected by default (Run mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided or multiplied by 2 (f_{OSC2}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 23. Power saving mode transitions



8.2 Slow mode

This mode has two targets:

- To reduce power consumption by decreasing the internal clock in the device,
- To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

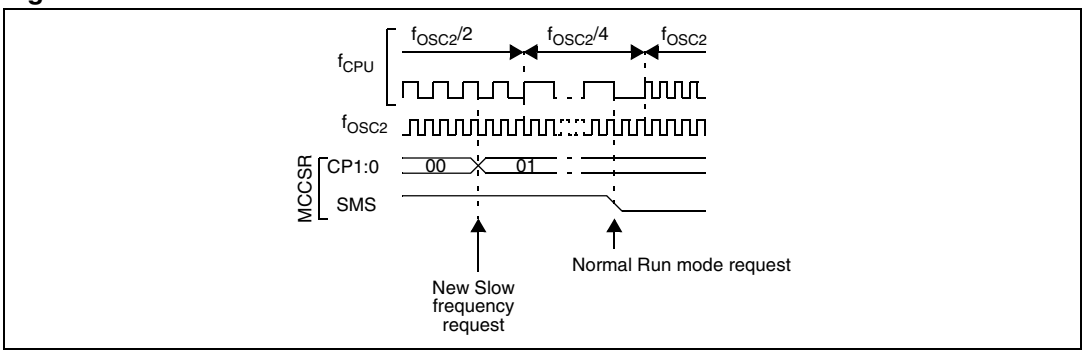
Slow mode is controlled by three bits in the MCCSR register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency (f_{CPU}).

In this mode, the master clock frequency (f_{OSC2}) can be divided by 2, 4, 8 or 16. The CPU and peripherals are clocked at this lower frequency (f_{CPU}).

Note: Slow-Wait mode is activated when entering the Wait mode while the device is already in Slow mode.

Downloaded from Arrow.com.

Figure 24. Slow mode clock transitions



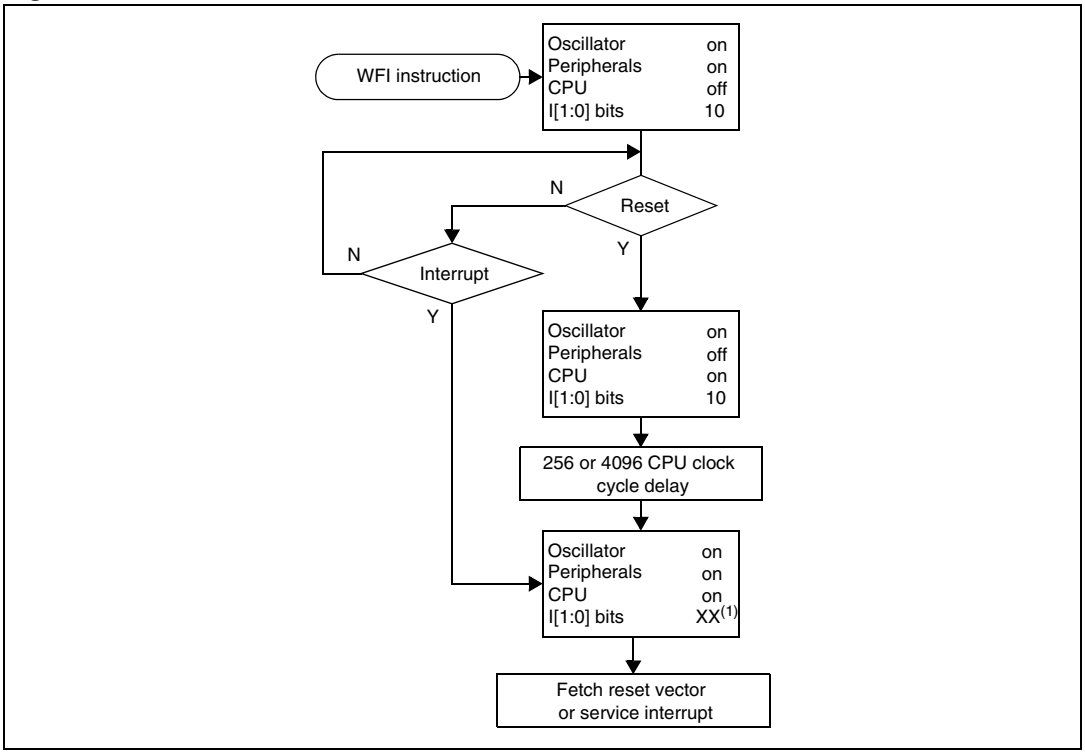
8.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the 'WFI' instruction.

All peripherals remain active. During Wait mode, the I[1:0] bits of the CC register are forced to '10', to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or reset occurs, whereupon the Program Counter branches to the starting address of the interrupt or reset service routine. The MCU will remain in Wait mode until a reset or an interrupt occurs, causing it to wake up. Refer to *Figure 25*.

Figure 25. Wait mode flowchart



1. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

Downloaded from Arrow.com.

8.4 Active-halt and Halt modes

Active-halt and Halt modes are the two lowest power consumption modes of the MCU. They are both entered by executing the 'HALT' instruction. The decision to enter either in Active-halt or Halt mode is given by the MCC/RTC interrupt enable flag (OIE bit in the MCCSR register).

Table 26. MCC/RTC low power mode selection

MCCSR OIE bit	Power saving mode entered when HALT instruction is executed
0	Halt mode
1	Active-halt mode

8.4.1 Active-halt mode

Active-halt mode is the lowest power consumption mode of the MCU with a real-time clock available. It is entered by executing the 'HALT' instruction when the OIE bit of the Main Clock Controller Status register (MCCSR) is set (see *Section 10.2: Main clock controller with real-time clock and beeper (MCC/RTC) on page 69* for more details on the MCCSR register).

The MCU can exit Active-halt mode on reception of either an MCC/RTC interrupt, a specific interrupt (see *Table 25: Interrupt mapping*) or a reset. When exiting Active-halt mode by means of an interrupt, no 256 or 4096 CPU cycle delay occurs. The CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see *Figure 27*).

When entering Active-halt mode, the I[1:0] bits in the CC register are forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

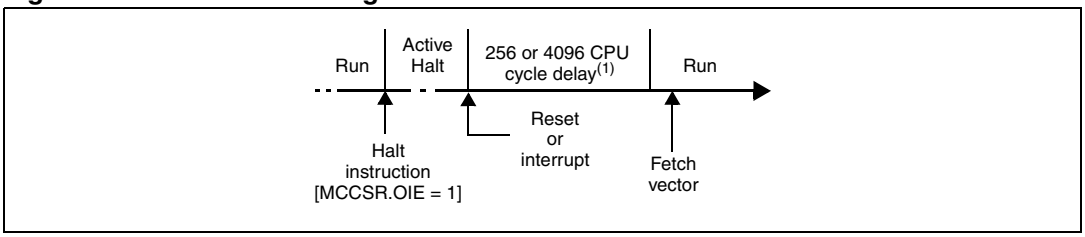
In Active-halt mode, only the main oscillator and its associated counter (MCC/RTC) are running to keep a wakeup time base. All other peripherals are not clocked except those which get their clock supply from another clock generator (such as external or auxiliary oscillator).

The safeguard against staying locked in Active-halt mode is provided by the oscillator interrupt.

Note: As soon as the interrupt capability of one of the oscillators is selected (MCCSR.OIE bit set), entering Active-halt mode while the Watchdog is active does not generate a reset. This means that the device cannot spend more than a defined delay in this power saving mode.

Caution: When exiting Active-halt mode following an interrupt, OIE bit of MCCSR register must not be cleared before t_{DELAY} after the interrupt occurs (t_{DELAY} = 256 or 4096 t_{CPU} delay depending on option byte). Otherwise, the ST7 enters Halt mode for the remaining t_{DELAY} period.

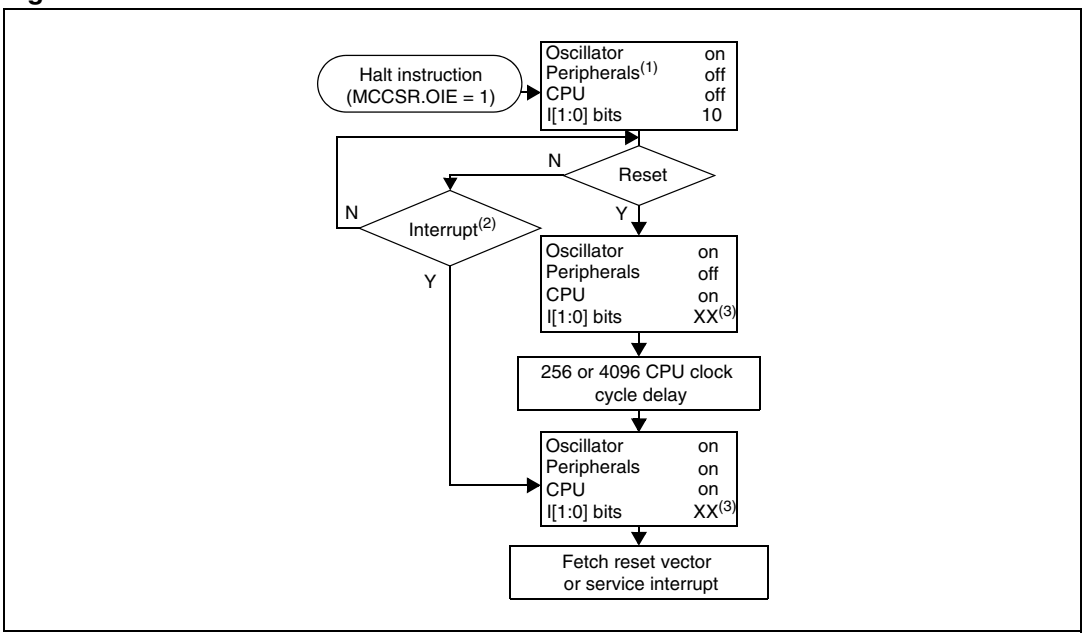
Figure 26. Active-halt timing overview



1. This delay occurs only if the MCU exits Active-halt mode by means of a reset.

Downloaded from Arrow.com.

Figure 27. Active-halt mode flowchart



1. Peripheral clocked with an external clock source can still be active.
2. Only the MCC/RTC interrupt and some specific interrupts can exit the MCU from Active-halt mode (such as external interrupt). Refer to *Table 25: Interrupt mapping on page 51* for more details.
3. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and restored when the CC register is popped.

8.4.2 Halt mode

The Halt mode is the lowest power consumption mode of the MCU. It is entered by executing the 'HALT' instruction when the OIE bit of the Main Clock Controller Status register (MCCSR) is cleared (see *Section 10.2: Main clock controller with real-time clock and beeper (MCC/RTC) on page 69* for more details on the MCCSR register).

The MCU can exit Halt mode on reception of either a specific interrupt (see *Table 25: Interrupt mapping*) or a reset. When exiting Halt mode by means of a reset or an interrupt, the oscillator is immediately turned on and the 256 or 4096 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see *Figure 29*).

When entering Halt mode, the I[1:0] bits in the CC register are forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In Halt mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with Halt mode is configured by the "WDGHALT" option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog reset (see *Section 14.1 on page 179*) for more details.

Downloaded from Arrow.com.

Figure 28. Halt timing overview

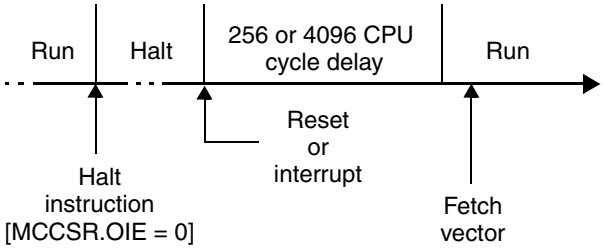
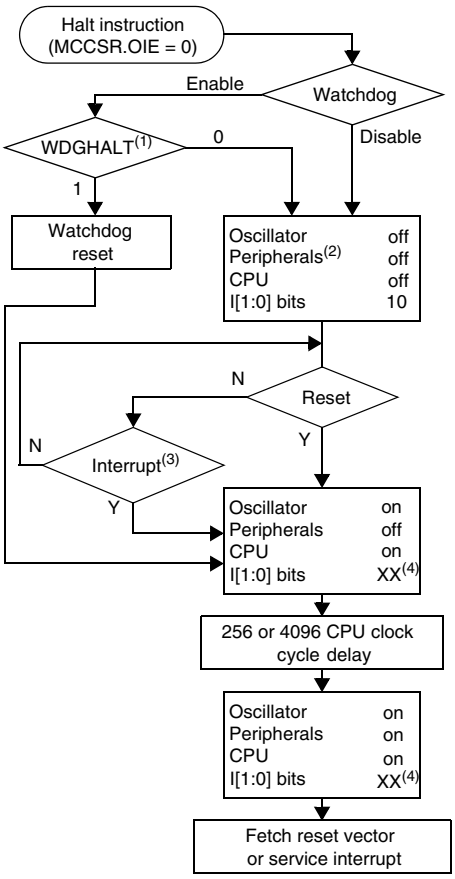


Figure 29. Halt mode flowchart



1. WDGHALT is an option bit. See *Section 14.1 on page 179* for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only some specific interrupts can exit the MCU from Halt mode (such as external interrupt). Refer to *Table 25: Interrupt mapping* for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

Downloaded from Arrow.com.

Halt mode recommendations

- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- When using an external interrupt to wake up the microcontroller, reinitialize the corresponding I/O as “Input Pull-up with Interrupt” before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.
- For the same reason, reinitialize the sensitivity level of each external interrupt as a precautionary measure.
- The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant in ROM with the value 0x8E.
- As the HALT instruction clears the interrupt mask in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wakeup event (reset or external interrupt).

Downloaded from Arrow.com.

9 I/O ports

9.1 Introduction

The I/O ports offer different functional modes:

- transfer of data through digital inputs and outputs,

and for specific pins:

- external interrupt generation,
- alternate signal input/output for the on-chip peripherals.

An I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

9.2 Functional description

Each port has two main registers:

- Data Register (DR)
- Data Direction Register (DDR)

and one optional register:

- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, (for specific ports which do not provide this register refer to *Section 9.3: I/O port implementation on page 62*). The generic I/O block diagram is shown in *Figure 30.*

9.2.1 Input modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Note: 1 Writing the DR register modifies the latch value but does not affect the pin status.

2 When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

3 Do not use read/modify/write instructions (BSET or BRES) to modify the DR register as this might corrupt the DR content for I/Os configured as input.

Downloaded from Arrow.com.

External interrupt function

When an I/O is configured as 'Input with Interrupt', an event on this I/O can generate an external interrupt request to the CPU.

Each pin can independently generate an interrupt request. The interrupt sensitivity is independently programmable using the sensitivity bits in the EICR register.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see pinout description and interrupt section). If several input pins are selected simultaneously as interrupt sources, these are first detected according to the sensitivity bits in the EICR register and then logically ORed.

The external interrupts are hardware interrupts, which means that the request latch (not accessible directly by the application) is automatically cleared when the corresponding interrupt vector is fetched. To clear an unwanted pending interrupt by software, the sensitivity bits in the EICR register must be modified.

9.2.2 Output modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

Table 27. DR register value and output pin status

DR	Push-pull	Open-drain
0	V_{SS}	V_{SS}
1	V_{DD}	Floating

9.2.3 Alternate functions

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over the standard I/O programming.

When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin must be configured in input mode. In this case, the pin state is also digitally readable by addressing the DR register.

Note: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on-chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.

Downloaded from Arrow.com.

Figure 30. I/O port general block diagram

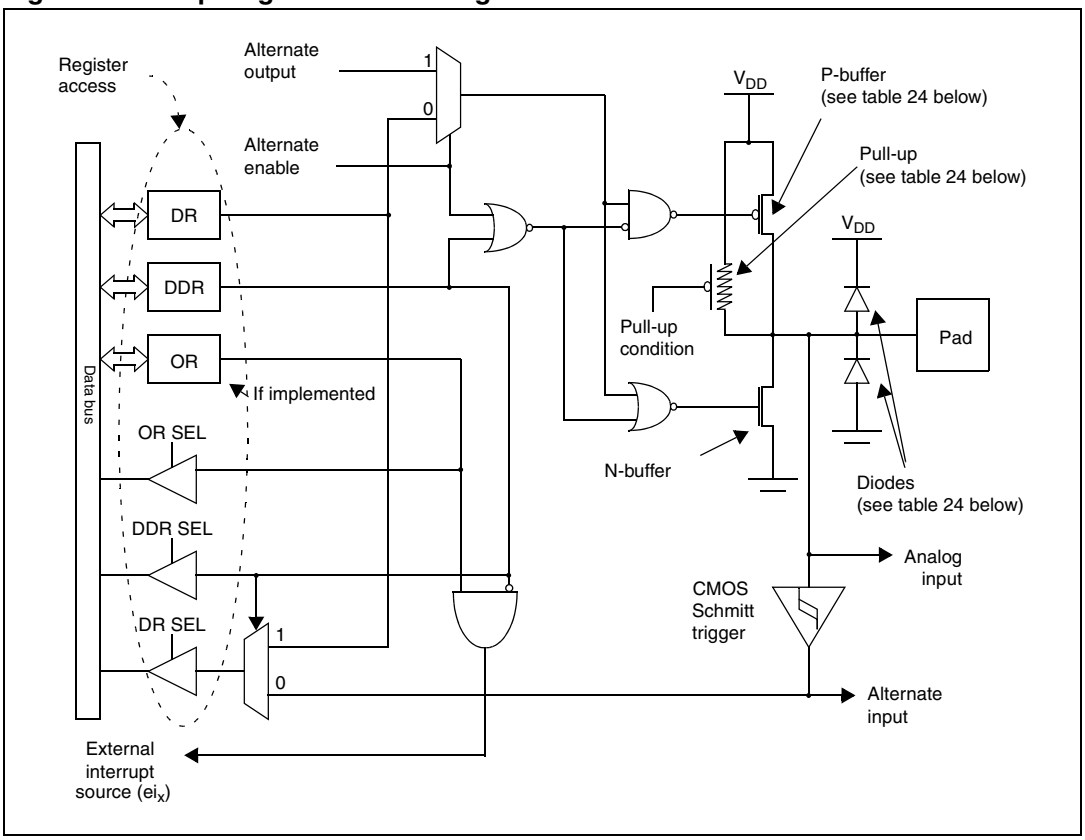


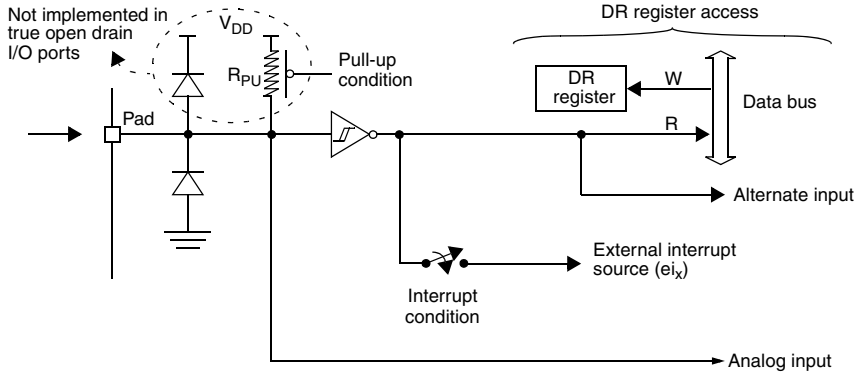
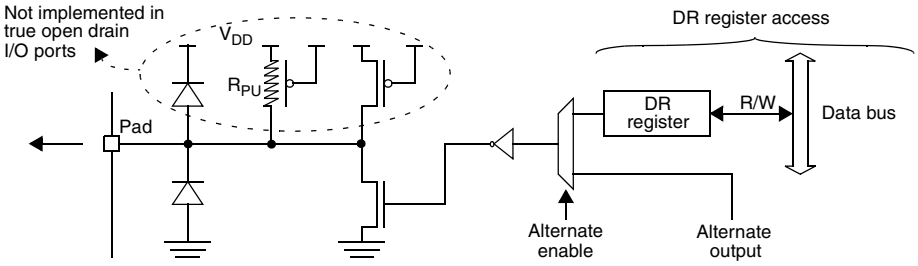
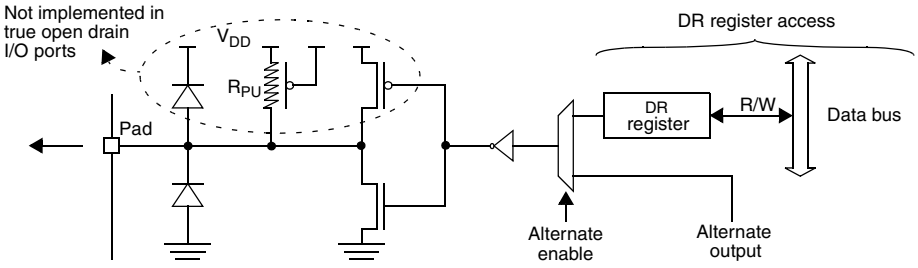
Table 28. I/O port mode options

Configuration mode		Pull-up	P-buffer	Diodes	
				to V_{DD}(1)	to V_{SS}(2)
Input	Floating with/without Interrupt	Off(3)	Off	On	On
	Pull-up with/without Interrupt	On(4)			
Output	Push-pull	Off	On		
	Open drain (logic level)		Off		
	True open drain	NI	NI	NI(5)	

1. The diode to V_{DD} is not implemented in the true open drain pads.
2. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.
3. Off = implemented not activated.
4. On = implemented and activated.
5. NI = not implemented

Downloaded from Arrow.com.

Table 29. I/O port configurations

	Hardware configuration
Input(1)	
Open-drain output(2)	
PUSH-pull output(2)	

1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

Caution: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

Downloaded from Arrow.com.

Analog alternate function

When the pin is used as an ADC input, the I/O must be configured as floating input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail which is connected to the ADC input.

It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

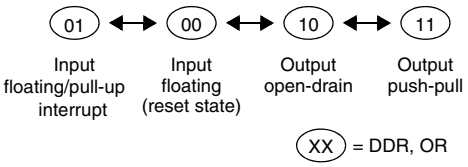
Warning: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

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 feature of the I/O port such as ADC Input or true 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 31.* Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 31. Interrupt I/O port state transitions



9.4 Low power modes

Table 30. Effect of low power modes on I/O ports

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.

9.5 Interrupts

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and the interrupt mask in the CC register is not active (RIM instruction).

Downloaded from Arrow.com.

Table 31. I/O port interrupt control/wakeup capability

Interrupt event	Event flag	Enable Control bit	Exit from Wait	Exit from Halt
External interrupt on selected external event	-	DDRx, ORx	Yes	Yes

9.5.1 I/O port implementation

The I/O port register configurations are summarized *Table 32*.

Table 32. Port configuration

Port	Pin name	Input (DDR = 0)		Output (DDR = 1)	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7:6	Floating		True open-drain (high sink)	
	PA5:4	Floating	Pull-up	Open drain	Push-pull
	PA3	Floating	Floating interrupt	Open drain	Push-pull
Port B	PB3	Floating	Floating interrupt	Open drain	Push-pull
	PB4, PB2:0	Floating	Pull-up	Open drain	Push-pull
Port C	PC7:0	Floating	Pull-up	Open drain	Push-pull
Port D	PD5:0	Floating	Pull-up	Open drain	Push-pull
Port E	PE1:0	Floating	Pull-up	Open drain	Push-pull
Port F	PF7:6, 4	Floating	Pull-up	Open drain	Push-pull
	PF2:0	Floating	Pull-up	Open drain	Push-pull

Table 33. I/O port register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Reset value of all I/O port registers		0	0	0	0	0	0	0	0
0000h	PADR	MSB							LSB
0001h	PADDR								
0002h	PAOR								
0003h	PBDR	MSB							LSB
0004h	PBDDR								
0005h	PBOR								
0006h	PCDR	MSB							LSB
0007h	PCDDR								
0008h	PCOR								
0009h	PDDR	MSB							LSB
000Ah	PDDDR								
000Bh	PDOR								

Downloaded from Arrow.com.

Table 33. I/O port register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
000Ch	PEDR	MSB							LSB
000Dh	PEDDR								
000Eh	PEOR								
000Fh	PFDR	MSB							LSB
0010h	PFDDR								
0011h	PFOR								

Downloaded from Arrow.com.

10 On-chip peripherals

10.1 Watchdog timer (WDG)

10.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

10.1.2 Main features

- Programmable free-running downcounter
- Programmable reset
- Reset (if Watchdog activated) when the T6 bit reaches zero
- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte

10.1.3 Functional description

The counter value stored in the Watchdog Control register (WDGCR bits T[6:0]), is decremented every 16384 f_{OSC2} cycles (approx.), and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 30µs.

The application program must write in the WDGCR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: it counts down even if the watchdog is disabled. The value to be stored in the WDGCR register must be between FFh and C0h:

- The WDGA bit is set (Watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the Watchdog produces a reset (see *Figure 33: Approximate timeout duration*). The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WDGCR register (see *Figure 34*).

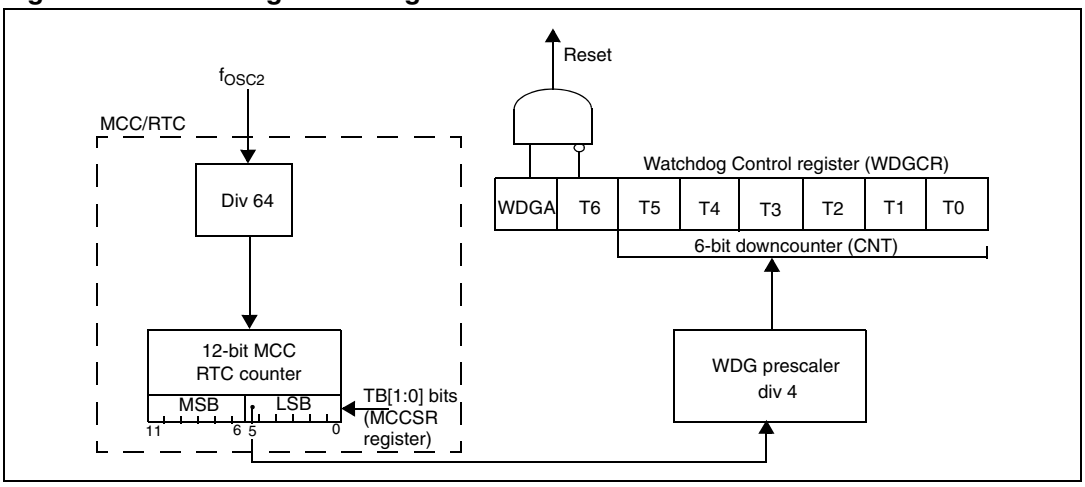
Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the Watchdog is activated, the HALT instruction generates a reset.

Downloaded from Arrow.com.

Figure 32. Watchdog block diagram

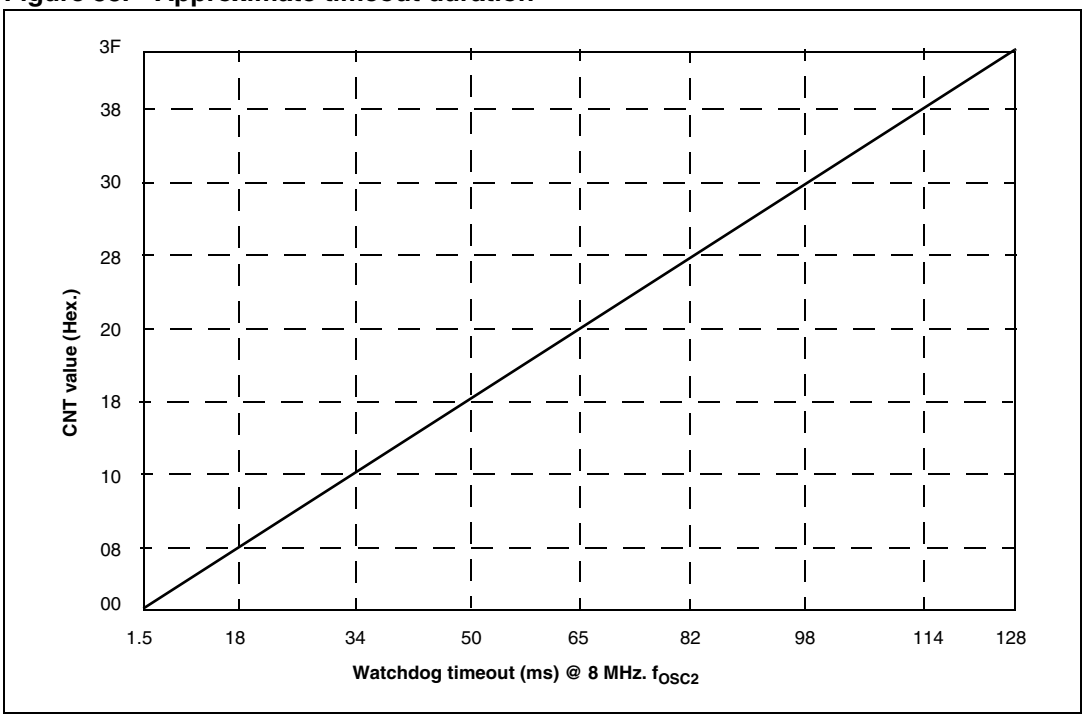


10.1.4 How to program the Watchdog timeout

Figure 33 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 more precision is needed, use the formulae in *Figure 34*.

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

Figure 33. Approximate timeout duration



Downloaded from Arrow.com.

Figure 34. Exact timeout duration (t_{min} and t_{max})

WHERE:

t_{min0} = (LSB + 128) x 64 x t_{OSC2}

t_{max0} = 16384 x t_{OSC2}

t_{OSC2} = 125 ns if f_{OSC2} = 8 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	2 ms	4	59
0	1	4 ms	8	53
1	0	10 ms	20	35
1	1	25 ms	49	54

To calculate the minimum Watchdog timeout (t_{min}):

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

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

To calculate the maximum Watchdog timeout (t_{max}):

IF $CNT \leq \left\lceil \frac{MSB}{4} \right\rceil$ **THEN** $t_{max} = t_{max0} + 16384 \times CNT \times t_{osc2}$

ELSE $t_{max} = t_{max0} + \left[16384 \times \left(CNT - \left\lceil \frac{4CNT}{MSB} \right\rceil \right) + (192 + LSB) \times 64 \times \left\lceil \frac{4CNT}{MSB} \right\rceil \right] \times t_{osc2}$

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

EXAMPLE: With 2ms timeout selected in MCCSR register

Value of T[5:0] bits in WDGCR register (Hex.)	Min. Watchdog timeout (ms) t_{min}	Max. Watchdog timeout (ms) t_{max}
00	1.496	2.048
3F	128	128.552

Downloaded from Arrow.com.

10.1.5 Low power modes

Table 34. Effect of lower power modes on Watchdog

Mode	Description		
Slow	No effect on Watchdog		
Wait			
Halt	**OIE bit in MCCSR register**	**WDGHALT bit in option byte**	
	0	0	No Watchdog reset is generated. The MCU enters Halt mode. The Watchdog counter is decremented once and then stops counting and is no longer able to generate a watchdog reset until the MCU receives an external interrupt or a reset. If an external interrupt is received, the Watchdog restarts counting after 256 or 4096 CPU clocks. If a reset is generated, the Watchdog is disabled (reset state) unless Hardware Watchdog is selected by option byte. For application recommendations, see *Section 10.1.7* below.
	0	1	A reset is generated.
	1	x	No reset is generated. The MCU enters Active-halt mode. The Watchdog counter is not decremented. It stop counting. When the MCU receives an oscillator interrupt or external interrupt, the Watchdog restarts counting immediately. When the MCU receives a reset the Watchdog restarts counting after 256 or 4096 CPU clocks.

10.1.6 Hardware Watchdog option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the WDGCR is not used. Refer to the option byte description in *Section 14.1: Flash devices*.

10.1.7 Using Halt mode with the WDG (WDGHALT option)

The following recommendation applies if Halt mode is used when the watchdog is enabled: Before executing the HALT instruction, refresh the WDG counter to avoid an unexpected WDG reset immediately after waking up the microcontroller.

10.1.8 Interrupts

None.

Downloaded from Arrow.com.

10.1.9 Control register (WDGCR)

WDGCR Reset value: 0111 1111 (7Fh)

7	6	5	4	3	2	1	0
WDGA	T[6:0]						
R/W	R/W						

Table 35. WDGCR register description

Bit	Name	Function
7	WDGA	Activation bit This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset. 0: Watchdog disabled 1: Watchdog enabled *Note: This bit is not used if the hardware watchdog option is enabled by option byte.*
6:0	T[6:0]	7-bit counter (MSB to LSB) These bits contain the value of the Watchdog counter, which is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 is cleared).

Table 36. Watchdog timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Ah	WDGCR reset value	WDGA 0	T6 1	T5 1	T4 1	T3 1	T2 1	T1 1	T0 1

10.2 Main clock controller with real-time clock and beeper (MCC/RTC)

The main clock controller consists of three different functions:

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

Each function can be used independently and simultaneously.

10.2.1 Programmable CPU clock prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages Slow power saving mode (see *Section 8.2: Slow mode on page 52* for more details).

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

Downloaded from Arrow.com.

10.2.2 Clock-out capability

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

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

10.2.3 Real-time clock (RTC) timer

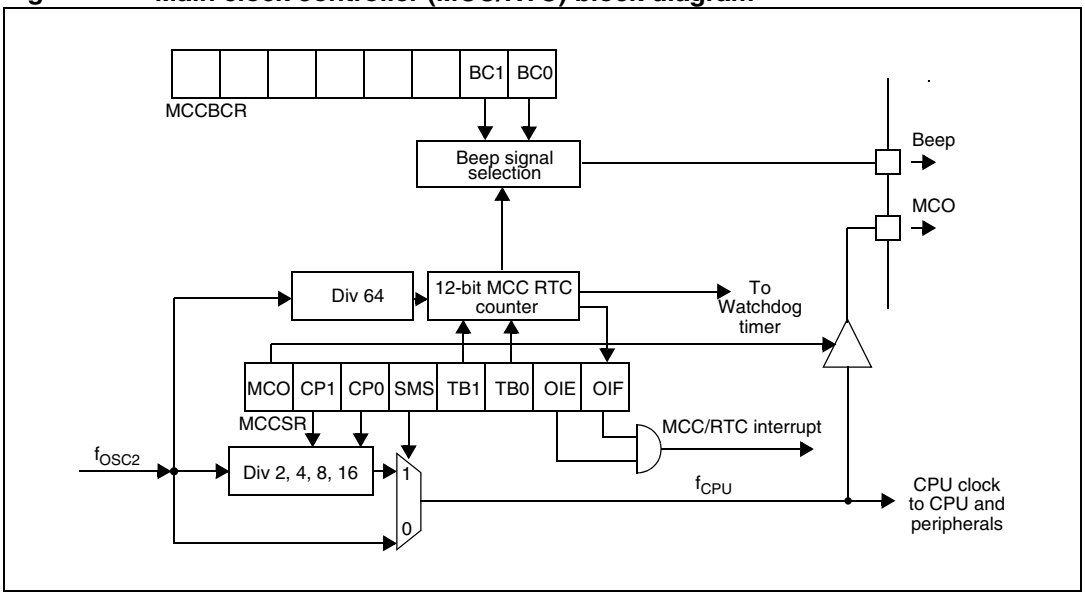
The counter of the real-time clock timer allows an interrupt to be generated based on an accurate real-time clock. Four different time bases depending directly on f_{OSC2} are available. The whole functionality is controlled by four bits of the MCCSR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters Active-halt mode when the HALT instruction is executed. See *Section 8.4: Active-halt and Halt modes on page 54* for more details.

10.2.4 Beeper

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

Figure 35. Main clock controller (MCC/RTC) block diagram



Downloaded from Arrow.com.

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/wakeup 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.*

Downloaded from Arrow.com.

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	4 ms	2 ms	0	0
32000	8 ms	4 ms	0	1
80000	20 ms	10 ms	1	0
200000	50 ms	25 ms	1	1

Downloaded from Arrow.com.

MCC beep control register (MCCBCR)

MCCBCR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
Reserved						BC[1:0]	
-						R/W	

Table 41. MCCBCR register description

Bit	Name	Function
7:2	-	Reserved, must be kept cleared
1:0	BC[1:0]	Beep Control These 2 bits select the PF1 pin beep capability (see *Table 42*). The beep output signal is available in Active-halt mode but has to be disabled to reduce the consumption.

Table 42. Beep frequency selection

BC1	BC0	Beep mode with f_{OSC2} = 8 MHz	
0	0	Off	
0	1	~2 kHz	Output Beep signal ~50% duty cycle
1	0	~1 kHz	
1	1	~500 Hz	

Table 43. Main clock controller register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
002Bh	SICSR Reset value	0	AVDIE 0	AVDF 0	LVDRF x	0	0	0	WDGRF x
002Ch	MCCSR Reset value	MCO 0	CP1 0	CP0 0	SMS 0	TB1 0	TB0 0	OIE 0	OIF 0
002Dh	MCCBCR Reset value	0	0	0	0	0	0	BC1 0	BC0 0

Downloaded from Arrow.com.

10.3 16-bit timer

10.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (input capture) or generation of up to two output waveforms (output compare and PWM).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

Some ST7 devices have two on-chip 16-bit timers. They are completely independent, and do not share any resources. They are synchronized after a MCU reset as long as the timer clock frequencies are not modified.

This description covers one or two 16-bit timers. In ST7 devices with two timers, register names are prefixed with TA (Timer A) or TB (Timer B).

10.3.2 Main features

- Programmable prescaler: f_{CPU} divided by 2, 4 or 8
- Overflow status flag and maskable interrupt
- External clock input (must be at least four times slower than the CPU clock speed) with the choice of active edge
- 1 or 2 output compare functions each with:
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- 1 or 2 input capture functions each with:
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- Reduced power mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)(c)

The timer block diagram is shown in *Figure 36*.

c. Some timer pins may not be available (not bonded) in some ST7 devices. Refer to *Section 2: Pin description*. When reading an input signal on a non-bonded pin, the value will always be '1'.

Downloaded from Arrow.com.

10.3.3 Functional description

Counter

The main block of the programmable timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high and low.

- Counter Register (CR)
  - Counter High Register (CHR) is the most significant byte (MSB)
  - Counter Low Register (CLR) is the least significant byte (LSB)
- Alternate Counter Register (ACR)
  - Alternate Counter High Register (ACHR) is the most significant byte (MSB)
  - Alternate Counter Low Register (ACLR) is the least significant byte (LSB)

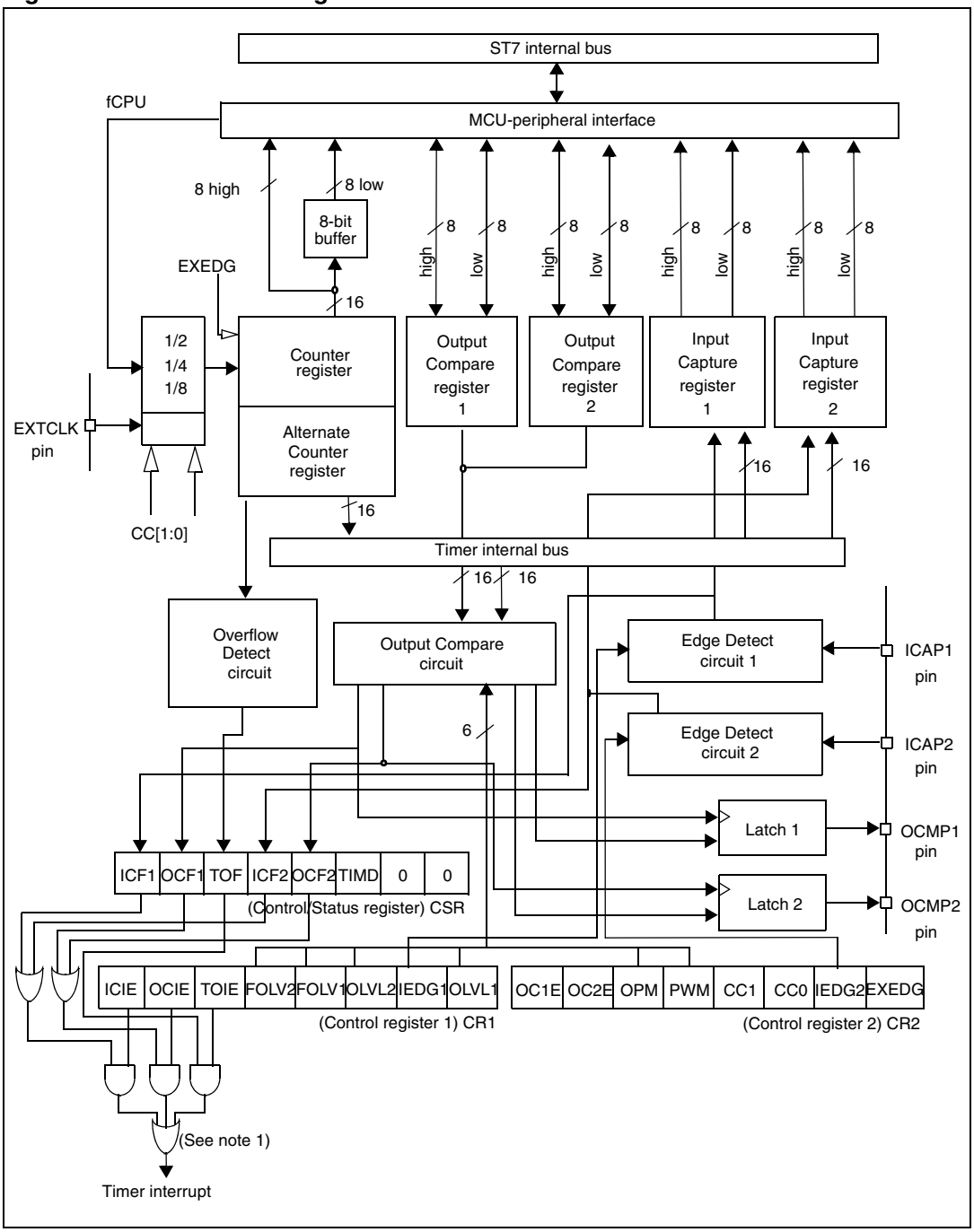
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (timer overflow flag), located in the Status register (SR) (see note at the end of paragraph entitled *16-bit read sequence)*.

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value. Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in one pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in *Table 50*. The value in the counter register repeats every 131072, 262144 or 524288 CPU clock cycles depending on the CC[1:0] bits. The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

Downloaded from Arrow.com.

Figure 36. Timer block diagram



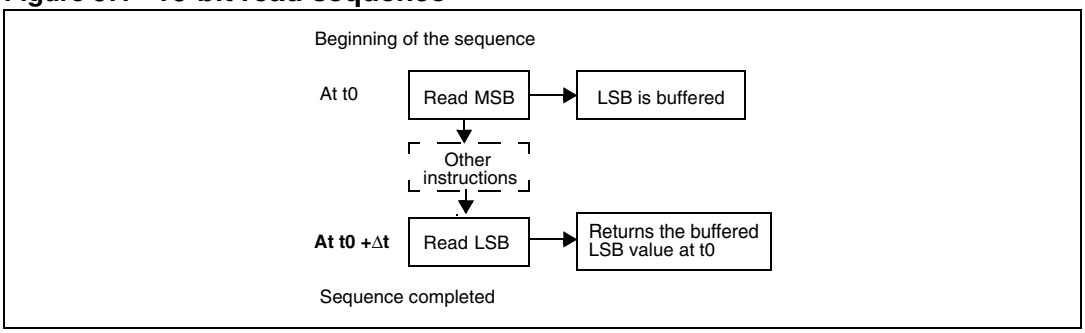
1. If IC, OC and TO interrupt requests have separate vectors then the last OR is not present (see *Table 25: Interrupt mapping on page 51*).

Downloaded from Arrow.com.

16-bit read sequence

The 16-bit read sequence (from either the Counter register or the Alternate Counter register) is illustrated in the following *Figure 37*.

Figure 37. 16-bit read sequence



The user must first read the MSB, afterwhich the LSB value is automatically buffered.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LSB of the count value at the time of the read.

Whatever the timer mode used (input capture, output compare, one pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h then:

- The TOF bit of the SR register is set.
- A timer interrupt is generated if:
  - TOIE bit of the CR1 register is set and
  - I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Note: The TOF bit is not cleared by access to the ACLR register. The advantage of accessing the ACLR register rather than the CLR register is that it allows simultaneous use of the overflow function and reading the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by Wait mode.

In Halt mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a reset).

Downloaded from Arrow.com.

External clock

The external clock (where available) is selected if CC0 = 1 and CC1 = 1 in the CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronized with the falling edge of the internal CPU clock.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

Figure 38. Counter timing diagram, internal clock divided by 2

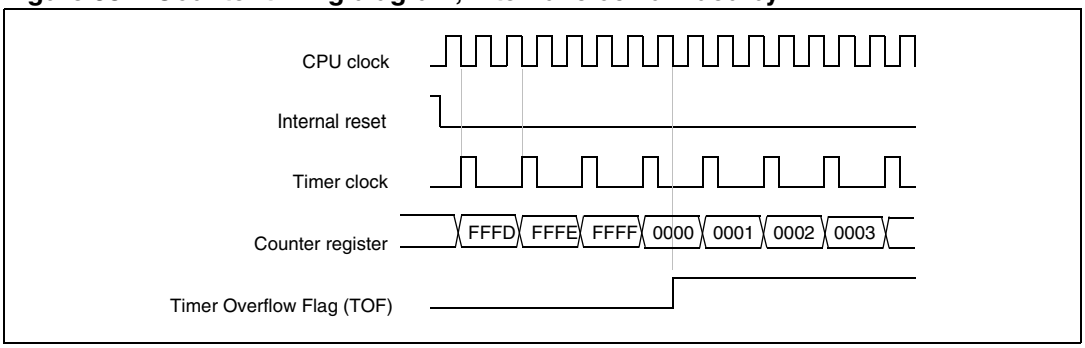


Figure 39. Counter timing diagram, internal clock divided by 4

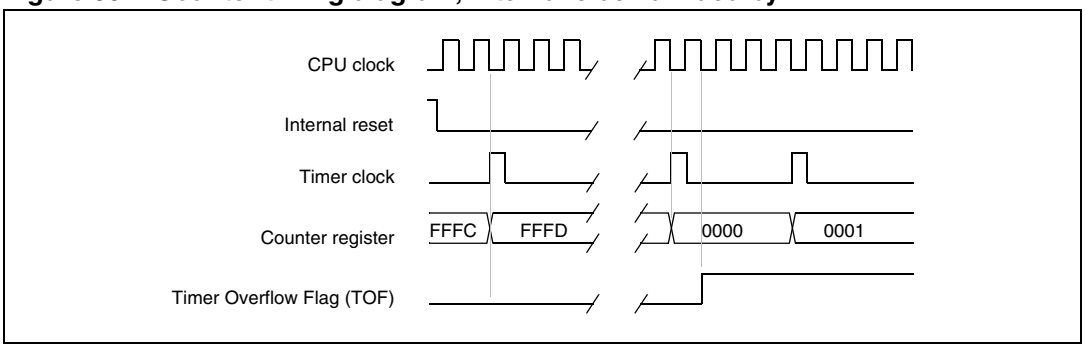
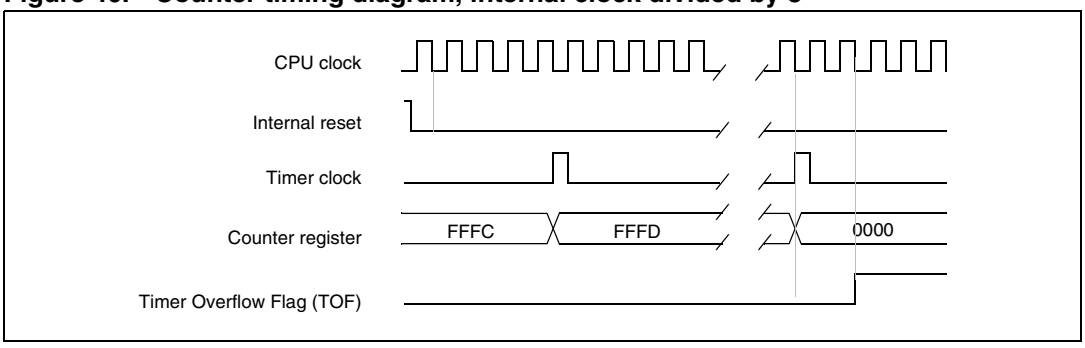


Figure 40. Counter timing diagram, internal clock divided by 8



Note: The MCU is in reset state when the internal reset signal is high, when it is low the MCU is running.

Downloaded from Arrow.com.

Input capture

In this section, the index, *i*, may be 1 or 2 because there are two input capture functions in the 16-bit timer.

The two 16-bit input capture registers (IC1R/IC2R) are used to latch the value of the free running counter after a transition is detected on the ICAP*i* pin (see *Figure 42*).

Table 44. Input capture byte distribution

Register	MS byte	LS byte
ICiR	IC*i*HR	IC*i*LR

The ICiR registers are read-only registers.

The active transition is software programmable through the IEDG*i* bit of Control Registers (CR*i*).

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure

To use the input capture function select the following in the CR2 register:

- Select the timer clock (CC[1:0]) (see *Table 50*).
- Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

Select the following in the CR1 register:

- Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin
- Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

When an input capture occurs:

- ICF*i* bit is set.
- The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see *Figure 42*).
- A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set
2. An access (read or write) to the IC*i*LR register

Downloaded from Arrow.com.

Note: 1 After reading the ICiHR register, transfer of input capture data is inhibited and ICFi will never be set until the ICiLR register is also read.

2 The ICiR register contains the free running counter value which corresponds to the most recent input capture.

3 The two input capture functions can be used together even if the timer also uses the two output compare functions.

4 In One pulse mode and PWM mode only Input Capture 2 can be used.

5 The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activates the input capture function.
Moreover if one of the ICAPi pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.
This can be avoided if the input capture function i is disabled by reading the ICiHR (see note 1).

6 The TOF bit can be used with interrupt generation in order to measure events that go beyond the timer range (FFFFh).

Figure 41. Input capture block diagram

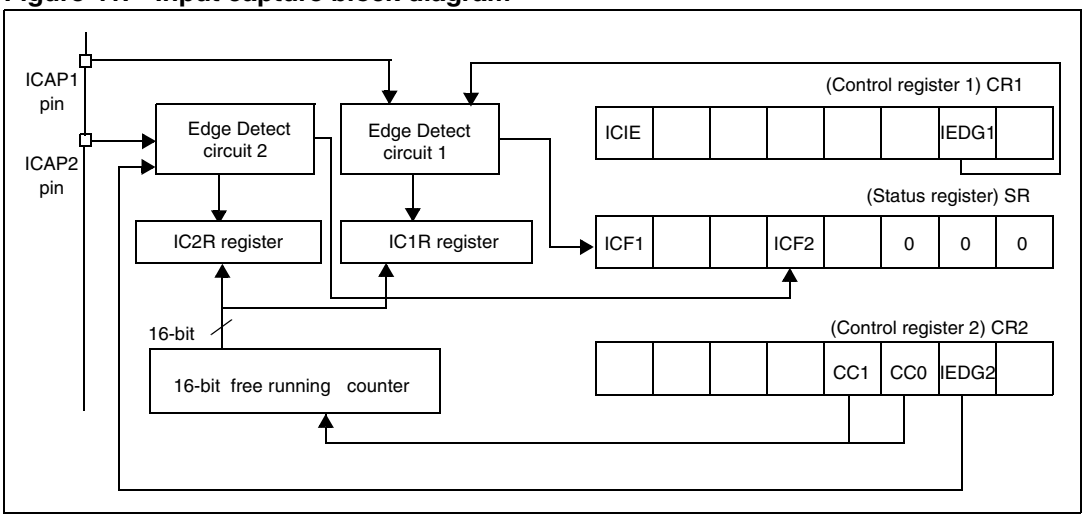
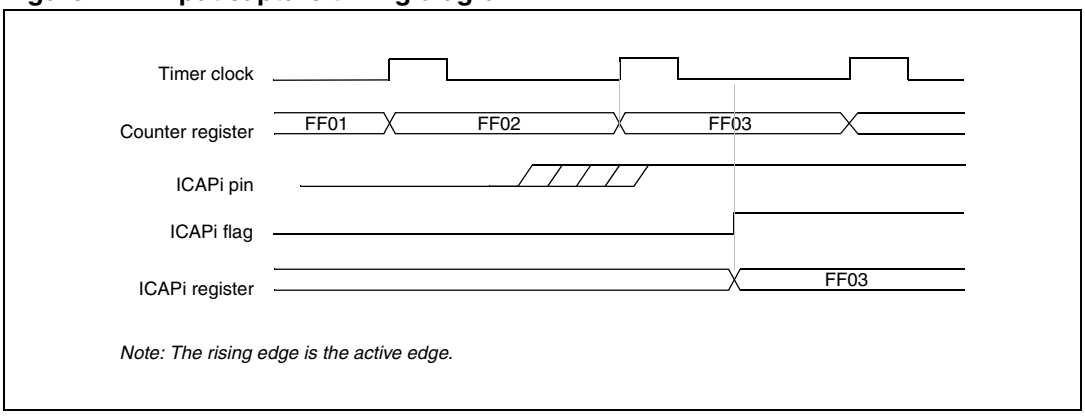


Figure 42. Input capture timing diagram



Note: The rising edge is the active edge.

Downloaded from Arrow.com.

Output compare

In this section, the index, *i*, may be 1 or 2 because there are two output compare functions in the 16-bit timer.

This function can be used to control an output waveform or indicate when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OC*i*E bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare register 1 (OC1R) and Output Compare register 2 (OC2R) contain the value to be compared to the counter register each timer clock cycle.

Table 45. Output compare byte distribution

Register	MS byte	LS byte
OCiR	OC*i*HR	OC*i*LR

These registers are readable and witable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure

To use the Output Compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* signal.
- Select the timer clock (CC[1:0]) (see *Table 50*).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When a match is found between OCRi register and CR register:

- OCF*i* bit is set
- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset)
- A timer interrupt is generated if the OCIE bit is set in the CR1 register and the I bit is cleared in the CC register (CC).

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{ OC}i\text{R} = \frac{\Delta t * f_{CPU}}{\text{PRESC}}$$

Where:

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

Downloaded from Arrow.com.

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

$$\Delta \text{ OC}i\text{R} = \Delta t * f_{EXT}$$

Where:

Δt = Output compare period (in seconds)
f_{EXT} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (that is, clearing the OCF*i* bit) is done by:

1. Reading the SR register while the OCF*i* bit is set.
2. An access (read or write) to the OC*i*LR register.

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

Note: 1 After a processor write cycle to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 If the OCiE bit is not set, the OCMPi pin is a general I/O port and the OLVLi bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.

3 In both internal and external clock modes, OCFi and OCMPi are set while the counter value equals the OCiR register value (see Figure 44 on page 83 for an example with $f_{CPU}/2$ and Figure 45 on page 83 for an example with $f_{CPU}/4$). This behavior is the same in OPM or PWM mode.

4 The output compare functions can be used both for generating external events on the OCMPi pins even if the input capture mode is also used.

5 The value in the 16-bit OCiR register and the OLVi bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

Forced output compare capability

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit = 1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

The FOLVL*i* bits have no effect in both one pulse mode and PWM mode.

Downloaded from Arrow.com.

Figure 43. Output compare block diagram

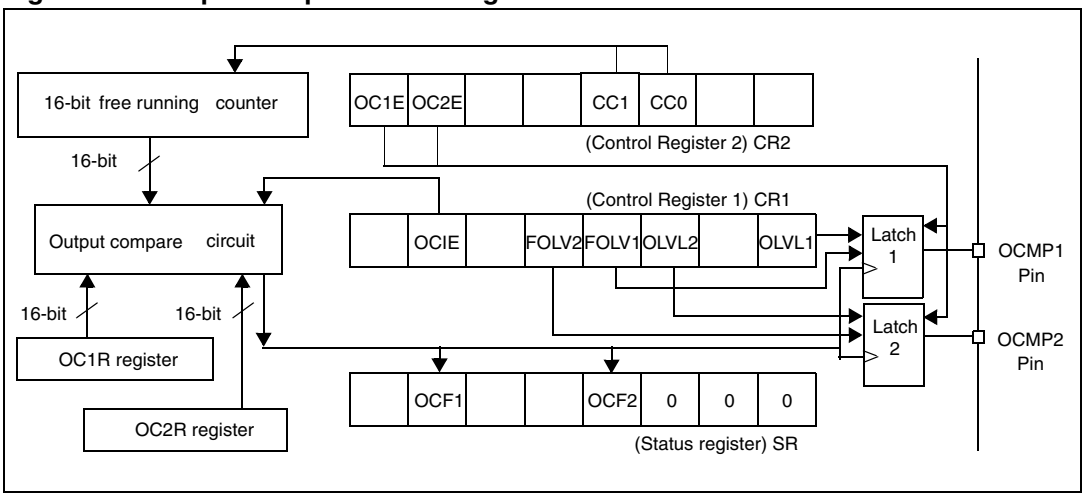


Figure 44. Output compare timing diagram, $f_{TIMER} = f_{CPU}/2$

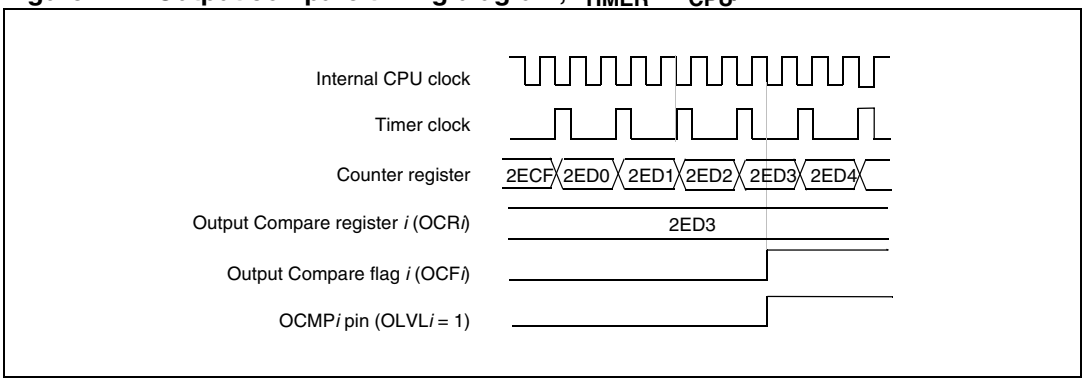
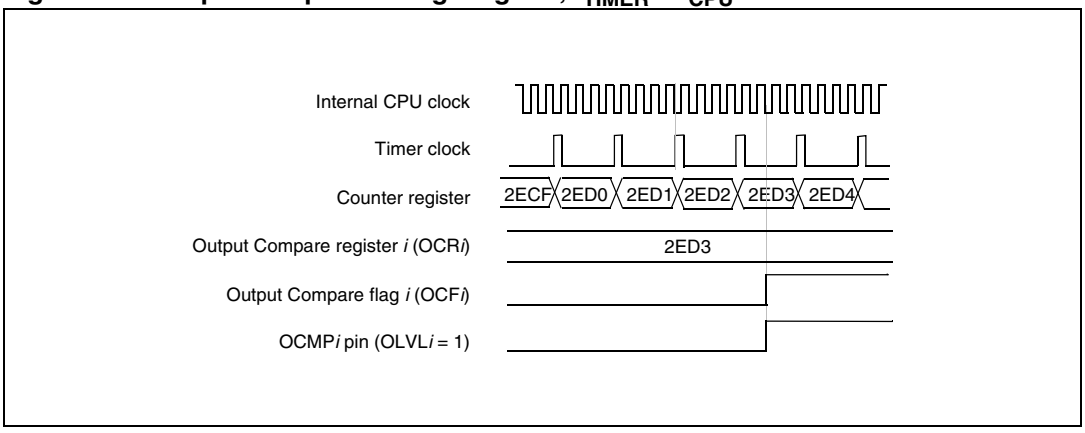


Figure 45. Output compare timing diagram, $f_{TIMER} = f_{CPU}/4$



Downloaded from Arrow.com.

One Pulse mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

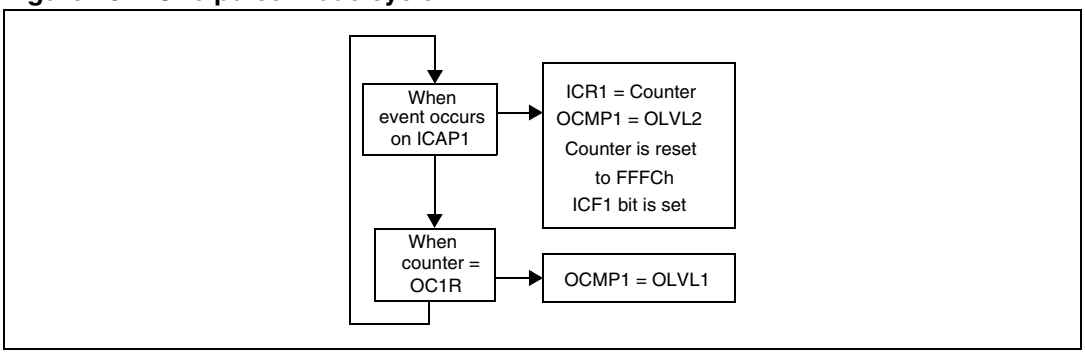
The one pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure

To use One Pulse mode:

1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula below).
2. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
   - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
   - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
   - Set the OPM bit.
   - Select the timer clock CC[1:0] (see *Table 50*).

Figure 46. One pulse mode cycle



Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

Downloaded from Arrow.com.

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 47*).

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

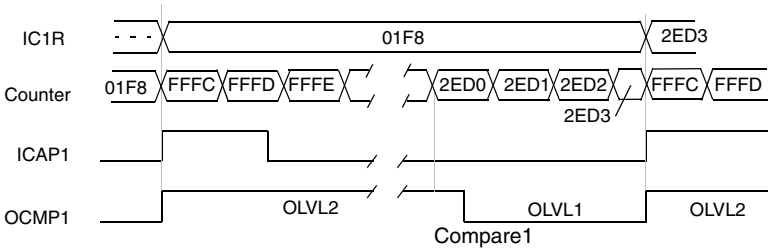
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

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.

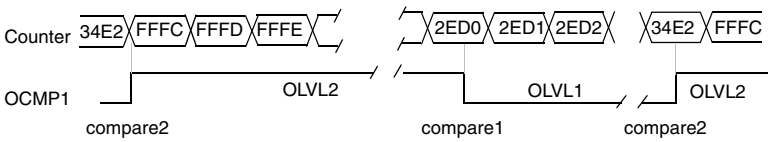
Figure 47. One Pulse mode timing example(1)



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

Downloaded from Arrow.com.

Figure 48. Pulse width modulation mode timing example with two output compare functions(1)(2)



1. OC1R = 2ED0h, OC2R = 34E2, OLVL1 = 0, OLVL2 = 1
2. On timers with only one Output Compare register, a fixed frequency PWM signal can be generated using the output compare and the counter overflow to define the pulse length.

Pulse Width Modulation mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

In PWM mode, double buffering is implemented on the output compare registers. Any new values written in the OC1R and OC2R registers are taken into account only at the end of the PWM period (OC2) to avoid spikes on the PWM output pin (OCMP1).

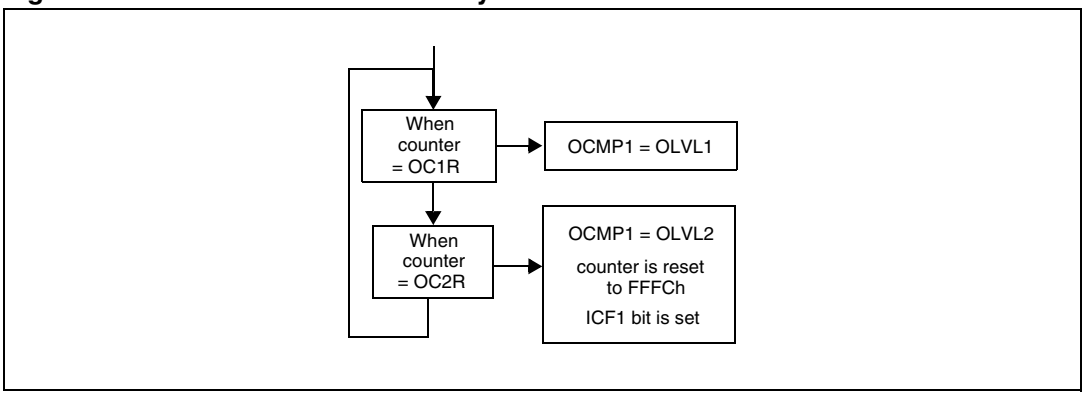
Procedure

To use Pulse Width Modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the formula below.
2. Load the OC1R register with the value corresponding to the period of the pulse if (OLVL1 = 0 and OLVL2 = 1) using the formula in the opposite column.
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see *Table 50*).

Downloaded from Arrow.com.

Figure 49. Pulse width modulation cycle



If OLVL1 = 1 and OLVL2 = 0, the length of the positive pulse is the difference between the OC2R and OC1R registers.

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

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 = Signal or 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 = Signal or pulse period (in seconds)
f_{EXT} = External timer clock frequency (in Hertz)

The Output Compare 2 event causes the counter to be initialized to FFFCh (see *Figure 48*).

Note: 1 After a write instruction to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.

3 The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.

4 In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected to the timer. 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 period and ICF1 can also generates interrupt if ICIE is set.

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

Downloaded from Arrow.com.

10.3.4 Low power modes

Table 46. Effect of low power modes on 16-bit timer

Mode	Description
Wait	No effect on 16-bit timer. Timer interrupts cause the device to exit from Wait mode.
Halt	16-bit timer registers are frozen. In Halt mode, the counter stops counting until Halt mode is exited. Counting resumes from the previous count when the MCU is woken up by an interrupt with Exit from Halt mode capability or from the counter reset value when the MCU is woken up by a reset. If an input capture event occurs on the ICAP*i* pin, the input capture detection circuitry is armed. Consequently, when the MCU is woken up by an interrupt with Exit from Halt mode capability, the ICF*i* bit is set, and the counter value present when exiting from Halt mode is captured into the IC*i*R register.

10.3.5 Interrupts

Table 47. 16-bit timer interrupt control/wakeup capability(1)

Interrupt event	Event flag	Enable Control bit	Exit from Wait	Exit from Halt
Input Capture 1 event/counter reset in PWM mode	ICF1	ICIE	Yes	No
Input Capture 2 event	ICF2			
Output Compare 1 event (not available in PWM mode)	OCF1	OCIE		
Output Compare 2 event (not available in PWM mode)	OCF2			
Timer Overflow event	TOF	TOIE		

1. The 16-bit timer interrupt events are connected to the same interrupt vector (see *Section 7: Interrupts*). These events generate an interrupt if the corresponding Enable Control bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Downloaded from Arrow.com.

10.3.6 Summary of timer modes

Table 48. Summary of timer modes

Mode	Timer resources			
	Input Capture 1	Input Capture 2	Output Compare 1	Output Compare 2
Input Capture (1 and/or 2)	Yes	Yes	Yes	Yes
Output Compare (1 and/or 2)				
One Pulse mode	No	Not recommended(1)	No	Partially(2)
PWM mode		Not recommended(3)		No

1. See note 4 in *One Pulse mode on page 84*.
2. See note 5 in *One Pulse mode on page 84*.
3. See note 4 in *Pulse Width Modulation mode on page 86*.

10.3.7 16-bit timer registers

Each timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

Control Register 1 (CR1)

CR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 49. CR1 register description

Bit	Name	Function
7	ICIE	Input Capture Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.
6	OCIE	Output Compare Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.
5	TOIE	Timer Overflow Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Downloaded from Arrow.com.

Table 49. CR1 register description (continued)

Bit	Name	Function
4	FOLV2	Forced Output compare 2 This bit is set and cleared by software. 0: No effect on the OCMP2 pin. 1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison.
3	FOLV1	Forced Output compare 1 This bit is set and cleared by software. 0: No effect on the OCMP1 pin. 1: Forces OLVL1 to be copied to the OCMP1 pin, if the OC1E bit is set and even if there is no successful comparison.
2	OLVL2	Output Level 2 This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse mode and Pulse Width modulation mode.
1	IEDG1	Input Edge 1 This bit determines which type of level transition on the ICAP1 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	OLVL1	Output Level 1 The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

Control Register 2 (CR2)

CR2 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
OC1E	OC2E	OPM	PWM	CC[1:0]		IEDG2	EXEDG
R/W	R/W	R/W	R/W	R/W		R/W	R/W

Table 50. CR2 register description

Bit	Name	Function
7	OCIE	Output compare 1 pin enable This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and One-Pulse mode). Whatever the value of the OC1E bit, the Output Compare 1 function of the timer remains active. 0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O). 1: OCMP1 pin alternate function enabled.
6	OC2E	Output compare 2 pin enable This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the Output Compare 2 function of the timer remains active. 0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O). 1: OCMP2 pin alternate function enabled.

Downloaded from Arrow.com.

Table 50. CR2 register description (continued)

Bit	Name	Function
5	OPM	One Pulse mode 0: One Pulse mode is not active. 1: One Pulse mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.
4	PWM	Pulse width modulation 0: PWM mode is not active. 1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.
3:2	CC[1:0]	Clock control The timer clock mode depends on these bits. 00: Timer clock = $f_{CPU}/4$ 01: Timer clock = $f_{CPU}/2$ 10: Timer clock = $f_{CPU}/8$ 11: Timer clock = external clock (where available) *Note: If the external clock pin is not available, programming the external clock configuration stops the counter.*
1	IEDG2	Input edge 2 This bit determines which type of level transition on the ICAP2 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	EXEDG	External clock edge This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register. 0: A falling edge triggers the counter register. 1: A rising edge triggers the counter register.

Control/Status Register (CSR)

CSR Reset value: xxxx x0xx (xxh)

7	6	5	4	3	2	1	0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	Reserved	
RO	RO	RO	RO	RO	R/W	-	

Table 51. CSR register description

Bit	Name	Function
7	ICF1	Input capture flag 1 0: No input capture (reset value). 1: An input capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Downloaded from Arrow.com.

Table 51. CSR register description (continued)

Bit	Name	Function
6	OCF1	Output compare flag 1 0: No match (reset value). 1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.
5	TOF	Timer overflow flag 0: No timer overflow (reset value). 1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register. *Note: Reading or writing the ACLR register does not clear TOF.*
4	ICF2	Input capture flag 2 0: No input capture (reset value). 1: An Input Capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.
3	OCF2	Output compare flag 2 0: No match (reset value). 1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.
2	TIMD	Timer disable This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed, or the counter reset, while it is disabled. 0: Timer enabled. 1: Timer prescaler, counter and outputs disabled.
1:0	-	Reserved, must be kept cleared.

Input capture 1 high register (IC1HR)

This is an 8-bit register that contains the high part of the counter value (transferred by the input capture 1 event).

IC1HR — Reset value: undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Downloaded from Arrow.com.

Input capture 1 low register (IC1LR)

This is an 8-bit register that contains the low part of the counter value (transferred by the input capture 1 event).

IC1LR — Reset value: undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Output compare 1 high register (OC1HR)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

OC1HR — Reset value: 1000 0000 (80h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output compare 1 low register (OC1LR)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

OC1LR — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output compare 2 high register (OC2HR)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

OC2HR — Reset value: 1000 0000 (80h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Downloaded from Arrow.com.

Output compare 2 low register (OC2LR)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

OC2LR — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
MSB							LSB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Counter high register (CHR)

This is an 8-bit register that contains the high part of the counter value.

CHR — Reset value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Counter low register (CLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.

CLR — Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Alternate counter high register (ACHR)

This is an 8-bit register that contains the high part of the counter value.

ACHR — Reset value: 1111 1111 (FFh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Downloaded from Arrow.com.

Alternate counter low register (ACLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

ACLR — Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Input capture 2 high register (IC2HR)

This is an 8-bit register that contains the high part of the counter value (transferred by the Input Capture 2 event).

1C2HR — Reset value: undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Input capture 2 low register (IC2LR)

This is an 8-bit register that contains the low part of the counter value (transferred by the Input Capture 2 event).

1C2LR — Reset value: undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Table 52. 16-bit timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Timer A: 32 Timer B: 42	CR1 Reset value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
Timer A: 31 Timer B: 41	CR2 Reset value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
Timer A: 33 Timer B: 43	CSR Reset value	ICF1 x	OCF1 x	TOF x	ICF2 x	OCF2 x	TIMD 0	- x	- x
Timer A: 34 Timer B: 44	IC1HR Reset value	MSB x	x	x	x	x	x	x	LSB x
Timer A: 35 Timer B: 45	IC1LR Reset value	MSB x	x	x	x	x	x	x	LSB x

Downloaded from Arrow.com.

Table 52. 16-bit timer register map and reset values (continued)

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Timer A: 36 Timer B: 46	OC1HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
Timer A: 37 Timer B: 47	OC1LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
Timer A: 3E Timer B: 4E	OC2HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
Timer A: 3F Timer B: 4F	OC2LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
Timer A: 38 Timer B: 48	CHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 39 Timer B: 49	CLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3A Timer B: 4A	ACHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 3B Timer B: 4B	ACLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3C Timer B: 4C	IC2HR Reset value	MSB x	x	x	x	x	x	x	LSB x
Timer A: 3D Timer B: 4D	IC2LR Reset value	MSB x	x	x	x	x	x	x	LSB x

10.4 Serial peripheral interface (SPI)

10.4.1 Introduction

The serial peripheral interface (SPI) allows full-duplex, synchronous, serial communication with external devices. An SPI system may consist of a master and one or more slaves. However, the SPI interface can not be a master in a multi-master system.

10.4.2 Main features

- Full duplex synchronous transfers (on 3 lines)
- Simplex synchronous transfers (on 2 lines)
- Master or slave operation
- 6 master mode frequencies (f_{CPU}/4 max.)
- f_{CPU}/2 max. slave mode frequency (see note)
- $\overline{SS}$ Management by software or hardware
- Programmable clock polarity and phase
- End of transfer interrupt flag
- Write collision, Master mode fault and Overrun flags

Note: In slave mode, continuous transmission is not possible at maximum frequency due to the software overhead for clearing status flags and to initiate the next transmission sequence.

Downloaded from Arrow.com.

10.4.3 General description

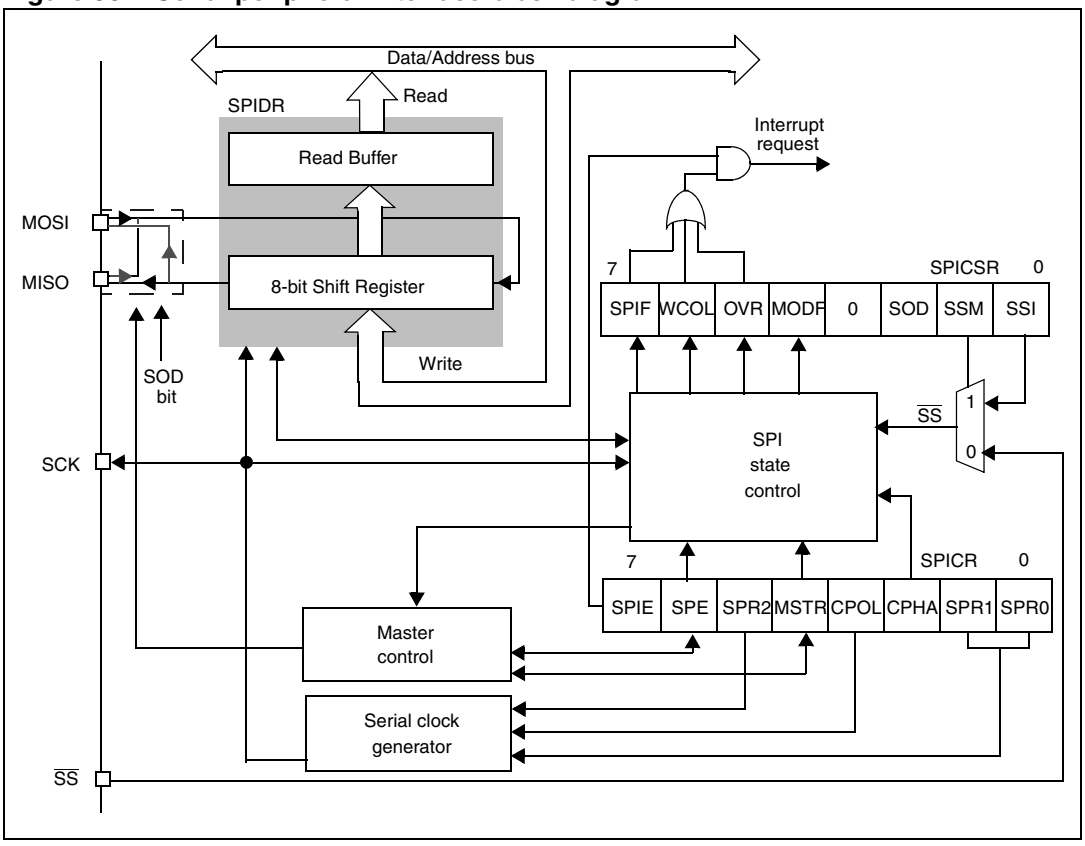
Figure 50 shows the serial peripheral interface (SPI) block diagram. The SPI has three registers:

- SPI Control Register (SPICR)
- SPI Control/Status Register (SPICSR)
- SPI Data Register (SPIDR)

The SPI is connected to external devices through four pins:

- MISO: Master In / Slave Out data
- MOSI: Master Out / Slave In data
- SCK: Serial Clock out by SPI masters and input by SPI slaves
- $\overline{SS}$: Slave select: This input signal acts as a 'chip select' to let the SPI master communicate with slaves individually and to avoid contention on the data lines. Slave $\overline{SS}$ inputs can be driven by standard I/O ports on the master MCU.

Figure 50. Serial peripheral interface block diagram



Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 51*.

The MOSI pins are connected together and the MISO pins are connected together. In this way data is transferred serially between master and slave (most significant bit first).

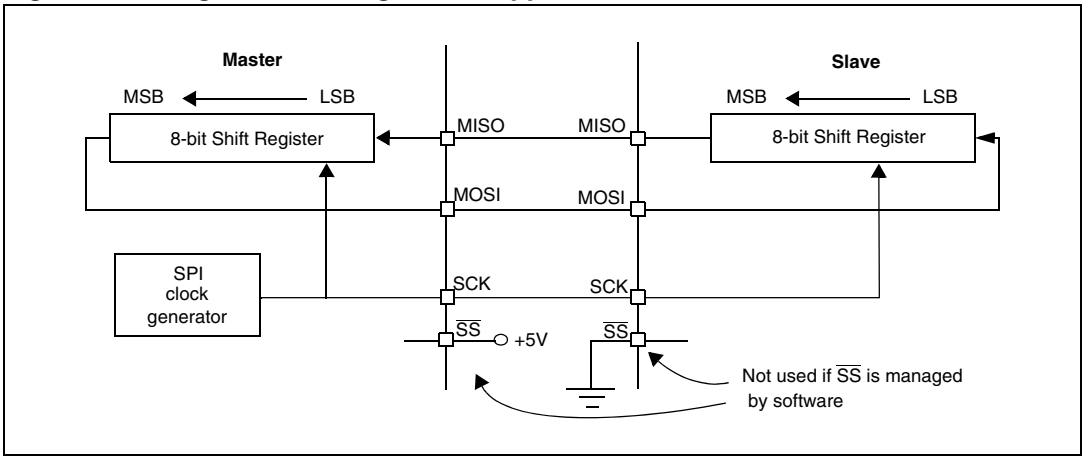
Downloaded from Arrow.com.

The communication is always initiated by the master. When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

To use a single data line, the MISO and MOSI pins must be connected at each node (in this case only simplex communication is possible).

Four possible data/clock timing relationships may be chosen (see *Figure 54*) but master and slave must be programmed with the same timing mode.

Figure 51. Single master/single slave application



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 53*).

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

Depending on the data/clock timing relationship, there are two cases in Slave mode (see *Figure 52*):

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 *Write collision error (WCOL) on page 102*).

Downloaded from Arrow.com.

Figure 52. Generic $\overline{SS}$ timing diagram

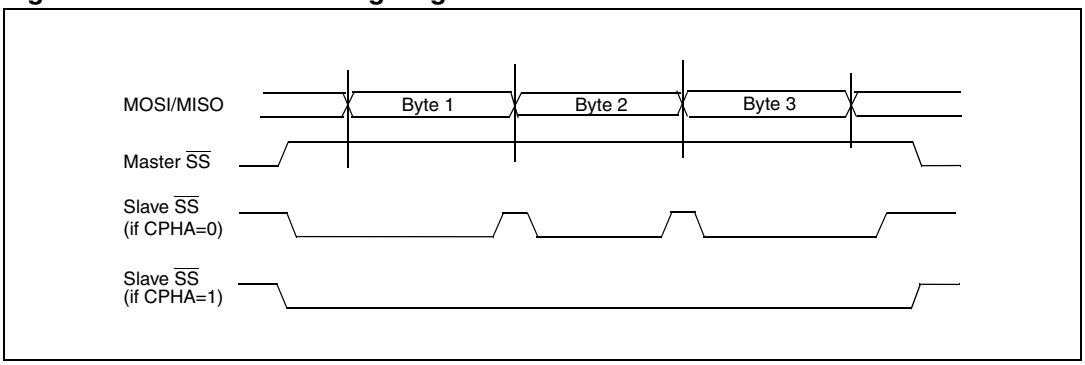
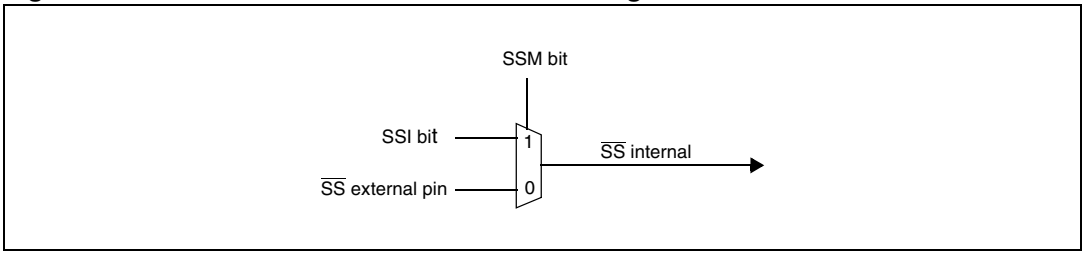


Figure 53. Hardware/software slave select management



Master mode operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

1. Write to the SPICR register:
   - Select the clock frequency by configuring the SPR[2:0] bits.
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. *Figure 54* shows the four possible configurations.
     Note: The slave must have the same CPOL and CPHA settings as the master.
2. Write to the SPICSR register:
   - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the $\overline{SS}$ pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
   - Set the MSTR and SPE bits.
     Note: MSTR and SPE bits remain set only if $\overline{SS}$ is high.

Caution: If the SPICSR register is not written first, the SPICR register setting (MSTR bit) might not be taken into account.

The transmit sequence begins when software writes a byte in the SPIDR register.

Downloaded from Arrow.com.

Master mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set.
2. A read to the SPIDR register.

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

Slave mode operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see *Figure 54*). The slave must have the same CPOL and CPHA settings as the master.
   - Manage the $\overline{SS}$ pin as described in *Slave select management on page 98* and *Figure 52*. If CPHA = 1, $\overline{SS}$ must be held low continuously. If CPHA = 0, $\overline{SS}$ must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

Slave mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set.
2. A write or a read to the SPIDR register.

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

Downloaded from Arrow.com.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see *Overrun condition (OVR) on page 102*).

10.4.4 Clock phase and clock polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (see *Figure 54*).

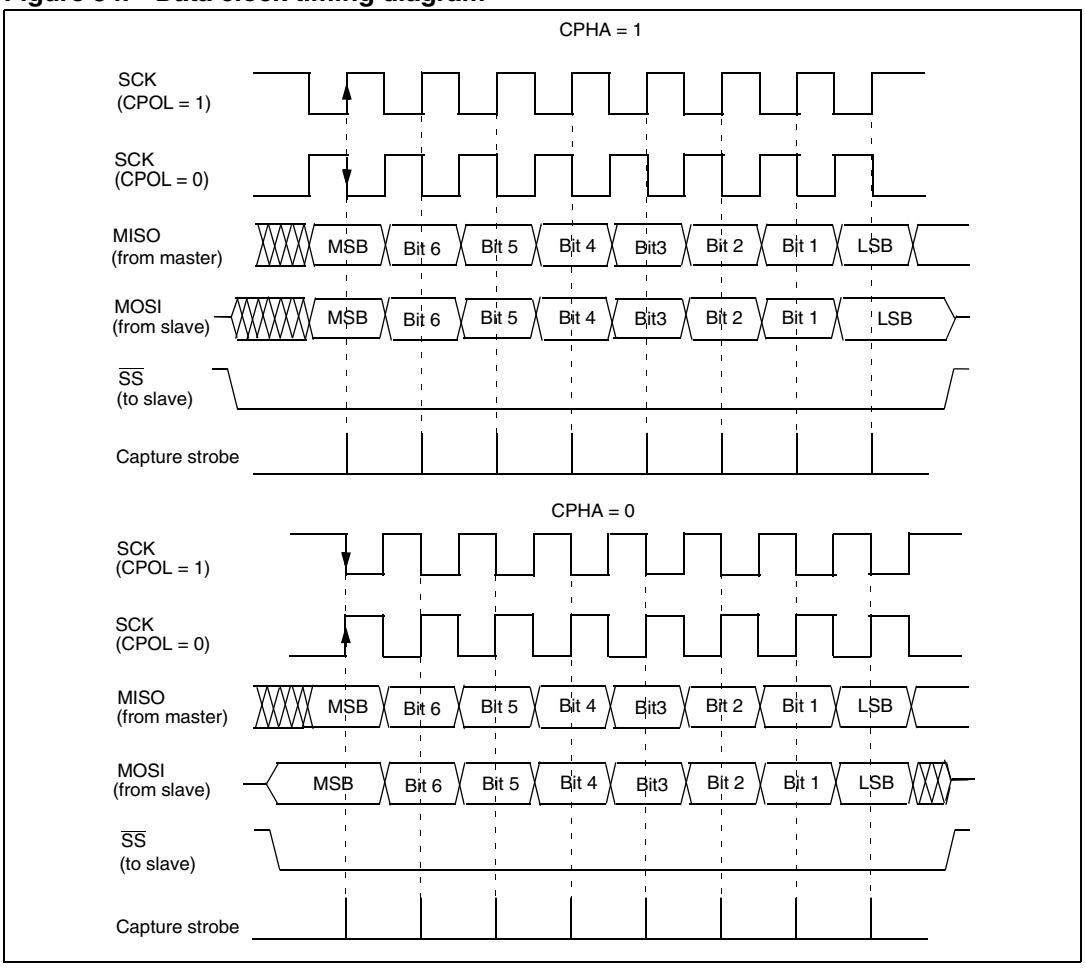
Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

The combination of the CPOL clock polarity and CPHA (clock phase) bits selects the data capture clock edge

Figure 54 shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK, MISO and MOSI pins are directly connected between the master and the slave device.

Note: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE bit.

Figure 54. Data clock timing diagram(1)



1. This figure should not be used as a replacement for parametric information. Refer to the Electrical Characteristics chapter.

Downloaded from Arrow.com.

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 98*.

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.

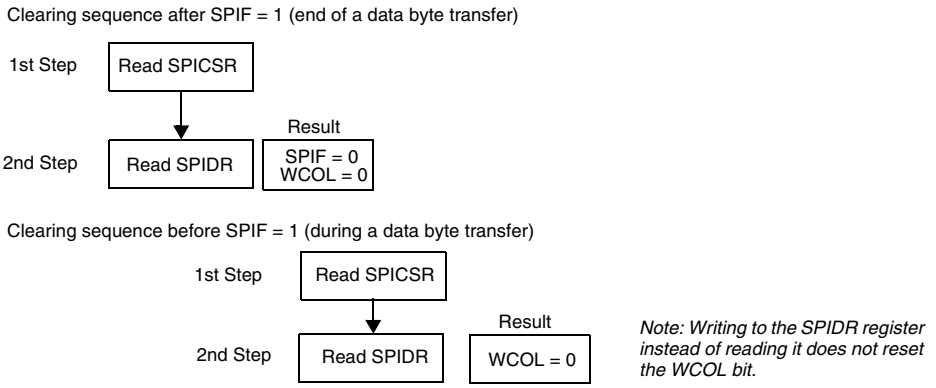
The WCOL bit in the SPICSR register is set if a write collision occurs.

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 55*).

Downloaded from Arrow.com.

Figure 55. Clearing the WCOL bit (Write collision flag) software sequence



Single master systems

A typical single master system may be configured, using an MCU as the master and four MCUs as slaves (see *Figure 56*).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

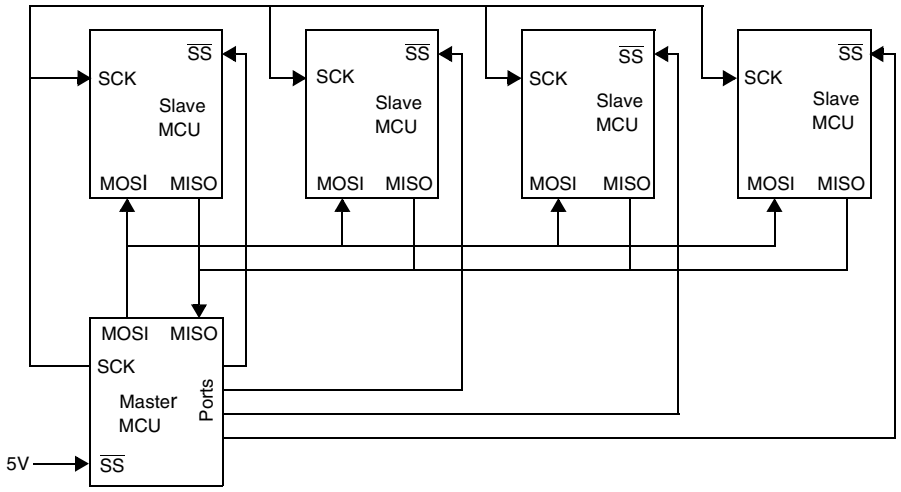
The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line the master allows only one active slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written to its SPIDR register.

Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Figure 56. Single master/multiple slave configuration



Downloaded from Arrow.com.

10.4.6 Low power modes

Table 53. Effect of low power modes on SPI

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 MCU 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 wakeup event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the device.

Using the SPI to wake up the MCU from Halt mode

In slave configuration, the SPI is able to wake up the ST7 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 ST7 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 ST7 enters Halt mode. Therefore, if Slave selection is configured as external (see *Slave select management on page 98*), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters Halt mode.

10.4.7 Interrupts

Table 54. SPI interrupt control/wakeup capability(1)

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			

1. The SPI interrupt events are connected to the same interrupt vector (see *Section 7: Interrupts*). They generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

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	

Downloaded from Arrow.com.

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 102*). 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 102*). 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.*
2	CPHA	Clock Phase This bit is set and cleared by software. 0: The first clock transition is the first data capture edge. 1: The second clock transition is the first capture edge. *Note: The slave must have the same CPOL and CPHA settings as the master.*
1:0	SPR[1:0]	Serial clock frequency These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode (see *Table 56*). *Note: These 2 bits have no effect in slave mode.*

Table 56. SPI master mode SCK frequency

Serial clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0

Downloaded from Arrow.com.

Table 56. SPI master mode SCK frequency (continued)

Serial clock	SPR2	SPR1	SPR0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

SPI control/status register (SPICSR)

SPICSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SPIF	WCOL	OVR	MODF	Reserved	SOD	SSM	SSI
RO	RO	RO	RO	-	R/W	R/W	R/W

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 55*). 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 102*). 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 102*). 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.

Downloaded from Arrow.com.

Table 57. SPICSR register description (continued)

Bit	Name	Function
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 98*. 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.

SPI data I/O register (SPIDR)

SPIDR — Reset value: undefined

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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.

Note: 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 50*).

Downloaded from Arrow.com.

Table 58. SPI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0021h	SPIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0022h	SPICR Reset value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0023h	SPICSR Reset value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

10.5 Serial communications interface (SCI)

10.5.1 Introduction

The serial communications interface (SCI) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The SCI offers a very wide range of baud rates using two baud rate generator systems.

10.5.2 Main features

- Full duplex, asynchronous communications
- NRZ standard format (mark/space)
- Dual baud rate generator systems
- Independently programmable transmit and receive baud rates up to 500K baud.
- Programmable data word length (8 or 9 bits)
- Receive buffer full, Transmit buffer empty and End of Transmission flags
- 2 receiver wakeup modes
  - Address bit (MSB)
  - Idle line
- Muting function for multiprocessor configurations
- Separate enable bits for Transmitter and Receiver
- 4 error detection flags
  - Overrun error
  - Noise error
  - Frame error
  - Parity error
- 5 interrupt sources with flags
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Idle line received
  - Overrun error detected

Downloaded from Arrow.com.

- Parity control
  - Transmits parity bit
  - Checks parity of received data byte
- Reduced power consumption mode

10.5.3 General description

The interface is externally connected to another device by two pins (see *Figure 58*):

- TDO: Transmit Data Output. When the transmitter and the receiver are disabled, the output pin returns to its I/O port configuration. When the transmitter and/or the receiver are enabled and nothing is to be transmitted, the TDO pin is at high level.
- RDI: Receive Data Input is the serial data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

Through these pins, serial data is transmitted and received as frames comprising:

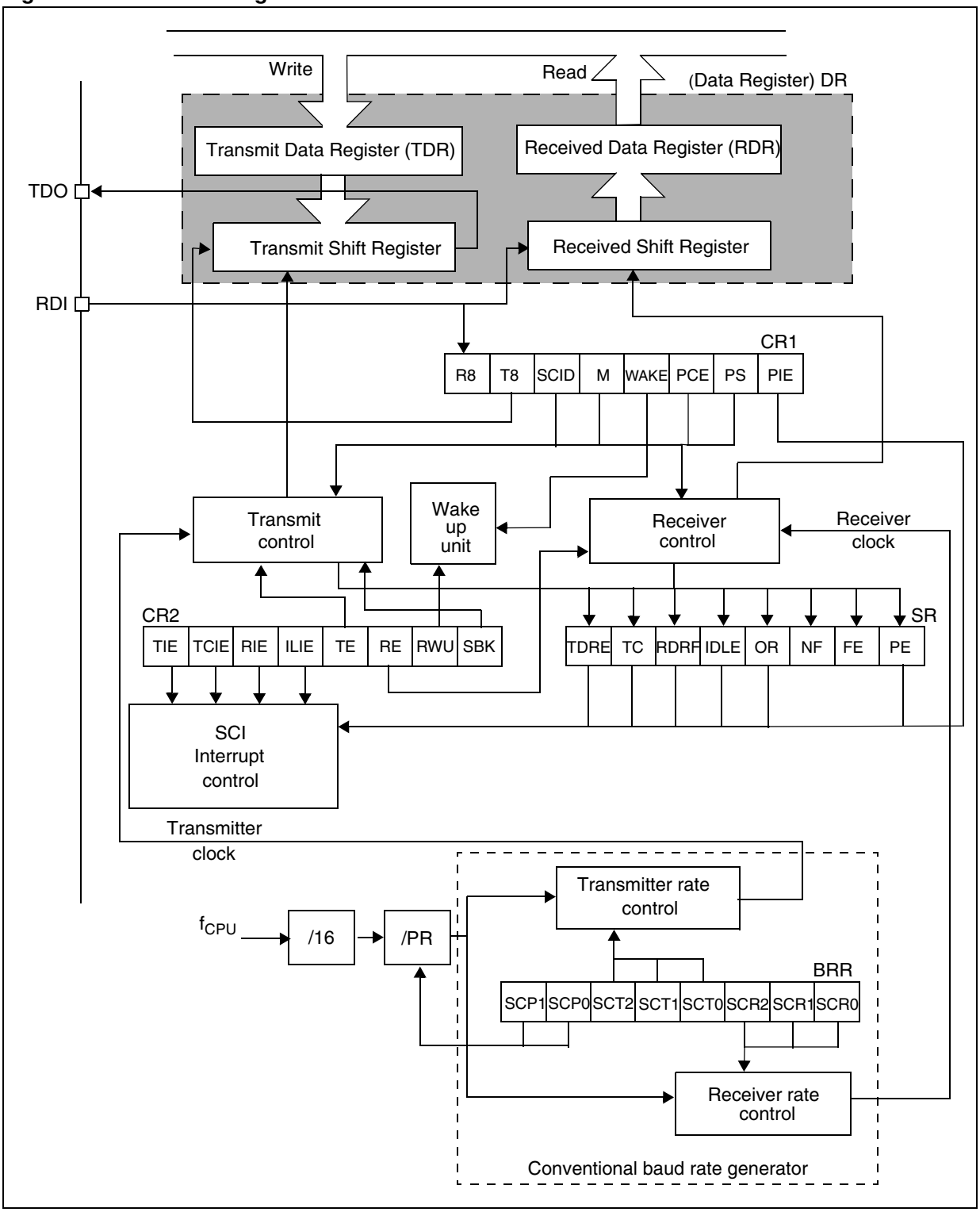
- an Idle Line prior to transmission or reception
- a start bit
- a data word (8 or 9 bits) least significant bit first
- a Stop bit indicating that the frame is complete

This interface uses two types of baud rate generator:

- a conventional type for commonly-used baud rates
- an extended type with a prescaler offering a very wide range of baud rates even with non-standard oscillator frequencies

Downloaded from Arrow.com.

Figure 57. SCI block diagram



Downloaded from Arrow.com.

10.5.4 Functional description

The block diagram of the serial control interface is shown in *Figure 57*. It contains six dedicated registers:

- 2 control registers (SCICR1 and SCICR2)
- a status register (SCISR)
- a baud rate register (SCIBRR)
- an extended prescaler receiver register (SCIERPR)
- an extended prescaler transmitter register (SCIETPR)

Refer to the register descriptions in *Section 10.5.7* for the definitions of each bit.

Serial data format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see *Figure 57*).

The TDO pin is in low state during the start bit.

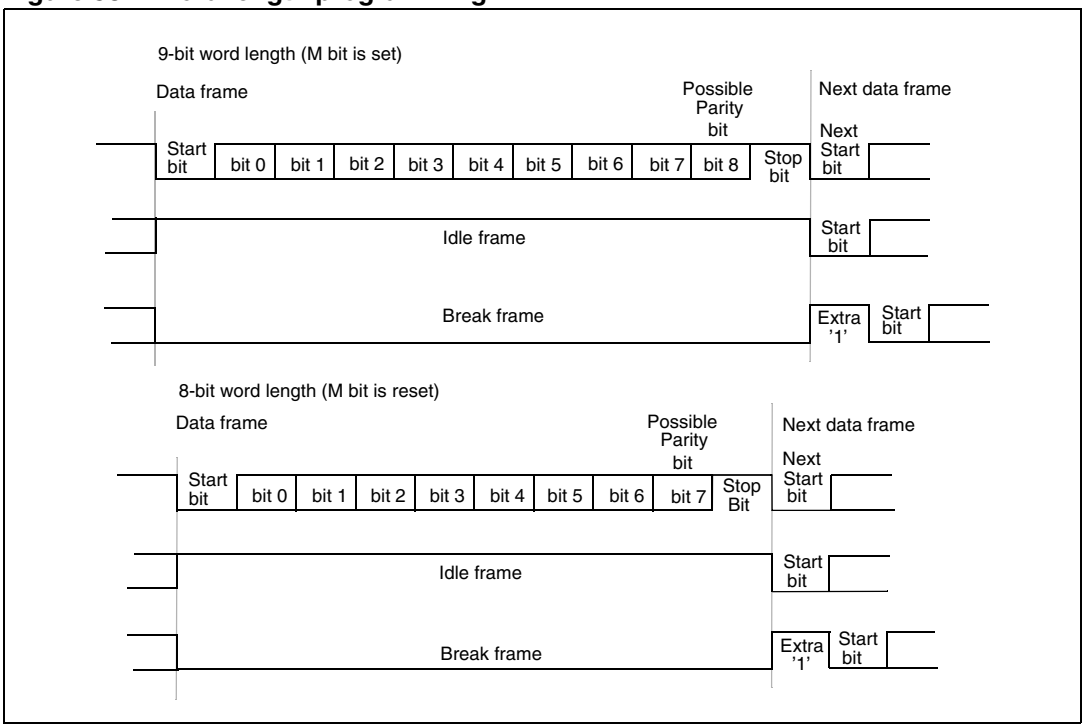
The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of ‘1’s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving ‘0’s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra ‘1’ bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.

Figure 58. Word length programming



Downloaded from Arrow.com.

Transmitter

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

Character transmission

During an SCI transmission, data shifts out LSB first on the TDO pin. In this mode, the SCIDR register consists of a buffer (TDR) between the internal bus and the transmit shift register (see *Figure 57*).

Procedure

1. Select the M bit to define the word length.
2. Select the desired baud rate using the SCIBRR and the SCIETPR registers.
3. Set the TE bit to assign the TDO pin to the alternate function and to send a idle frame as first transmission.
4. Access the SCISR register and write the data to send in the SCIDR register (this sequence clears the TDRE bit). Repeat this sequence for each data to be transmitted.

Clearing the TDRE bit is always performed by the following software sequence:

1. An access to the SCISR register
2. A write to the SCIDR register

The TDRE bit is set by hardware and it indicates:

- The TDR register is empty.
- The data transfer is beginning.
- The next data can be written in the SCIDR register without overwriting the previous data.

This flag generates an interrupt if the TIE bit is set and the I bit is cleared in the CCR register.

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

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

When a frame transmission is complete (after the stop bit) the TC bit is set and an interrupt is generated if the TCIE is set and the I bit is cleared in the CCR register.

Clearing the TC bit is performed by the following software sequence:

1. An access to the SCISR register
2. A write to the SCIDR register

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

Downloaded from Arrow.com.

Break characters

Setting the SBK bit loads the shift register with a break character. The break frame length depends on the M bit (see *Figure 58*).

As long as the SBK bit is set, the SCI send break frames to the TDO pin. After clearing this bit by software the SCI insert a logic 1 bit at the end of the last break frame to guarantee the recognition of the start bit of the next frame.

Idle characters

Setting the TE bit drives the SCI to send an idle frame before the first data frame.

Clearing and then setting the TE bit during a transmission sends an idle frame after the current word.

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

Receiver

The SCI can receive data words of either 8 or 9 bits. When the M bit is set, word length is 9 bits and the MSB is stored in the R8 bit in the SCICR1 register.

Character reception

During a SCI reception, data shifts in least significant bit first through the RDI pin. In this mode, the SCIDR register consists or a buffer (RDR) between the internal bus and the received shift register (see *Figure 57*).

Procedure

1. Select the M bit to define the word length.
2. Select the desired baud rate using the SCIBRR and the SCIERPR registers.
3. Set the RE bit, this enables the receiver which begins searching for a start bit.

When a character is received:

- The RDRF bit is set. It indicates that the content of the shift register is transferred to the RDR.
- An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.
- The error flags can be set if a frame error, noise or an overrun error has been detected during reception.

Clearing the RDRF bit is performed by the following software sequence done by:

1. An access to the SCISR register

2. A read to the SCIDR register.

The RDRF bit must be cleared before the end of the reception of the next character to avoid an overrun error.

Break character

When a break character is received, the SCI handles it as a framing error.

Idle character

When a idle frame is detected, there is the same procedure as a data received character plus an interrupt if the ILIE bit is set and the I bit is cleared in the CCR register.

Downloaded from Arrow.com.

Overrun error

An overrun error occurs when a character is received when RDRF has not been reset. Data can not be transferred from the shift register to the RDR register as long as the RDRF bit is not cleared.

When a overrun error occurs:

- The OR bit is set.
- The RDR content will not be lost.
- The shift register will be overwritten.
- An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

The OR bit is reset by an access to the SCISR register followed by a SCIDR register read operation.

Noise error

Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise. Normal data bits are considered valid if three consecutive samples (8th, 9th, 10th) have the same bit value, otherwise the NF flag is set. In the case of start bit detection, the NF flag is set on the basis of an algorithm combining both valid edge detection and three samples (8th, 9th, 10th). Therefore, to prevent the NF flag from being set during start bit reception, there should be a valid edge detection as well as three valid samples.

When noise is detected in a frame:

- The NF flag is set at the rising edge of the RDRF bit.
- 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 NF flag is reset by a SCISR register read operation followed by a SCIDR register read operation.

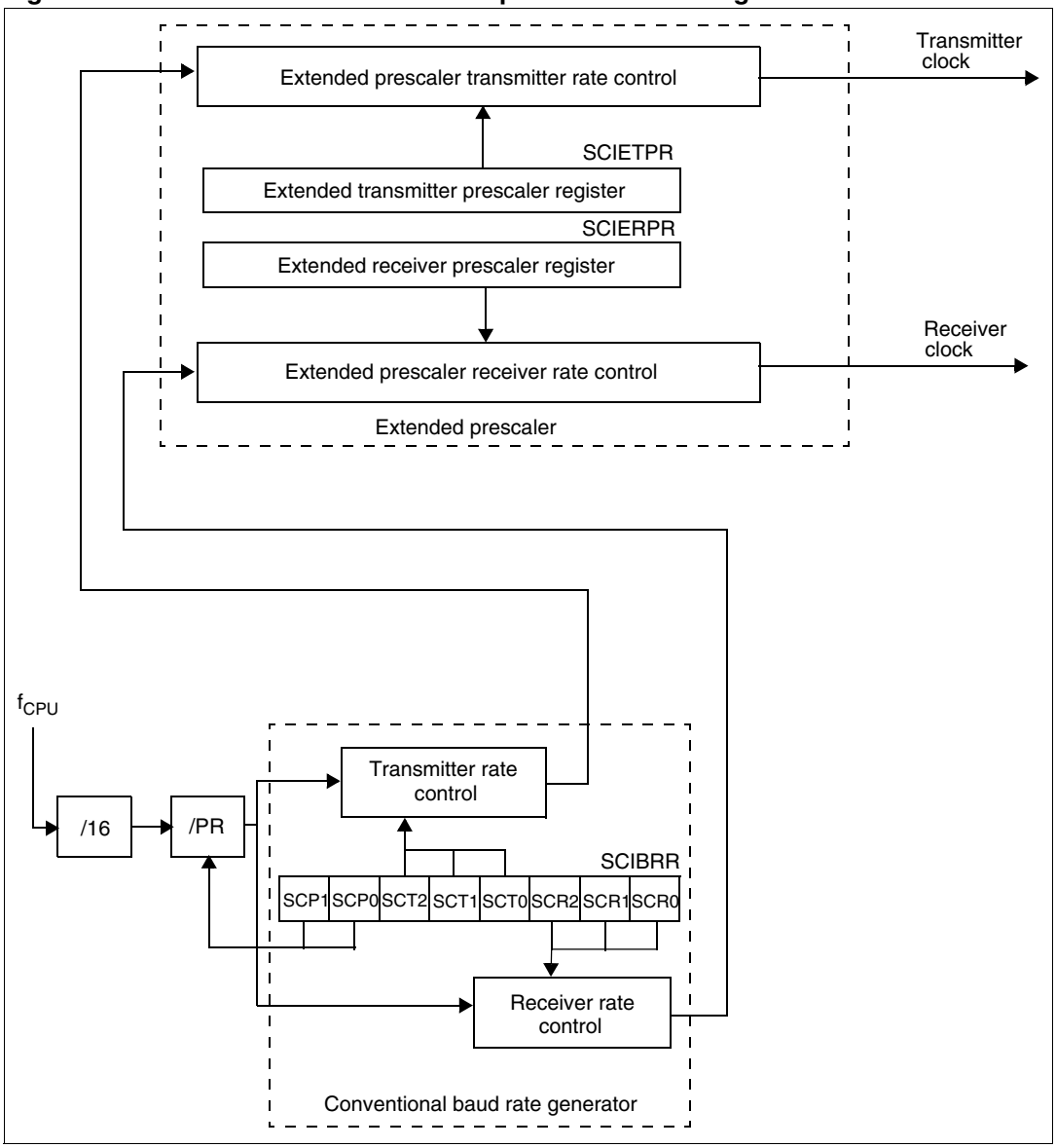
During reception, if a false start bit is detected (for example, 8th, 9th, 10th samples are 011,101,110), the frame is discarded and the receiving sequence is not started for this frame. There is no RDRF bit set for this frame and the NF flag is set internally (not accessible to the user). This NF flag is accessible along with the RDRF bit when a next valid frame is received.

Note: If the application Start bit is not long enough to match the above requirements, then the NF Flag may get set due to the short Start bit. In this case, the NF flag may be ignored by the application software when the first valid byte is received.

See also *Noise error causes on page 119*.

Downloaded from Arrow.com.

Figure 59. SCI baud rate and extended prescaler block diagram



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.

Downloaded from Arrow.com.

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 *SCI baud rate register (SCIBRR) on page 125*.

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.

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 *Figure 59*.

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.

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:

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

with:

ETPR = 1,..,255, see *SCI extended transmit prescaler division register (SCIETPR) on page 126*.

ERPR = 1,.. 255, see *SCI extended receive prescaler division register (SCIERPR) on page 125*.

Downloaded from Arrow.com.

Receiver muting and wakeup 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 cannot 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.

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

A 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 an address mark wakeup 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.

Parity control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in *Table 59*.

Table 59. Frame formats(1)(2)

M bit	PCE bit	SCI frame
0	0	| SB | 8 bit data | STB |
0	1	| SB | 7-bit data | PB | STB |
1	0	| SB | 9-bit data | STB |
1	1	| SB | 8-bit data PB | STB |

1. SB = Start bit, STB = Stop bit, and PB = Parity bit.
2. In case of wakeup by an address mark, the MSB bit of the data is taken into account and not the Parity bit.

Downloaded from Arrow.com.

Even parity

The parity bit is calculated to obtain an even number of ‘1’s inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit, for example, data = 00110101; 4 bits set => Parity bit will be 0 if Even parity is selected (PS bit = 0).

Odd parity

The parity bit is calculated to obtain an odd number of ‘1’s inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit, for example, data = 00110101; 4 bits set => Parity bit will be 1 if Odd parity is selected (PS bit = 1).

Transmission mode

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

Reception mode

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

SCI clock tolerance

During reception, each bit is sampled 16 times. The majority of the 8th, 9th and 10th samples is considered as the bit value. For a valid bit detection, all the three samples should have the same value otherwise the noise flag (NF) is set. For example: If the 8th, 9th and 10th samples are 0, 1 and 1 respectively, then the bit value will be ‘1’, but the Noise flag bit is set because the three samples values are not the same.

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 kbaud (bit length is 64 µs), then the 8th, 9th and 10th samples will be at 28 µs, 32 µs and 36 µs respectively (the first sample starting ideally at 0 µs). But if the falling edge of the internal clock occurs just before the pin value changes, the samples would then be out of sync by ~4 µs. This means the entire bit length must be at least 40µs (36 µs for the 10th sample + 4 µs for synchronization with the internal sampling clock).

Downloaded from Arrow.com.

Clock deviation causes

The causes which contribute to the total deviation are:

- D_{TRA}: Deviation due to transmitter error (local oscillator error of the transmitter or the transmitter is transmitting at a different baud rate).
- D_{QUANT}: Error due to the baud rate quantization of the receiver.
- D_{REC}: Deviation of the local oscillator of the receiver: This deviation can occur during the reception of one complete SCI message assuming that the deviation has been compensated at the beginning of the message.
- D_{TCL}: Deviation due to the transmission line (generally due to the transceivers)

All the deviations of the system should be added and compared to the SCI clock tolerance:

$$D_{TRA} + D_{QUANT} + D_{REC} + D_{TCL} < 3.75\%$$

Noise error causes

See also the description of Noise error in *Receiver on page 113*.

Start bit

The Noise Flag (NF) is set during start bit reception if one of the following conditions occurs:

1. A valid falling edge is not detected. A falling edge is considered to be valid if the three consecutive samples before the falling edge occurs are detected as ‘1’ and, after the falling edge occurs, during the sampling of the 16 samples, if one of the samples numbered 3, 5 or 7 is detected as a ‘1’.
2. During sampling of the 16 samples, if one of the samples numbered 8, 9 or 10 is detected as a ‘1’.

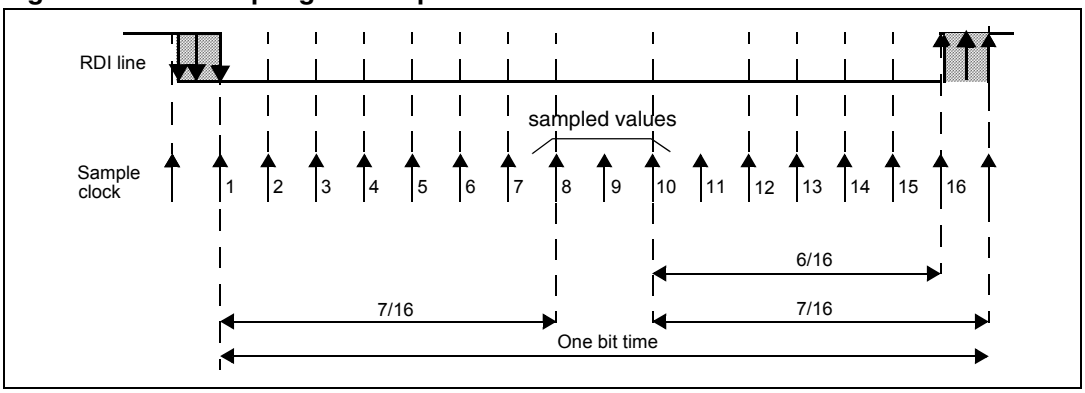
Therefore, a valid Start bit must satisfy both the above conditions to prevent the Noise Flag from being set.

Data bits

The Noise Flag (NF) is set during normal data bit reception if the following condition occurs: During the sampling of 16 samples, if all three samples numbered 8, 9 and10 are not the same. The majority of the 8th, 9th and 10th samples is considered as the bit value.

Therefore, a valid Data bit must have samples 8, 9 and 10 at the same value to prevent the Noise Flag from being set.

Figure 60. Bit sampling in Reception mode



Downloaded from Arrow.com.

10.5.5 Low power modes

Table 60. Effect of low power modes on SCI

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

10.5.6 Interrupts

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 61. SCI interrupt control/wakeup capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Transmit data register empty	TDRE	TIE	Yes	No
Transmission complete	TC	TCIE	Yes	No
Received data ready to be read	RDRF	RIE	Yes	No
Overrun error detected	OR		Yes	No
Idle line detected	IDLE	ILIE	Yes	No
Parity error	PE	PIE	Yes	No

10.5.7 SCI registers

SCI status register (SCISR)

SCISR Reset value: 1100 0000 (C0h)

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	PE
RO	RO	RO	RO	RO	RO	RO	RO

Table 62. SCISR register description

Bit	Name	Function
7	TDRE	Transmit Data Register Empty This bit is set by hardware when the content of the TDR register has been transferred into the shift register. An interrupt is generated if the TIE bit = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Data is not transferred to the shift register. 1: Data is transferred to the shift register. *Note: Data will not be transferred to the shift register unless the TDRE bit is cleared.*

Downloaded from Arrow.com.

Table 62. SCISR register description (continued)

Bit	Name	Function
6	TC	Transmission complete This bit is set by hardware when transmission of a frame containing data is complete. An interrupt is generated if TCIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a write to the SCIDR register). 0: Transmission is not complete 1: Transmission is complete *Note: TC is not set after the transmission of a Preamble or a Break.*
5	RDRF	Received data ready flag This bit is set by hardware when the content of the RDR register has been transferred to the SCIDR register. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: Data is not received 1: Received data is ready to be read
4	IDLE	Idle line detect This bit is set by hardware when a Idle Line is detected. An interrupt is generated if the ILIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No idle line is detected 1: Idle line is detected *Note: The IDLE bit is not reset until the RDRF bit has itself been set (that is, a new idle line occurs).*
3	OR	Overrun error This bit is set by hardware when the word currently being received in the shift register is ready to be transferred into the RDR register while RDRF = 1. An interrupt is generated if RIE = 1 in the SCICR2 register. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No overrun error 1: Overrun error is detected *Note: When this bit is set RDR register content is not lost but the shift register is overwritten.*
2	NF	Noise flag This bit is set by hardware when noise is detected on a received frame. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No noise is detected 1: Noise is detected *Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt.*
1	FE	Framing error This bit is set by hardware when a desynchronization, excessive noise or a break character is detected. It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register). 0: No framing error is detected 1: Framing error or break character is detected *Note: This bit does not generate interrupt as it appears at the same time as the RDRF bit which itself generates an interrupt. If the word currently being transferred causes both Frame Error and Overrun error, it is transferred and only the OR bit will be set.*

Downloaded from Arrow.com.

Table 62. SCISR register description (continued)

Bit	Name	Function
0	PE	Parity error This bit is set by hardware when a parity error occurs in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE = 1 in the SCICR1 register. 0: No parity error 1: Parity error

SCI Control Register 1 (SCICR1)

SCICR1 Reset value: x000 0000 (x0h)

7	6	5	4	3	2	1	0
R8	T8	SCID	M	WAKE	PCE	PS	PIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 63. SCICR1 register description

Bit	Name	Function
7	R8	Receive data bit 8 This bit is used to store the 9th bit of the received word when M = 1.
6	T8	Transmit data bit 8 This bit is used to store the 9th bit of the transmitted word when M = 1.
5	SCID	Disabled for low power consumption When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption.This bit is set and cleared by software. 0: SCI enabled 1: SCI prescaler and outputs disabled
4	M	Word length This bit determines the word length. It is set or cleared by software. 0: 1 Start bit, 8 data bits, 1 Stop bit 1: 1 Start bit, 9 data bits, 1 Stop bit *Note: The M bit must not be modified during a data transfer (both transmission and reception).*
3	WAKE	Wakeup method This bit determines the SCI wakeup method, it is set or cleared by software. 0: Idle line 1: Address mark
2	PCE	Parity control enable This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M = 1; 8th bit if M = 0) and parity is checked on the received data. This bit is set and cleared by software. Once it is set, PCE is active after the current byte (in reception and in transmission). 0: Parity control disabled 1: Parity control enabled

Downloaded from Arrow.com.

Table 63. SCICR1 register description (continued)

Bit	Name	Function
1	PS	Parity selection This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte. 0: Even parity 1: Odd parity
0	PIE	Parity interrupt enable This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software. 0: Parity error interrupt disabled 1: Parity error interrupt enabled

SCI control register 2 (SCICR2)

SCICR2 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 64. SCICR2 register description

Bit	Name	Function
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.
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.
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.
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.

Downloaded from Arrow.com.

Table 64. SCICR2 register description (continued)

Bit	Name	Function
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).
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 *Note: Before selecting Mute mode (setting the RWU bit), the SCI must first receive some data, otherwise it cannot function in Mute mode with wakeup by Idle line detection.*
1	RWU	Receiver wakeup 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 wakeup sequence is recognized. 0: Receiver in Active mode 1: Receiver in Mute mode
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.*

SCI data register (SCIDR)

This register contains the received or transmitted data character, depending on whether it is read from or written to.

SCIDR — Reset value: undefined

7	6	5	4	3	2	1	0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).

The TDR register provides the parallel interface between the internal bus and the output shift register (see *Figure 57*). The RDR register provides the parallel interface between the input shift register and the internal bus (see *Figure 57*).

Downloaded from Arrow.com.

SCI baud rate register (SCIBRR)

SCIBRR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SCP[1:0]		SCT[2:0]			SCR[2:0]		
R/W		R/W			R/W		

Table 65. SCIBRR register description

Bit	Name	Function
7:6	SCP[1:0]	First SCI prescaler These 2 prescaling bits allow several standard clock division ranges. 00: PR prescaling factor = 1 01: PR prescaling factor = 3 10: PR prescaling factor = 4 11: PR prescaling factor = 13
5:3	SCT[2:0]	SCI Transmitter rate divisor These 3 bits, in conjunction with the SCP1 and SCP0 bits, define the total division applied to the bus clock to yield the transmit rate clock in conventional baud rate generator mode. 000: TR dividing factor = 1 001: TR dividing factor = 2 010: TR dividing factor = 4 011: TR dividing factor = 8 100: TR dividing factor = 16 101: TR dividing factor = 32 110: TR dividing factor = 64 111: TR dividing factor = 128
2:0	SCR[2:0]	SCI Receiver rate divisor These 3 bits, in conjunction with the SCP[1:0] bits, define the total division applied to the bus clock to yield the receive rate clock in conventional baud rate generator mode. 000: RR dividing factor = 1 001: RR dividing factor = 2 010: RR dividing factor = 4 011: RR dividing factor = 8 100: RR dividing factor = 16 101: RR dividing factor = 32 110: RR dividing factor = 64 111: RR dividing factor = 128

SCI extended receive prescaler division register (SCIERPR)

This register is used to set the Extended Prescaler rate division factor for the receive circuit.

SCIERPR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ERPR[7:0]							
R/W							

Downloaded from Arrow.com.

Table 66. SCIERPR register description

Bit	Name	Function
7:0	ERPR[7:0]	8-bit extended receive prescaler register The extended baud rate generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see *Figure 59*) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255). The extended baud rate generator is not used after a reset.

SCI extended transmit prescaler division register (SCIETPR)

This register is used to set the External Prescaler rate division factor for the transmit circuit.

SCIETPR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ETPR[7:0]							
R/W							

Table 67. SCIETPR register description

Bit	Name	Function
7:0	ETPR[7:0]	8-bit Extended Transmit Prescaler Register The extended baud rate generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see *Figure 59*) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255). The extended baud rate generator is not used after a reset.

Table 68. Baud rate selection

Symbol	Parameter	Conditions			Standard	Baud rate	Unit
		f_{CPU}	Accuracy vs. Standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8 MHz	~0.16%	Conventional mode TR (or RR) = 128, PR = 13 TR (or RR) = 32, PR = 13 TR (or RR) = 16, PR = 13 TR (or RR) = 8, PR = 13 TR (or RR) = 4, PR = 13 TR (or RR) = 16, PR = 3 TR (or RR) = 2, PR =13 TR (or RR) = 1, PR = 13	300 1200 2400 4800 9600 10400 19200 38400	~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230.77 ~38461.54	Hz
			~0.79%	Extended mode ETPR (or ERPR) = 35, TR (or RR)= 1, PR = 1	14400	~14285.71	

Downloaded from Arrow.com.

Table 69. SCI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0050h	SCISR Reset value	TDRE 1	TC 1	RDRF 0	IDLE 0	OR 0	NF 0	FE 0	PE 0
0051h	SCIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0052h	SCIBRR Reset value	SCP1 0	SCP0 0	SCT2 0	SCT1 0	SCT0 0	SCR2 0	SCR1 0	SCR0 0
0053h	SCICR1 Reset value	R8 x	T8 0	SCID 0	M 0	WAKE 0	PCE 0	PS 0	PIE 0
0054h	SCICR2 Reset value	TIE 0	TCIE 0	RIE 0	ILIE 0	TE 0	RE 0	RWU 0	SBK 0
0055h	SCIERPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
0057h	SCIPETPR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

Downloaded from Arrow.com.

10.6 10-bit A/D converter (ADC)

10.6.1 Introduction

The on-chip analog-to-digital converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

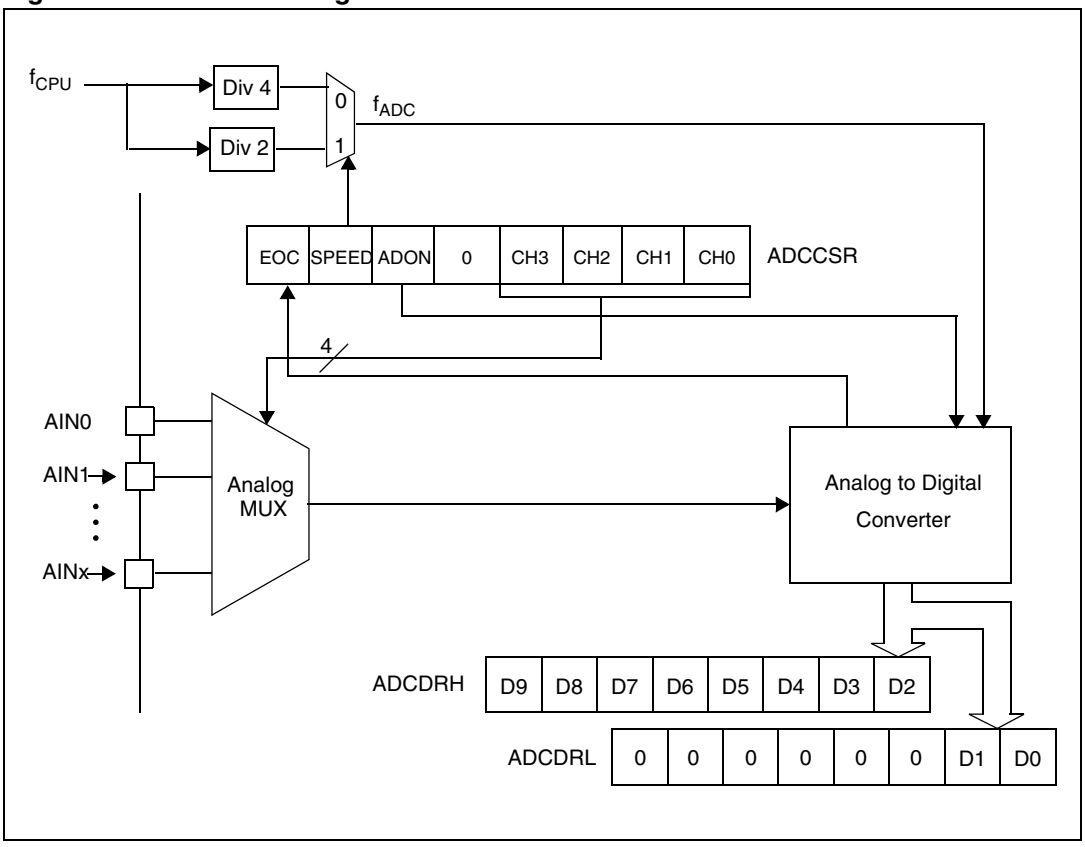
The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

10.6.2 Main features

- 10-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in *Figure 61*.

Figure 61. ADC block diagram



Downloaded from Arrow.com.

10.6.3 Functional description

The conversion is monotonic, meaning that the result never decreases if the analog input does not increase.

If the input voltage (V_{AIN}) is greater than V_{AREF} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the Electrical Characteristics Section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the allotted time.

A/D converter configuration

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to *Section 9: I/O ports*. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

Select the CS[3:0] bits to assign the analog channel to convert.

Starting the conversion

In the ADCCSR register:

Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

- the EOC bit is set by hardware
- the result is in the ADCDR registers

A read to the ADCDRH or a write to any bit of the ADCCSR register resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll the EOC bit.
2. Read the ADCDRL register
3. Read the ADCDRH register. This clears EOC automatically.

Note: The data is not latched, so both the low and the high data register must be read before the next conversion is complete. Therefore, it is recommended to disable interrupts while reading the conversion result.

To read only 8 bits, perform the following steps:

1. Poll the EOC bit.
2. Read the ADCDRH register. This clears EOC automatically.

Downloaded from Arrow.com.

Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

10.6.4 Low power modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed.

Table 70. Effect of low power modes on ADC

Mode	Description
Wait	No effect on A/D converter.
Halt	A/D converter disabled. After wakeup from Halt mode, the A/D converter requires a stabilization time t_{STAB} (see *Section 12: Electrical characteristics*) before accurate conversions can be performed.

10.6.5 Interrupts

None.

10.6.6 ADC registers

ADC control/status register (ADCCSR)

ADCCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EOC	SPEED	ADON	Reserved	CH[3:0]			
RO	R/W	RW	-	RW			

Table 71. ADCCSR register description

Bit	Name	Function
7	EOC	End of Conversion This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register. 0: Conversion is not complete 1: Conversion complete
6	SPEED	ADC clock selection This bit is set and cleared by software. 0: $f_{ADC} = f_{CPU}/4$ 1: $f_{ADC} = f_{CPU}/2$
5	ADON	A/D Converter on This bit is set and cleared by software. 0: Disable ADC and stop conversion 1: Enable ADC and start conversion

Downloaded from Arrow.com.

Table 71. ADCCSR register description (continued)

Bit	Name	Function
4	-	Reserved, must be kept cleared.
3:0	CH[3:0]	Channel selection These bits are set and cleared by software. They select the analog input to convert. 0000: Channel pin = AIN0 0001: Channel pin = AIN1 0010: Channel pin = AIN2 0011: Channel pin = AIN3 0100: Channel pin = AIN4 0101: Channel pin = AIN5 0110: Channel pin = AIN6 0111: Channel pin = AIN7 1000: Channel pin = AIN8 1001: Channel pin = AIN9 1010: Channel pin = AIN10 1011: Channel pin = AIN11 1100: Channel pin = AIN12 1101: Channel pin = AIN13 1110: Channel pin = AIN14 1111: Channel pin = AIN15 *Note: The number of channels is device dependent. Refer to Section 2: Pin description.*

ADC data register high (ADCDRH)

ADCDRH Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
D[9:2]							
RO							

Table 72. ADCDRH register description

Bit	Name	Function
7:0	D[9:2]	MSB of converted analog value

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

11 Instruction set

11.1 CPU addressing modes

The CPU features 17 different addressing modes which can be classified in 7 main groups (see *Table 75*).

Table 75. Addressing mode groups

Addressing mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The CPU Instruction Set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be divided in two submodes called long and short:

- Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 76. CPU addressing mode overview

Mode			Syntax	Destination	Pointer address (Hex.)	Pointer size (Hex.)	Length (bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No offset	Direct	Indexed	ld A,(X)	00..FF			+ 0
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2

Downloaded from Arrow.com.

Table 76. CPU addressing mode overview (continued)

Relative	Direct		jrne loop	PC+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

11.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Table 77. Inherent instructions

Instruction	Function
NOP	No operation
TRAP	S/W interrupt
WFI	Wait for interrupt (low power mode)
HALT	Halt oscillator (lowest power mode)
RET	Sub-routine return
IRET	Interrupt sub-routine return
SIM	Set interrupt mask (level 3)
RIM	Reset interrupt mask (level 0)
SCF	Set carry flag
RCF	Reset carry flag
RSP	Reset stack pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/decrement
TNZ	Test negative or zero
CPL, NEG	1 or 2 complement
MUL	Byte multiplication
SLL, SRL, SRA, RLC, RRC	Shift and rotate operations
SWAP	Swap nibbles

11.1.2 Immediate

Immediate instructions have two bytes: The first byte contains the opcode and the second byte contains the operand value.

Downloaded from Arrow.com.

Table 78. Immediate instructions

Instruction	Function
LD	Load
CP	Compare
BCP	Bit compare
AND, OR, XOR	Logical operations
ADC, ADD, SUB, SBC	Arithmetic operations

11.1.3 Direct

In Direct instructions, the operands are referenced by their memory address. The direct addressing mode consists of two submodes:

Direct (short)

The address is a byte, thus requiring only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

11.1.4 Indexed (no offset, short, long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indexed addressing mode consists of three submodes:

Indexed (no offset)

There is no offset, (no extra byte after the opcode), and it allows 00 - FF addressing space.

Indexed (short)

The offset is a byte, thus requiring only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

11.1.5 Indirect (short, long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two submodes:

Downloaded from Arrow.com.

Indirect (short)

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

Indirect (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

11.1.6 Indirect indexed (short, long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two submodes:

Indirect indexed (short)

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

Indirect indexed (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 79. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes

Instructions		Function
Long and short	LD	Load
	CP	Compare
	AND, OR, XOR	Logical operations
	ADC, ADD, SUB, SBC	Arithmetic additions/subtractions operations
	BCP	Bit Compare
Short only	CLR	Clear
	INC, DEC	Increment/decrement
	TNZ	Test negative or zero
	CPL, NEG	1 or 2 complement
	BSET, BRES	Bit operations
	BTJT, BTJF	Bit test and jump operations
	SLL, SRL, SRA, RLC, RRC	Shift and rotate operations
	SWAP	Swap nibbles
	CALL, JP	Call or jump sub-routine

Downloaded from Arrow.com.

11.1.7 Relative mode (direct, indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Table 80. Relative direct and indirect instructions and functions

Available relative direct/indirect instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (direct)

The offset follows the opcode.

Relative (indirect)

The offset is defined in the memory, the address of which follows the opcode.

11.2 Instruction groups

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Table 81. Instruction groups

Group	Instructions							
Load and transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/decrement	INC	DEC						
Compare and tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit operation	BSET	BRES						
Conditional bit test and branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional jump or call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Condition code flag modification	SIM	RIM	SCF	RCF				

Downloaded from Arrow.com.

Using a prebyte

The instructions are described with one to four opcodes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC Opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable the instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

Downloaded from Arrow.com.

Table 82. Instruction set overview

Mnemo	Description	Function/example	Dst	Src	I1	H	I0	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M		H		N	Z	C
ADD	Addition	A = A + M	A	M		H		N	Z	C
AND	Logical And	A = A . M	A	M				N	Z	
BCP	Bit compare A, memory	tst (A . M)	A	M				N	Z	
BRES	Bit reset	bres Byte, #3	M							
BSET	Bit set	bset Byte, #3	M							
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M							C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M							C
CALL	Call sub-routine									
CALLR	Call sub-routine relative									
CLR	Clear		reg, M					0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M				N	Z	C
CPL	One Complement	A = FFH-A	reg, M					N	Z	1
DEC	Decrement	dec Y	reg, M					N	Z	
HALT	Halt				1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC			I1	H	I0	N	Z	C
INC	Increment	inc X	reg, M					N	Z	
JP	Absolute Jump	jp [TBL.w]								
JRA	Jump relative always									
JRT	Jump relative									
JRF	Never jump	jrf *								
JRIH	Jump if ext. INT pin = 1	(ext. INT pin high)								
JRIL	Jump if ext. INT pin = 0	(ext. INT pin low)								
JRH	Jump if H = 1	H = 1 ?								
JRNH	Jump if H = 0	H = 0 ?								
JRM	Jump if I1:0 = 11	I1:0 = 11 ?								
JRNM	Jump if I1:0 <> 11	I1:0 <> 11 ?								
JRMI	Jump if N = 1 (minus)	N = 1 ?								
JRPL	Jump if N = 0 (plus)	N = 0 ?								
JREQ	Jump if Z = 1 (equal)	Z = 1 ?								
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?								
JRC	Jump if C = 1	C = 1 ?								
JRNC	Jump if C = 0	C = 0 ?								
JRULT	Jump if C = 1	Unsigned <								
JRUGE	Jump if C = 0	Jmp if unsigned >=								

Downloaded from Arrow.com.

Table 82. Instruction set overview (continued)

Mnemo	Description	Function/example	Dst	Src	I1	H	I0	N	Z	C
JRUGT	Jump if (C + Z = 0)	Unsigned >								
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	I1	H	I0	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC						
RCF	Reset carry flag	C = 0								0
RIM	Enable Interrupts	I1:0 = 10 (level 0)			1		0			
RLC	Rotate Left true C	C <= A <= C	reg, M					N	Z	C
RRC	Rotate Right true C	C => A => 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	I1:0 = 11 (level 3)			1		1			
SLA	Shift Left Arithmetic	C <= A <= 0	reg, M					N	Z	C
SLL	Shift Left Logic	C <= A <= 0	reg, M					N	Z	C
SRL	Shift Right Logic	0 => A => C	reg, M					0	Z	C
SRA	Shift Right Arithmetic	A7 => A => C	reg, M					N	Z	C
SUB	Subtraction	A = A - M	A	M				N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M					N	Z	
TNZ	Test for Neg and Zero	tnz lbl1						N	Z	
TRAP	S/W TRAP	S/W interrupt			1		1			
WFI	Wait for Interrupt				1		0			
XOR	Exclusive OR	A = A XOR M	A	M				N	Z	

Downloaded from Arrow.com.

12 Electrical characteristics

12.1 Parameter conditions

Unless otherwise specified, all voltages are referred to V_{SS}.

12.1.1 Minimum and maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at T_A = 25°C and T_A = $T_A$max (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean ±3Σ).

12.1.2 Typical values

Unless otherwise specified, typical data are based on T_A = 25 °C, V_{DD} = 5 V. They are given only as design guidelines and are not tested.

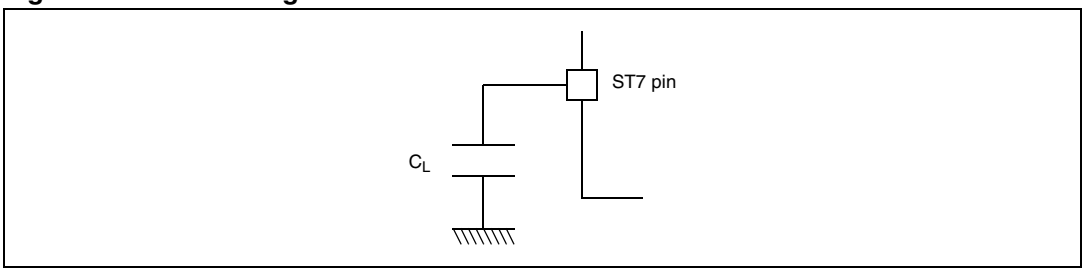
12.1.3 Typical curves

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

12.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 62*.

Figure 62. Pin loading conditions

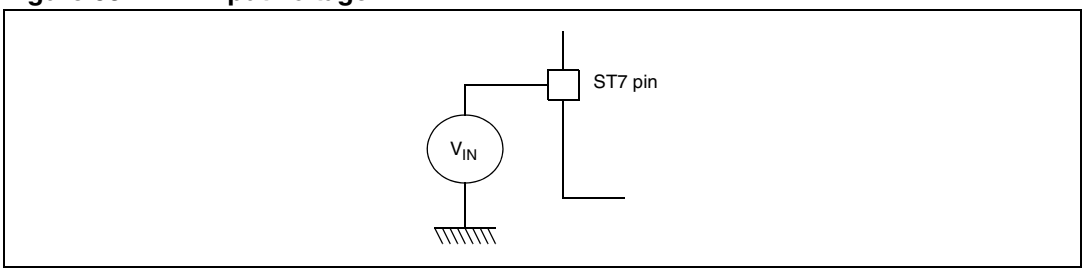


Downloaded from Arrow.com.

12.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 63*.

Figure 63. Pin input voltage



12.2 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

12.2.1 Voltage characteristics

Table 83. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.5	V
V_{PP} - V_{SS}	Programming voltage	13	
V_{IN} (1)(2)	Input voltage on true open drain pin	V_{SS} - 0.3 to 6.5	
	Input voltage on any other pin	V_{SS} - 0.3 to V_{DD} + 0.3	
$\lvert\Delta V_{DDx}\rvert$ and $\lvert\Delta V_{SSx}\rvert$	Variations between different digital power pins	50	mV
$\lvert V_{SSA} - V_{SSx}\rvert$	Variations between digital and analog ground pins	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see *Section 12.8.3 on page 157*	
$V_{ESD(MM)}$	Electrostatic discharge voltage (machine model)		

1. Directly connecting the $\overline{\text{RESET}}$ and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7 kΩ for $\overline{\text{RESET}}$, 10 kΩ for I/Os). For the same reason, unused I/O pins must not be directly tied to V_{DD} or V_{SS}.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly ensured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected.

Downloaded from Arrow.com.

12.2.2 Current characteristics

Table 84. Current characteristics

Symbol	Ratings		Max value	Unit
I_{VDD}	Total current into V_{DD} power lines (source)(1)	32-pin devices	75	mA
		44-pin devices	150	
I_{VSS}	Total current out of V_{SS} ground lines (sink)(1)	32-pin devices	75	
		44-pin devices	150	
I_{IO}	Output current sunk by any standard I/O and control pin		20	
	Output current sunk by any high sink I/O pin		40	
	Output current source by any I/Os and control pin		- 25	
$I_{INJ(PIN)}$(2)(3)	Injected current on V_{PP} pin		± 5	
	Injected current on $\overline{\text{RESET}}$ pin		± 5	
	Injected current on OSC1 and OSC2 pins		± 5	
	Injected current on ROM and 32 Kbyte Flash devices PB0 pin		± 5	
	Injected current on 8/16 Kbyte Flash devices PB0 pin		+ 5	
	Injected current on any other pin(4)(5)		± 5	
$\Sigma I_{INJ(PIN)}$(2)	Total injected current (sum of all I/O and control pins)(4)		± 25	

1. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected.
3. Negative injection degrades the analog performance of the device. See note in *Section 12.13.3: ADC accuracy on page 171*. If the current injection limits given in *Table 106: General characteristics on page 158* are exceeded, general device malfunction may result.
4. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.
5. True open drain I/O port pins do not accept positive injection.

12.2.3 Thermal characteristics

Table 85. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see *Section 13.3: Thermal characteristics*)		

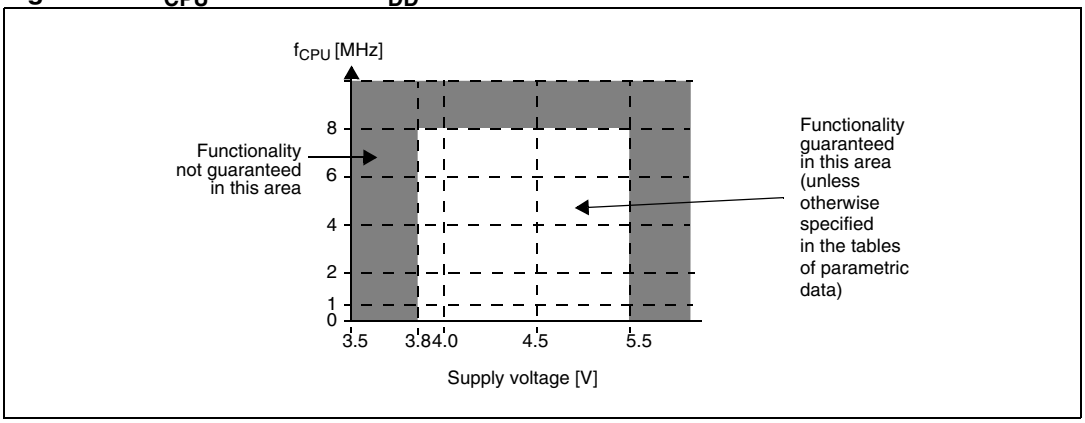
Downloaded from Arrow.com.

12.3 Operating conditions

Table 86. Operating conditions

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CPU}	Internal clock frequency		0	8	MHz
V_{DD}	Operating voltage (except Flash Write/Erase)		3.8	5.5	V
	Operating Voltage for Flash Write/Erase	V_{PP} = 11.4 to 12.6 V	4.5	5.5	
T_A	Ambient temperature range	1-suffix version	0	70	°C
		5-suffix version	-10	85	
		6-suffix version	-40	85	
		7-suffix version	-40	105	
		3-suffix version	-40	125	

Figure 64. f_{CPU} max versus V_{DD}



Note: Some temperature ranges are only available with a specific package and memory size. Refer to Section 14: Device configuration and ordering information.

Warning: Do not connect 12 V to V_{PP} before V_{DD} is powered on, as this may damage the device.

Downloaded from Arrow.com.

12.4 LVD/AVD characteristics

12.4.1 Operating conditions with LVD

Subject to general operating conditions for T_A.

Table 87. Operating conditions with LVD

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)	VD level = high in option byte	4.0(1)	4.2	4.5	V
		VD level = med. in option byte(2)	3.55(1)	3.75	4.0(1)	
		VD level = low in option byte(2)	2.95(1)	3.15	3.35(1)	
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)	VD level = high in option byte	3.8	4.0	4.25(1)	
		VD level = med. in option byte(2)	3.35(1)	3.55	3.75(1)	
		VD level = low in option byte(2)	2.8(1)	3.0	3.15(1)	
$V_{hys(LVD)}$	LVD voltage threshold hysteresis(1)	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$	150	200	250	mV
Vt_{POR}	V_{DD} rise time(1)	Flash devices	6µs/V		100ms/V	
		8/16 Kbyte ROM devices			20ms/V	
		32 Kbyte ROM devices			∝ ms/V	
$t_{g(VDD)}$	Filtered glitch delay on V_{DD}(1)	Not detected by the LVD			40	ns

1. Data based on characterization results, tested in production for ROM devices only.
2. If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range.

12.4.2 Auxiliary voltage detector (AVD) thresholds

Subject to general operating conditions for T_A.

Table 88. AVD thresholds

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(AVD)}$	1 ⇒0 AVDF flag toggle threshold (V_{DD} rise)	VD level = high in option byte	4.4(1)	4.6	4.9	V
		VD level = med. in option byte	3.95(1)	4.15	4.4(1)	
		VD level = low in option byte	3.4(1)	3.6	3.8(1)	
$V_{IT-(AVD)}$	0 ⇒1 AVDF flag toggle threshold (V_{DD} fall)	VD level = high in option byte	4.2	4.4	4.65(1)	
		VD level = med. in option byte	3.75(1)	4.0	4.2(1)	
		VD level = low in option byte	3.2(1)	3.4	3.6(1)	
$V_{hys(AVD)}$	AVD voltage threshold hysteresis	$V_{IT+(AVD)}$-$V_{IT-(AVD)}$		200		mV
ΔV_{IT-}	Voltage drop between AVD flag set and LVD reset activated	$V_{IT-(AVD)}$-$V_{IT-(LVD)}$		450		

1. Data based on characterization results, tested in production for ROM devices only.

Downloaded from Arrow.com.

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	32 Kbyte ROM devices		16/8 Kbyte 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*).

Downloaded from Arrow.com.

12.5.2 Flash current consumption

Table 90. Flash current consumption

Symbol	Parameter	Conditions	32 Kbyte Flash		16/8 Kbyte Flash		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	1.3 2.0 3.6 7.1	3.0 5.0 8.0 15.0	1 1.4 2.4 4.4	2.3 3.5 5.3 7.0	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	0.6 0.7 0.8 1.1	2.7 3.0 3.6 4.0	0.48 0.53 0.63 0.80	1 1.1 1.2 1.4	
	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	0.8 1.2 2.0 3.5	3.0 4.0 5.0 7.0	0.6 0.9 1.3 2.3	1.8 2.2 2.6 3.6	
	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	580 650 770 1050	1200 1300 1800 2000	430 470 530 660	950 1000 1050 1200	µA
	Supply current in Halt mode(3)	-40°C ≤ T_A ≤ +85°C	<1	10	<1	10	
		-40°C ≤ T_A ≤ +125°C	5	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	365 380 410 500	475 500 550 650	315 330 360 460	425 450 500 600	

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*).

Downloaded from Arrow.com.

12.5.3 Supply and clock managers

The previous 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).

Table 91. Oscillators, PLL and LVD current consumption

Symbol	Parameter	Conditions	Typ	Max	Unit
$I_{DD(RCINT)}$	Supply current of internal RC oscillator		625		µA
$I_{DD(RES)}$	Supply current of resonator oscillator(1)(2)		see Section 12.6.3 on page 150		
$I_{DD(PLL)}$	PLL supply current	V_{DD} = 5V	360		
$I_{DD(LVD)}$	LVD supply current		150	300	

1. Data based on characterization results done with the external components specified in Section 12.6.3, not tested in production.
2. As the oscillator is based on a current source, the consumption does not depend on the voltage.

12.5.4 On-chip peripherals

Table 92. On-chip peripherals current consumption

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(TIM)}$	16-bit timer supply current(1)	T_A = 25 °C, f_{CPU} = 4 MHz, V_{DD} = 5.0 V	50	µA
$I_{DD(SPI)}$	SPI supply current(2)		400	
$I_{DD(SCI)}$	SCI supply current(3)			
$I_{DD(ADC)}$	ADC supply current when converting(4)			

1. Data based on a differential I_{DD} measurement between reset configuration (timer counter running at $f_{CPU}/4$) and timer counter stopped (only TIMD bit set). Data valid for one timer.
2. Data based on a differential I_{DD} measurement between reset configuration (SPI disabled) and a permanent SPI master communication at maximum speed (data sent equal to 55h). This measurement includes the pad toggling consumption.
3. Data based on a differential I_{DD} measurement between SCI low power state (SCID = 1) and a permanent SCI data transmit sequence.
4. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

Downloaded from Arrow.com.

12.6 Clock and timing characteristics

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A.

12.6.1 General timings

Table 93. General timings

Symbol	Parameter	Conditions	Min	Typ(1)	Max	Unit
$t_{c(INST)}$	Instruction cycle time		2	3	12	t_{CPU}
		f_{CPU} = 8 MHz	250	375	1500	ns
$t_{v(IT)}$	Interrupt reaction time $t_{v(IT)} = \Delta t_{c(INST)} + 10$(2)		10		22	t_{CPU}
		f_{CPU} = 8 MHz	1.25		2.75	µs

1. Data based on typical application software.
2. Time measured between interrupt event and interrupt vector fetch. $\Delta t_{c(INST)}$ is the number of t_{CPU} cycles needed to finish the current instruction execution.

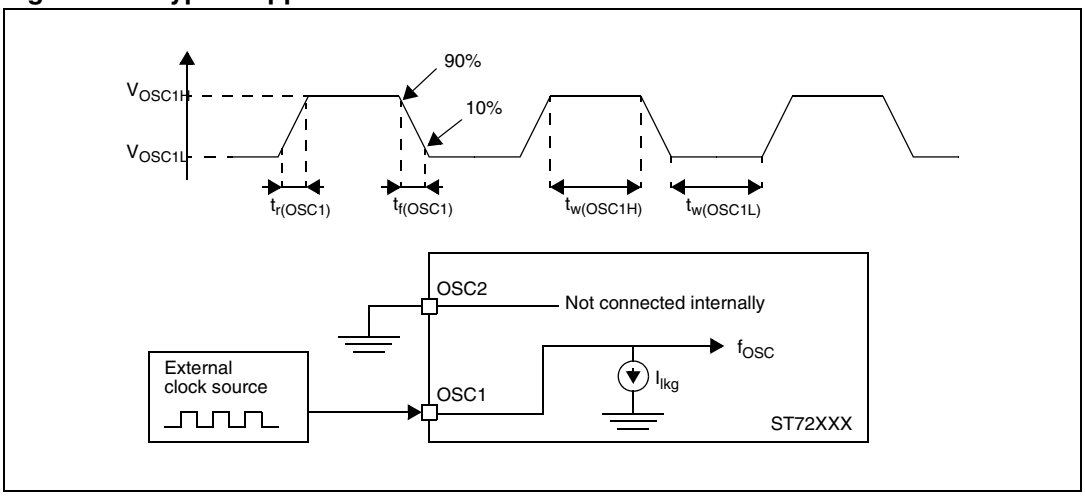
12.6.2 External clock source

Table 94. External clock source

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OSC1H}	OSC1 input pin high level voltage	See *Figure 65.*	V_{DD}-1		V_{DD}	V
V_{OSC1L}	OSC1 input pin low level voltage		V_{SS}		V_{SS}+1	
$t_{w(OSC1H)}$ $t_{w(OSC1L)}$	OSC1 high or low time(1)		5			ns
$t_{r(OSC1)}$ $t_{f(OSC1)}$	OSC1 rise or fall time(1)				15	
I_{lkg}	OSC1 input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$			±1	µA

1. Data based on design simulation and/or technology characteristics, not tested in production.

Figure 65. Typical application with an external clock source



Downloaded from Arrow.com.

12.6.3 Crystal and ceramic resonator oscillators

The ST7 internal clock can be supplied with four different crystal/ceramic resonator oscillators. All the information given in this paragraph are based on characterization results with specified typical external components. In the application, 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. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

8/16 Kbyte Flash and ROM devices

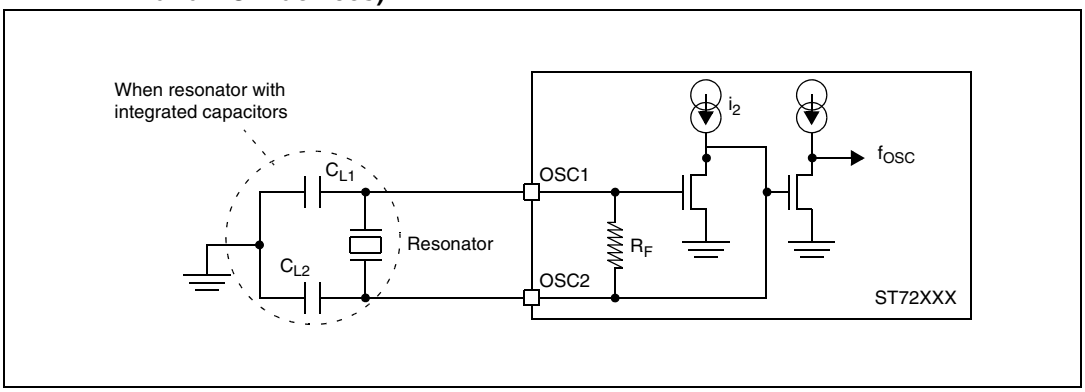
Table 95. Crystal and ceramic resonator oscillators (8/16 Kbyte Flash and ROM devices)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	Oscillator frequency(1)	LP: low power oscillator MP: medium power oscillator MS: medium speed oscillator HS: high speed oscillator	1 >2 >4 >8		2 4 8 16	MHz
R_F	Feedback resistor(2)		20		40	kΩ
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)(3)	R_S = 200 Ω LP oscillator R_S = 200 Ω MP oscillator R_S = 200 Ω MS oscillator R_S = 100 Ω HS oscillator	22 22 18 15		56 46 33 33	pF
i_2	OSC2 driving current	V_{DD} = 5V, V_{IN} = V_{SS} LP oscillator MP oscillator MS oscillator HS oscillator		 80 160 310 610	 150 250 460 910	µA

1. The oscillator selection can be optimized in terms of supply current using an high quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.
2. Data based on characterization results, not tested in production. The relatively low value of the RF resistor, offers a good protection against issues resulting from use in a humid environment, due to the induced leakage and the bias condition change. However, it is recommended to take this point into account if the microcontroller is used in tough humidity conditions.
3. For C_{L1} and C_{L2} it is recommended to use high-quality ceramic capacitors in the 5 pF to 25 pF range (typ.) designed for high-frequency applications and selected to match the requirements of the crystal or resonator. C_{L1} and C_{L2}, are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of C_{L1} and C_{L2}. PCB and MCU pin capacitance must be included when sizing C_{L1} and C_{L2} (10 pF can be used as a rough estimate of the combined pin and board capacitance).

Downloaded from Arrow.com.

Figure 66. Typical application with a crystal or ceramic resonator (8/16 Kbyte Flash and ROM devices)



32 Kbyte Flash and ROM devices

Table 96. Crystal and ceramic resonator oscillators (32 Kbyte Flash and ROM devices)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	Oscillator frequency(1)		1		16	MHz
R_F	Feedback resistor(2)		20		40	kΩ
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)(3)	f_{OSC} = 1 to 2 MHz f_{OSC} = 2 to 4 MHz f_{OSC} = 4 to 8 MHz f_{OSC} = 8 to 16 MHz	20 20 15 15		60 50 35 35	pF
i_2	OSC2 driving current	V_{DD} = 5V, V_{IN} = V_{SS} LP oscillator MP oscillator MS oscillator HS oscillator		 80 160 310 610	 150 250 460 910	µA

1. The oscillator selection can be optimized in terms of supply current using an high quality resonator with small RS value. Refer to crystal/ceramic resonator manufacturer for more details.
2. Data based on characterization results, not tested in production. The relatively low value of the RF resistor, offers a good protection against issues resulting from use in a humid environment, due to the induced leakage and the bias condition change. However, it is recommended to take this point into account if the microcontroller is used in tough humidity conditions.
3. For C_{L1} and C_{L2} it is recommended to use high-quality ceramic capacitors in the 5-pF to 25-pF range (typ.) designed for high-frequency applications and selected to match the requirements of the crystal or resonator. C_{L1} and C_{L2}, are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of C_{L1} and C_{L2}. PCB and MCU pin capacitance must be included when sizing C_{L1} and C_{L2} (10 pF can be used as a rough estimate of the combined pin and board capacitance).

Downloaded from Arrow.com.

Figure 67. Typical application with a crystal or ceramic resonator (32 Kbyte Flash and ROM devices)

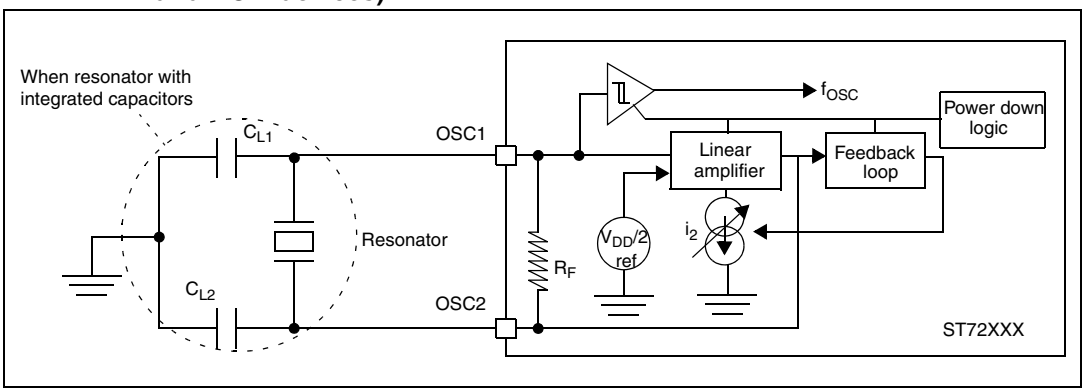


Table 97. OSCRANGE selection for typical resonators

Supplier	f_{OSC} (MHz)	Typical ceramic resonators(1)	
		Reference	Recommended OSCRANGE option bit configuration
Murata	2	CSTCC2M00G56A-R0	MP mode(2)
	4	CSTCR4M00G55B-R0	MS mode
	8	CSTCE8M00G52A-R0	HS mode
	16	CSTCE16M0V51A-R0	HS mode

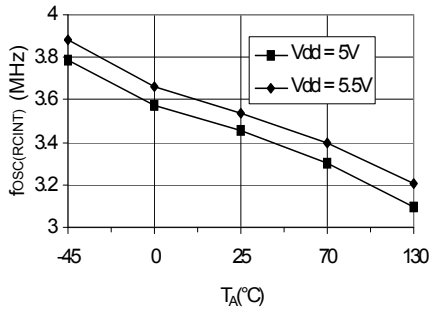
1. Resonator characteristics given by the ceramic resonator manufacturer.
2. LP mode is not recommended for 2 MHz resonator because the peak to peak amplitude is too small (>0.8 V). For more information on these resonators, please consult www.murata.com.

12.6.4 RC oscillators

Table 98. RC oscillators

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{OSC\ (RCINT)}$	Internal RC oscillator frequency (see *Figure 68)*	T_A = 25 °C, V_{DD} = 5 V	2	3.5	5.6	MHz

Figure 68. Typical $f_{OSC(RCINT)}$ vs T_A



Downloaded from Arrow.com.

Note: To reduce disturbance to the RC oscillator, it is recommended to place decoupling capacitors between V_{DD} and V_{SS} as shown in Figure 87 on page 170.

12.6.5 PLL characteristics

Table 99. PLL characteristics

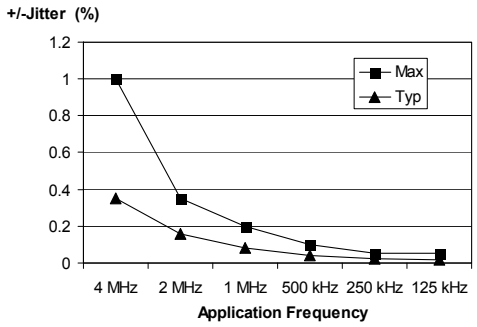
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{OSC}	PLL input frequency range		2		4	MHz
$\Delta f_{CPU}/f_{CPU}$	Instantaneous PLL jitter(1)	f_{OSC} = 4 MHz		0.7	2	%

1. Data characterized but not tested

The user must take the PLL jitter into account in the application (for example in serial communication or sampling of high frequency signals). The PLL jitter is a periodic effect, which is integrated over several CPU cycles. Therefore the longer the period of the application signal, the less it will be impacted by the PLL jitter.

Figure 69 shows the PLL jitter integrated on application signals in the range 125 kHz to 2 MHz. At frequencies of less than 125 kHz, the jitter is negligible.

Figure 69. Integrated PLL jitter vs signal frequency(1)



1. Measurement conditions: f_{CPU} = 8 MHz

12.7 Memory characteristics

12.7.1 RAM and hardware registers

Table 100. RAM and hardware registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode(1)	Halt mode (or reset)	1.6			V

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in Halt mode or under reset) or in hardware registers (only in Halt mode). Not tested in production.

Downloaded from Arrow.com.

12.7.2 Flash memory

Table 101. Dual voltage HDFlash memory

Symbol	Parameter	Conditions	Min(1)	Typ	Max(1)	Unit
f_{CPU}	Operating frequency	Read mode	0		8	MHz
		Write/Erase mode	1		8	
V_{PP}	Programming voltage(2)	4.5 V ≤ V_{DD} ≤ 5.5 V	11.4		12.6	V
I_{DD}	Supply current(3)	Write/Erase		<10		µA
I_{PP}	V_{PP} current(3)	Read (V_{PP} = 12 V)			200	µA
		Write/Erase			30	mA
t_{VPP}	Internal V_{PP} stabilization time			10		µs
t_{RET}	Data retention	T_A = 85 °C	40			years
		T_A = 105 °C	15			
		T_A = 125 °C	7			
N_{RW}	Write erase cycles	T_A=85 °C	100			cycles
		T_A=55 °C	1000			cycles
T_{PROG} T_{ERASE}	Programming or erasing temperature range		-40	25	85	°C

1. Data based on characterization results, not tested in production.
2. V_{PP} must be applied only during the programming or erasing operation and not permanently for reliability reasons.
3. Data based on simulation results, not tested in production.

Downloaded from Arrow.com.

12.8 EMC characteristics

Susceptibility tests are performed on a sample basis during product characterization.

12.8.1 Functional electromagnetic susceptibility (EMS)

Based on a simple running application on the product (toggling two LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- **ESD**: Electrostatic discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- **FTB**: A burst of fast transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100 pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results given in *Table 102 on page 156* are based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations

The software flowchart must include the management of runaway conditions such as:

- corrupted program counter
- unexpected reset
- critical data corruption (control registers...)

Prequalification trials

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the reset pin or the oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Downloaded from Arrow.com.

Table 102. EMS test results

Symbol	Parameter	Conditions	Level/class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	32 Kbyte Flash or ROM device: V_{DD} = 5 V, T_A = +25 °C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	3B
		8 or 16 Kbyte ROM device: V_{DD} = 5 V, T_A = +25°C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	4A
		8 or 16 Kbyte Flash device: V_{DD} = 5 V, T_A = +25 °C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	4B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100 pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD} = 5 V, T_A = +25 °C, f_{OSC} = 8 MHz conforms to IEC 1000-4-4	4A

12.8.2 Electromagnetic interference (EMI)

Based on a simple application running on the product (toggling two LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/3 which specifies the board and the loading of each pin.

Table 103. EMI emissions

Symbol	Parameter	Conditions	Device/package(1)	Monitored frequency band	Max vs [f_{OSC}/f_{CPU}]		Unit
					8/4 MHz	16/8 MHz	
S_{EMI}	Peak level(2)	V_{DD} = 5 V T_A = +25 °C conforming to SAE J 1752/3	8/16 Kbyte Flash LQFP32 and LQFP44	0.1 MHz to 30 MHz	12	18	dBµV
				30 MHz to 130 MHz	19	25	
				130 MHz to 1 GHz	15	22	
				SAE EMI Level	3	3.5	-
			32 Kbyte Flash LQFP32 and LQFP44	0.1 MHz to 30 MHz	13	14	dBµV
				30 MHz to 130 MHz	20	25	
				130 MHz to 1 GHz	16	21	
				SAE EMI Level	3.0	3.5	-
			8/16 Kbyte ROM LQFP32 and LQFP44	0.1 MHz to 30 MHz	12	15	dBµV
				30 MHz to 130 MHz	23	26	
				130 MHz to 1 GHz	15	20	
				SAE EMI Level	3.0	3.5	-
			32 Kbyte ROM LQFP32 and LQFP44	0.1 MHz to 30 MHz	17	21	dBµV
				30 MHz to 130 MHz	24	30	
				130 MHz to 1 GHz	18	23	
				SAE EMI Level	3.0	3.5	-

1. Refer to application note AN1709 for data on other package types.
2. Not tested in production.

Downloaded from Arrow.com.

12.8.3 Absolute maximum ratings (electrical sensitivity)

Based on two different tests (ESD and LU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity. For more details, refer to the application note AN1181.

Electrostatic discharge (ESD)

Electrostatic discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts*(n+1) supply pin). Two models can be simulated: human body model and machine model. This test conforms to the JESD22-A114A/A115A standard.

Table 104. Absolute maximum ratings

Symbol	Ratings	Conditions	Maximum value(1)	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	T_A = +25 °C	2000	V
$V_{ESD(CDM)}$	Electrostatic discharge voltage (charged device model)		750	

1. Data based on characterization results, not tested in production.

Static latch-up

- **LU**: two complementary static tests are required on 6 parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin) and a current injection (applied to each input, output and configurable I/O pin) are performed on each sample. This test conforms to the EIA/JESD 78 IC latch-up standard.

Table 105. Electrical sensitivities

Symbol	Parameter	Conditions	Test specification	Class
LU	Static latch-up class	T_A = +25 °C T_A = +85 °C T_A = +125 °C	JESD 78	II level A

Downloaded from Arrow.com.

12.9 I/O port pin characteristics

12.9.1 General characteristics

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

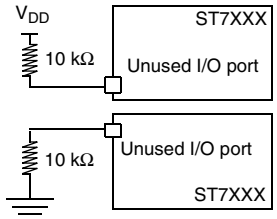
Table 106. General characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage (standard voltage devices)(1)				$0.3xV_{DD}$	V
V_{IH}	Input high level voltage(1)		$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis(2)			0.7		
$I_{INJ(PIN)}$(3)	Injected current on I/O pins other than pin PB0(4)	V_{DD} = 5 V			±4	mA
	Injected current on ROM and 32 Kbyte Flash devices pin PB0					
	Injected current on 8/16 Kbyte Flash devices pin PB0		0		+4	
$\Sigma I_{INJ(PIN)}$(3)	Total injected current (sum of all I/O and control pins)	V_{DD} = 5 V			±25	mA
I_{lkg}	Input leakage current	$V_{SS} \le V_{IN} \le V_{DD}$			±1	µA
I_S	Static current consumption induced by each floating input pin	Floating input mode(5)(6)		200		
R_{PU}	Weak pull-up equivalent resistor(7)	$V_{IN} = V_{SS}$, V_{DD} = 5 V	50	120	250	kΩ
C_{IO}	I/O pin capacitance			5		pF
$t_{f(IO)out}$	Output high to low level fall time(1)	C_L = 50 pF between 10% and 90%		25		ns
$t_{r(IO)out}$	Output low to high level rise time(1)			25		
$t_{w(IT)in}$	External interrupt pulse time(8)		1			t_{CPU}

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
3. When the current limitation is not possible, the V_{IN} maximum must be respected, otherwise refer to the $I_{INJ(PIN)}$ specification. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. Refer to *Section 12.2.2 on page 143* for more details.
4. No negative current injection allowed on 8/16 Kbyte Flash devices
5. Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.
6. The Schmitt trigger that is connected to every I/O port is disabled for analog inputs only when ADON bit is ON and the particular ADC channel is selected (with port configured in input floating mode). When the ADON bit is OFF, static current consumption may result. This can be avoided by keeping the input voltage of this pin close to V_{DD} or V_{SS}.
7. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in *Figure 71*).
8. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

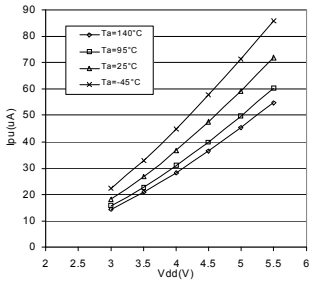
Downloaded from Arrow.com.

Figure 70. Unused I/O pins configured as input(1)



1. I/O can be left unconnected if it is configured as output (0 or 1) by the software. This has the advantage of greater EMC robustness and lower cost.

Figure 71. Typical I_{PU} vs. V_{DD} with $V_{IN} = V_{SS}$



12.9.2 Output driving current

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

Table 107. Output driving current

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL}(1)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see *Figure 72*)	V_{DD} = 5V	I_{IO} = +5 mA		1.2	V
			I_{IO} = +2 mA		0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 73* and *Figure 75*)		I_{IO} = +20 mA $T_A \le 85$ °C $T_A > 85$ °C		 1.3 1.5	
			I_{IO} = +8 mA		0.6	
V_{OH}(2)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 74* and *Figure 77*)		I_{IO} = -5 mA, $T_A \le 85$ °C $T_A > 85$ °C	 V_{DD} - 1.4 V_{DD} - 1.6		
			I_{IO} = -2 mA	V_{DD} - 0.7		

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

Downloaded from Arrow.com.

Figure 72. Typical V_{OL} at V_{DD} = 5 V (standard ports)

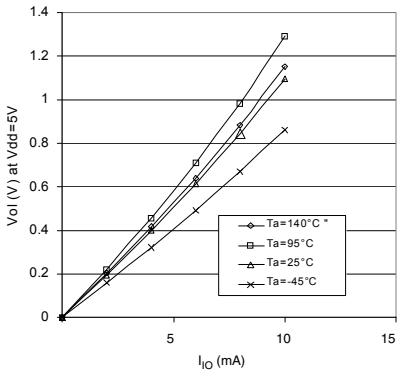


Figure 73. Typical V_{OL} at V_{DD} = 5 V (high-sink ports)

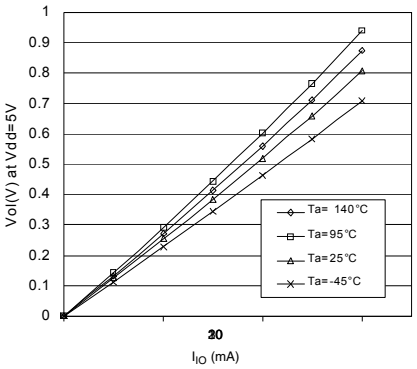
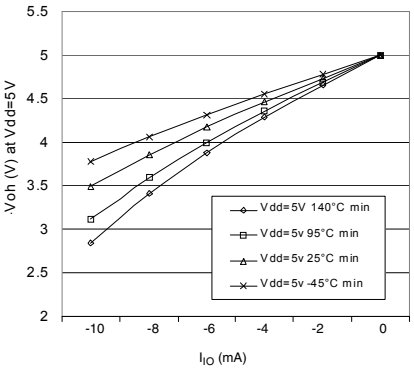


Figure 74. Typical V_{OH} at V_{DD} = 5 V



Downloaded from Arrow.com.

Figure 75. Typical V_{OL} vs. V_{DD} (standard ports)

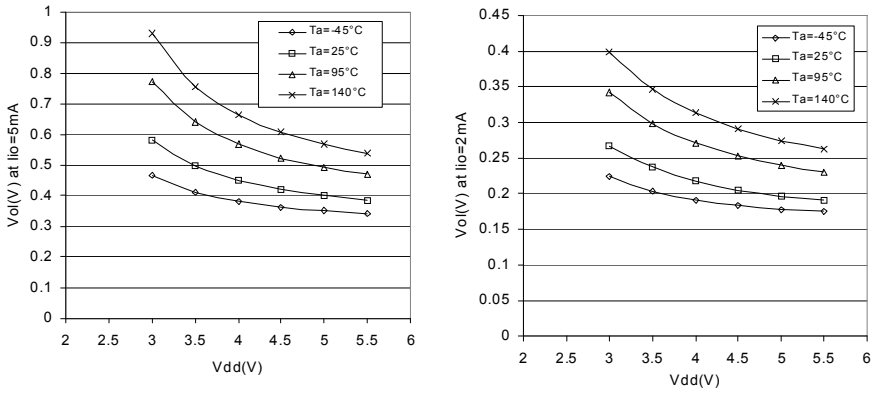


Figure 76. Typical V_{OL} vs. V_{DD} (high-sink ports)

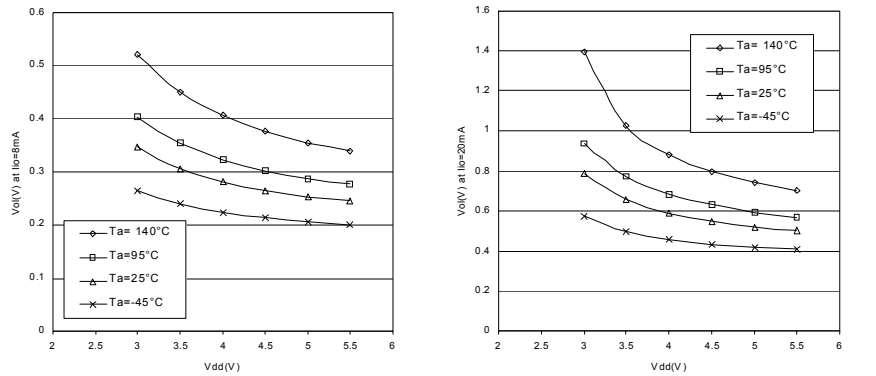
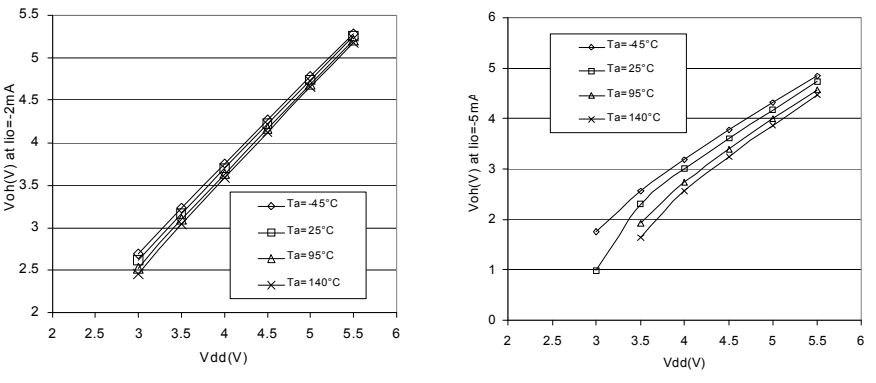


Figure 77. Typical V_{OH} vs. V_{DD}



Downloaded from Arrow.com.

12.10 Control pin characteristics

12.10.1 Asynchronous $\overline{\text{RESET}}$ pin

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

Table 108. Asynchronous $\overline{\text{RESET}}$ pin

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage(1)				$0.3xV_{DD}$	V
V_{IH}	Input high level voltage(1)		$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis(2)			2.5		
V_{OL}	Output low level voltage(3)	V_{DD} = 5 V, I_{IO} = +2 mA		0.2	0.5	V
I_{IO}	Driving current on $\overline{\text{RESET}}$ pin			2		mA
R_{ON}	Weak pull-up equivalent resistor	V_{DD} = 5V	20	30	120	kΩ
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset sources	20	30	42(4)	µs
$t_{h(RSTL)in}$	External reset pulse hold time(5)		2.5			µs
$t_{g(RSTL)in}$	Filtered glitch duration(6)			200		ns

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels.
3. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 12.2.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
4. Data guaranteed by design, not tested in production.
5. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on the $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.
6. The reset network (the resistor and two capacitors) protects the device against parasitic resets, especially in noisy environments.

Downloaded from Arrow.com.

$\overline{\text{RESET}}$ pin protection when LVD is enabled

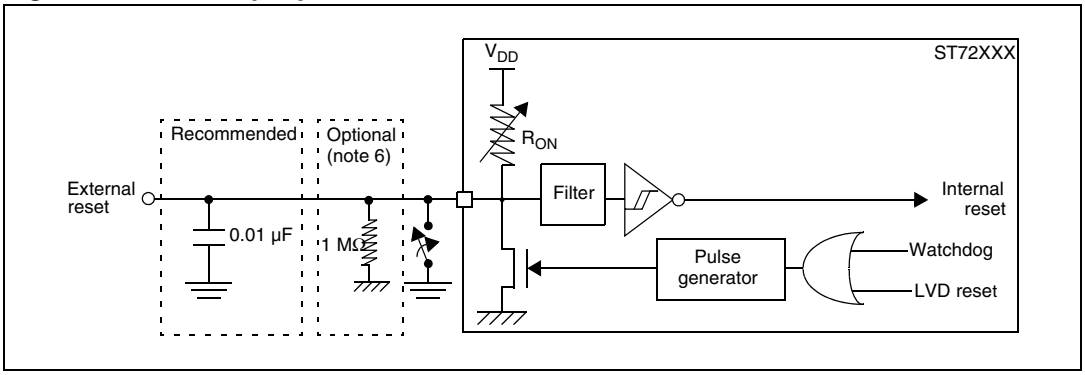
When the LVD is enabled, it is recommended to protect the $\overline{\text{RESET}}$ pin as shown in *Figure 78* and follow these guidelines:

1. The reset network protects the device against parasitic resets.
2. The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
3. Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in *Section 12.10.1*. Otherwise the reset will not be taken into account internally.
4. Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin (by an external pull-up for example) is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 12.2.2 on page 143*.
5. When the LVD is enabled, it is mandatory not to connect a pull-up resistor. A 10nF pull-down capacitor is recommended to filter noise on the reset line.
6. In case a capacitive power supply is used, it is recommended to connect a 1M ohm pull-down resistor to the $\overline{\text{RESET}}$ pin to discharge any residual voltage induced by this capacitive power supply (this will add 5 µA to the power consumption of the MCU).

Tips when using the LVD:

- Check that all recommendations related to reset circuit have been applied (see section above)
- Check that the power supply is properly decoupled (100 nF + 10 µF close to the MCU). Refer to AN1709. If this cannot be done, it is recommended to put a 100 nF + 1 M Ohm pull-down on the $\overline{\text{RESET}}$ pin.
- The capacitors connected on the $\overline{\text{RESET}}$ pin and also the power supply are key to avoiding any start-up marginality. In most cases, steps 1 and 2 above are sufficient for a robust solution. Otherwise: Replace 10 nF pull-down on the $\overline{\text{RESET}}$ pin with a 5 µF to 20 µF capacitor.

Figure 78. $\overline{\text{RESET}}$ pin protection when LVD is enabled



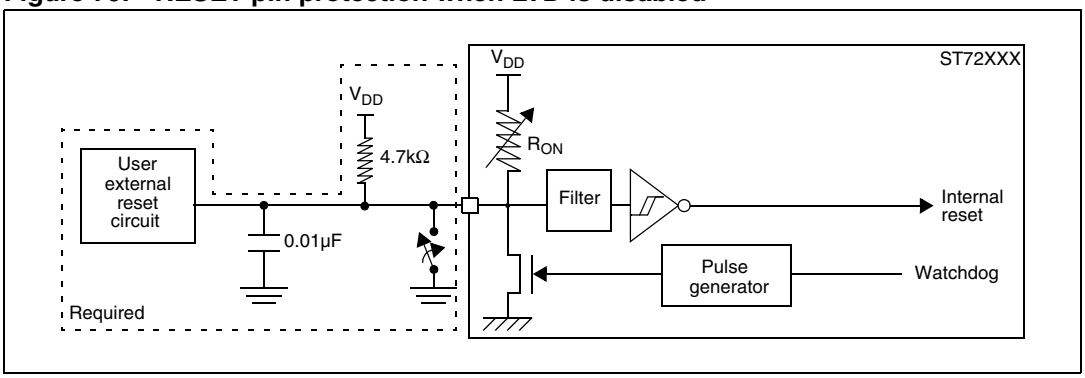
Downloaded from Arrow.com.

$\overline{\text{RESET}}$ pin protection when LVD is disabled

When the LVD is disabled, it is recommended to protect the $\overline{\text{RESET}}$ pin as shown in *Figure 79* and follow these guidelines:

1. The reset network protects the device against parasitic resets.
2. The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
3. Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in *Section 12.10.1*. Otherwise the reset will not be taken into account internally.
4. Because the reset circuit is designed to allow the internal RESET to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin (by an external pull-up for example) is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in *Section 12.2.2 on page 143*.

Figure 79. $\overline{\text{RESET}}$ pin protection when LVD is disabled



Downloaded from Arrow.com.

12.10.2 ICCSEL/V_{PP} pin

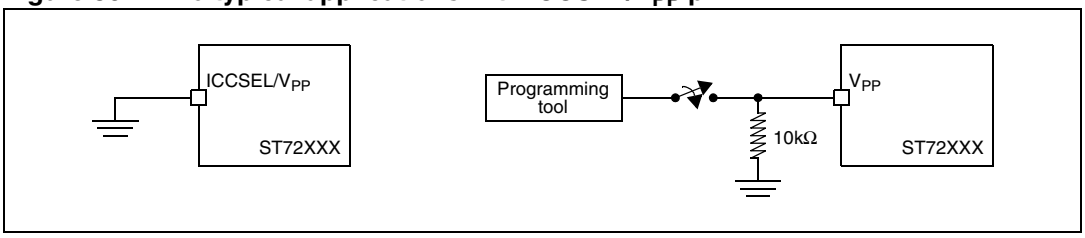
Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified.

Table 109. ICCSEL/V_{PP} pin

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input low level voltage(1)	Flash versions	V_{SS}	0.2	V
		ROM versions	V_{SS}	0.3 x V_{DD}	
V_{IH}	Input high level voltage(1)	Flash versions	V_{DD} - 0.1	12.6	
		ROM versions	0.7 x V_{DD}	V_{DD}	
I_{lkg}	Input leakage current	$V_{IN} = V_{SS}$		±1	µA

1. Data based on design simulation and/or technology characteristics, not tested in production.

Figure 80. Two typical applications with ICCSEL/V_{PP} pin(1)



1. When ICC mode is not required by the application ICCSEL/V_{PP} pin must be tied to V_{SS}.

12.11 Timer peripheral characteristics

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 (output compare, input capture, external clock, PWM output...).

Data based on design simulation and/or characterization results, not tested in production.

12.11.1 16-bit timer

Table 110. 16-bit timer

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{w(ICAP)in}$	Input capture pulse time		1			t_{CPU}
$t_{res(PWM)}$	PWM resolution time		2			t_{CPU}
		f_{CPU} = 8 MHz	250			ns
f_{EXT}	Timer external clock frequency		0		$f_{CPU}/4$	MHz
f_{PWM}	PWM repetition rate					
Res_{PWM}	PWM resolution				16	bit

Downloaded from Arrow.com.

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).

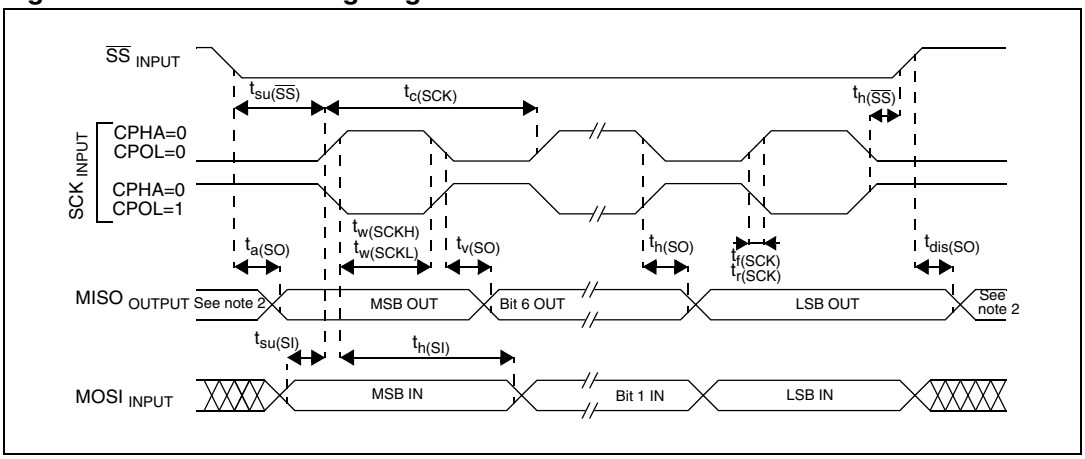
Table 111. SPI characteristics

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 $t_{CPU} = 1 / f_{CPU}$ = 125 ns and $t_{su(SS)}$ = 175 ns.

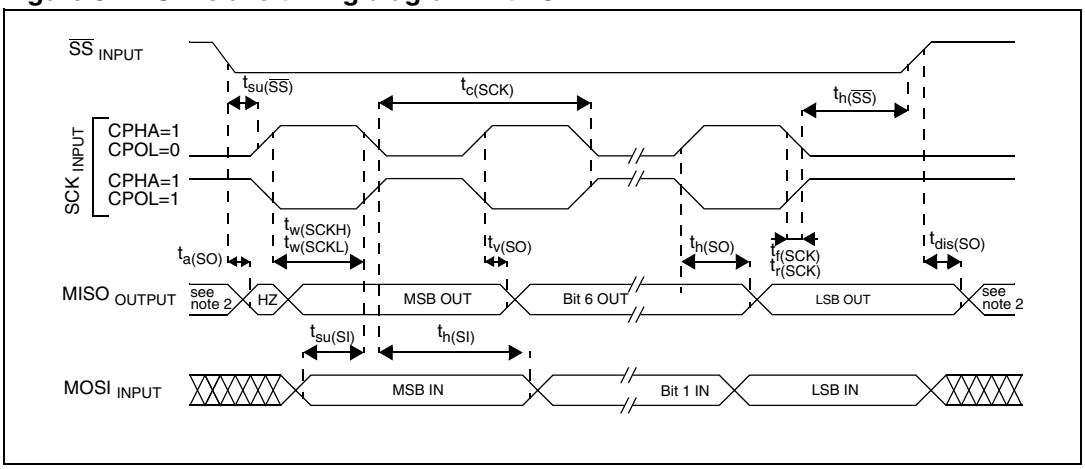
Downloaded from Arrow.com.

Figure 81. SPI slave timing diagram with CPHA = 0(1)



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

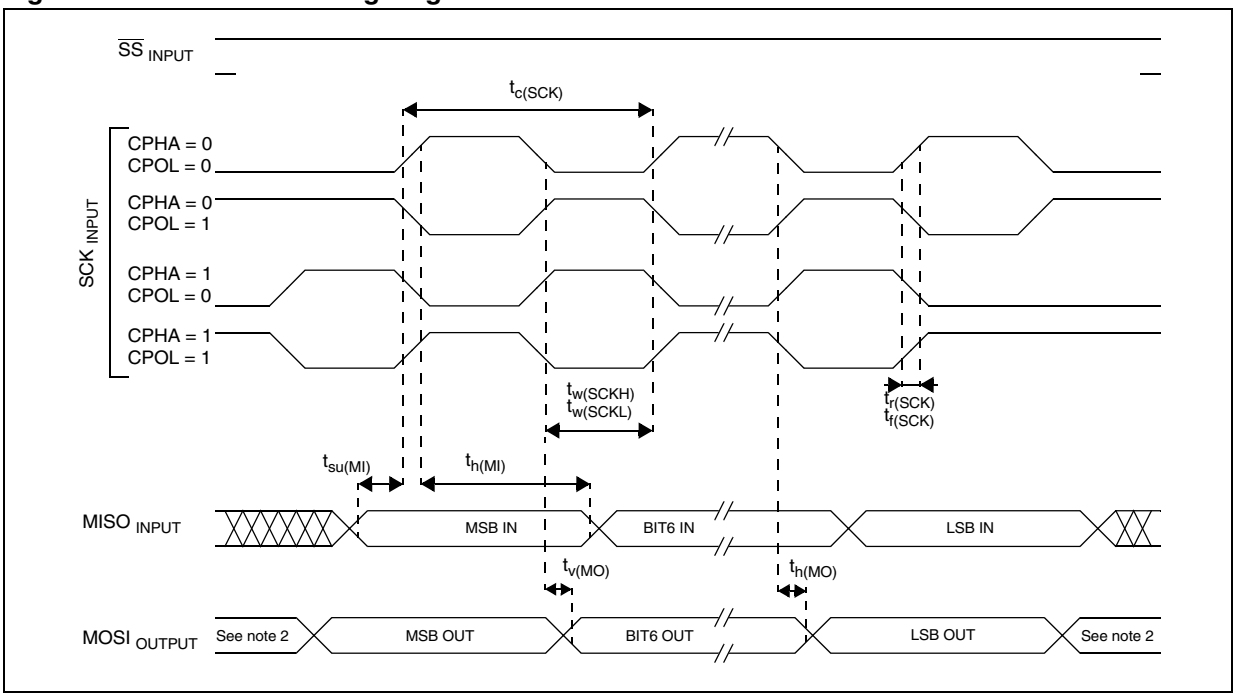
Figure 82. SPI slave timing diagram with CPHA = 1(1)



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

Downloaded from Arrow.com.

Figure 83. SPI master timing diagram(1)



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

12.13 10-bit ADC characteristics

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

Table 112. 10-bit ADC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{ADC}	ADC clock frequency		0.4		2	MHz
V_{AREF}	Analog reference voltage	$0.7*V_{DD} \leq V_{AREF} \leq V_{DD}$	3.8		V_{DD}	V
V_{AIN}	Conversion voltage range(1)		V_{SSA}		V_{AREF}	
I_{lkg}	Input leakage current for analog input(2)	-40 °C ≤ T_A ≤ +85 °C			±250	nA
		Other T_A ranges			±1	µA
R_{AIN}	External input impedance				See figures 84 and 85	kΩ
C_{AIN}	External capacitor on analog input					pF
f_{AIN}	Variation freq. of analog input signal					Hz
C_{ADC}	Internal sample and hold capacitor			12		pF

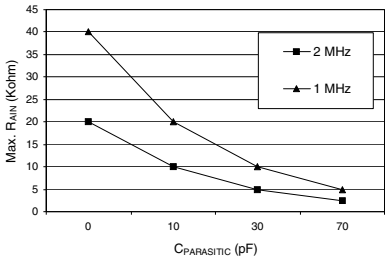
Downloaded from Arrow.com.

Table 112. 10-bit ADC characteristics (continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{ADC}	Conversion time (Sample + Hold) f_{CPU} = 8 MHz, Speed = 0, f_{ADC} = 2 MHz			7.5		µs
	No. of sample capacitor loading cycles No. of Hold conversion cycles			4 11		$1/f_{ADC}$

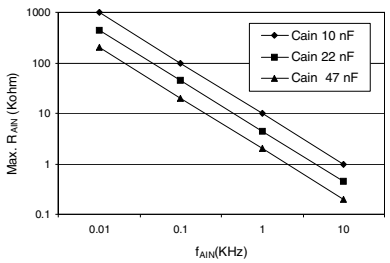
1. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.
2. Injecting negative current on adjacent pins may result in increased leakage currents. Software filtering of the converted analog value is recommended.

Figure 84. R_{AIN} max. vs f_{ADC} with C_{AIN} = 0 pF(1)



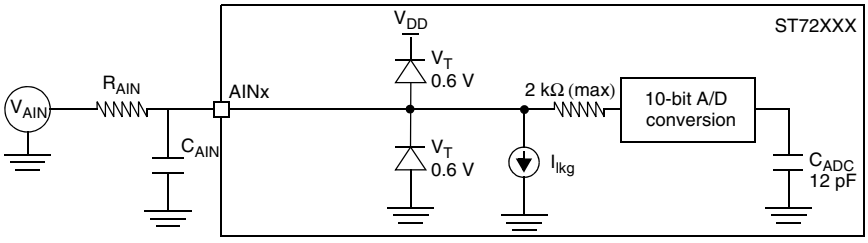
1. $C_{PARASITIC}$ represents the capacitance of the PCB (dependent on soldering and PCB layout quality) plus the pad capacitance (3 pF). A high $C_{PARASITIC}$ value will downgrade conversion accuracy. To remedy this, f_{ADC} should be reduced.

Figure 85. Recommended C_{AIN} and R_{AIN} values(1)



1. This graph shows that, depending on the input signal variation (f_{AIN}), C_{AIN} can be increased for stabilization time and decreased to allow the use of a larger serial resistor (R_{AIN}).

Figure 86. Typical A/D converter application



Downloaded from Arrow.com.

12.13.1 Analog power supply and reference pins

Depending on the MCU pin count, the package may feature separate V_{AREF} and V_{SSA} analog power supply pins. These pins supply power to the A/D converter cell and function as the high and low reference voltages for the conversion. In some packages, V_{AREF} and V_{SSA} pins are not available (refer to *Section 2 on page 15*). In this case the analog supply and reference pads are internally bonded to the V_{DD} and V_{SS} pins.

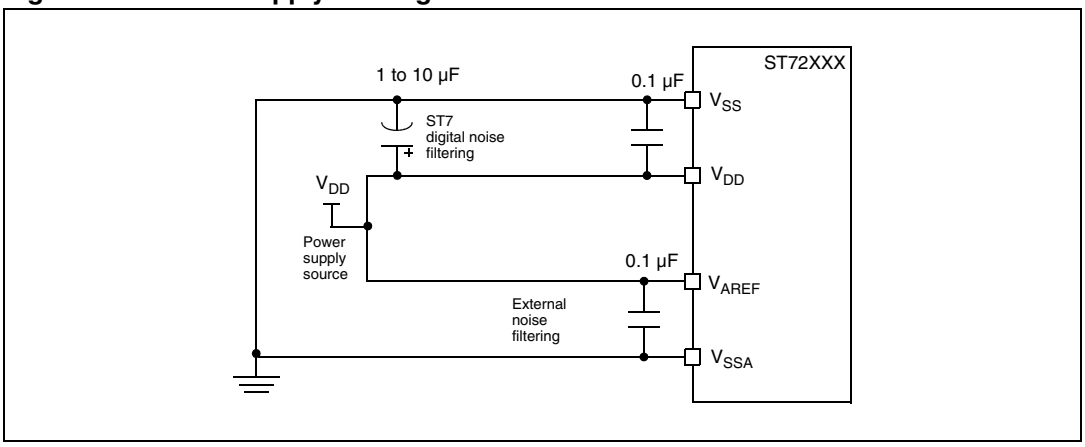
Separation of the digital and analog power pins allow board designers to improve A/D performance. Conversion accuracy can be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines (see *Section 12.13.2: General PCB design guidelines*).

12.13.2 General PCB design guidelines

To obtain best results, some general design and layout rules should be followed when designing the application PCB to shield the noise-sensitive, analog physical interface from noise-generating CMOS logic signals.

- Use separate digital and analog planes. The analog ground plane should be connected to the digital ground plane via a single point on the PCB.
- Filter power to the analog power planes. It is recommended to connect capacitors, with good high frequency characteristics, between the power and ground lines, placing 0.1µF and optionally, if needed 10 pF capacitors as close as possible to the ST7 power supply pins and a 1 to 10 µF capacitor close to the power source (see *Figure 87*).
- The analog and digital power supplies should be connected in a star network. Do not use a resistor, as V_{AREF} is used as a reference voltage by the A/D converter and any resistance would cause a voltage drop and a loss of accuracy.
- Properly place components and route the signal traces on the PCB to shield the analog inputs. Analog signals paths should run over the analog ground plane and be as short as possible. Isolate analog signals from digital signals that may switch while the analog inputs are being sampled by the A/D converter. Do not toggle digital outputs on the same I/O port as the A/D input being converted.

Figure 87. Power supply filtering



Downloaded from Arrow.com.

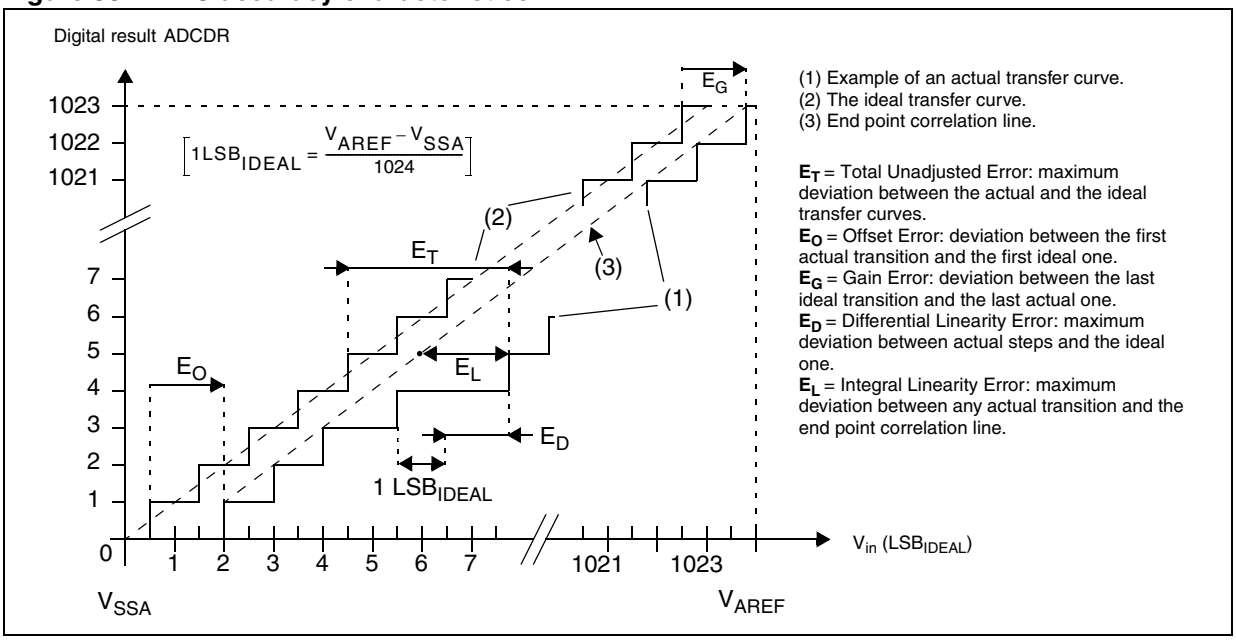
12.13.3 ADC accuracy

Table 113. ADC accuracy

Symbol	Parameter	Conditions		Typ	Max(1)		Unit
					ROM and 8/16 Kbyte Flash	32 Kbyte Flash	
$\|E_T\|$	Total unadjusted error(2)	V_{DD} = 5 V(2)	CPU in run mode @ f_{ADC} 2 MHz	3	4	6	LSB
$\|E_O\|$	Offset error(2)			2	3	5	
$\|E_G\|$	Gain error(2)			0.5	3	4.5	
$\|E_D\|$	Differential linearity error(2)			1	2	2	
$\|E_L\|$	Integral linearity error(2)					3	

1. Data based on characterization results, monitored in production to guarantee 99.73% within ± max value from -40°C to 125 °C (± 3σ distribution limits).
2. ADC accuracy vs. negative injection current: Injecting negative current may reduce the accuracy of the conversion being performed on another analog input. Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section 12.9* does not affect the ADC accuracy.

Figure 88. ADC accuracy characteristics



Downloaded from Arrow.com.

13 Package characteristics

13.1 ECOPACK

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com. ECOPACK® is an ST trademark.

13.2 Package mechanical data

13.2.1 LQFP44 package mechanical data

Figure 89. 44-pin low profile quad flat package outline

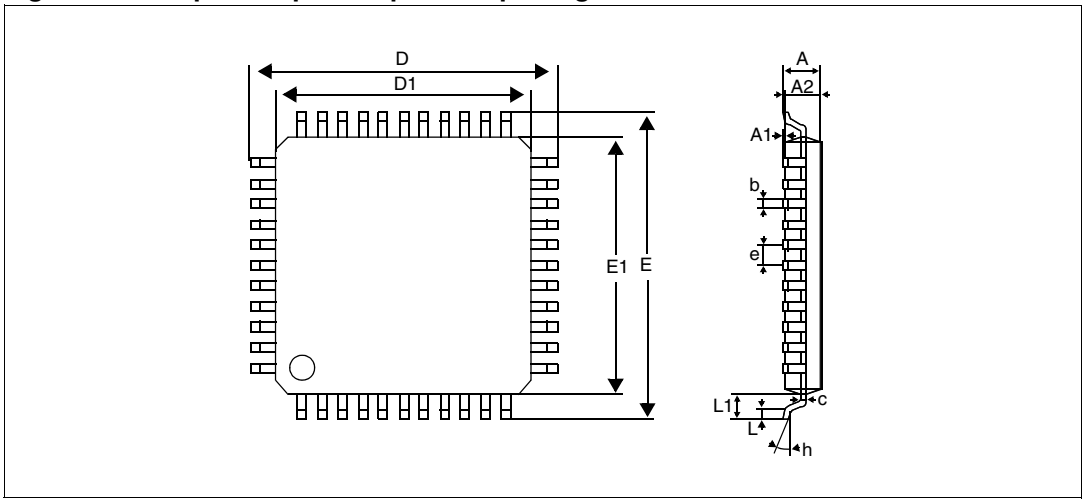


Table 114. 44-pin low profile quad flat package mechanical data

Dim.	mm			inches (1)		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.0630
A1	0.05		0.15	0.0020		0.0059
A2	1.35	1.40	1.45	0.0531	0.0551	0.0571
b	0.30	0.37	0.45	0.0118	0.0146	0.0177
C	0.09		0.20	0.0035		0.0079
D		12.00			0.4724	
D1		10.00			0.3937	
E		12.00			0.4724	

Downloaded from Arrow.com.

Table 114. 44-pin low profile quad flat package mechanical data (continued)

Dim.	mm			inches (1)		
	Min	Typ	Max	Min	Typ	Max
E1		10.00			0.3937	
e		0.80			0.0315	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.45	0.60	0.75	0.0177	0.0236	0.0295
L1		1.00			0.0394	
Number of pins						
N	44					

1. Values in inches are converted from mm and rounded to 4 decimal digits.

13.2.2 SDIP42 package mechanical data

Figure 90. 42-pin plastic dual in-line package, shrink 600-mil width

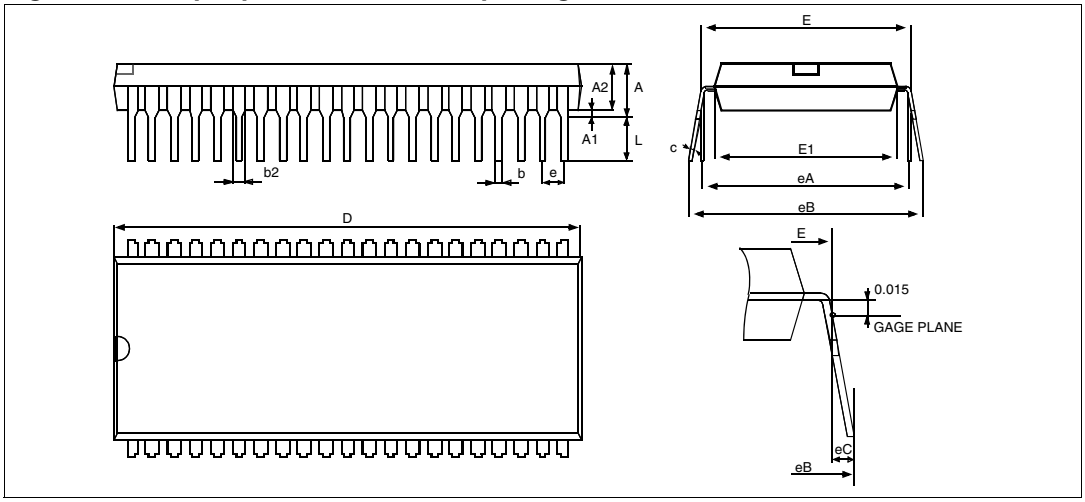


Table 115. 42-pin dual in line package mechanical data

Dim.	mm			inches (1)		
	Min	Typ	Max	Min	Typ	Max
A			5.08			0.2000
A1	0.51			0.0201		
A2	3.05	3.81	4.57	0.1201	0.1500	0.1799
b	0.38	0.46	0.56	0.0150	0.0181	0.0220
b2	0.89	1.02	1.14	0.0350	0.0402	0.0449
c	0.23	0.25	0.38	0.0091	0.0098	0.0150
D	36.58	36.83	37.08	1.4402	1.4500	1.4598

Downloaded from Arrow.com.

Table 115. 42-pin dual in line package mechanical data

Dim.	mm			inches (1)		
	Min	Typ	Max	Min	Typ	Max
E	15.24		16.00	0.6000		0.6299
E1	12.70	13.72	14.48	0.5000	0.5402	0.5701
e		1.78			0.0701	
eA		15.24			0.6000	
eB			18.54			0.7299
eC			1.52			0.0598
L	2.54	3.30	3.56	0.1000	0.1299	0.1402
Number of pins						
N	42					

1. Values in inches are converted from mm and rounded to 4 decimal digits.

13.2.3 LQFP32 package mechanical data

Figure 91. 32-pin low profile quad flat package outline

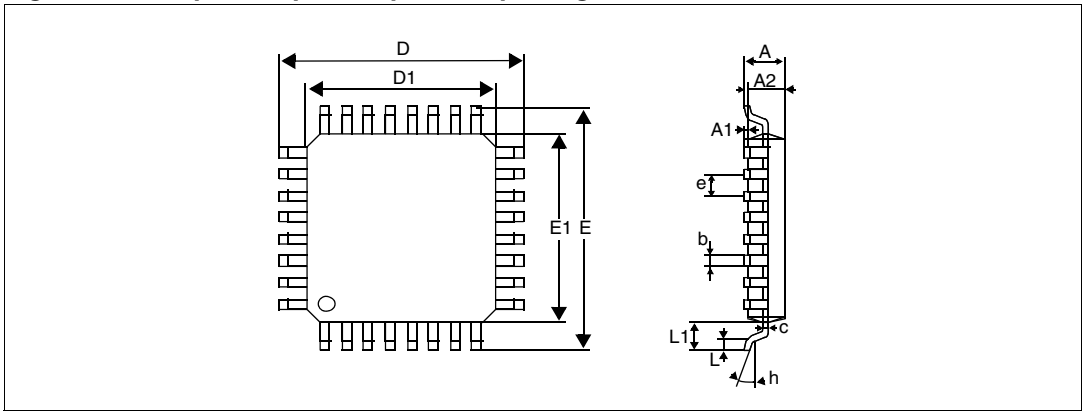


Table 116. 32-pin low profile quad flat package mechanical data

Dim.	mm			inches (1)		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.0630
A1	0.05		0.15	0.0020		0.0059
A2	1.35	1.40	1.45	0.0531	0.0551	0.0571
b	0.30	0.37	0.45	0.0118	0.0146	0.0177
C	0.09		0.20	0.0035		0.0079
D		9.00			0.3543	
D1		7.00			0.2756	

Downloaded from Arrow.com.

Table 116. 32-pin low profile quad flat package mechanical data (continued)

Dim.	mm			inches (1)		
	Min	Typ	Max	Min	Typ	Max
E		9.00			0.3543	
E1		7.00			0.2756	
e		0.80			0.0315	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.45	0.60	0.75	0.0177	0.0236	0.0295
L1		1.00			0.0394	
Number of pins						
N	32					

1. Values in inches are converted from mm and rounded to 4 decimal digits.

13.2.4 SDIP32 package mechanical data

Figure 92. 32-pin plastic dual in-line package, shrink 400-mil width

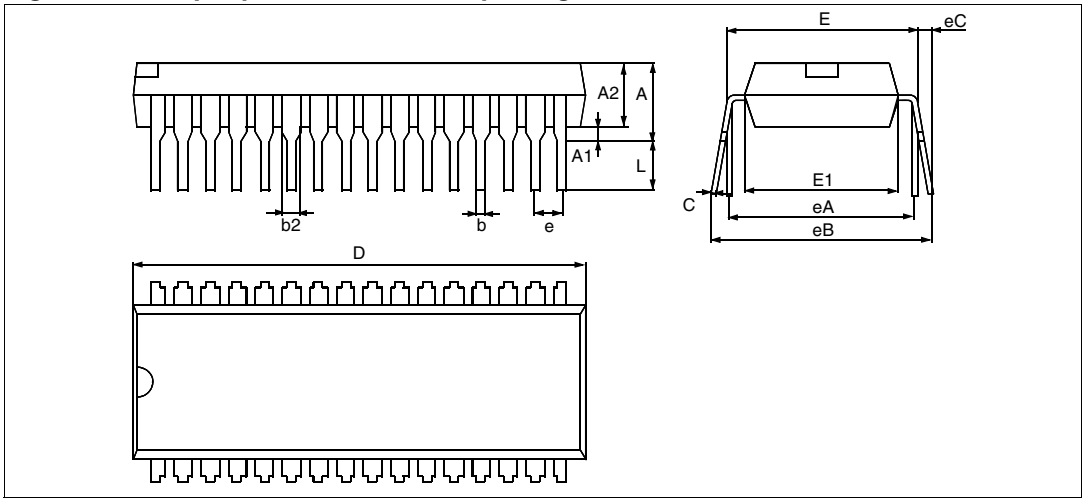


Table 117. 32-pin dual in-line package mechanical data

Dim.	mm			inches(1)		
	Min	Typ	Max	Min	Typ	Max
A	3.56	3.76	5.08	0.1402	0.1480	0.2000
A1	0.51			0.0201		
A2	3.05	3.56	4.57	0.1201	0.1402	0.1799
b	0.36	0.46	0.58	0.0142	0.0181	0.0228
b1	0.76	1.02	1.40	0.0299	0.0402	0.0551
C	0.20	0.25	0.36	0.0079	0.0098	0.0142
D	27.43		28.45	1.0799		1.1201

Downloaded from Arrow.com.

Table 117. 32-pin dual in-line package mechanical data (continued)

Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
E	9.91	10.41	11.05	0.3902	0.4098	0.4350
E1	7.62	8.89	9.40	0.3000	0.3500	0.3701
e		1.78			0.0701	
eA		10.16			0.4000	
eB			12.70			0.5000
eC			1.40			0.0551
L	2.54	3.05	3.81	0.1000	0.1201	0.1500
Number of pins						
N	42					

1. Values in inches are converted from mm and rounded to 4 decimal digits.

Downloaded from Arrow.com.

13.3 Thermal characteristics

Table 118. Thermal characteristics

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient): LQFP44 10x10 LQFP32 7x7 DIP42 600mil SDIP32 200mil	 52 70 55 50	°C/W
P_D	Power dissipation(1)	500	mW
T_{Jmax}	Maximum junction temperature(2)	150	°C

1. The maximum power dissipation is obtained from the formula $P_D = (T_J - T_A) / R_{thJA}$. The power dissipation of an application can be defined by the user with the formula: $P_D = P_{INT} + P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD} \times V_{DD}$) and P_{PORT} is the port power dissipation depending on the ports used in the application.
2. The maximum chip-junction temperature is based on technology characteristics.

Downloaded from Arrow.com.

14 Device configuration and ordering information

Each device is available for production in user programmable versions (Flash) as well as in factory coded versions (ROM/FASTROM).

ST72324Bxx devices are ROM versions. ST72P324B devices are Factory Advanced Service Technique ROM (FASTROM) versions: They are factory-programmed HDFlash devices. Flash devices are shipped to customers with a default content (FFh), while ROM factory coded parts contain the code supplied by the customer. This implies that Flash devices have to be configured by the customer using the Option bytes while the ROM devices are factory-configured.

Figure 93. ST72324Bxx ordering information scheme

Example: ST72 F 324B K 2 B 5

Family

ST7 microcontroller family

Version

F = Flash
P = FASTROM
Blank = ROM

Sub-family

324B

No. of pins

K = 32
J = 42 or 44

Memory size

2 = 8 Kbytes
4 = 16 Kbytes
6 = 32 Kbytes

Package

B = DIP [1)]
M = SO
U = DFN

Temperature range

1 = 0 to +70 °C
5 = -10 to +85 °C
6 = -40 to +85 °C
7 = -40 to +105 °C
3 = -40 to +125 °C

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 contact the ST sales office nearest to you.

Downloaded from Arrow.com.

14.1 Flash devices

14.1.1 Flash configuration

Table 119. Flash option bytes

	Static option byte 0								Static option byte 1							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	WDG		Res	VD		Reserved		FMP_R	PKG1	RSTC	OSCTYPE		OSCRANGE			PLLOFF
	HALT	SW		1	0						1	0	2	1	0	
Default	1	1	1	0	0	1	1	1	See note 1	1	1	0	0	1	1	1

1. Depends on device type as defined in *Table 122: Package selection (OPT7) on page 181.*

The option bytes allow the hardware configuration of the microcontroller to be selected. They have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the Flash is fixed to FFh. To program directly the Flash devices using ICP, Flash devices are shipped to customers with the internal RC clock source. In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see option list).

Table 120. Option byte 0 bit description

Bit	Name	Function
OPT7	WDG HALT	Watchdog reset on Halt This option bit determines if a reset is generated when entering Halt mode while the Watchdog is active. 0: No reset generation when entering Halt mode 1: Reset generation when entering Halt mode
OPT6	WDG SW	Hardware or software Watchdog This option bit selects the Watchdog type. 0: Hardware (Watchdog always enabled) 1: Software (Watchdog to be enabled by software)
OPT5	-	Reserved, must be kept at default value.
OPT4:3	VD[1:0]	Voltage detection These option bits enable the voltage detection block (LVD and AVD) with a selected threshold for the LVD and AVD. 00: Selected LVD = Highest threshold (V_{DD}~4V). 01: Selected LVD = Medium threshold (V_{DD}~3.5V). 10: Selected LVD = Lowest threshold (V_{DD}~3V). 11: LVD and AVD off **Caution:** 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. For details on the AVD and LVD threshold levels refer to *Section 12.4.1 on page 145*.
OPT2:1	-	Reserved, must be kept at default value

Downloaded from Arrow.com.

Table 120. Option byte 0 bit description (continued)

Bit	Name	Function
OPT0	FMP_R	Flash memory readout protection Readout protection, when selected, provides a protection against program memory content extraction and against write access to Flash memory. Erasing the option bytes when the FMP_R option is selected causes the whole user memory to be erased first, afterwhich the device can be reprogrammed. Refer to *Section 4.3.1 on page 24* and the *ST7 Flash Programming Reference Manual* for more details. 0: Readout protection enabled 1: Readout protection disabled

Table 121. Option byte 1 bit description

Bit	Name	Function
OPT7	PKG1	Pin package selection bit This option bit selects the package (see *Table 122*). *Note: On the chip, each I/O port has eight pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.*
OPT6	RSTC	Reset clock cycle selection This option bit selects the number of CPU cycles applied during the reset phase and when exiting Halt mode. For resonator oscillators, it is advised to select 4096 due to the long crystal stabilization time. 0: Reset phase with 4096 CPU cycles 1: Reset phase with 256 CPU cycles
OPT5:4	OSCTYPE[1:0]	Oscillator type These option bits select the ST7 main clock source type. 00: Clock source = Resonator oscillator 01: Reserved 10: Clock source = Internal RC oscillator 11: Clock source = External source
OPT3:1	OSCRANGE[2:0]	Oscillator range When the resonator oscillator type is selected, these option bits select the resonator oscillator current source corresponding to the frequency range of the used resonator. When the external clock source is selected, these bits are set to medium power (2 ~ 4 MHz). 000: Typ. frequency range (LP) = 1 ~ 2 MHz 001: Typ. frequency range (MP) = 2 ~ 4 MHz 010: Typ. frequency range (MS) = 4 ~ 8 MHz 011: Typ. frequency range (HS) = 8 ~ 16 MHz

Downloaded from Arrow.com.

Table 121. Option byte 1 bit description (continued)

Bit	Name	Function
OPT0	PLL OFF	PLL activation This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL must not be used with the internal RC oscillator. The PLL is guaranteed only with an input frequency between 2 and 4 MHz. 0: PLL x2 enabled 1: PLL x2 disabled **Caution**: The PLL can be enabled only if the “OSCRANGE” (OPT3:1) bits are configured to “MP - 2~4 MHz”. Otherwise, the device functionality is not guaranteed.

Table 122. Package selection (OPT7)

Version	Selected package	PKG1
J	LQFP44/SDIP42	1
K	LQFP32/SDIP32	0

14.2 ROM devices

14.2.1 Transfer of customer code

Customer code is made up of the ROM/FASTROM contents and the list of the selected options (if any). The ROM/FASTROM contents are to be sent with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh. Complete the appended *ST72324Bxx MICROCONTROLLER OPTION LIST on page 182* to communicate the selected options to STMicroelectronics.

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

Figure 93: ST72324Bxx ordering information scheme on page 178 serves as a guide for ordering. The STMicroelectronics sales organization will be pleased to provide detailed information on contractual points.

Caution: The readout protection binary value is inverted between ROM and Flash products. The option byte checksum differs between ROM and Flash.

Downloaded from Arrow.com.

ST72324Bxx MICROCONTROLLER OPTION LIST
(Last update: March 2009)

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

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

ROM DEVICE:	32K	16K	8K
LQFP32:	[ ]	[ ]	[ ]
DIP32:	[ ]	[ ]	[ ]
LQFP44 :	[ ]	[ ]	[ ]
DIP42:	[ ]	[ ]	[ ]

DIE FORM:	32K	16K	8K
32-pin:	[ ]	[ ]	[ ]
44-pin:	[ ]	[ ]	[ ]

Conditioning (check only one option):

Packaged product	Die product (dice tested at 25 °C only)
LQFP: [ ] Tape & reel [ ] Tray	[ ] Tape & Reel
DIP: [ ] Tube	[ ] Inked wafer
	[ ] Sawn wafer on sticky foil

Power supply range: [ ] 3.8 to 5.5 V
Temp. range (do not check for die product).

[ ] 0 °C to +70 °C
[ ] -10 °C to +85 °C
[ ] -40 °C to +85 °C
[ ] -40 °C to +105 °C
[ ] -40 °C to +125 °C

Special marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ " (LQFP32 7 char., other pkg. 10 char. max)
Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Clock source selection:
[ ] Resonator: [ ] LP: Low power resonator (1 to 2 MHz)
[ ] MP: Medium power resonator (2 to 4 MHz)
[ ] MS: Medium speed resonator (4 to 8 MHz)
[ ] HS: High speed resonator (8 to 16 MHz)
[ ] Internal RC
[ ] External clock

PLL [ ] Disabled [ ] Enabled
LVD Reset [ ] Disabled [ ] High threshold [ ] Med. threshold [ ] Low threshold
Reset Delay [ ] 256 Cycles [ ] 4096 Cycles

Watchdog selection: [ ] Software activation [ ] Hardware activation
Watchdog Reset on Halt: [ ] Reset [ ] No Reset

Readout protection: [ ] Disabled [ ] Enabled

Date . . . . . .
Signature . . . . . .

Caution: The readout protection binary value is inverted between ROM and Flash products. The option byte checksum will differ between ROM and Flash.

Downloaded from Arrow.com.

14.3 Development tools

14.3.1 Introduction

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

14.3.2 Evaluation tools and starter kits

ST offers complete, affordable **starter kits** and full-featured **evaluation boards** that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete, affordable hardware/software tool packages that include features and samples to help you quickly start developing your application. ST evaluation boards are open-design, embedded systems, which are developed and documented to serve as references for your application design. They include sample application software to help you demonstrate, learn about and implement your ST7's features.

14.3.3 Development and debugging tools

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

The range of hardware tools includes cost effective ST7-DVP3 series emulators. These tools are supported by the ST7 Toolset from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

14.3.4 Programming tools

During the development cycle, the ST7-DVP3 and ST7-EMU3 series emulators and the RLink provide in-circuit programming capability for programming the Flash microcontroller on your application board.

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

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

For additional ordering codes for spare parts, accessories and tools available for the ST7 (including from third party manufacturers), refer to the online product selector at www.st.com/mcu.

Downloaded from Arrow.com.

Table 123. STMicroelectronics development tools

Supported products	Emulation				Programming
	ST7 DVP3 series		ST7 EMU3 series		ICC socket board
	Emulator	Connection kit	Emulator	Active probe and TEB	
ST72324BJ, ST72F324BJ	ST7MDT20-DVP3	ST7MDT20-T44/DVP	ST7MDT20J-EMU3	ST7MDT20J-TEB	ST7SB20J/xx(1)
ST72324BK, ST72F324BK		ST7MDT20-T32/DVP			

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

14.3.5 Socket and emulator adapter information

For information on the type of socket that is supplied with the emulator, refer to the suggested list of sockets in *Table 124*.

Note: Before designing the board layout, it is recommended to check the overall dimensions of the socket as they may be greater than the dimensions of the device.

For footprint and other mechanical information about these sockets and adapters, refer to the manufacturer's datasheet (www.yamaichi.de for LQFP44 10x10 and www.ironwoodelectronics.com for LQFP32 7x7).

Table 124. Suggested list of socket types

Device	Socket (supplied with ST7MDT20J-EMU3)	Emulator adapter (supplied with ST7MDT20J-EMU3)
LQFP32 7X7	IRONWOOD SF-QFE32SA-L-01	IRONWOOD SK-UGA06/32A-01
LQFP44 10X10	YAMAICHI IC149-044-*52-*5	YAMAICHI ICP-044-5

14.4 ST7 Application notes

All relevant ST7 application notes can be found on www.st.com.

Downloaded from Arrow.com.

15 Known limitations

15.1 All Flash and ROM devices

15.1.1 Safe connection of OSC1/OSC2 pins

The OSC1 and/or OSC2 pins must not be left unconnected, otherwise the ST7 main oscillator may start and, in this configuration, could generate an f_{OSC} clock frequency in excess of the allowed maximum (> 16 MHz), putting the ST7 in an unsafe/undefined state. Refer to *Section 6.3 on page 32*.

15.1.2 External interrupt missed

To avoid any risk of generating a parasitic interrupt, the edge detector is automatically disabled for one clock cycle during an access to either DDR and OR. Any input signal edge during this period will not be detected and will not generate an interrupt.

This case can typically occur if the application refreshes the port configuration registers at intervals during runtime.

Workaround

The workaround is based on software checking the level on the interrupt pin before and after writing to the PxOR or PxDDR registers. If there is a level change (depending on the sensitivity programmed for this pin) the interrupt routine is invoked using the call instruction with three extra PUSH instructions before executing the interrupt routine (this is to make the call compatible with the IRET instruction at the end of the interrupt service routine).

But detection of the level change does not make sure that edge occurs during the critical one cycle duration and the interrupt has been missed. This may lead to occurrence of same interrupt twice (one hardware and another with software call).

To avoid this, a semaphore is set to ‘1’ before checking the level change. The semaphore is changed to level '0' inside the interrupt routine. When a level change is detected, the semaphore status is checked and if it is ‘1’ this means that the last interrupt has been missed. In this case, the interrupt routine is invoked with the call instruction.

There is another possible case that is, if writing to PxOR or PxDDR is done with global interrupts disabled (interrupt mask bit set). In this case, the semaphore is changed to ‘1’ when the level change is detected. Detecting a missed interrupt is done after the global interrupts are enabled (interrupt mask bit reset) and by checking the status of the semaphore. If it is ‘1’ this means that the last interrupt was missed and the interrupt routine is invoked with the call instruction.

To implement the workaround, the following software sequence is to be followed for writing into the PxOR/PxDDR registers. The example is for Port PF1 with falling edge interrupt sensitivity. The software sequence is given for both cases (global interrupt disabled/enabled).

Downloaded from Arrow.com.

Case 1: Writing to PxOR or PxDDR with global interrupts enabled:

```
LD A,#01
LD sema,A; set the semaphore to '1'
LD A,PFDR
AND A,#02
LD  X,A; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A ; Write to PFDDR
LD A,#$ff
LD  PFOR,A ; Write to PFOR
LD A,PFDR
AND A,#02
LD Y,A; store the level after writing to PxOR/PxDDR
LD A,X; check for falling edge
cp A,#02
jrne OUT
TNZ Y
jrne OUT
LD A,sema ; check the semaphore status if edge is detected
CP A,#01
jrne OUT
call call_routine ; call the interrupt routine
OUT:LD A,#00
LD sema,A
.call_routine ; entry to call_routine
PUSH A
PUSH X
PUSH CC
.ext1_rt ; entry to interrupt routine
LD A,#00
LD sema,A
IRET
```

Case 2: Writing to PxOR or PxDDR with global interrupts disabled:

```
SIM ; set the interrupt mask
LD A,PFDR
AND A,#$02
LD  X,A ; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A ; Write into PFDDR
LD A,#$ff
LD PFOR,A ; Write to PFOR
LD A,PFDR
AND A,#$02
LD Y,A ; store the level after writing to PxOR/PxDDR
LD A,X ; check for falling edge
cp A,#$02
jrne OUT
TNZ Y
jrne OUT
LD A,#$01
LD sema,A ; set the semaphore to '1' if edge is detected
```


Downloaded from Arrow.com.

```
RIM ; reset the interrupt mask
LD A,sema ; check the semaphore status
CP A,#$01
jrne OUT
call call_routine ; call the interrupt routine
RIM
OUT:RIM
JP while_loop
.call_routine ; entry to call_routine
PUSH A
PUSH X
PUSH CC
.ext1_rt ; entry to interrupt routine
LD A,#$00
LD sema,A
IRET
```

15.1.3 Unexpected reset fetch

If an interrupt request occurs while a “POP CC” instruction is executed, the interrupt controller does not recognize the source of the interrupt and, by default, passes the reset vector address to the CPU.

Workaround

To solve this issue, a “POP CC” instruction must always be preceded by a “SIM” instruction.

15.1.4 Clearing active interrupts outside interrupt routine

When an active interrupt request occurs at the same time as the related flag is being cleared, an unwanted reset may occur.

Note: Clearing the related interrupt mask will not generate an unwanted reset.

Concurrent 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
- 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:

Perform SIM and RIM operation before and after resetting an active interrupt request. Example:

– SIM
– Reset interrupt flag
– RIM

Downloaded from Arrow.com.

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

Downloaded from Arrow.com.

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 125* 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 125*.

Downloaded from Arrow.com.

Table 125. Port A and F configuration

PLL	PA3	PF4	PF1	PF0	Clock disturbance
Off	0	1	0	Toggling	Maximum 2 clock cycles lost at each rising or falling edge of PF0
On	0	1	0	1	Maximum 1 clock cycle lost out of every 16

As a consequence, for cycle-accurate operations, these configurations are prohibited in either input or output mode.

Workaround

To avoid this from occurring, it is recommended to connect one of these pins to GND (PF4 or PF0) or V_{DD} (PA3 or PF1).

Downloaded from Arrow.com.

16 Revision history

Table 126. Document revision history

Date	Revision	Changes
05-May-2004	2.0	Merged ST72F324 Flash with ST72324B ROM datasheet. Vt POR max modified in *Section 12.4 on page 145* Added *Figure 79 on page 164* Modified V_{AREF} min in *"10-bit ADC characteristics" on page 168* Modified I INJ for PB0 in *Section 12.9 on page 158* Added *"Clearing active interrupts outside interrupt routine" on page 187* Modified "32K ROM DEVICES ONLY" on page 165
30-Mar-2005	3	Removed Clock Security System (CSS) throughout document Added notes on ST72F324B 8K/16K Flash devices in Table 27 Corrected MCO description in *Section 10.2 on page 69* Modified VtPOR in *Section 12.4 on page 145* Static current consumption modified in *Section 12.9 on page 158* Updated footnote and *Figure 78 on page 163* and *Figure 79 on page 164* Modified Soldering information in *Section 13.6* Updated *Section 14 on page 178* Added Table 27 Modified *Figure 8 on page 25* and note 4 in *"Flash program memory" on page 23* Added limitation on ICC entry mode with 39 pulses to *"Known limitations" on page 185* Added Section 16 on page 166 for ST72F324B 8K/16K Flash devices Modified "Internal Sales Types on box label" in Table 29 on page 157
12-Sep-2005	4	Removed notes related to ST72F324, refer to datasheet rev 3 for specifications on older devices. Note: This datasheet rev refers only to ST72F324B and ST72324B. Changed character transmission procedure in *Section on page 112* Updated Vt POR max in *Section 12.4 on page 145* Updated Current Consumption for in *Section 12.5 on page 146* Added oscillator diagram and table to *Section 12.6.3 on page 150* Increased Data retention max. parameter in *Section 12.7.2 on page 154* Updated ordering Section 14.3 on page 155 and Section 14.5 on page 157 Updated Development tools *Section 14.3 on page 183* Added "external interrupt missed" in *Section 15.1 on page 185*
06-Feb-2006	5	Added description of SICSR register at address 2Bh in *Table 3 on page 20* Changed description on port PF2 to add internal pull-up in *Section 9.5.1 on page 63* Highlighted note in SPI *"Master mode operation" on page 99* Changed *"Static latch-up" on page 157* Added note 5 on analog input static current consumption *"General characteristics" on page 158* Updated notes in *"Thermal characteristics" on page 177*

Downloaded from Arrow.com.

Table 126. Document revision history (continued)

Date	Revision	Changes
10-Oct-2007	6	Removed references to automotive versions (these are covered by separate ST72324B-Auto datasheet). Changed Flash endurance to 1 Kcycles at 55°C Replaced TQFP with LQFP in package outline and device summary on page *1* *Figure 1 on page 14*: Replaced 60 Kbytes with 32 Kbytes in program memory block Replaced TQFP with LQFP in *Figure 2 on page 15*, in *Figure 4 on page 16* and in *Table 2 on page 17* Changed note 3 in *Section 9.2.1 on page 58* Changed *Section 10.1.3 on page 65* Changed *Master mode operation on page 99* Added unit of measure to LVD supply current in *Section 12.5.3 on page 148* Replaced TQFP with LQFP in *Section 12.8.2 on page 156* Changed note 4 in *Section 12.9.1 on page 158* Changed *Figure 78 on page 163* Removed EMC protective circuitry in *Figure 79 on page 164* (device works correctly without these components) Changed titles of *Figure 89 on page 172* and *Figure 91 on page 174* Replaced TQFP with LQFP in *Section 13.3 on page 177* Changed *Section 13.6 on page 171* Replaced TQFP with LQFP in *Section 14.1 on page 179*, in *Table 122 on page 181*, in *Section Table 122. on page 182* and in *Section 14.3.5 on page 184*
17-Mar-2009	7	Removed soldering information section. In *Section 10.6.3: Functional description on page 129*, modified “Starting the conversion” paragraph: added “ or a write to any bit of the ADCCSR register”. Modified t_{RET} values in *Table 101: Dual voltage HDFlash memory on page 154* . *Section 13.2: Package mechanical data on page 172* modified (values in inches rounded to 4 decimal digits). Modified *Section 12.8.3: Absolute maximum ratings (electrical sensitivity) on page 157* (removed DLU and $V_{ESD\ (MM)}$). Added *Section 13.1: ECOPACK on page 172*. Modified *“Device configuration and ordering information” on page 178*.

Downloaded from Arrow.com.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2009 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

Downloaded from Arrow.com.