

zilog[®]

Embedded in Life

An  IXYS Company

High-Performance 8-Bit Microcontrollers

**Z8 Encore! XP[®] F0823
Series**

Product Specification

PS024315-1011



Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2011 Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore! and Z8 Encore! XP are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

| Date | Revision Level | Chapter/Section | Description | Page No. |
|----------|----------------|---|---|--|
| Sep | 15 | LED Drive Enable Register | Clarified statement surrounding the Alternate Function Register as it relates to the LED function; revised Flash Sector Protect Register description; revised Packaging chapter. | 51 , 144 , 210 |
| Mar 2008 | 14 | n/a | Changed branding to <i>Z8 Encore! XP F0823 Series</i> where appropriate. | All |
| Dec 2007 | 13 | Pin Description, General-Purpose Input/Output, Interrupt Controller, Watchdog Timer, Electrical Characteristics, and Ordering Information | Updated title from <i>Z8 Encore! 8K and 4K Series</i> to <i>Z8 Encore! XP Z8F0823 Series</i> . Updated Figure 3, Table 15, Table 35, Tables 59 through 61, Table 119 and Part Number Suffix Designations section. | 8 , 36 , 60 , 95 , 199 , and 220 |
| Aug 2007 | 12 | Part Selection Guide, External Clock Setup, and Program Memory | Updated Table 1, Table 16, and Program Memory section. | 2 , 35 , and 13 |
| Jun 2007 | 11 | n/a | Updated to combine Z8 Encore! 8K and Z8 Encore! 4K Series. | All |
| Dec 2006 | 10 | Ordering Information | Updated Ordering Information chapter. | 211 |

Table of Contents

| | |
|---|------|
| Revision History | .iii |
| List of Figures | x |
| List of Tables | xii |
| Overview | 1 |
| Features | 1 |
| Part Selection Guide | 2 |
| Block Diagram | 3 |
| CPU and Peripheral Overview | 4 |
| General-Purpose I/O | 4 |
| Flash Controller | 4 |
| Internal Precision Oscillator | 5 |
| 10-Bit Analog-to-Digital Converter | 5 |
| Analog Comparator | 5 |
| Universal Asynchronous Receiver/Transmitter | 5 |
| Timers | 5 |
| Interrupt Controller | 5 |
| Reset Controller | 5 |
| On-Chip Debugger | 6 |
| Pin Description | 7 |
| Available Packages | 7 |
| Pin Configurations | 7 |
| Signal Descriptions | 9 |
| Pin Characteristics | 11 |
| Address Space | 13 |
| Register File | 13 |
| Program Memory | 13 |
| Data Memory | 15 |
| Flash Information Area | 15 |
| Register Map | 16 |
| Reset and Stop Mode Recovery | 21 |
| Reset Types | 21 |
| Reset Sources | 23 |
| Power-On Reset | 23 |
| Voltage Brown-Out Reset | 24 |
| Watchdog Timer Reset | 25 |

| | |
|---|----|
| External Reset Input | 25 |
| External Reset Indicator | 26 |
| On-Chip Debugger Initiated Reset | 26 |
| Stop Mode Recovery | 26 |
| Stop Mode Recovery Using Watchdog Timer Time-Out | 27 |
| Stop Mode Recovery Using a GPIO Port Pin Transition | 27 |
| Stop Mode Recovery Using the External RESET Pin | 28 |
| Reset Register Definitions | 28 |
| Low-Power Modes | 30 |
| STOP Mode | 30 |
| HALT Mode | 31 |
| Peripheral-Level Power Control | 31 |
| Power Control Register Definitions | 31 |
| General-Purpose Input/Output | 33 |
| GPIO Port Availability By Device | 33 |
| Architecture | 34 |
| GPIO Alternate Functions | 34 |
| Direct LED Drive | 38 |
| Shared Reset Pin | 39 |
| Shared Debug Pin | 39 |
| Crystal Oscillator Override | 39 |
| 5V Tolerance | 39 |
| External Clock Setup | 40 |
| GPIO Interrupts | 40 |
| GPIO Control Register Definitions | 40 |
| Port A–C Address Registers | 41 |
| Port A–C Control Registers | 42 |
| Port A–C Data Direction Subregisters | 43 |
| Port A–C Alternate Function Subregisters | 43 |
| Port A–C Input Data Registers | 50 |
| Port A–C Output Data Register | 51 |
| LED Drive Enable Register | 51 |
| LED Drive Level High Register | 52 |
| LED Drive Level Low Register | 53 |
| Interrupt Controller | 54 |
| Interrupt Vector Listing | 54 |
| Architecture | 56 |
| Operation | 56 |
| Master Interrupt Enable | 56 |
| Interrupt Vectors and Priority | 57 |

| | |
|--|-----|
| Interrupt Assertion | 57 |
| Software Interrupt Assertion | 58 |
| Watchdog Timer Interrupt Assertion | 58 |
| Interrupt Control Register Definitions | 59 |
| Interrupt Request 0 Register | 59 |
| Interrupt Request 1 Register | 60 |
| Interrupt Request 2 Register | 61 |
| IRQ0 Enable High and Low Bit Registers | 61 |
| IRQ1 Enable High and Low Bit Registers | 63 |
| IRQ2 Enable High and Low Bit Registers | 65 |
| Interrupt Edge Select Register | 66 |
| Shared Interrupt Select Register | 67 |
| Interrupt Control Register | 68 |
| Timers | 69 |
| Architecture | 70 |
| Operation | 70 |
| Timer Operating Modes | 70 |
| Reading the Timer Count Values | 83 |
| Timer Pin Signal Operation | 83 |
| Timer Control Register Definitions | 83 |
| Timer 0–1 High and Low Byte Registers | 83 |
| Timer Reload High and Low Byte Registers | 84 |
| Timer 0–1 PWM High and Low Byte Registers | 86 |
| Timer 0–1 Control Registers | 86 |
| Watchdog Timer | 91 |
| Operation | 91 |
| Watchdog Timer Refresh | 92 |
| Watchdog Timer Time-Out Response | 92 |
| Watchdog Timer Reload Unlock Sequence | 93 |
| Watchdog Timer Control Register Definitions | 93 |
| Watchdog Timer Control Register | 94 |
| Watchdog Timer Reload Upper, High and Low Byte Registers | 94 |
| Universal Asynchronous Receiver/Transmitter | 97 |
| Architecture | 97 |
| Operation | 98 |
| Transmitting Data Using the Polled Method | 99 |
| Transmitting Data Using the Interrupt-Driven Method | 100 |
| Receiving Data Using the Polled Method | 101 |
| Receiving Data Using the Interrupt-Driven Method | 102 |
| Clear To Send (CTS) Operation | 103 |

| | |
|--|-----|
| MULTIPROCESSOR (9-Bit) Mode | 103 |
| External Driver Enable | 105 |
| UART Interrupts | 105 |
| UART Baud Rate Generator | 108 |
| UART Control Register Definitions | 108 |
| UART Transmit Data Register | 109 |
| UART Receive Data Register | 109 |
| UART Status 0 Register | 110 |
| UART Status 1 Register | 111 |
| UART Control 0 and Control 1 Registers | 112 |
| UART Address Compare Register | 115 |
| UART Baud Rate High and Low Byte Registers | 115 |
| Infrared Encoder/Decoder | 117 |
| Architecture | 117 |
| Operation | 117 |
| Transmitting IrDA Data | 118 |
| Receiving IrDA Data | 119 |
| Infrared Encoder/Decoder Control Register Definitions | 120 |
| Analog-to-Digital Converter | 121 |
| Architecture | 121 |
| Operation | 122 |
| Automatic Powerdown | 123 |
| Single-Shot Conversion | 123 |
| Continuous Conversion | 124 |
| Interrupts | 125 |
| Calibration and Compensation | 125 |
| ADC Control Register Definitions | 126 |
| ADC Control Register 0 | 126 |
| ADC Control/Status Register 1 | 129 |
| ADC Data High Byte Register | 130 |
| ADC Data Low Bits Register | 131 |
| Comparator | 132 |
| Operation | 132 |
| Comparator Control Register Definition | 133 |
| Flash Memory | 134 |
| Flash Information Area | 135 |
| Operation | 135 |
| Flash Operation Timing Using the Flash Frequency Registers | 137 |
| Flash Code Protection Against External Access | 137 |

| | |
|--|-----|
| Flash Code Protection Against Accidental Program and Erasure | 137 |
| Byte Programming | 139 |
| Page Erase | 139 |
| Mass Erase | 139 |
| Flash Controller Bypass | 140 |
| Flash Controller Behavior in DEBUG Mode | 140 |
| Flash Control Register Definitions | 141 |
| Flash Control Register | 141 |
| Flash Status Register | 142 |
| Flash Page Select Register | 142 |
| Flash Sector Protect Register | 144 |
| Flash Frequency High and Low Byte Registers | 144 |
| Flash Option Bits | 146 |
| Operation | 146 |
| Option Bit Configuration By Reset | 146 |
| Option Bit Types | 147 |
| Reading the Flash Information Page | 148 |
| Flash Option Bit Control Register Definitions | 148 |
| Trim Bit Address Register | 148 |
| Trim Bit Data Register | 149 |
| Flash Option Bit Address Space | 149 |
| Trim Bit Address Space | 151 |
| Zilog Calibration Data | 152 |
| ADC Calibration Data | 153 |
| Serialization Data | 154 |
| Randomized Lot Identifier | 154 |
| On-Chip Debugger | 156 |
| Architecture | 156 |
| Operation | 157 |
| OCD Interface | 157 |
| DEBUG Mode | 158 |
| OCD Data Format | 159 |
| OCD Autobaud Detector/Generator | 159 |
| OCD Serial Errors | 160 |
| OCD Unlock Sequence (8-Pin Devices Only) | 161 |
| Breakpoints | 161 |
| Runtime Counter | 161 |
| On-Chip Debugger Commands | 162 |
| On-Chip Debugger Control Register Definitions | 166 |
| OCD Control Register | 166 |

| | |
|---|-----|
| OCD Status Register | 168 |
| Oscillator Control | 169 |
| Operation | 169 |
| System Clock Selection | 169 |
| Clock Failure Detection and Recovery | 170 |
| Oscillator Control Register Definitions | 171 |
| Internal Precision Oscillator | 173 |
| Operation | 173 |
| eZ8 CPU Instruction Set | 174 |
| Assembly Language Programming Introduction | 174 |
| Assembly Language Syntax | 175 |
| eZ8 CPU Instruction Notation | 176 |
| eZ8 CPU Instruction Classes | 178 |
| eZ8 CPU Instruction Summary | 182 |
| Opcode Maps | 192 |
| Electrical Characteristics | 196 |
| Absolute Maximum Ratings | 196 |
| DC Characteristics | 197 |
| AC Characteristics | 200 |
| On-Chip Peripheral AC and DC Electrical Characteristics | 201 |
| General Purpose I/O Port Input Data Sample Timing | 204 |
| General Purpose I/O Port Output Timing | 206 |
| On-Chip Debugger Timing | 207 |
| UART Timing | 208 |
| Packaging | 210 |
| Ordering Information | 211 |
| Part Number Suffix Designations | 220 |
| Index | 221 |
| Customer Support | 230 |

List of Figures

| | | |
|------------|--|-----|
| Figure 1. | Z8 Encore! XP F0823 Series Block Diagram | 3 |
| Figure 2. | Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 8-Pin SOIC, QFN/MLF-S, or PDIP Package | 8 |
| Figure 3. | Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 20-Pin SOIC, SSOP or PDIP Package | 8 |
| Figure 4. | Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 28-Pin SOIC, SSOP or PDIP Package | 8 |
| Figure 5. | Power-On Reset Operation | 24 |
| Figure 6. | Voltage Brown-Out Reset Operation | 25 |
| Figure 7. | GPIO Port Pin Block Diagram | 34 |
| Figure 8. | Interrupt Controller Block Diagram | 56 |
| Figure 9. | Timer Block Diagram | 70 |
| Figure 10. | UART Block Diagram | 98 |
| Figure 11. | UART Asynchronous Data Format without Parity | 99 |
| Figure 12. | UART Asynchronous Data Format with Parity | 99 |
| Figure 13. | UART Asynchronous MULTIPROCESSOR Mode Data Format | 103 |
| Figure 14. | UART Driver Enable Signal Timing (shown with 1 Stop Bit and Parity) | 105 |
| Figure 15. | UART Receiver Interrupt Service Routine Flow | 107 |
| Figure 16. | Infrared Data Communication System Block Diagram | 117 |
| Figure 17. | Infrared Data Transmission | 118 |
| Figure 18. | IrDA Data Reception | 119 |
| Figure 19. | Analog-to-Digital Converter Block Diagram | 122 |
| Figure 20. | Flash Memory Arrangement | 135 |
| Figure 21. | Flash Controller Operation Flowchart | 136 |
| Figure 22. | On-Chip Debugger Block Diagram | 156 |
| Figure 23. | Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 1 of 2 | 157 |
| Figure 24. | Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 2 of 2 | 158 |
| Figure 25. | OCD Data Format | 159 |
| Figure 26. | Opcode Map Cell Description | 192 |
| Figure 27. | First Opcode Map | 194 |
| Figure 28. | Second Opcode Map after 1FH | 195 |
| Figure 29. | Port Input Sample Timing | 205 |

Figure 30. GPIO Port Output Timing 206
Figure 31. On-Chip Debugger Timing 207
Figure 32. UART Timing With CTS 208
Figure 33. UART Timing Without CTS 209

List of Tables

| | | |
|-----------|---|----|
| Table 1. | F0823 Series Family Part Selection Guide | 2 |
| Table 2. | F0823 Series Package Options | 7 |
| Table 3. | Signal Descriptions | 9 |
| Table 4. | Pin Characteristics (20- and 28-pin Devices)* | 11 |
| Table 5. | Pin Characteristics (8-Pin Devices) | 12 |
| Table 6. | Z8 Encore! XP F0823 Series Program Memory Maps | 14 |
| Table 7. | F0823 Series Flash Memory Information Area Map | 15 |
| Table 8. | Register File Address Map | 16 |
| Table 9. | Reset and Stop Mode Recovery Characteristics and Latency | 21 |
| Table 10. | Reset Sources and Resulting Reset Type | 23 |
| Table 11. | Stop Mode Recovery Sources and Resulting Action | 27 |
| Table 12. | Reset Status Register (RSTSTAT) | 28 |
| Table 13. | POR Indicator Values | 29 |
| Table 14. | Power Control Register 0 (PWRCTL0) | 32 |
| Table 15. | Port Availability by Device and Package Type | 33 |
| Table 16. | Port Alternate Function Mapping (8-Pin Parts) | 35 |
| Table 17. | Port Alternate Function Mapping (Non 8-Pin Parts) | 36 |
| Table 18. | GPIO Port Registers and Subregisters | 40 |
| Table 19. | Port A–C GPIO Address Registers (PxADDR) | 41 |
| Table 20. | PADDR[7:0] Subregister Functions | 41 |
| Table 21. | Port A–C Control Registers (PxCTL) | 42 |
| Table 22. | Port A–C Data Direction Subregisters (PxDD) | 43 |
| Table 23. | Port A–C Alternate Function Subregisters (PxAF) | 44 |
| Table 24. | Port A–C Output Control Subregisters (PxOC) | 44 |
| Table 25. | Port A–C High Drive Enable Subregisters (PHDEx) | 45 |
| Table 26. | Port A–C Stop Mode Recovery Source Enable Subregisters (PSMREx) | 46 |
| Table 27. | Port A–C Pull-Up Enable Subregisters (PPUEx) | 47 |
| Table 28. | Port A–C Alternate Function Set 1 Subregisters (PAFS1x) | 48 |
| Table 29. | Port A–C Alternate Function Set 2 Subregisters (PxAFS2) | 49 |
| Table 30. | Port A–C Input Data Registers (PxIN) | 50 |
| Table 31. | Port A–C Output Data Register (PxOUT) | 51 |
| Table 32. | LED Drive Enable (LEDEN) | 51 |
| Table 33. | LED Drive Level High Register (LEDLVLH) | 52 |

| | | |
|-----------|--|-----|
| Table 34. | LED Drive Level Low Register (LEDLVLL) | 53 |
| Table 35. | Trap and Interrupt Vectors in Order of Priority | 55 |
| Table 36. | Interrupt Request 0 Register (IRQ0) | 59 |
| Table 37. | Interrupt Request 1 Register (IRQ1) | 60 |
| Table 38. | Interrupt Request 2 Register (IRQ2) | 61 |
| Table 39. | IRQ0 Enable and Priority Encoding | 61 |
| Table 40. | IRQ0 Enable High Bit Register (IRQ0ENH) | 62 |
| Table 41. | IRQ0 Enable Low Bit Register (IRQ0ENL) | 62 |
| Table 42. | IRQ1 Enable and Priority Encoding | 63 |
| Table 43. | IRQ1 Enable Low Bit Register (IRQ1ENL) | 64 |
| Table 44. | IRQ1 Enable High Bit Register (IRQ1ENH) | 64 |
| Table 45. | IRQ2 Enable and Priority Encoding | 65 |
| Table 46. | IRQ2 Enable High Bit Register (IRQ2ENH) | 65 |
| Table 47. | IRQ2 Enable Low Bit Register (IRQ2ENL) | 66 |
| Table 48. | Interrupt Edge Select Register (IRQES) | 66 |
| Table 49. | Shared Interrupt Select Register (IRQSS) | 67 |
| Table 50. | Interrupt Control Register (IRQCTL) | 68 |
| Table 51. | Timer 0–1 High Byte Register (TxH) | 84 |
| Table 52. | Timer 0–1 Low Byte Register (TxL) | 84 |
| Table 53. | Timer 0–1 Reload High Byte Register (TxRH) | 85 |
| Table 54. | Timer 0–1 Reload Low Byte Register (TxRL) | 85 |
| Table 55. | Timer 0–1 PWM High Byte Register (TxPWMH) | 86 |
| Table 56. | Timer 0–1 PWM Low Byte Register (TxPWML) | 86 |
| Table 57. | Timer 0–1 Control Register 0 (TxCTL0) | 87 |
| Table 58. | Timer 0–1 Control Register 1 (TxCTL1) | 88 |
| Table 59. | Watchdog Timer Approximate Time-Out Delays | 91 |
| Table 60. | Watchdog Timer Control Register (WDTCTL) | 94 |
| Table 61. | Watchdog Timer Reload Upper Byte Register (WDTU) | 95 |
| Table 62. | Watchdog Timer Reload High Byte Register (WDTH) | 95 |
| Table 63. | Watchdog Timer Reload Low Byte Register (WDTL) | 95 |
| Table 64. | UART Transmit Data Register (U0TXD) | 109 |
| Table 65. | UART Receive Data Register (U0RXD) | 109 |
| Table 66. | UART Status 0 Register (U0STAT0) | 110 |
| Table 67. | UART Status 1 Register (U0STAT1) | 111 |
| Table 68. | UART Control 0 Register (U0CTL0) | 112 |
| Table 69. | UART Control 1 Register (U0CTL1) | 113 |

| | | |
|------------|--|-----|
| Table 70. | UART Address Compare Register (U0ADDR) | 115 |
| Table 71. | UART Baud Rate High Byte Register (U0BRH) | 115 |
| Table 72. | UART Baud Rate Low Byte Register (U0BRL) | 115 |
| Table 73. | UART Baud Rates | 116 |
| Table 74. | ADC Control Register 0 (ADCCTL0) | 127 |
| Table 75. | ADC Control/Status Register 1 (ADCCTL1) | 129 |
| Table 76. | ADC Data High Byte Register (ADCD_H) | 130 |
| Table 77. | ADC Data Low Bits Register (ADCD_L) | 131 |
| Table 78. | Comparator Control Register (CMP0) | 133 |
| Table 79. | Z8 Encore! XP F0823 Series Flash Memory Configurations | 134 |
| Table 80. | Flash Code Protection Using the Flash Option Bits | 138 |
| Table 81. | Flash Control Register (FCTL) | 141 |
| Table 82. | Flash Status Register (FSTAT) | 142 |
| Table 83. | Flash Page Select Register (FPS) | 143 |
| Table 84. | Flash Sector Protect Register (FPROT) | 144 |
| Table 85. | Flash Frequency High Byte Register (FFREQH) | 145 |
| Table 86. | Flash Frequency Low Byte Register (FFREQL) | 145 |
| Table 87. | Trim Bit Address Register (TRMADR) | 148 |
| Table 88. | Trim Bit Data Register (TRMDR) | 149 |
| Table 89. | Flash Option Bits at Program Memory Address 0000H | 149 |
| Table 90. | Flash Options Bits at Program Memory Address 0001H | 150 |
| Table 91. | Trim Options Bits at Address 0000H | 151 |
| Table 92. | Trim Option Bits at 0001H | 152 |
| Table 93. | Trim Option Bits at 0002H (TIPO) | 152 |
| Table 94. | ADC Calibration Bits | 153 |
| Table 95. | ADC Calibration Data Location | 153 |
| Table 96. | Serial Number at 001C–001F (S_NUM) | 154 |
| Table 97. | Serialization Data Locations | 154 |
| Table 98. | Lot Identification Number (RAND_LOT) | 154 |
| Table 99. | Randomized Lot ID Locations | 155 |
| Table 100. | OCD Baud-Rate Limits | 160 |
| Table 101. | OCD Commands | 162 |
| Table 102. | OCD Control Register (OCDCTL) | 167 |
| Table 103. | OCD Status Register (OCDSTAT) | 168 |
| Table 104. | Oscillator Configuration and Selection | 169 |
| Table 105. | Oscillator Control Register (OSCCTL) | 172 |

| | | |
|------------|---|-----|
| Table 106. | Assembly Language Syntax Example 1 | 175 |
| Table 107. | Assembly Language Syntax Example 2 | 176 |
| Table 108. | Notational Shorthand | 176 |
| Table 109. | Additional Symbols | 177 |
| Table 110. | Arithmetic Instructions | 178 |
| Table 111. | Bit Manipulation Instructions | 179 |
| Table 112. | Block Transfer Instructions | 179 |
| Table 113. | CPU Control Instructions | 180 |
| Table 114. | Load Instructions | 180 |
| Table 115. | Logical Instructions | 181 |
| Table 116. | Program Control Instructions | 181 |
| Table 117. | Rotate and Shift Instructions | 181 |
| Table 118. | eZ8 CPU Instruction Summary | 182 |
| Table 119. | Opcode Map Abbreviations | 193 |
| Table 120. | Absolute Maximum Ratings | 196 |
| Table 121. | DC Characteristics | 197 |
| Table 122. | Power Consumption | 199 |
| Table 123. | AC Characteristics | 200 |
| Table 124. | Internal Precision Oscillator Electrical Characteristics | 200 |
| Table 125. | Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing | 201 |
| Table 126. | Flash Memory Electrical Characteristics and Timing | 202 |
| Table 127. | Watchdog Timer Electrical Characteristics and Timing | 202 |
| Table 128. | Analog-to-Digital Converter Electrical Characteristics and Timing | 203 |
| Table 129. | Comparator Electrical Characteristics | 204 |
| Table 130. | GPIO Port Input Timing | 205 |
| Table 131. | GPIO Port Output Timing | 206 |
| Table 132. | On-Chip Debugger Timing | 207 |
| Table 133. | UART Timing With CTS | 208 |
| Table 134. | UART Timing Without CTS | 209 |
| Table 135. | Z8 Encore! XP F0823 Series Ordering Matrix | 211 |

Overview

Zilog's Z8 Encore! XP microcontroller unit (MCU) family of products are the first Zilog microcontroller products based on the 8-bit eZ8 CPU core. Z8 Encore! XP F0823 Series products expand upon Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming capability allows for faster development time and program changes in the field. The new eZ8 CPU is upward compatible with existing Z8 instructions. The rich peripheral set of Z8 Encore! XP F0823 Series makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Features

The key features of Z8 Encore! XP F0823 Series include:

- 5MHz eZ8 CPU
- 1 KB, 2KB, 4KB, or 8KB Flash memory with in-circuit programming capability
- 256B, 512B, or 1 KB register RAM
- 6 to 24 I/O pins depending upon package
- Internal precision oscillator (IPO)
- Full-duplex UART
- The universal asynchronous receiver/transmitter (UART) baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Infrared data association (IrDA)-compliant infrared encoder/decoders, integrated with UART
- Two enhanced 16-bit timers with capture, compare, and PWM capability
- Watchdog Timer (WDT) with dedicated internal RC oscillator
- On-Chip Debugger (OCD)
- Optional 8-channel, 10-bit Analog-to-Digital Converter (ADC)
- On-Chip analog comparator
- Up to 20 vectored interrupts
- Direct LED drive with programmable drive strengths
- Voltage Brown-Out (VBO) protection
- Power-On Reset (POR)

- 2.7V to 3.6V operating voltage
- Up to thirteen 5V-tolerant input pins
- 8-, 20-, and 28-pin packages
- 0°C to +70°C and –40°C to +105°C for operating temperature ranges

Part Selection Guide

Table 1 lists the basic features and package styles available for each device within the Z8 Encore! XP® F0823 Series product line.

Table 1. F0823 Series Family Part Selection Guide

| Part Number | Flash (KB) | RAM (B) | I/O | ADC Inputs | Packages |
|-------------|------------|---------|------|------------|----------------------|
| Z8F0823 | 8 | 1024 | 6–22 | 4–8 | 8-, 20-, and 28-pins |
| Z8F0813 | 8 | 1024 | 6–24 | 0 | 8-, 20-, and 28-pins |
| Z8F0423 | 4 | 1024 | 6–22 | 4–8 | 8-, 20-, and 28-pins |
| Z8F0413 | 4 | 1024 | 6–24 | 0 | 8-, 20-, and 28-pins |
| Z8F0223 | 2 | 512 | 6–22 | 4–8 | 8-, 20-, and 28-pins |
| Z8F0213 | 2 | 512 | 6–24 | 0 | 8-, 20-, and 28-pins |
| Z8F0123 | 1 | 256 | 6–22 | 4–8 | 8-, 20-, and 28-pins |
| Z8F0113 | 1 | 256 | 6–24 | 0 | 8-, 20-, and 28-pins |

Block Diagram

Figure 1 displays a block diagram of the F0823 Series architecture.



Figure 1. Z8 Encore! XP F0823 Series Block Diagram

CPU and Peripheral Overview

The eZ8 CPU, Zilog's latest 8-bit central processing unit (CPU), meets the continuing demand for faster and code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8 instruction set. The eZ8 CPU features include:

- Direct register-to-register architecture allows each register to function as an accumulator, improving execution time and decreasing the required program memory
- Software stack allows much greater depth in subroutine calls and interrupts than hardware stacks
- Compatible with existing Z8 code
- Expanded internal Register File allows access of up to 4 KB
- New instructions improve execution efficiency for code developed using higher-level programming languages, including C
- Pipelined instruction fetch and execution
- New instructions for improved performance including BIT, BSWAP, BTJ, CPC, LDC, LDCI, LEA, MULT, and SRL
- New instructions support 12-bit linear addressing of the Register File
- Up to 10 MIPS operation
- C-Compiler friendly
- 2 to 9 clock cycles per instruction

For more information about the eZ8 CPU, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#) available for download at www.zilog.com.

General-Purpose I/O

F0823 Series features 6 to 24 port pins (Ports A–C) for general-purpose I/O (GPIO). The number of GPIO pins available is a function of package. Each pin is individually programmable. 5V-tolerant input pins are available on all I/Os on 8-pin devices, most I/Os on other package types.

Flash Controller

The Flash Controller programs and erases Flash memory. The Flash Controller supports protection against accidental program and erasure, as well as factory serialization and read protection.

Internal Precision Oscillator

The internal precision oscillator (IPO) is a trimmable clock source that requires no external components.

10-Bit Analog-to-Digital Converter

The optional analog-to-digital converter (ADC) converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from eight different analog input pins in both single-ended and differential modes.

Analog Comparator

The analog comparator compares the signal at an input pin with either an internal programmable voltage reference or a second input pin. The comparator output can be used to drive either an output pin or to generate an interrupt.

Universal Asynchronous Receiver/Transmitter

The UART is full-duplex and capable of handling asynchronous data transfers. The UART supports 8- and 9-bit data modes and selectable parity. The UART also supports multi-drop address processing in hardware. The UART baud rate generator can be configured and used as a basic 16-bit timer.

Timers

Two enhanced 16-bit reloadable timers can be used for timing/counting events or for motor control operations. These timers provide a 16-bit programmable reload counter and operate in ONE-SHOT, CONTINUOUS, GATED, CAPTURE, CAPTURE RESTART, COMPARE, CAPTURE AND COMPARE, PWM SINGLE OUTPUT, and PWM DUAL OUTPUT modes.

Interrupt Controller

Z8 Encore! XP[®] F0823 Series products support up to 20 interrupts. These interrupts consist of eight internal peripheral interrupts and 12 general-purpose I/O pin interrupt sources. The interrupts have three levels of programmable interrupt priority.

Reset Controller

Z8 Encore! XP[®] F0823 Series products can be reset using the $\overline{\text{RESET}}$ pin, POR, WDT time-out, STOP Mode exit, or Voltage Brown-Out warning signal. The $\overline{\text{RESET}}$ pin is bidirectional, that is, it functions as reset source as well as a reset indicator.

On-Chip Debugger

F0823 Series products feature an integrated On-Chip Debugger. The OCD provides a rich-set of debugging capabilities, such as reading and writing registers, programming Flash memory, setting breakpoints and executing code. A single-pin interface provides communication to the OCD.

Pin Description

Z8 Encore! XP F0823 Series products are available in a variety of package styles and pin configurations. This chapter describes the signals and pin configurations available for each of the package styles. For information about physical package specifications, see [the Packaging chapter on page 210](#).

Available Packages

Table 2 lists the package styles that are available for each device in the F0823 Series product line.

Table 2. F0823 Series Package Options

| Part Number | ADC | 8-pin PDIP | 8-pin SOIC | 20-pin PDIP | 20-pin SOIC | 20-pin SSOP | 28-pin PDIP | 28-pin SOIC | 28-pin SSOP | 8-pin QFN/MLF-S |
|-------------|-----|------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------------|
| Z8F0823 | Yes | X | X | X | X | X | X | X | X | X |
| Z8F0813 | No | X | X | X | X | X | X | X | X | X |
| Z8F0423 | Yes | X | X | X | X | X | X | X | X | X |
| Z8F0413 | No | X | X | X | X | X | X | X | X | X |
| Z8F0223 | Yes | X | X | X | X | X | X | X | X | X |
| Z8F0213 | No | X | X | X | X | X | X | X | X | X |
| Z8F0123 | Yes | X | X | X | X | X | X | X | X | X |
| Z8F0113 | No | X | X | X | X | X | X | X | X | X |

Pin Configurations

Figures 2 through 4 display the pin configurations for all packages available in the F0823 Series. For description of signals, see Table 3. The analog input alternate functions (ANAx) are not available on the Z8F0x13 devices. The analog supply pins (AV_{DD} and AV_{SS}) are also not available on these parts, and are replaced by PB6 and PB7.

At reset, all pins of Ports A, B, and C default to an input state. In addition, any alternate functionality is not enabled, so the pins function as general-purpose input ports until programmed otherwise.

The pin configurations listed are preliminary and subject to change based on manufacturing limitations.



Figure 2. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 8-Pin SOIC, QFN/MLF-S, or PDIP Package*

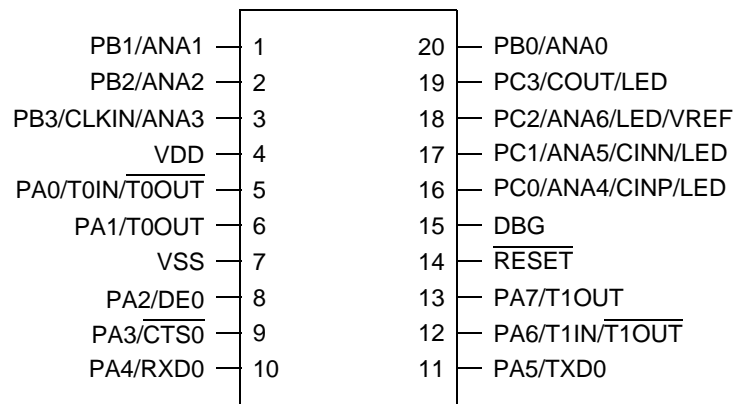


Figure 3. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 20-Pin SOIC, SSOP or PDIP Package*



Figure 4. Z8F08x3, Z8F04x3, F02x3 and Z8F01x3 in 28-Pin SOIC, SSOP or PDIP Package*

► **Note:** *Analog input alternate functions (ANA) are not available on Z8F0x13 devices.

Signal Descriptions

Table 3 lists the Z8 Encore! XP F0823 Series signals. To determine the signals available for the specific package styles, see the [Pin Configurations](#) section on page 7.

Table 3. Signal Descriptions

| Signal Mnemonic | I/O | Description |
|--------------------------------------|-----|---|
| General-Purpose I/O Ports A–D | | |
| PA[7:0] | I/O | Port A. These pins are used for general-purpose I/O. |
| PB[7:0] ¹ | I/O | Port B. These pins are used for general-purpose I/O. PB6 and PB7 are available only in those devices without an ADC. |
| PC[7:0] | I/O | Port C. These pins are used for general-purpose I/O. |
| UART Controllers | | |
| TXD0 | O | Transmit Data. This signal is the transmit output from the UART and IrDA. |
| RXD0 | I | Receive Data. This signal is the receive input for the UART and IrDA. |
| $\overline{\text{CTS0}}$ | I | Clear To Send. This signal is the flow control input for the UART. |
| DE | O | Driver Enable. This signal allows automatic control of external RS-485 drivers. This signal is approximately the inverse of the TXE (Transmit Empty) bit in the UART Status 0 Register. The DE signal can be used to ensure the external RS-485 driver is enabled when data is transmitted by the UART. |
| Timers | | |
| $\overline{\text{T0OUT/T1OUT}}$ | O | Timer Output 0–1. These signals are output from the timers. |
| $\overline{\text{T0OUT/T1OUT}}$ | O | Timer Complement Output 0–1. These signals are output from the timers in PWM DUAL OUTPUT Mode. |
| T0IN/T1IN | I | Timer Input 0–1. These signals are used as the capture, gating and counter inputs. The T0IN signal is multiplexed $\overline{\text{T0OUT}}$ signals. |
| Comparator | | |
| CINP/CINN | I | Comparator Inputs. These signals are the positive and negative inputs to the comparator. |

Notes:

1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV_{DD} and AV_{SS}.
2. The AV_{DD} and AV_{SS} signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC.

Table 3. Signal Descriptions (Continued)

| Signal Mnemonic | I/O | Description |
|---|-----|---|
| COUT | O | Comparator Output. This is the output of the comparator. |
| Analog | | |
| ANA[7:0] | I | Analog port. These signals are used as inputs to the ADC. The ANA0, ANA1, and ANA2 pins can also access the inputs and output of the integrated transimpedance amplifier. |
| VREF | I/O | Analog-to-Digital Converter reference voltage input. |
| Clock Input | | |
| CLKIN | I | Clock Input Signal. This pin can be used to input a TTL-level signal to be used as the system clock. |
| LED Drivers | | |
| LED | O | Direct LED drive capability. All port C pins have the capability to drive an LED without any other external components. These pins have programmable drive strengths set by the GPIO block. |
| On-Chip Debugger | | |
| DBG | I/O | Debug. This signal is the control and data input and output to and from the OCD. Caution: The DBG pin is open-drain and requires an external pull-up resistor to ensure proper operation. |
| Reset | | |
| RESET | I/O | RESET. Generates a reset when asserted (driven Low). Also serves as a reset indicator; the Z8 Encore! XP forces this pin Low when in reset. This pin is open-drain and features an enabled internal pull-up resistor. |
| Power Supply | | |
| V _{DD} | I | Digital Power Supply. |
| AV _{DD} ² | I | Analog Power Supply. |
| V _{SS} | I | Digital Ground. |
| AV _{SS} | I | Analog Ground. |
| Notes: | | |
| 1. PB6 and PB7 are only available in 28-pin packages without ADC. In 28-pin packages with ADC, they are replaced by AV _{DD} and AV _{SS} . | | |
| 2. The AV _{DD} and AV _{SS} signals are available only in 28-pin packages with ADC. They are replaced by PB6 and PB7 on 28-pin packages without ADC. | | |

Pin Characteristics

Table 4 provides detailed information about the characteristics for each pin available on Z8 Encore! XP F0823 Series 20- and 28-pin devices. Data in Table 4 is sorted alphabetically by the pin symbol mnemonic.

► **Note:** All six I/O pins on the 8-pin packages are 5 V-tolerant (unless the pull-up devices are enabled). The right-most column in Table 4 describes 5 V tolerance for the 20- and 28-pin packages only.

Table 4. Pin Characteristics (20- and 28-pin Devices)*

| Symbol Mnemonic | Direction | Reset Direction | Active Low or Active High | Tristate Output | Internal Pull-up or Pull-down | Schmitt-Trigger Input | Open Drain Output | 5V Tolerance |
|-----------------|-----------|-------------------------|---------------------------|-----------------|-------------------------------|-----------------------|---------------------|--------------|
| AVDD | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| AVSS | N/A | N/A | N/A | N/A | N/A | N/A | N/A | NA |
| DBG | I/O | I | N/A | Yes | No | Yes | Yes | Yes |
| PA[7:0] | I/O | I | N/A | Yes | Program-mable Pull-up | Yes | Yes, Programmable | PA[7:2] only |
| PB[7:0] | I/O | I | N/A | Yes | Program-mable Pull-up | Yes | Yes, Programmable | PB[7:6] only |
| PC[7:0] | I/O | I | N/A | Yes | Program-mable Pull-up | Yes | Yes, Programmable | PC[7:3] only |
| RESET | I/O | I/O (defaults to RESET) | Low (in Reset mode) | Yes (PDO only) | Always on for RESET | Yes | Always on for RESET | Yes |
| VDD | N/A | N/A | N/A | N/A | | | N/A | N/A |
| VSS | N/A | N/A | N/A | N/A | | | N/A | N/A |

Note: PB6 and PB7 are available only in the devices without ADC.

Table 5 provides detailed information about the characteristics for each pin available on Z8 Encore! XP F0823 Series 8-pin devices.

Table 5. Pin Characteristics (8-Pin Devices)

| Symbol Mnemonic | Direction | Reset Direction | Active Low or Active High | Tristate Output | Internal Pull-up or Pull-down | Schmitt-Trigger Input | Open Drain Output | 5V Tolerance |
|-----------------|-----------|--|---------------------------|-----------------|---|-----------------------|---|------------------------------|
| PA0/DBG | I/O | I (but can change during reset if key sequence detected) | N/A | Yes | Programmable Pull-up | Yes | Yes, Programmable | Yes, unless pull-ups enabled |
| PA1 | I/O | I | N/A | Yes | Programmable Pull-up | Yes | Yes, Programmable | Yes, unless pull-ups enabled |
| RESET/PA2 | I/O | I/O (defaults to $\overline{\text{RESET}}$) | N/A | Yes | Programmable for PA2; always on for $\overline{\text{RESET}}$ | Yes | Programmable for PA2; always on for $\overline{\text{RESET}}$ | Yes, unless pull-ups enabled |
| PA[5:3] | I/O | I | N/A | Yes | Programmable Pull-up | Yes | Yes, Programmable | Yes, unless pull-ups enabled |
| VDD | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| VSS | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |

Address Space

The eZ8 CPU can access three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral, and general-purpose I/O Port Control Registers
- The Program Memory contains addresses for all memory locations having executable code and/or data
- The Data Memory contains addresses for all memory locations that contain data only

These three address spaces are covered briefly in the following subsections. For more detailed information regarding the eZ8 CPU and its address space, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#), available for download at www.zilog.com.

Register File

The Register File address space in the Z8 Encore! XP[™] MCU is 4KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read, and registers defined as destinations are written. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256 B control register section are reserved (unavailable). Reading from a reserved Register File address returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. Z8 Encore! XP F0823 Series devices contain 256B–1KB of on-chip RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

Program Memory

The eZ8 CPU supports 64KB of Program Memory address space. F0823 Series devices contain 1KB to 8KB of on-chip Flash memory in the Program Memory address space. Reading from Program Memory addresses outside the available Flash memory addresses

returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 6 describes the Program Memory maps for the Z8 Encore! XP F0823 Series products.

Table 6. Z8 Encore! XP F0823 Series Program Memory Maps

| Program Memory Address (Hex) | Function |
|-------------------------------------|--------------------------|
| Z8F0823 and Z8F0813 Products | |
| 0000–0001 | Flash Option Bits |
| 0002–0003 | Reset Vector |
| 0004–0005 | WDT Interrupt Vector |
| 0006–0007 | Illegal Instruction Trap |
| 0008–0037 | Interrupt Vectors* |
| 0038–003D | Oscillator Fail Traps* |
| 003E–0FFF | Program Memory |
| Z8F0423 and Z8F0413 Products | |
| 0000–0001 | Flash Option Bits |
| 0002–0003 | Reset Vector |
| 0004–0005 | WDT Interrupt Vector |
| 0006–0007 | Illegal Instruction Trap |
| 0008–0037 | Interrupt Vectors* |
| 0038–003D | Oscillator Fail Traps* |
| 003E–0FFF | Program Memory |
| Z8F0223 and Z8F0213 Products | |
| 0000–0001 | Flash Option Bits |
| 0002–0003 | Reset Vector |
| 0004–0005 | WDT Interrupt Vector |
| 0006–0007 | Illegal Instruction Trap |
| 0008–0037 | Interrupt Vectors* |
| 0038–003D | Oscillator Fail Traps* |
| 003E–07FF | Program Memory |

Note: *See [the Trap and Interrupt Vectors in Order of Priority section on page 55](#) for a list of the interrupt vectors and traps.

Table 6. Z8 Encore! XP F0823 Series Program Memory Maps (Continued)

| Program Memory Address (Hex) | Function |
|-------------------------------------|--------------------------|
| Z8F0123 and Z8F0113 Products | |
| 0000–0001 | Flash Option Bits |
| 0002–0003 | Reset Vector |
| 0004–0005 | WDT Interrupt Vector |
| 0006–0007 | Illegal Instruction Trap |
| 0008–0037 | Interrupt Vectors* |
| 0038–003D | Oscillator Fail Traps* |
| 003E–03FF | Program Memory |

Note: *See [the Trap and Interrupt Vectors in Order of Priority section on page 55](#) for a list of the interrupt vectors and traps.

Data Memory

Z8 Encore! XP F0823 Series does not use the eZ8 CPU's 64KB Data Memory address space.

Flash Information Area

Table 7 lists the F0823 Series Flash Information Area. This 128B Information Area is accessed by setting bit 7 of the Flash Page Select Register to 1. When access is enabled, the Flash Information Area is mapped into the Program Memory and overlays the 128 bytes at addresses FE00H to FF7FH. When the Information Area access is enabled, all reads from these Program Memory addresses return the Information Area data rather than the Program Memory data. Access to the Flash Information Area is read-only.

Table 7. F0823 Series Flash Memory Information Area Map

| Program Memory Address (Hex) | Function |
|------------------------------|--|
| FE00–FE3F | Zilog Option Bits. |
| FE40–FE53 | Part Number. 20-character ASCII alphanumeric code Left-justified and filled with FH. |
| FE54–FE5F | Reserved. |
| FE60–FE7F | Zilog Calibration Data. |
| FE80–FFFF | Reserved. |

Register Map

Table 8 lists an address map of the Z8 Encore! XP F0823 Series Register File. Not all devices and package styles in the Z8 Encore! XP F0823 Series support the ADC, nor all GPIO ports. Consider registers for unimplemented peripherals to be reserved.

Table 8. Register File Address Map

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|--------------------------------|-----------------------------------|----------|-------------|--------------------|
| General-Purpose RAM | | | | |
| Z8F0823/Z8F0813 Devices | | | | |
| 000–3FF | General-Purpose Register File RAM | — | XX | |
| 400–EFF | Reserved | — | XX | |
| Z8F0423/Z8F0413 Devices | | | | |
| 000–3FF | General-Purpose Register File RAM | — | XX | |
| 400–EFF | Reserved | — | XX | |
| Z8F0223/Z8F0213 Devices | | | | |
| 000–1FF | General-Purpose Register File RAM | — | XX | |
| 200–EFF | Reserved | — | XX | |
| Z8F0123/Z8F0113 Devices | | | | |
| 000–0FF | General-Purpose Register File RAM | — | XX | |
| 100–EFF | Reserved | — | XX | |
| Timer 0 | | | | |
| F00 | Timer 0 High Byte | T0H | 00 | 84 |
| F01 | Timer 0 Low Byte | T0L | 01 | 84 |
| F02 | Timer 0 Reload High Byte | T0RH | FF | 85 |
| F03 | Timer 0 Reload Low Byte | T0RL | FF | 85 |
| F04 | Timer 0 PWM High Byte | T0PWMH | 00 | 86 |
| F05 | Timer 0 PWM Low Byte | T0PWML | 00 | 86 |
| F06 | Timer 0 Control 0 | T0CTL0 | 00 | 87 |
| F07 | Timer 0 Control 1 | T0CTL1 | 00 | 88 |
| Timer 1 | | | | |
| F08 | Timer 1 High Byte | T1H | 00 | 84 |
| F09 | Timer 1 Low Byte | T1L | 01 | 84 |

Note: XX=Undefined.

Table 8. Register File Address Map (Continued)

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|--|---------------------------|----------|-------------|---------------------|
| Timer 1 (cont'd) | | | | |
| F0A | Timer 1 Reload High Byte | T1RH | FF | 85 |
| F0B | Timer 1 Reload Low Byte | T1RL | FF | 85 |
| F0C | Timer 1 PWM High Byte | T1PWMH | 00 | 86 |
| F0D | Timer 1 PWM Low Byte | T1PWML | 00 | 86 |
| F0E | Timer 1 Control 0 | T1CTL0 | 00 | 87 |
| F0F | Timer 1 Control 1 | T1CTL1 | 00 | 84 |
| F10–F3F | Reserved | — | XX | |
| UART | | | | |
| F40 | UART0 Transmit Data | U0TXD | XX | 109 |
| | UART0 Receive Data | U0RXD | XX | 109 |
| F41 | UART0 Status 0 | U0STAT0 | 0000011Xb | 110 |
| F42 | UART0 Control 0 | U0CTL0 | 00 | 112 |
| F43 | UART0 Control 1 | U0CTL1 | 00 | 112 |
| F44 | UART0 Status 1 | U0STAT1 | 00 | 111 |
| F45 | UART0 Address Compare | U0ADDR | 00 | 115 |
| F46 | UART0 Baud Rate High Byte | U0BRH | FF | 115 |
| F47 | UART0 Baud Rate Low Byte | U0BRL | FF | 115 |
| F48–F6F | Reserved | — | XX | |
| Analog-to-Digital Converter (ADC) | | | | |
| F70 | ADC Control 0 | ADCCTL0 | 00 | 127 |
| F71 | ADC Control 1 | ADCCTL1 | 80 | 127 |
| F72 | ADC Data High Byte | ADCD_H | XX | 130 |
| F73 | ADC Data Low Bits | ADCD_L | XX | 130 |
| F74–F7F | Reserved | — | XX | |
| Low Power Control | | | | |
| F80 | Power Control 0 | PWRCTL0 | 80 | 32 |
| F81 | Reserved | — | XX | |
| LED Controller | | | | |
| F82 | LED Drive Enable | LEDEN | 00 | 51 |
| F83 | LED Drive Level High Byte | LEDLVLH | 00 | 52 |

Note: XX=Undefined.

Table 8. Register File Address Map (Continued)

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|--------------------------------|--------------------------|----------|-------------|---------------------|
| LED Controller (cont'd) | | | | |
| F84 | LED Drive Level Low Byte | LEDLVLL | 00 | 53 |
| F85 | Reserved | — | XX | |
| Oscillator Control | | | | |
| F86 | Oscillator Control | OSCCTL | A0 | 172 |
| F87–F8F | Reserved | — | XX | |
| Comparator 0 | | | | |
| F90 | Comparator 0 Control | CMP0 | 14 | 133 |
| F91–FBF | Reserved | — | XX | |
| Interrupt Controller | | | | |
| FC0 | Interrupt Request 0 | IRQ0 | 00 | 59 |
| FC1 | IRQ0 Enable High Bit | IRQ0ENH | 00 | 62 |
| FC2 | IRQ0 Enable Low Bit | IRQ0ENL | 00 | 62 |
| FC3 | Interrupt Request 1 | IRQ1 | 00 | 60 |
| FC4 | IRQ1 Enable High Bit | IRQ1ENH | 00 | 64 |
| FC5 | IRQ1 Enable Low Bit | IRQ1ENL | 00 | 64 |
| FC6 | Interrupt Request 2 | IRQ2 | 00 | 61 |
| FC7 | IRQ2 Enable High Bit | IRQ2ENH | 00 | 65 |
| FC8 | IRQ2 Enable Low Bit | IRQ2ENL | 00 | 66 |
| FC9–FCC | Reserved | — | XX | |
| FCD | Interrupt Edge Select | IRQES | 00 | 67 |
| FCE | Shared Interrupt Select | IRQSS | 00 | 67 |
| FCF | Interrupt Control | IRQCTL | 00 | 68 |
| GPIO Port A | | | | |
| FD0 | Port A Address | PAADDR | 00 | 40 |
| FD1 | Port A Control | PACTL | 00 | 42 |
| FD2 | Port A Input Data | PAIN | XX | 43 |
| FD3 | Port A Output Data | PAOUT | 00 | 43 |
| GPIO Port B | | | | |
| FD4 | Port B Address | PBADDR | 00 | 40 |
| FD5 | Port B Control | PBCTL | 00 | 42 |

Note: XX=Undefined.

Table 8. Register File Address Map (Continued)

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|--------------------------------|---------------------------------------|----------|-------------|---------------------|
| GPIO Port B (cont'd) | | | | |
| FD6 | Port B Input Data | PBIN | XX | 43 |
| FD7 | Port B Output Data | PBOUT | 00 | 43 |
| GPIO Port C | | | | |
| FD8 | Port C Address | PCADDR | 00 | 40 |
| FD9 | Port C Control | PCCTL | 00 | 42 |
| FDA | Port C Input Data | PCIN | XX | 43 |
| FDB | Port C Output Data | PCOUT | 00 | 43 |
| FDC–FEF | Reserved | — | XX | |
| Watchdog Timer (WDT) | | | | |
| FF0 | Reset Status | RSTSTAT | XX | 94 |
| | Watchdog Timer Control | WDTCTL | XX | 94 |
| FF1 | Watchdog Timer Reload Upper Byte | WDTU | FF | 95 |
| FF2 | Watchdog Timer Reload High Byte | WDTH | FF | 95 |
| FF3 | Watchdog Timer Reload Low Byte | WDTL | FF | 95 |
| FF4–FF5 | Reserved | — | XX | |
| Trim Bit Control | | | | |
| FF6 | Trim Bit Address | TRMADR | 00 | 148 |
| FF7 | Trim Data | TRMDR | XX | 149 |
| Flash Memory Controller | | | | |
| FF8 | Flash Control | FCTL | 00 | 141 |
| FF8 | Flash Status | FSTAT | 00 | 142 |
| FF9 | Flash Page Select | FPS | 00 | 143 |
| | Flash Sector Protect | FPROT | 00 | 144 |
| FFA | Flash Programming Frequency High Byte | FFREQH | 00 | 145 |
| FFB | Flash Programming Frequency Low Byte | FFREQL | 00 | 145 |

Note: XX=Undefined.

Table 8. Register File Address Map (Continued)

| Address (Hex) | Register Description | Mnemonic | Reset (Hex) | Page No. |
|---------------------|-------------------------|----------|-------------|--|
| eZ8 CPU | | | | |
| FFC | Flags | — | XX | Refer to the eZ8 CPU Core User Manual (UM0128) |
| FFD | Register Pointer | RP | XX | |
| FFE | Stack Pointer High Byte | SPH | XX | |
| FFF | Stack Pointer Low Byte | SPL | XX | |
| Note: XX=Undefined. | | | | |

Reset and Stop Mode Recovery

The Reset Controller within the Z8 Encore! XP F0823 Series controls Reset and Stop Mode Recovery operation and provides indication of low supply voltage conditions. In typical operation, the following events cause a Reset:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO)
- Watchdog Timer time-out (when configured by the WDT_RES Flash Option Bit to initiate a reset)
- External $\overline{\text{RESET}}$ pin assertion (when the alternate RESET function is enabled by the GPIO register)
- On-chip Debugger initiated Reset (OCDCTL[0] set to 1)

When the device is in STOP Mode, a Stop Mode Recovery is initiated by either of the following:

- Watchdog Timer time-out
- GPIO port input pin transition on an enabled Stop Mode Recovery source

The VBO circuitry on the device performs the following function:

- Generates the VBO reset when the supply voltage drops below a minimum safe level

Reset Types

F0823 Series MCUs provide several different types of Reset operations. Stop Mode Recovery is considered a form of Reset. Table 9 lists the types of Reset and their operating characteristics. The duration of a System Reset is longer if the external crystal oscillator is enabled by the Flash option bits; this configuration allows additional time for oscillator startup.

Table 9. Reset and Stop Mode Recovery Characteristics and Latency

| Reset Characteristics and Latency | | | |
|-----------------------------------|--|---------|--|
| Reset Type | Control Registers | eZ8 CPU | Reset Latency (Delay) |
| System Reset | Reset (as applicable) | Reset | 66 Internal Precision Oscillator Cycles |
| Stop Mode Recovery | Unaffected, except WDT_CTL and OSC_CTL registers | Reset | 66 Internal Precision Oscillator Cycles + IPO startup time |

During a System Reset or Stop Mode Recovery, the IPO requires 4 μ s to start up. Then the Z8 Encore! XP F0823 Series device is held in Reset for 66 cycles of the Internal Precision Oscillator. If the crystal oscillator is enabled in the Flash option bits, this reset period is increased to 5000 IPO cycles. When a reset occurs because of a low voltage condition or Power-On Reset, this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the reset period, the device remains in reset until the pin is deasserted.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address.

When the control registers are re-initialized by a system reset, the system clock after reset is always the IPO. The software must reconfigure the oscillator control block, such that the correct system clock source is enabled and selected.

Reset Sources

Table 10 lists the possible sources of a System Reset.

Table 10. Reset Sources and Resulting Reset Type

| Operating Mode | Reset Source | Special Conditions |
|----------------------|--|---|
| NORMAL or HALT modes | Power-On Reset/Voltage Brown-Out. | Reset delay begins after supply voltage exceeds POR level. |
| | Watchdog Timer time-out when configured for Reset. | None. |
| | RESET pin assertion. | All reset pulses less than three system clocks in width are ignored. |
| | OCD initiated Reset (OCDCTL[0] set to 1). | System Reset, except the OCD is unaffected by the reset. |
| STOP Mode | Power-On Reset/Voltage Brown-Out. | Reset delay begins after supply voltage exceeds POR level. |
| | RESET pin assertion. | All reset pulses less than the specified analog delay are ignored. See the Electrical Characteristics chapter on page 196 . |
| | DBG pin driven Low. | None. |

Power-On Reset

Each device in the Z8 Encore! XP F0823 Series contains an internal POR circuit. The POR circuit monitors the supply voltage and holds the device in the Reset state until the supply voltage reaches a safe operating level. After the supply voltage exceeds the POR voltage threshold (V_{POR}), the device is held in the Reset state until the POR Counter has timed out. If the crystal oscillator is enabled by the option bits, this time-out is longer.

After the Z8 Encore! XP F0823 Series device exits the POR state, the eZ8 CPU fetches the Reset vector. Following the POR, the POR status bit in Watchdog Timer Control (WDTCTL) Register is set to 1.

Figure 5 displays POR operation. For the POR threshold voltage (V_{POR}), see the [Electrical Characteristics](#) chapter on page 196.



Figure 5. Power-On Reset Operation

Voltage Brown-Out Reset

The devices in the Z8 Encore! XP F0823 Series provide low VBO protection. The VBO circuit senses when the supply voltage drops to an unsafe level (below the VBO threshold voltage) and forces the device into the Reset state. While the supply voltage remains below the POR voltage threshold (V_{POR}), the VBO block holds the device in the Reset.

After the supply voltage again exceeds the Power-On Reset voltage threshold, the device progresses through a full System Reset sequence, as described in the [Power-On Reset](#) section on page 23. Following POR, the POR status bit in the Reset Status (RSTSTAT) Register is set to 1. Figure 6 displays Voltage Brown-Out operation. For the VBO and POR threshold voltages (V_{VBO} and V_{POR}), see the [Electrical Characteristics](#) chapter on page 196.

The VBO circuit can be either enabled or disabled during STOP Mode. Operation during STOP Mode is set by the VBO_AO Flash Option bit. For information about configuring VBO_AO, see the [Flash Option Bits](#) chapter on page 146.



Figure 6. Voltage Brown-Out Reset Operation

The POR level is greater than the VBO level by the specified hysteresis value. This ensures that the device undergoes a POR after recovering from a VBO condition.

Watchdog Timer Reset

If the device is in NORMAL or STOP Mode, the Watchdog Timer can initiate a System Reset at time-out if the WDT_RES Flash Option Bit is programmed to 1. This is the unprogrammed state of the WDT_RES Flash Option Bit. If the bit is programmed to 0, it configures the Watchdog Timer to cause an interrupt, not a System Reset, at time-out.

The WDT status bit in the WDT Control Register is set to signify that the reset was initiated by the Watchdog Timer.

External Reset Input

The $\overline{\text{RESET}}$ pin has a Schmitt-Triggered input and an internal pull-up resistor. Once the $\overline{\text{RESET}}$ pin is asserted for a minimum of four system clock cycles, the device progresses through the System Reset sequence. Because of the possible asynchronicity of the system

clock and reset signals, the required reset duration can be as short as three clock periods and as long as four. A reset pulse three clock cycles in duration might trigger a reset; a pulse four cycles in duration always triggers a reset.

While the $\overline{\text{RESET}}$ input pin is asserted Low, the Z8 Encore! XP F0823 Series devices remain in the Reset state. If the $\overline{\text{RESET}}$ pin is held Low beyond the System Reset time-out, the device exits the Reset state on the system clock rising edge following $\overline{\text{RESET}}$ pin deassertion. Following a System Reset initiated by the external $\overline{\text{RESET}}$ pin, the EXT status bit in the WDT Control (WDTCTL) register is set to 1.

External Reset Indicator

During System Reset or when enabled by the GPIO logic (see [the Port A–C Control Registers section on page 42](#)), the $\overline{\text{RESET}}$ pin functions as an open-drain (active Low) reset mode indicator in addition to the input functionality. This reset output feature allows an Z8 Encore! XP F0823 Series device to reset other components to which it is connected, even if that reset is caused by internal sources such as POR, VBO, or WDT events.

After an internal reset event occurs, the internal circuitry begins driving the $\overline{\text{RESET}}$ pin Low. The $\overline{\text{RESET}}$ pin is held Low by the internal circuitry until the appropriate delay listed in Table 9 has elapsed.

On-Chip Debugger Initiated Reset

A POR is initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The OCD block is not reset but the rest of the chip goes through a normal system reset. The RST bit automatically clears during the System Reset. Following the System Reset, the POR bit in the Reset Status (RSTSTAT) Register is set.

Stop Mode Recovery

The device enters into STOP Mode when eZ8 CPU executes a STOP instruction. For more details about STOP Mode, see [the Low-Power Modes section on page 30](#). During Stop Mode Recovery, the CPU is held in reset for 66 IPO cycles if the crystal oscillator is disabled or 5000 cycles if it is enabled. The SMR delay also included the time required to start up the IPO.

Stop Mode Recovery does not affect on-chip registers other than the Watchdog Timer Control Register (WDTCTL) and the Oscillator Control Register (OSCCTL). After any Stop Mode Recovery, the IPO is enabled and selected as the system clock. If another system clock source is required or IPO disabling is required, the Stop Mode Recovery code must reconfigure the oscillator control block such that the correct system clock source is enabled and selected.

The eZ8 CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the STOP bit in the Watchdog Timer Control Register is set to 1. Table 11 lists the Stop Mode Recovery sources and resulting actions. The section following the table provides more detailed information about each of the Stop Mode Recovery sources.

Table 11. Stop Mode Recovery Sources and Resulting Action

| Operating Mode | Stop Mode Recovery Source | Action |
|----------------|---|--|
| STOP Mode | Watchdog Timer time-out when configured for Reset | Stop Mode Recovery |
| | Watchdog Timer time-out when configured for interrupt | Stop Mode Recovery followed by interrupt (if interrupts are enabled) |
| | Data transition on any GPIO port pin enabled as a Stop Mode Recovery source | Stop Mode Recovery |
| | Assertion of external RESET Pin | System Reset |
| | Debug Pin driven Low | System Reset |

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control Register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and Z8 Encore! XP F0823 Series device is configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery.

Note: The SMR pulses shorter than specified does not trigger a recovery. When this happens, the STOP bit in the Reset Status (RSTSTAT) Register is set to 1.

Caution: In STOP Mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the port transition only if the signal stays on the port pin through the end of the Stop Mode Recovery delay. As a result, short pulses on the port pin can initiate Stop Mode Recovery without being written to the Port Input Data Register or without initiating an interrupt (if enabled for that pin).

Stop Mode Recovery Using the External $\overline{\text{RESET}}$ Pin

When a Z8 Encore! XP F0823 Series device is in STOP Mode and the external $\overline{\text{RESET}}$ pin is driven Low, a system reset occurs. Because of a glitch filter operating on the $\overline{\text{RESET}}$ pin, the Low pulse must be greater than the minimum width specified, or it is ignored. For more details, see the [Electrical Characteristics](#) chapter on page 196.

Reset Register Definitions

The following sections define the Reset registers.

Reset Status Register

The Reset Status (RSTSTAT) Register is a read-only register that indicates the source of the most recent Reset event, indicates a Stop Mode Recovery event, and indicates a Watchdog Timer time-out. Reading this register resets the upper four bits to 0.

This register shares its address with the Watchdog Timer Control Register, which is write-only; see Table 12.

Table 12. Reset Status Register (RSTSTAT)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------------------------|------|-----|-----|----------|---|---|---|
| Field | POR | STOP | WDT | EXT | Reserved | | | |
| RESET | See descriptions in Table 13 | | | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |
| Address | FF0H | | | | | | | |

| Bit | Description |
|-------------|--|
| [7] POR | Power-On Reset Indicator If this bit is set to 1, a Power-On Reset event has occurred. This bit is reset to 0 if a WDT time-out or Stop Mode Recovery occurs. This bit is also reset to 0 when the register is read. For POR/Stop Mode Recover event values, please see Table 13. |
| [6] STOP | Stop Mode Recovery Indicator If this bit is set to 1, a Stop Mode Recovery is occurred. If the STOP and WDT bits are both set to 1, the Stop Mode Recovery occurred because of a WDT time-out. If the STOP bit is 1 and the WDT bit is 0, the Stop Mode Recovery was not caused by a WDT time-out. This bit is reset by a POR or a WDT time-out that occurred while not in STOP Mode. Reading this register also resets this bit. For POR/Stop Mode Recover event values, please see Table 13. |
| [5] WDT | Watchdog Timer Time-Out Indicator If this bit is set to 1, a WDT time-out has occurred. A POR resets this pin. A Stop Mode Recovery from a change in an input pin also resets this bit. Reading this register resets this bit; this read must occur before clearing the WDT interrupt. For POR/Stop Mode Recover event values, please see Table 13. |

| Bit | Description (Continued) |
|------------|---|
| [4] EXT | External Reset Indicator If this bit is set to 1, a Reset initiated by the external $\overline{\text{RESET}}$ pin occurred. A Power-On Reset or a Stop Mode Recovery from a change in an input pin resets this bit. Reading this register resets this bit. For POR/Stop Mode Recover event values, please see Table 13. |
| [3:0] | Reserved These bits are reserved and must be programmed to 0000 when read. |

Table 13. POR Indicator Values

| Reset or Stop Mode Recovery Event | POR | STOP | WDT | EXT |
|---|-----|------|-----|-----|
| Power-On Reset | 1 | 0 | 0 | 0 |
| Reset using $\overline{\text{RESET}}$ pin assertion | 0 | 0 | 0 | 1 |
| Reset using WDT time-out | 0 | 0 | 1 | 0 |
| Reset using the OCD (OCTCTL[1] set to 1) | 1 | 0 | 0 | 0 |
| Reset from STOP Mode using DBG Pin driven Low | 1 | 0 | 0 | 0 |
| Stop Mode Recovery using GPIO pin transition | 0 | 1 | 0 | 0 |
| Stop Mode Recovery using WDT time-out | 0 | 1 | 1 | 0 |

Low-Power Modes

Z8 Encore! XP F0823 Series products contain power-saving features. The highest level of power reduction is provided by the STOP Mode, in which nearly all device functions are powered down. The next lower level of power reduction is provided by the HALT Mode, in which the CPU is powered down.

Further power savings can be implemented by disabling individual peripheral blocks while in ACTIVE mode (defined as being in neither STOP nor HALT Mode).

STOP Mode

Executing the eZ8 CPU's Stop instruction places the device into STOP Mode, powering down all peripherals except the Voltage Brown-Out detector, and the Watchdog Timer. These two blocks may also be disabled for additional power savings. In STOP Mode, the operating characteristics are:

- Primary crystal oscillator and internal precision oscillator are stopped; X_{IN} and X_{OUT} (if previously enabled) are disabled, and PA0/PA1 revert to the states programmed by the GPIO registers
- System clock is stopped
- eZ8 CPU is stopped
- Program counter (PC) stops incrementing
- Watchdog Timer's internal RC oscillator continues to operate if enabled by the Oscillator Control Register
- If enabled, the Watchdog Timer logic continues to operate
- If enabled for operation in STOP Mode by the associated Flash Option Bit, the Voltage Brown-Out protection circuit continues to operate
- All other on-chip peripherals are idle

To minimize current in STOP Mode, all GPIO pins that are configured as digital inputs must be driven to one of the supply rails (V_{CC} or GND). Additionally, any GPIOs configured as outputs must also be driven to one of the supply rails. The device can be brought out of STOP Mode using Stop Mode Recovery. For more information about Stop Mode Recovery, see [the Reset and Stop Mode Recovery](#) chapter on page 21.

HALT Mode

Executing the eZ8 CPU's HALT instruction places the device into HALT Mode, which powers down the CPU but leaves all other peripherals active. In HALT Mode, the operating characteristics are:

- Primary oscillator is enabled and continues to operate
- System clock is enabled and continues to operate
- eZ8 CPU is stopped
- Program counter stops incrementing
- Watchdog Timer's internal RC oscillator continues to operate
- If enabled, the Watchdog Timer continues to operate
- All other on-chip peripherals continue to operate

The eZ8 CPU can be brought out of HALT Mode by any of the following operations:

- Interrupt
- Watchdog Timer time-out (interrupt or reset)
- Power-On Reset
- Voltage Brown-Out reset
- External $\overline{\text{RESET}}$ pin assertion

To minimize current in HALT Mode, all GPIO pins that are configured as inputs must be driven to one of the supply rails (V_{CC} or GND).

Peripheral-Level Power Control

In addition to the STOP and HALT modes, it is possible to disable each peripheral on each of the Z8 Encore! XP F0823 Series devices. Disabling a given peripheral minimizes its power consumption.

Power Control Register Definitions

The following sections describe the power control registers.

Power Control Register 0

Each bit of the following registers disables a peripheral block, either by gating its system clock input or by removing power from the block.

► **Note:** This register is only reset during a Power-On Reset sequence. Other System Reset events do not affect it.

Table 14. Power Control Register 0 (PWRCTL0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|----------|-----|-----|----------|-----|------|----------|
| Field | Reserved | Reserved | | VBO | Reserved | ADC | COMP | Reserved |
| RESET | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F80H | | | | | | | |

| Bit | Description |
|-------------|---|
| [7] | Reserved This bit is reserved and must be programmed to 1. |
| [6:5] | Reserved These bits are reserved and must be programmed to 00. |
| [4] VBO | Voltage Brown-Out Detector Disable This bit and the VBO_AO Flash option bit must both enable the VBO for the VBO to be active. 0 = VBO enabled. 1 = VBO disabled. |
| [3] | Reserved This bit is reserved and must be programmed to 0. |
| [2] ADC | Analog-to-Digital Converter Disable 0 = Analog-to-Digital Converter enabled. 1 = Analog-to-Digital Converter disabled. |
| [1] COMP | Comparator Disable 0 = Comparator is enabled. 1 = Comparator is disabled. |
| [0] | Reserved This bit is reserved and must be programmed to 0. |

General-Purpose Input/Output

Z8 Encore! XP F0823 Series products support a maximum of 24 port pins (Ports A–C) for general-purpose input/output (GPIO) operations. Each port contains control and data registers. The GPIO control registers determine data direction, open-drain, output drive current, programmable pull-ups, Stop Mode Recovery functionality, and alternate pin functions. Each port pin is individually programmable. In addition, the Port C pins are capable of direct LED drive at programmable drive strengths.

GPIO Port Availability By Device

Table 15 lists the port pins available with each device and package type.

Table 15. Port Availability by Device and Package Type

| Devices | Package | 10-Bit ADC | Port A | Port B | Port C | Total I/O |
|--|---------|------------|--------|--------|--------|-----------|
| Z8F0823SB, Z8F0823PB Z8F0423SB, Z8F0423PB Z8F0223SB, Z8F0223PB Z8F0123SB, Z8F0123PB | 8-pin | Yes | [5:0] | No | No | 6 |
| Z8F0813SB, Z8F0813PB Z8F0413SB, Z8F0413PB Z8F0213SB, Z8F0213PB Z8F0113SB, Z8F011vPB | 8-pin | No | [5:0] | No | No | 6 |
| Z8F0823PH, Z8F0823HH Z8F0423PH, Z8F0423HH Z8F0223PH, Z8F0223HH Z8F0123PH, Z8F0123HH | 20-pin | Yes | [7:0] | [3:0] | [3:0] | 16 |
| Z8F0813PH, Z8F0813HH Z8F0413PH, Z8F0413HH Z8F0213PH, Z8F0213HH Z8F0113PH, Z8F0113HH | 20-pin | No | [7:0] | [3:0] | [3:0] | 16 |
| Z8F0823PJ, Z8F0823SJ Z8F0423PJ, Z8F0423SJ Z8F0223PJ, Z8F0223SJ Z8F0123PJ, Z8F0123SJ | 28-pin | Yes | [7:0] | [5:0] | [7:0] | 22 |
| Z8F0813PJ, Z8F0813SJ Z8F0413PJ, Z8F0413SJ Z8F0213PJ, Z8F0213SJ Z8F0113PJ, Z8F0113SJ | 28-pin | No | [7:0] | [7:0] | [7:0] | 24 |

Architecture

Figure 7 displays a simplified block diagram of a GPIO port pin. In this figure, the ability to accommodate alternate functions and variable port current drive strength is not displayed.



Figure 7. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins are used for general-purpose I/O and access to on-chip peripheral functions such as the timers and serial communication devices. The port A–D Alternate Function subregisters configure these pins for either GPIO or alternate function operation. When a pin is configured for alternate function, control of the port pin direction (input/output) is passed from the Port A–D Data Direction registers to the alternate function assigned to this pin. Tables 16 and 17 list the alternate functions possible with each port pin for 8-pin and non-8-pin parts, respectively. The alternate function associated at a pin is defined through Alternate Function Sets subregisters AFS1 and AFS2.

The crystal oscillator functionality is not controlled by the GPIO block. When the crystal oscillator is enabled in the oscillator control block, the GPIO functionality of PA0 and PA1 is overridden. In that case, those pins function as input and output for the crystal oscillator.

PA0 and PA6 contain two different timer functions, a timer input and a complementary timer output. Both of these functions require the same GPIO configuration, the selection between the two is based on the timer mode. For more details, see the [Timers](#) chapter on page 69.

 **Caution:** For pins with multiple alternate functions, Zilog recommends writing to the AFS1 and AFS2 subregisters before enabling the alternate function via the AF Subregister. This prevents spurious transitions through unwanted alternate function modes.

Table 16. Port Alternate Function Mapping (8-Pin Parts)

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Select Register AFS1 | Alternate Function Select Register AFS2 |
|--------|-------------------|---------------------------|-----------------------------------|---|---|
| Port A | PA0 | T0IN | Timer 0 Input | AFS1[0]: 0 | AFS2[0]: 0 |
| | | Reserved | | AFS1[0]: 0 | AFS2[0]: 1 |
| | | Reserved | | AFS1[0]: 1 | AFS2[0]: 0 |
| | | T0OUT | Timer 0 Output Complement | AFS1[0]: 1 | AFS2[0]: 1 |
| | PA1 | T0OUT | Timer 0 Output | AFS1[1]: 0 | AFS2[1]: 0 |
| | | Reserved | | AFS1[1]: 0 | AFS2[1]: 1 |
| | | CLKIN | External Clock Input | AFS1[1]: 1 | AFS2[1]: 0 |
| | | Analog Functions* | ADC Analog Input/V _{REF} | AFS1[1]: 1 | AFS2[1]: 1 |
| | PA2 | DE0 | UART 0 Driver Enable | AFS1[2]: 0 | AFS2[2]: 0 |
| | | RESET | External Reset | AFS1[2]: 0 | AFS2[2]: 1 |
| | | T1OUT | Timer 1 Output | AFS1[2]: 1 | AFS2[2]: 0 |
| | | Reserved | | AFS1[2]: 1 | AFS2[2]: 1 |
| | PA3 | CTS0 | UART 0 Clear to Send | AFS1[3]: 0 | AFS2[3]: 0 |
| | | COUT | Comparator Output | AFS1[3]: 0 | AFS2[3]: 1 |
| | | T1IN | Timer 1 Input | AFS1[3]: 1 | AFS2[3]: 0 |
| | | Analog Functions* | ADC Analog Input | AFS1[3]: 1 | AFS2[3]: 1 |
| PA4 | RXD0 | UART 0 Receive Data | AFS1[4]: 0 | AFS2[4]: 0 | |
| | Reserved | | AFS1[4]: 0 | AFS2[4]: 1 | |
| | Reserved | | AFS1[4]: 1 | AFS2[4]: 0 | |
| | Analog Functions* | ADC/Comparator Input (N) | AFS1[4]: 1 | AFS2[4]: 1 | |
| PA5 | TXD0 | UART 0 Transmit Data | AFS1[5]: 0 | AFS2[5]: 0 | |
| | T1OUT | Timer 1 Output Complement | AFS1[5]: 0 | AFS2[5]: 1 | |
| | Reserved | | AFS1[5]: 1 | AFS2[5]: 0 | |
| | Analog Functions* | ADC/Comparator Input (P) | AFS1[5]: 1 | AFS2[5]: 1 | |

Note: *Analog Functions include ADC inputs, ADC reference and comparator inputs. Also, alternate function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must be enabled.

Table 17. Port Alternate Function Mapping (Non 8-Pin Parts)

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Set Register AFS1 |
|---------------------|----------|-------------------------|---|--------------------------------------|
| Port A ¹ | PA0 | T0IN/T0OUT | Timer 0 Input/Timer 0 Output Complement | N/A |
| | | Reserved | | |
| | PA1 | T0OUT | Timer 0 Output | |
| | | Reserved | | |
| | PA2 | DE0 | UART 0 Driver Enable | |
| | | Reserved | | |
| | PA3 | CTS0 | UART 0 Clear to Send | |
| | | Reserved | | |
| | PA4 | RXD0/IRRX0 | UART 0 / IrDA 0 Receive Data | |
| | | Reserved | | |
| | PA5 | TXD0/IRTX0 | UART 0 / IrDA 0 Transmit Data | |
| | | Reserved | | |
| | PA6 | T1IN/T1OUT ² | Timer 1 Input/Timer 1 Output Complement | |
| | | Reserved | | |
| PA7 | T1OUT | Timer 1 Output | | |
| | Reserved | | | |

Notes:

1. Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 automatically enables the associated alternate function.
2. Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in the [Timer Pin Signal Operation](#) section on page 83.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, alternate function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must also be enabled.
4. V_{REF} is available on PB5 in 28-pin products only.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, Alternate Function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must also be enabled.
6. V_{REF} is available on PC2 in 20-pin parts only.

Table 17. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Set Register AFS1 |
|---------------------|------|-------------------------------|--------------------------------|--------------------------------------|
| Port B ³ | PB03 | Reserved | | AFS1[0]: 0 |
| | | ANA0 | ADC Analog Input | AFS1[0]: 1 |
| | PB1 | Reserved | | AFS1[1]: 0 |
| | | ANA1 | ADC Analog Input | AFS1[1]: 1 |
| | PB2 | Reserved | | AFS1[2]: 0 |
| | | ANA2 | ADC Analog Input | AFS1[2]: 1 |
| | PB3 | CLKIN | External Clock Input | AFS1[3]: 0 |
| | | ANA3 | ADC Analog Input | AFS1[3]: 1 |
| | PB4 | Reserved | | AFS1[4]: 0 |
| | | ANA7 | ADC Analog Input | AFS1[4]: 1 |
| | PB5 | Reserved | | AFS1[5]: 0 |
| | | V _{REF} ⁴ | ADC Voltage Reference | AFS1[5]: 1 |
| | PB6 | Reserved | | AFS1[6]: 0 |
| | | Reserved | | AFS1[6]: 1 |
| | PB7 | Reserved | | AFS1[7]: 0 |
| | | Reserved | | AFS1[7]: 1 |

Notes:

1. Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 automatically enables the associated alternate function.
2. Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in the [Timer Pin Signal Operation](#) section on page 83.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, alternate function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must also be enabled.
4. V_{REF} is available on PB5 in 28-pin products only.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, Alternate Function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must also be enabled.
6. V_{REF} is available on PC2 in 20-pin parts only.

Table 17. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

| Port | Pin | Mnemonic | Alternate Function Description | Alternate Function Set Register AFS1 |
|---------------------|----------|------------------------------------|---|--------------------------------------|
| Port C ⁴ | PC0 | Reserved | | AFS1[0]: 0 |
| | | ANA4/CINP | ADC or Comparator Input | AFS1[0]: 1 |
| | PC1 | Reserved | | AFS1[1]: 0 |
| | | ANA5/CINN | ADC or Comparator Input | AFS1[1]: 1 |
| | PC2 | Reserved | | AFS1[2]: 0 |
| | | ANA6/V _{REF} ⁶ | ADC Analog Input or ADC Voltage Reference | AFS1[2]: 1 |
| | PC3 | COUT | Comparator Output | AFS1[3]: 0 |
| | | Reserved | | AFS1[3]: 1 |
| | PC4 | Reserved | | AFS1[4]: 0 |
| | | | AFS1[4]: 1 | |
| | PC5 | Reserved | | AFS1[5]: 0 |
| | | | AFS1[5]: 1 | |
| | PC6 | Reserved | | AFS1[6]: 0 |
| | | | AFS1[6]: 1 | |
| PC7 | Reserved | | AFS1[7]: 0 | |
| | | AFS1[7]: 1 | | |

Notes:

1. Because there is only a single alternate function for each Port A pin, the Alternate Function Set registers are not implemented for Port A. Enabling alternate function selections as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 automatically enables the associated alternate function.
2. Whether PA0/PA6 take on the timer input or timer output complement function depends on the timer configuration as described in the [Timer Pin Signal Operation](#) section on page 83.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, alternate function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must also be enabled.
4. V_{REF} is available on PB5 in 28-pin products only.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set register AFS2 is implemented but not used to select the function. Also, Alternate Function selection as described in the [Port A–C Alternate Function Subregisters](#) section on page 43 must also be enabled.
6. V_{REF} is available on PC2 in 20-pin parts only.

Direct LED Drive

The Port C pins provide a current sinked output capable of driving an LED without requiring an external resistor. The output sinks current at programmable levels of 3 mA, 7 mA, 13 mA, and 20 mA. This mode is enabled through the LED control registers. The LED Drive Enable (LEDEN) register turns on the drivers. The LED Drive Level (LEDLVLH and LEDLVLL) registers select the sink current.

For correct operation, the LED anode must be connected to V_{DD} and the cathode must be connected to the GPIO pin. Using all Port C pins in LED Drive Mode with maximum current can result in excessive total current. For the maximum total current for the applicable package, see the [Electrical Characteristics](#) chapter on page 196.

Shared Reset Pin

On the 8-pin product versions, the reset pin is shared with PA2, but the pin is not limited to output-only when in GPIO Mode.



Caution: If PA2 on the 8-pin product is reconfigured as an input, ensure that no external stimulus drives the pin Low during any reset sequence. Because PA2 returns to its RESET alternate function during system resets, driving it Low holds the chip in a reset state until the pin is released.

Shared Debug Pin

On the 8-pin version of this device only, the Debug pin shares function with the PA0 GPIO pin. This pin performs as a general purpose input pin on power-up, but the debug logic monitors this pin during the reset sequence to determine if the unlock sequence occurs. If the unlock sequence is present, the debug function is unlocked and the pin no longer functions as a GPIO pin. If it is not present, the debug feature is disabled until/unless another reset event occurs. For more details, see the [On-Chip Debugger](#) chapter on page 156.

Crystal Oscillator Override

For systems using a crystal oscillator, PA0 and PA1 are used to connect the crystal. When the crystal oscillator is enabled (see the [Oscillator Control Register Definitions](#) section on page 171), the GPIO settings are overridden and PA0 and PA1 are disabled.

5V Tolerance

All six I/O pins on the 8-pin devices are 5 V-tolerant, unless the programmable pull-ups are enabled. If the pull-ups are enabled and inputs higher than V_{DD} are applied to these parts, excessive current flows through those pull-up devices and can damage the chip.



Note: In the 20- and 28-pin versions of this device, any pin which shares functionality with an ADC, crystal or comparator port is not 5 V-tolerant, including PA[1:0], PB[5:0], and

PC[2:0]. All other signal pins are 5 V-tolerant, and can safely handle inputs higher than V_{DD} even with the pull-ups enabled.

External Clock Setup

For systems using an external TTL drive, PB3 is the clock source for 20- and 28-pin devices. In this case, configure PB3 for alternate function CLKIN. Write the Oscillator Control Register (see the [Oscillator Control Register Definitions](#) section on page 171) such that the external oscillator is selected as the system clock. For 8-pin devices, use PA1 instead of PB3.

GPIO Interrupts

Many of the GPIO port pins are used as interrupt sources. Some port pins are configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port pin interrupt sources generate an interrupt when any edge occurs (both rising and falling). For more information about interrupts using the GPIO pins, see the [Interrupt Controller](#) chapter on page 54.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data, and output data. Table 18 lists these port registers. Use the Port A–D Address and Control registers together to provide access to subregisters for port configuration and control.

Table 18. GPIO Port Registers and Subregisters

| Port Register Mnemonic | Port Register Name |
|---------------------------|--|
| PxADDR | Port A–C Address Register (Selects subregisters). |
| PxCTL | Port A–C Control Register (Provides access to subregisters). |
| PxIN | Port A–C Input Data Register. |
| PxOUT | Port A–C Output Data Register. |
| Port Subregister Mnemonic | Port Register Name |
| PxDD | Data Direction. |
| PxAF | Alternate Function. |
| PxOC | Output Control (Open-Drain). |

Table 18. GPIO Port Registers and Subregisters (Continued)

| Port Register Mnemonic | Port Register Name |
|------------------------|-----------------------------------|
| PxHDE | High Drive Enable. |
| PxSMRE | Stop Mode Recovery Source Enable. |
| PxPUE | Pull-up Enable. |
| PxAFS1 | Alternate Function Set 1. |
| PxAFS2 | Alternate Function Set 2. |

Port A–C Address Registers

The Port A–C Address registers select the GPIO port functionality accessible through the Port A–C Control registers. The Port A–C Address and Control registers combine to provide access to all GPIO port controls (Table 19).

Table 19. Port A–C GPIO Address Registers (PxADDR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | PADDR[7:0] | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FD0H, FD4H, FD8H | | | | | | | |

| Bit | Description |
|----------------|---|
| [7:0] PADDR | Port Address The Port Address selects one of the subregisters accessible through the Port Control Register. See Table 20 for each subregister function. |

Table 20. PADDR[7:0] Subregister Functions

| PADDR[7:0] | Port Control Subregister Accessible Using the Port A–C Control Registers |
|------------|--|
| 00H | No function. Provides some protection against accidental Port reconfiguration. |
| 01H | Data Direction. |
| 02H | Alternate Function. |
| 03H | Output Control (Open-Drain). |
| 04H | High Drive Enable. |

Table 20. PADDR[7:0] Subregister Functions

| PADDR[7:0] | Port Control Subregister Accessible Using the Port A–C Control Registers |
|------------|--|
| 05H | Stop Mode Recovery Source Enable. |
| 06H | Pull-up Enable. |
| 07H | Alternate Function Set 1. |
| 08H | Alternate Function Set 2. |
| 09H–FFH | No function. |

Port A–C Control Registers

The Port A–C Control registers set the GPIO port operation. The value in the corresponding Port A–C Address Register determines which subregister is read from or written to by a Port A–C Control Register transaction; see Table 21.

Table 21. Port A–C Control Registers (PxCTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | PCTL | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FD1H, FD5H, FD9H | | | | | | | |

| Bit | Description |
|-------|---|
| [7:0] | Port Control |
| PCTL | The Port Control Register provides access to all subregisters that configure the GPIO Port operation. |

Port A–C Data Direction Subregisters

The Port A–C Data Direction Subregister is accessed through the Port A–C Control Register by writing 01H to the Port A–C Address Register; see Table 22.

Table 22. Port A–C Data Direction Subregisters (PxDD)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--|-----|-----|-----|-----|-----|-----|-----|
| Field | DD7 | DD6 | DD5 | DD4 | DD3 | DD2 | DD1 | DD0 |
| RESET | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 01H in Port A–C Address Register, accessible through the Port A–C Control Register. | | | | | | | |

| Bit | Description |
|--------------|--|
| [7:0] DDx | <p>Data Direction</p> <p>These bits control the direction of the associated port pin. Port Alternate Function operation overrides the Data Direction register setting.</p> <p>0 = Output. Data in the Port A–C Output Data Register is driven onto the port pin.</p> <p>1 = Input. The port pin is sampled and the value written into the Port A–C Input Data Register. The output driver is tristated.</p> |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Alternate Function Subregisters

The Port A–C Alternate Function Subregister (Table 23) is accessed through the Port A–C Control Register by writing 02H to the Port A–C Address Register. The Port A–C Alternate Function subregisters enable the alternate function selection on pins. If disabled, pins function as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the [Port A–C Alternate Function Set 1 Subregisters](#) section on page 48 and the [Port A–C Alternate Function Set 2 Subregisters](#) section on page 49. See the [GPIO Alternate Functions](#) section on page 34 to determine the alternate function associated with each port pin.



Caution: Do not enable alternate functions for GPIO port pins for which there is no associated alternate function. Failure to follow this guideline can result in unpredictable operation.

Table 23. Port A–C Alternate Function Subregisters (PxAF)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|-----|-----|-----|-----|-----|-----|-----|
| Field | AF7 | AF6 | AF5 | AF4 | AF3 | AF2 | AF1 | AF0 |
| RESET | 00H (Ports A–C); 04H (Port A of 8-pin device) | | | | | | | |
| R/W | R/W | | | | | | | |
| Address | If 02H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|--------------|---|
| [7:0] AFx | Port Alternate Function enabled 0 = The port pin is in NORMAL Mode and the DDx bit in the Port A–C Data Direction Subregister determines the direction of the pin. 1 = The alternate function selected through Alternate Function Set subregisters is enabled. Port pin operation is controlled by the alternate function. |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Output Control Subregisters

The Port A–C Output Control Subregister (Table 24) is accessed through the Port A–C Control Register by writing 03H to the Port A–C Address Register. Setting the bits in the Port A–C Output Control subregisters to 1 configures the specified port pins for open-drain operation. These subregisters affect the pins directly and, as a result, alternate functions are also affected.

Table 24. Port A–C Output Control Subregisters (PxOC)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|------|------|------|------|------|------|------|
| Field | POC7 | POC6 | POC5 | POC4 | POC3 | POC2 | POC1 | POC0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 03H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|---------------|---|
| [7:0] POCx | Port Output Control These bits function independently of the alternate function bit and always disable the drains if set to 1. 0 = The drains are enabled for any output mode (unless overridden by the alternate function). 1 = The drain of the associated pin is disabled (open-drain mode). |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C High Drive Enable Subregisters

The Port A–C High Drive Enable Subregister (Table 25) is accessed through the Port A–C Control Register by writing 04H to the Port A–C Address Register. Setting the bits in the Port A–C High Drive Enable subregisters to 1 configures the specified port pins for high-current output drive operation. The Port A–C High Drive Enable Subregister affects the pins directly and, as a result, alternate functions are also affected.

Table 25. Port A–C High Drive Enable Subregisters (PHDE_x)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|-------|-------|-------|-------|-------|-------|-------|
| Field | PHDE7 | PHDE6 | PHDE5 | PHDE4 | PHDE3 | PHDE2 | PHDE1 | PHDE0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 04H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|-------------------|--|
| [7:0] | Port High Drive Enabled. |
| PHDE _x | 0 = The Port pin is configured for standard output current drive. 1 = The Port pin is configured for high output current drive. |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Stop Mode Recovery Source Enable Subregisters

The Port A–C Stop Mode Recovery Source Enable Subregister (Table 26) is accessed through the Port A–C Control Register by writing 05H to the Port A–C Address Register. Setting the bits in the Port A–C Stop Mode Recovery Source Enable subregisters to 1 configures the specified Port pins as a Stop Mode Recovery source. During STOP Mode, any logic transition on a Port pin enabled as a Stop Mode Recovery source initiates Stop Mode Recovery.

Table 26. Port A–C Stop Mode Recovery Source Enable Subregisters (PSMREx)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|--------|--------|--------|--------|--------|--------|--------|
| Field | PSMRE7 | PSMRE6 | PSMRE5 | PSMRE4 | PSMRE3 | PSMRE2 | PSMRE1 | PSMRE0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 05H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|--------|---|
| [7:0] | Port Stop Mode Recovery Source Enabled. |
| PSMREx | 0 = The Port pin is not configured as a Stop Mode Recovery source. Transitions on this pin during STOP Mode do not initiate Stop Mode Recovery. 1 = The Port pin is configured as a Stop Mode Recovery source. Any logic transition on this pin during STOP Mode initiates Stop Mode Recovery. |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Pull-up Enable Subregisters

The Port A–C Pull-up Enable Subregister (Table 27) is accessed through the Port A–C Control Register by writing 06H to the Port A–C Address Register. Setting the bits in the Port A–C Pull-up Enable subregisters enables a weak internal resistive pull-up on the specified Port pins.

Table 27. Port A–C Pull-Up Enable Subregisters (PPUE_x)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|-------|-------|-------|-------|-------|-------|-------|
| Field | PPUE7 | PPUE6 | PPUE5 | PPUE4 | PPUE3 | PPUE2 | PPUE1 | PPUE0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 06H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|----------------------------|--|
| [7:0] PPUE _x | Port Pull-up Enabled 0 = The weak pull-up on the Port pin is disabled. 1 = The weak pull-up on the Port pin is enabled. |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Alternate Function Set 1 Subregisters

The Port A–C Alternate Function Set1 Subregister (Table 28) is accessed through the Port A–C Control Register by writing 07H to the Port A–C Address Register. The Alternate Function Set 1 subregisters selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register are defined in “GPIO Alternate Functions” on page 34.

► **Note:** Alternate function selection on port pins must also be enabled as described in the [Port A–C Alternate Function Subregisters](#) section on page 43.

Table 28. Port A–C Alternate Function Set 1 Subregisters (PAFS1x)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|--------|--------|--------|--------|--------|--------|--------|
| Field | PAFS17 | PAFS16 | PAFS15 | PAFS14 | PAFS13 | PAFS12 | PAFS11 | PAFS10 |
| RESET | 00H (all ports of 20/28 pin devices); 04H (Port A of 8-pin device) | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 07H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|--------|--|
| [7:0] | Port Alternate Function Set to 1 |
| PAFS1x | 0 = Port Alternate Function selected as defined in Table 15 (see the GPIO Alternate Functions section on page 34). 1 = Port Alternate Function selected as defined in Table 15 (see the GPIO Alternate Functions section on page 34). |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Alternate Function Set 2 Subregisters

The Port A–C Alternate Function Set 2 Subregister (Table 29) is accessed through the Port A–C Control Register by writing 08H to the Port A–C Address Register. The Alternate Function Set 2 subregisters selects the alternate function available at a port pin. Alternate Functions selected by setting or clearing bits of this register is defined in Table 15 in the section the [GPIO Alternate Functions](#) section on page 34.

Table 29. Port A–C Alternate Function Set 2 Subregisters (PxAFS2)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|--------|--------|--------|--------|--------|--------|--------|
| Field | PAFS27 | PAFS26 | PAFS25 | PAFS24 | PAFS23 | PAFS22 | PAFS21 | PAFS20 |
| RESET | 00H (all ports of 20/28 pin devices); 04H (Port A of 8-pin device) | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | If 08H in Port A–C Address Register, accessible through the Port A–C Control Register | | | | | | | |

| Bit | Description |
|--------|--|
| [7:0] | Port Alternate Function Set 2 |
| PAFS2x | 0 = Port Alternate Function selected as defined in Table 15 on page 33; also see the GPIO Alternate Functions section on page 34). 1 = Port Alternate Function selected as defined in Table 15. |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Input Data Registers

Reading from the Port A–C Input Data registers (Table 30) returns the sampled values from the corresponding port pins. The Port A–C Input Data registers are read-only. The value returned for any unused ports is 0. Unused ports include those missing on the 8- and 28-pin packages, as well as those missing on the ADC-enabled 28-pin packages.

Table 30. Port A–C Input Data Registers (PxIN)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------------|------|------|------|------|------|------|------|
| Field | PIN7 | PIN6 | PIN5 | PIN4 | PIN3 | PIN2 | PIN1 | PIN0 |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R | R | R | R | R | R | R | R |
| Address | FD2H, FD6H, FDAH | | | | | | | |

| Bit | Description |
|---------------|--|
| [7:0] PxIN | Port Input Data Sampled data from the corresponding port pin input. 0 = Input data is logical 0 (Low). 1 = Input data is logical 1 (High). |

Note: x indicates the specific GPIO port pin number (7–0).

Port A–C Output Data Register

The Port A–C Output Data Register (Table 31) controls the output data to the pins.

Table 31. Port A–C Output Data Register (PxOUT)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------------|-------|-------|-------|-------|-------|-------|-------|
| Field | POUT7 | POUT6 | POUT5 | POUT4 | POUT3 | POUT2 | POUT1 | POUT0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FD3H, FD7H, FDBH | | | | | | | |

| Bit | Description |
|----------------|--|
| [7:0] PxOUT | <p>Port Output Data</p> <p>These bits contain the data to be driven to the port pins. The values are only driven if the corresponding pin is configured as an output and the pin is not configured for alternate function operation.</p> <p>0 = Drive a logical 0 (Low).</p> <p>1 = Drive a logical 1 (High). High value is not driven if the drain has been disabled by setting the corresponding Port Output Control Register bit to 1.</p> |

Note: x indicates the specific GPIO port pin number (7–0).

LED Drive Enable Register

The LED Drive Enable Register, shown in Table 32, activates the controlled current drive. The Alternate Function Register has no control over the LED function; therefore, setting the Alternate Function Register to select the LED function is not required. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Table 32. LED Drive Enable (LEDEN)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | LEDEN[7:0] | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F82H | | | | | | | |

| Bit | Description |
|----------------|---|
| [7:0] LEDEN | <p>LED Drive Enable</p> <p>These bits determine which Port C pins are connected to an internal current sink.</p> <p>0 = Tristate the Port C pin.</p> <p>1 = Connect controlled current sink to the Port C pin.</p> |

LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 33). These two bits select between four programmable drive levels. Each pin is individually programmable.

Table 33. LED Drive Level High Register (LEDLVLH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------------|-----|-----|-----|-----|-----|-----|-----|
| Field | LEDLVLH[7:0] | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F83H | | | | | | | |

| Bit | Description |
|---------|--|
| [7:0] | LED Level High Bit |
| LEDLVLH | {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA. |

LED Drive Level Low Register

The LED Drive Level registers contain two control bits for each Port C pin (Table 34). These two bits select between four programmable drive levels. Each pin is individually programmable.

Table 34. LED Drive Level Low Register (LEDLVLL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------------|-----|-----|-----|-----|-----|-----|-----|
| Field | LEDLVLL[7:0] | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F84H | | | | | | | |

| Bit | Description |
|---------|--|
| [7:0] | LED Level High Bit |
| LEDLVLL | {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA. 01 = 7mA. 10 = 13mA. 11 = 20mA. |

Interrupt Controller

The interrupt controller on the Z8 Encore! XP F0823 Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 unique interrupt vectors
 - 12 GPIO port pin interrupt sources (two are shared)
 - 8 on-chip peripheral interrupt sources (two are shared)
- Flexible GPIO interrupts
 - Eight selectable rising and falling edge GPIO interrupts
 - Four dual-edge interrupts
- Three levels of individually programmable interrupt priority
- Watchdog Timer can be configured to generate an interrupt

Interrupt requests (IRQs) allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start an interrupt service routine (ISR). Usually this interrupt service routine is involved with the exchange of data, status information, or control information between the CPU and the interrupting peripheral. When the service routine is completed, the CPU returns to the operation from which it was interrupted.

The eZ8 CPU supports both vectored and polled interrupt handling. For polled interrupts, the interrupt controller has no effect on operation. For more information about interrupt servicing by the eZ8 CPU, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#) available for download at www.zilog.com.

Interrupt Vector Listing

Table 35 lists all of the interrupts available in order of priority. The interrupt vector is stored with the most-significant byte (MSB) at the even Program Memory address and the least-significant byte (LSB) at the following odd Program Memory address.

► **Note:** Some port interrupts are not available on the 8- and 20-pin packages. The ADC interrupt is unavailable on devices not containing an ADC.

Table 35. Trap and Interrupt Vectors in Order of Priority

| Priority | Program Memory Vector Address | Interrupt or Trap Source | |
|----------|-------------------------------|--|----------|
| Highest | 0002H | Reset (not an interrupt) | |
| | 0004H | Watchdog Timer (see the Watchdog Timer section on page 91) | |
| | 003AH | Primary Oscillator Fail Trap (not an interrupt) | |
| | 003CH | Watchdog Timer Oscillator Fail Trap (not an interrupt) | |
| | 0006H | Illegal Instruction Trap (not an interrupt) | |
| | 0008H | Reserved | |
| | 000AH | Timer 1 | |
| | 000CH | Timer 0 | |
| | 000EH | UART 0 receiver | |
| | 0010H | UART 0 transmitter | |
| | 0012H | Reserved | |
| | 0014H | Reserved | |
| | 0016H | ADC | |
| | 0018H | Port A Pin 7, selectable rising or falling input edge | |
| | 001AH | Port A Pin 6, selectable rising or falling input edge or Comparator Output | |
| | 001CH | Port A Pin 5, selectable rising or falling input edge | |
| | 001EH | Port A Pin 4, selectable rising or falling input edge | |
| | 0020H | Port A Pin 3 or Port D Pin 3, selectable rising or falling input edge | |
| | 0022H | Port A Pin 2 or Port D Pin 2, selectable rising or falling input edge | |
| | 0024H | Port A Pin 1, selectable rising or falling input edge | |
| | 0026H | Port A Pin 0, selectable rising or falling input edge | |
| | 0028H | Reserved | |
| | 002AH | Reserved | |
| | 002CH | Reserved | |
| | 002EH | Reserved | |
| | 0030H | Port C Pin 3, both input edges | |
| | 0032H | Port C Pin 2, both input edges | |
| | 0034H | Port C Pin 1, both input edges | |
| | 0036H | Port C Pin 0, both input edges | |
| | Lowest | 0038H | Reserved |

Architecture

Figure 8 displays the interrupt controller block diagram.



Figure 8. Interrupt Controller Block Diagram

Operation

This section describes the operational aspects of the following functions.

[Master Interrupt Enable](#): see page 56

[Interrupt Vectors and Priority](#): see page 57

[Interrupt Assertion](#): see page 57

[Software Interrupt Assertion](#): see page 58

[Watchdog Timer Interrupt Assertion](#): see page 58

Master Interrupt Enable

The master interrupt enable bit (IRQE) in the Interrupt Control Register globally enables and disables interrupts.

Interrupts are globally enabled by any of the following actions:

- Execution of an Enable Interrupt (EI) instruction

- Execution of an Return from Interrupt (IRET) instruction
- Writing a 1 to the IRQE bit in the Interrupt Control Register

Interrupts are globally disabled by any of the following actions:

- Execution of a Disable Interrupt (DI) instruction
- eZ8 CPU acknowledgement of an interrupt service request from the interrupt controller
- Writing a 0 to the IRQE bit in the Interrupt Control Register
- Reset
- Execution of a Trap instruction
- Illegal Instruction Trap
- Primary Oscillator Fail Trap
- Watchdog Timer Oscillator Fail Trap

Interrupt Vectors and Priority

The interrupt controller supports three levels of interrupt priority. Level 3 is the highest priority, Level 2 is the second highest priority, and Level 1 is the lowest priority. If all interrupts are enabled with identical interrupt priority (for example, all as Level 2 interrupts), the interrupt priority is assigned from highest to lowest as specified in [Table 35](#) on page 55. Level 3 interrupts are always assigned higher priority than Level 2 interrupts which, in turn, always are assigned higher priority than Level 1 interrupts. Within each interrupt priority level (Level 1, Level 2 or Level 3), priority is assigned as specified in Table 35. Reset, Watchdog Timer interrupt (if enabled), Primary Oscillator Fail Trap, Watchdog Timer Oscillator Fail Trap, and Illegal Instruction Trap always have highest (Level 3) priority.

Interrupt Assertion

Interrupt sources assert their interrupt requests for only a single system clock period (single pulse). When the interrupt request is acknowledged by the eZ8 CPU, the corresponding bit in the Interrupt Request register is cleared until the next interrupt occurs. Writing a 0 to the corresponding bit in the Interrupt Request register likewise clears the interrupt request.



Caution: Zilog recommends not using a coding style that clears bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 1, which follows.

Example 1. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
AND r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 2 to clear bits in the Interrupt Request 0 Register:

Example 2. A good coding style that avoids lost interrupt requests:

```
ANDX IRQ0, MASK
```

Software Interrupt Assertion

Program code generates interrupts directly. Writing a 1 to the correct bit in the Interrupt Request register triggers an interrupt (assuming that interrupt is enabled). When the interrupt request is acknowledged by the eZ8 CPU, the bit in the Interrupt Request register is automatically cleared to 0.



Caution: Zilog recommends not using a coding style to generate software interrupts by setting bits in the Interrupt Request registers. All incoming interrupts received between execution of the first LDX command and the final LDX command are lost. See Example 3, which follows.

Example 3. A poor coding style that can result in lost interrupt requests:

```
LDX r0, IRQ0
OR r0, MASK
LDX IRQ0, r0
```

To avoid missing interrupts, use the coding style in Example 4 to set bits in the Interrupt Request registers:

Example 4. A good coding style that avoids lost interrupt requests:

```
ORX IRQ0, MASK
```

Watchdog Timer Interrupt Assertion

The Watchdog Timer interrupt behavior is different from interrupts generated by other sources. The Watchdog Timer continues to assert an interrupt as long as the timeout condition continues. As it operates on a different (and usually slower) clock domain than the rest of the device, the Watchdog Timer continues to assert this interrupt for many system clocks until the counter rolls over.



Caution: To avoid retriggerings of the Watchdog Timer interrupt after exiting the associated interrupt service routine, Zilog recommends that the service routine continues to read from the RSTSTAT register until the WDT bit is cleared as shown in the following example.

```
CLEARWDT:
    LDX r0, RSTSTAT ; read reset status register to clear wdt bit
    BTJNZ 5, r0, CLEARWDT ; loop until bit is cleared
```

Interrupt Control Register Definitions

For all interrupts other than the Watchdog Timer interrupt, the Primary Oscillator Fail Trap, and the Watchdog Timer Oscillator Fail Trap, the interrupt control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) register (Table 36) stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU reads the Interrupt Request 0 register to determine if any interrupt requests are pending.

Table 36. Interrupt Request 0 Register (IRQ0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-----|-----|-------|-------|----------|---|------|
| Field | Reserved | T1I | T0I | U0RXI | U0TXI | Reserved | | ADCI |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Address | FC0H | | | | | | | |

| Bit | Description |
|------------|--|
| [7] | Reserved This bit is reserved and must be programmed to 0. |
| [6] T1I | Timer 1 Interrupt Request 0 = No interrupt request is pending for Timer 1. 1 = An interrupt request from Timer 1 is awaiting service. |
| [5] T0I | Timer 0 Interrupt Request 0 = No interrupt request is pending for Timer 0. 1 = An interrupt request from Timer 0 is awaiting service. |

| Bit | Description (Continued) |
|--------------|---|
| [4] U0RXI | UART 0 Receiver Interrupt Request 0 = No interrupt request is pending for the UART 0 receiver. 1 = An interrupt request from the UART 0 receiver is awaiting service. |
| [3] U0TXI | UART 0 Transmitter Interrupt Request 0 = No interrupt request is pending for the UART 0 transmitter. 1 = An interrupt request from the UART 0 transmitter is awaiting service. |
| [2:1] | Reserved These bits are reserved and must be programmed to 00. |
| [0] ADC1 | ADC Interrupt Request 0 = No interrupt request is pending for the ADC. 1 = An interrupt request from the ADC is awaiting service. |

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) register (Table 37) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ1 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU reads the Interrupt Request 1 Register to determine if any interrupt requests are pending.

Table 37. Interrupt Request 1 Register (IRQ1)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------|-------|------|------|------|------|------|------|
| Field | PA7VI | PA6CI | PA5I | PA4I | PA3I | PA2I | PA1I | PA0I |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FC3H | | | | | | | |

| Bit | Description |
|---------------|--|
| [7] PA7VI | Port A7 Interrupt Request 0 = No interrupt request is pending for GPIO Port A. 1 = An interrupt request from GPIO Port A. |
| [6] PA6CI | Port A6 or Comparator Interrupt Request 0 = No interrupt request is pending for GPIO Port A or Comparator. 1 = An interrupt request from GPIO Port A or Comparator. |
| [5:0] PAxI | Port A Pin x Interrupt Request 0 = No interrupt request is pending for GPIO Port A pin x. 1 = An interrupt request from GPIO Port A pin x is awaiting service. |

Note: x indicates the specific GPIO Port pin number (0–5).

Interrupt Request 2 Register

The Interrupt Request 2 (IRQ2) register (Table 38) stores interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ2 Register becomes 1. If interrupts are globally enabled (vectored interrupts), the interrupt controller passes an interrupt request to the eZ8 CPU. If interrupts are globally disabled (polled interrupts), the eZ8 CPU can read the Interrupt Request 2 Register to determine if any interrupt requests are pending.

Table 38. Interrupt Request 2 Register (IRQ2)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-----|-----|-----|------|------|------|------|
| Field | Reserved | | | | PC3I | PC2I | PC1I | PC0I |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FC6H | | | | | | | |

| Bit | Description |
|---------------|---|
| [7:4] | Reserved These bits are reserved and must be programmed to 0000. |
| [3:0] PCxI | Port C Pin x Interrupt Request 0 = No interrupt request is pending for GPIO Port C pin x. 1 = An interrupt request from GPIO Port C pin x is awaiting service. |

Note: x indicates the specific GPIO Port C pin number (3–0).

IRQ0 Enable High and Low Bit Registers

Table 39 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers (Table 40 and Table 41) form a priority-encoded enabling for interrupts in the Interrupt Request 0 Register. Priority is generated by setting bits in each register.

Table 39. IRQ0 Enable and Priority Encoding

| IRQ0ENH[x] | IRQ0ENL[x] | Priority | Description |
|------------|------------|----------|-------------|
| 0 | 0 | Disabled | Disabled |
| 0 | 1 | Level 1 | Low |
| 1 | 0 | Level 2 | Nominal |
| 1 | 1 | Level 3 | High |

Note: where x indicates the register bits from 0–7.

Table 40. IRQ0 Enable High Bit Register (IRQ0ENH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-------|-------|--------|--------|----------|---|--------|
| Field | Reserved | T1ENH | T0ENH | U0RENH | U0TENH | Reserved | | ADCENH |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Address | FC1H | | | | | | | |

| Bit | Description |
|---------------|--|
| [7] | Reserved This bit is reserved and must be programmed to 0. |
| [6] T1ENH | Timer 1 Interrupt Request Enable High Bit |
| [5] T0ENH | Timer 0 Interrupt Request Enable High Bit |
| [4] U0RENH | UART 0 Receive Interrupt Request Enable High Bit |
| [3] U0TENH | UART 0 Transmit Interrupt Request Enable High Bit |
| [2:1] | Reserved These bits are reserved and must be programmed to 00. |
| [0] ADCENH | ADC Interrupt Request Enable High Bit |

Table 41. IRQ0 Enable Low Bit Register (IRQ0ENL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-------|-------|--------|--------|----------|---|--------|
| Field | Reserved | T1ENL | T0ENL | U0RENL | U0TENL | Reserved | | ADCENL |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R | | R/W |
| Address | FC2H | | | | | | | |

| Bit | Description |
|--------------|--|
| [7] | Reserved This bit is reserved and must be programmed to 0 when read. |
| [6] T1ENL | Timer 1 Interrupt Request Enable Low Bit |

| Bit | Description (Continued) |
|---------------|--|
| [5] T0ENL | Timer 0 Interrupt Request Enable Low Bit |
| [4] U0RENL | UART 0 Receive Interrupt Request Enable Low Bit |
| [3] U0TENL | UART 0 Transmit Interrupt Request Enable Low Bit |
| [2:1] | Reserved These bits are reserved and must be programmed to 00. |
| [0] ADCENL | ADC Interrupt Request Enable Low Bit |

IRQ1 Enable High and Low Bit Registers

Table 42 describes the priority control for IRQ1. The IRQ1 Enable High and Low Bit registers (Table 43 and Table 44) form a priority-encoded enabling for interrupts in the Interrupt Request 1 Register. Priority is generated by setting bits in each register.

Table 42. IRQ1 Enable and Priority Encoding

| IRQ1ENH[x] | IRQ1ENL[x] | Priority | Description |
|------------|------------|----------|-------------|
| 0 | 0 | Disabled | Disabled |
| 0 | 1 | Level 1 | Low |
| 1 | 0 | Level 2 | Nominal |
| 1 | 1 | Level 3 | High |

Note: x indicates register bits 0–7.

Table 43. IRQ1 Enable High Bit Register (IRQ1ENH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| Field | PA7VENH | PA6CENH | PA5ENH | PA4ENH | PA3ENH | PA2ENH | PA1ENH | PA0ENH |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FC4H | | | | | | | |

| Bit | Description |
|-----------------|---|
| [7] PA7VENH | Port A Bit[7] Interrupt Request Enable High Bit |
| [6] PA6CENH | Port A Bit[7] or Comparator Interrupt Request Enable High Bit |
| [5:0] PAxENH | Port A Bit[x] Interrupt Request Enable High Bit For selection of Port A as the interrupt source, see the Shared Interrupt Select Register section on page 67. |

Note: x indicates the specific GPIO Port A pin number (5–0).

Table 44. IRQ1 Enable Low Bit Register (IRQ1ENL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| Field | PA7VENL | PA6CENL | PA5ENL | PA4ENL | PA3ENL | PA2ENL | PA1ENL | PA0ENL |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FC5H | | | | | | | |

| Bit | Description |
|-----------------|---|
| [7] PA7VENL | Port A Bit[7] Interrupt Request Enable Low Bit |
| [6] PA6CENL | Port A Bit[7] or Comparator Interrupt Request Enable Low Bit |
| [5:0] PAxENL | Port A Bit[x] Interrupt Request Enable Low Bit |

Note: x indicates the specific GPIO Port A pin number (5–0).

IRQ2 Enable High and Low Bit Registers

Table 45 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers (Table 46 and Table 47) form a priority encoded enabling for interrupts in the Interrupt Request 2 register. Priority is generated by setting bits in each register.

Table 45. IRQ2 Enable and Priority Encoding

| IRQ2ENH[x] | IRQ2ENL[x] | Priority | Description |
|------------|------------|----------|-------------|
| 0 | 0 | Disabled | Disabled |
| 0 | 1 | Level 1 | Low |
| 1 | 0 | Level 2 | Nominal |
| 1 | 1 | Level 3 | High |

Note: where x indicates the register bits from 0–7.

Table 46. IRQ2 Enable High Bit Register (IRQ2ENH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-----|-----|-----|-------|-------|-------|-------|
| Field | Reserved | | | | C3ENH | C2ENH | C1ENH | C0ENH |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FC7H | | | | | | | |

| Bit | Description |
|--------------|--|
| [7:4] | Reserved These bits are reserved and must be programmed to 0000. |
| [3] C3ENH | Port C3 Interrupt Request Enable High Bit |
| [2] C2ENH | Port C2 Interrupt Request Enable High Bit |
| [1] C1ENH | Port C1 Interrupt Request Enable High Bit |
| [0] C0ENH | Port C0 Interrupt Request Enable High Bit |

Table 47. IRQ2 Enable Low Bit Register (IRQ2ENL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-----|-----|-----|-------|-------|-------|-------|
| Field | Reserved | | | | C3ENL | C2ENL | C1ENL | C0ENL |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FC8H | | | | | | | |

| Bit | Description |
|--------------|--|
| [7:4] | Reserved These bits are reserved and must be programmed to 0000. |
| [3] C3ENL | Port C3 Interrupt Request Enable Low Bit |
| [2] C2ENL | Port C2 Interrupt Request Enable Low Bit |
| [1] C1ENL | Port C1 Interrupt Request Enable Low Bit |
| [0] C0ENL | Port C0 Interrupt Request Enable High Low |

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register (Table 48) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A or Port D input pin.

Table 48. Interrupt Edge Select Register (IRQES)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|------|------|------|------|------|------|------|
| Field | IES7 | IES6 | IES5 | IES4 | IES3 | IES2 | IES1 | IES0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FCDH | | | | | | | |

| Bit | Description |
|-------------|--|
| [7] IESx | Interrupt Edge Select x 0 = An interrupt request is generated on the falling edge of the PAX input or PDx. 1 = An interrupt request is generated on the rising edge of the PAX input PDx. |

Note: x indicates the specific GPIO port pin number (7–0).

Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) register (Table 49) determines the source of the PADxS interrupts. The Shared Interrupt Select register selects between Port A and alternate sources for the individual interrupts.

Because these shared interrupts are edge-triggered, it is possible to generate an interrupt just by switching from one shared source to another. For this reason, an interrupt must be disabled before switching between sources.

Table 49. Shared Interrupt Select Register (IRQSS)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|-------|----------|-----|-----|-----|-----|-----|
| Field | Reserved | PA6CS | Reserved | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FCEH | | | | | | | |

| Bit | Description |
|--------------|---|
| [7] | Reserved This bit is reserved and must be programmed to 0. |
| [6] PA6CS | PA6/Comparator Selection 0 = PA6 is used for the interrupt for PA6CS interrupt request. 1 = The comparator is used as an interrupt for PA6CS interrupt requests. |
| [5:0] | Reserved These bits are reserved and must be programmed to 000000. |

Interrupt Control Register

The Interrupt Control (IRQCTL) Register (Table 50) contains the master enable bit for all interrupts.

Table 50. Interrupt Control Register (IRQCTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|----------|---|---|---|---|---|---|
| Field | IRQE | Reserved | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R | R | R | R |
| Address | FCFH | | | | | | | |

| Bit | Description |
|-------------|--|
| [7] IRQE | <p>Interrupt Request Enable</p> <p>This bit is set to 1 by executing an Enable Interrupts (EI) or Interrupt Return (IRET) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, reset or by a direct register write of a 0 to this bit.</p> <p>0 = Interrupts are disabled. 1 = Interrupts are enabled.</p> |
| [6:0] | <p>Reserved</p> <p>These bits are reserved and must be programmed to 0000000 when read.</p> |

Timers

Z8 Encore! XP F0823 Series products contain up to two 16-bit reloadable timers that are used for timing, event counting or generation of PWM signals. The timers' features include:

- 16-bit reload counter
- Programmable prescaler with prescale values from 1 to 128
- PWM output generation
- Capture and compare capability
- External input pin for timer input, clock gating, or capture signal; external input pin signal frequency is limited to a maximum of one-fourth the system clock frequency
- Timer output pin
- Timer interrupt

In addition to the timers described in this chapter, the baud rate generator of the UART (if unused) also provides basic timing functionality. For information about using the baud rate generator as an additional timer, see the [Universal Asynchronous Receiver/Transmitter](#) chapter on page 97.

Architecture

Figure 9 displays the architecture of the timers.



Figure 9. Timer Block Diagram

Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value 0001H into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value 0000H into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches FFFFH, the timer rolls over to 0000H and continues counting.

Timer Operating Modes

The timers can be configured to operate in the following modes:

ONE-SHOT Mode

In ONE-SHOT Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt and the count value in the Timer High and Low Byte registers is reset to 0001H. The timer is automatically disabled and stops counting.

Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer reload. If it is appropriate to have the Timer Output make a state change at a One-Shot time-out (rather than a single cycle pulse), first set the TPOL bit in the Timer Control Register to the start value before enabling ONE-SHOT Mode. After starting the timer, set TPOL to the opposite bit value.

Observe the following steps to configure a timer for ONE-SHOT Mode and initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for ONE-SHOT Mode
 - Set the prescale value
 - Set the initial output level (High or Low) if using the Timer Output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control Register to enable the timer and initiate counting.

In ONE-SHOT Mode, the system clock always provides the timer input. The timer period is computed via the following equation:

$$\text{ONE-SHOT Mode Time-Out Period (s)} = \frac{(\text{Reload Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

CONTINUOUS Mode

In CONTINUOUS Mode, the timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps to configure a timer for CONTINUOUS Mode and to initiate the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for CONTINUOUS Mode
 - Set the prescale value
 - If using the Timer Output alternate function, set the initial output level (High or Low)
2. Write to the Timer High and Low Byte registers to set the starting count value (usually 0001H). This action only affects the first pass in CONTINUOUS Mode. After the first timer reload in CONTINUOUS Mode, counting always begins at the reset value of 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Enable the timer interrupt (if appropriate) and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin (if using the Timer Output function) for the Timer Output alternate function.
6. Write to the Timer Control Register to enable the timer and initiate counting.

In CONTINUOUS Mode, the system clock always provides the timer input. The timer period is computed via the following equation:

$$\text{CONTINUOUS Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT Mode equation to determine the first time-out period.

COUNTER Mode

In COUNTER Mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO port pin Timer Input alternate function. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the timer input signal. In COUNTER Mode, the prescaler is disabled.



Caution: The input frequency of the timer input signal must not exceed one-fourth the system clock frequency.

Upon reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is

enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps to configure a timer for COUNTER Mode and initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer.
 - Configure the timer for COUNTER Mode.
 - Select either the rising edge or falling edge of the Timer Input signal for the count. This selection also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function is not required to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in COUNTER Mode. After the first timer reload in COUNTER Mode, counting always begins at the reset value of 0001H. In COUNTER Mode the Timer High and Low Byte registers must be written with the value 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin for the Timer Input alternate function.
6. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control Register to enable the timer.

In COUNTER Mode, the number of timer input transitions since the timer start is computed via the following equation:

$$\text{COUNTER Mode Timer Input Transitions} = \text{Current Count Value} - \text{Start Value}$$

COMPARATOR COUNTER Mode

In COMPARATOR COUNTER Mode, the timer counts input transitions from the analog comparator output. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge or the falling edge of the comparator output signal. In COMPARATOR COUNTER Mode, the prescaler is disabled.



Caution: The frequency of the comparator output signal must not exceed one-fourth the system clock frequency.

After reaching the reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reload.

Observe the following steps to configure a timer for COMPARATOR COUNTER Mode and initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer.
 - Configure the timer for COMPARATOR COUNTER Mode.
 - Select either the rising edge or falling edge of the comparator output signal for the count. This also sets the initial logic level (High or Low) for the Timer Output alternate function. However, the Timer Output function is not required to be enabled.
2. Write to the Timer High and Low Byte registers to set the starting count value. This action only affects the first pass in COMPARATOR COUNTER Mode. After the first timer reload in COMPARATOR COUNTER Mode, counting always begins at the reset value of 0001H. Generally, in COMPARATOR COUNTER Mode the Timer High and Low Byte registers must be written with the value 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control Register to enable the timer.

In COMPARATOR COUNTER Mode, the number of comparator output transitions since the timer start is computed via the following equation:

$$\text{Comparator Output Transitions} = \text{Current Count Value} - \text{Start Value}$$

PWM SINGLE OUTPUT Mode

In PWM SINGLE OUTPUT Mode, the timer outputs a PWM output signal through a GPIO port pin. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

If the TPOL bit in the Timer Control Register is set to 1, the Timer Output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the reload value and is reset to 0001H.

If the TPOL bit in the Timer Control Register is set to 0, the Timer Output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the reload value and is reset to 0001H.

Observe the following steps to configure a timer for PWM Single Output mode and initiating the PWM operation:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for PWM Mode
 - Set the prescale value
 - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H); this write only affects the first pass in PWM Mode. After the first timer reset in PWM Mode, counting always begins at the reset value of 0001H.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
5. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, use the ONE-SHOT Mode equation to determine the first PWM time-out period. If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by the following equation:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

PWM Dual Output Mode

In PWM DUAL OUTPUT Mode, the timer outputs a PWM output signal pair (basic PWM signal and its complement) through two GPIO port pins. The timer input is the system clock. The timer first counts up to the 16-bit PWM match value stored in the Timer PWM High and Low Byte registers. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes.

If the TPOL bit in the Timer Control Register is set to 1, the Timer Output signal begins as a High (1) and transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the reload value and is reset to 0001H.

If the TPOL bit in the Timer Control Register is set to 0, the Timer Output signal begins as a Low (0) and transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the reload value and is reset to 0001H.

The timer also generates a second PWM output signal Timer Output Complement. The Timer Output Complement is the complement of the Timer Output PWM signal. A programmable deadband delay can be configured to time delay (0 to 128 system clock cycles) PWM output transitions on these two pins from a low to a high (inactive to active). This ensures a time gap between the deassertion of one PWM output to the assertion of its complement.

Observe the following steps to configure a timer for PWM DUAL OUTPUT Mode and initiating the PWM operation:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for PWM DUAL OUTPUT Mode. Setting the mode also involves writing to the TMODEHI bit in the TxCTL1 Register
 - Set the prescale value
 - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output alternate function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H); this write only affects the first pass in PWM Mode. After the first timer reset in PWM Mode, counting always begins at the reset value of 0001H.
3. Write to the PWM High and Low Byte registers to set the PWM value.
4. Write to the PWM Control Register to set the PWM dead band delay value. The dead-band delay must be less than the duration of the positive phase of the PWM signal (as defined by the PWM high and low byte registers). It must also be less than the duration of the negative phase of the PWM signal (as defined by the difference between the PWM registers and the Timer Reload registers).
5. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
6. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
7. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions. The Timer Output Complement function is shared with the Timer Input function for both timers. Setting the timer mode to Dual PWM automatically switches the function from Timer In to Timer Out Complement.
8. Write to the Timer Control Register to enable the timer and initiate counting.

The PWM period is represented by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than 0001H is loaded into the Timer High and Low Byte registers, the ONE-SHOT Mode equation determines the first PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to 1, the ratio of the PWM output High time to the total period is represented by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

CAPTURE Mode

In CAPTURE Mode, the current timer count value is recorded when the appropriate external Timer Input transition occurs. The capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if the capture occurs on a rising edge or a falling edge of the Timer Input signal. When the capture event occurs, an interrupt is generated and the timer continues counting. The INPCAP bit in TxCTL1 Register is set to indicate the timer interrupt is because of an input capture event.

The timer continues counting up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt and continues counting. The INPCAP bit in TxCTL1 Register clears indicating the timer interrupt is not because of an input capture event.

Observe the following steps to configure a timer for CAPTURE Mode and initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for CAPTURE Mode
 - Set the prescale value
 - Set the capture edge (rising or falling) for the Timer Input
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Clear the Timer PWM High and Low Byte registers to 0000H. Clearing these registers allows the software to determine if interrupts were generated by either a capture or a reload event. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt was generated by a reload.
5. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both

input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.

6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

CAPTURE RESTART Mode

In CAPTURE RESTART Mode, the current timer count value is recorded when the acceptable external Timer Input transition occurs. The capture count value is written to the Timer PWM High and Low Byte registers. The timer input is the system clock. The TPOL bit in the Timer Control Register determines if the capture occurs on a rising edge or a falling edge of the Timer Input signal. When the capture event occurs, an interrupt is generated and the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in TxCTL1 Register is set to indicate the timer interrupt is because of an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in TxCTL1 Register is cleared to indicate the timer interrupt is not caused by an input capture event.

Observe the following steps to configure a timer for CAPTURE RESTART Mode and initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for CAPTURE RESTART Mode; setting the mode also involves writing to TMODEHI bit in TxCTL1 Register
 - Set the prescale value
 - Set the capture edge (rising or falling) for the Timer Input
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the reload value.

4. Clear the Timer PWM High and Low Byte registers to 0000H. Clearing these registers allows the software to determine if interrupts were generated by either a capture or a reload event. If the PWM High and Low Byte registers still contain 0000H after the interrupt, the interrupt was generated by a reload.
5. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.
6. Configure the associated GPIO port pin for the Timer Input alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

In CAPTURE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

COMPARE Mode

In COMPARE Mode, the timer counts up to the 16-bit maximum compare value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the compare value, the timer generates an interrupt and counting continues (the timer value is not reset to 0001H). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) upon compare.

If the Timer reaches FFFFH, the timer rolls over to 0000H and continue counting. Observe the following steps to configure a timer for COMPARE Mode and to initiate the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for COMPARE Mode
 - Set the prescale value
 - Set the initial logic level (High or Low) for the Timer Output alternate function, if appropriate
2. Write to the Timer High and Low Byte registers to set the starting count value.
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers.

5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function.
6. Write to the Timer Control Register to enable the timer and initiate counting.

In COMPARE Mode, the system clock always provides the timer input. The compare time can be calculated by the following equation:

$$\text{COMPARE Mode Time (s)} = \frac{(\text{Compare Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

GATED Mode

In GATED Mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control Register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is deasserted or a timer reload occurs. To determine if a Timer Input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. When reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes (assuming the Timer Input signal remains asserted). Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer reset.

Observe the following steps to configure a timer for GATED Mode and to initiate the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for Gated mode
 - Set the prescale value
2. Write to the Timer High and Low Byte registers to set the starting count value. Writing these registers only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt is generated for both input deassertion and reload events. If appropriate, configure the timer interrupt to be generated only at the input deassertion event or the reload event by setting TICONFIG field of the TxCTL1 Register.

5. Configure the associated GPIO port pin for the Timer Input alternate function.
6. Write to the Timer Control Register to enable the timer.
7. Assert the Timer Input signal to initiate the counting.

CAPTURE/COMPARE Mode

In CAPTURE/COMPARE Mode, the timer begins counting on the first external Timer Input transition. The acceptable transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the system clock.

Every subsequent acceptable transition (after the first) of the Timer Input signal captures the current count value. The capture value is written to the Timer PWM High and Low Byte registers. When the capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H, and counting resumes. The INPCAP bit in TxCTL1 Register is set to indicate the timer interrupt is caused by an input capture event.

If no capture event occurs, the timer counts up to the 16-bit compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in TxCTL1 Register is cleared to indicate the timer interrupt is not because of an input capture event.

Observe the following steps to configure a timer for CAPTURE/COMPARE Mode and initiating the count:

1. Write to the Timer Control Register to:
 - Disable the timer
 - Configure the timer for CAPTURE/COMPARE Mode
 - Set the prescale value
 - Set the capture edge (rising or falling) for the Timer Input
2. Write to the Timer High and Low Byte registers to set the starting count value (typically 0001H).
3. Write to the Timer Reload High and Low Byte registers to set the compare value.
4. Enable the timer interrupt, if appropriate, and set the timer interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt are generated for both input capture and reload events. If appropriate, configure the timer interrupt to be generated only at the input capture event or the reload event by setting TICONFIG field of the TxCTL1 Register.
5. Configure the associated GPIO port pin for the Timer Input alternate function.
6. Write to the Timer Control Register to enable the timer.

- Counting begins on the first appropriate transition of the Timer Input signal. No interrupt is generated by this first edge.

In CAPTURE/COMPARE Mode, the elapsed time from timer start to capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

Reading the Timer Count Values

The current count value in the timers can be read while counting (enabled). This capability has no effect on timer operation. When the timer is enabled and the Timer High Byte Register is read, the contents of the Timer Low Byte register are placed in a holding register. A subsequent read from the Timer Low Byte register returns the value in the holding register. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte register returns the actual value in the counter.

Timer Pin Signal Operation

Timer Output is a GPIO port pin alternate function. The Timer Output is toggled every time the counter is reloaded.

The timer input can be used as a selectable counting source. It shares the same pin as the complementary timer output. When selected by the GPIO Alternate Function registers, this pin functions as a timer input in all modes except for the DUAL PWM OUTPUT mode. For this mode, there is no timer input available.

Timer Control Register Definitions

This section defines the features of the following Timer Control registers.

[Timer 0–1 High and Low Byte Registers](#): see page 83

[Timer Reload High and Low Byte Registers](#): see page 84

[Timer 0–1 PWM High and Low Byte Registers](#): see page 86

[Timer 0–1 Control Registers](#): see page 86

Timer 0–1 High and Low Byte Registers

The Timer 0–1 High and Low Byte (TxH and TxL) registers (Table 51 and Table 52) contain the current 16-bit timer count value. When the timer is enabled, a read from TxH

causes the value in TxL to be stored in a temporary holding register. A read from TxL always returns this temporary register when the timers are enabled. When the timer is disabled, reads from the TxL reads the register directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations, so simultaneous 16-bit writes are not possible. If either the Timer High or Low Byte registers are written during counting, the 8-bit written value is placed in the counter (High or Low Byte) at the next clock edge. The counter continues counting from the new value.

Table 51. Timer 0–1 High Byte Register (TxH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | TH | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F00H, F08H | | | | | | | |

Table 52. Timer 0–1 Low Byte Register (TxL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | TL | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F01H, F09H | | | | | | | |

| Bit | Description |
|-----------------|---|
| [7:0] TH, TL | Timer High and Low Bytes These 2 bytes, {TH[7:0], TL[7:0]}, contain the current 16-bit timer count value. |

Timer Reload High and Low Byte Registers

The Timer 0–1 Reload High and Low Byte (TxRH and TxRL) registers (Table 53 and Table 54) store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte register are stored in a temporary holding register. When a write to the Timer Reload Low Byte register occurs, the temporary holding register value is written to the Timer High Byte register. This operation allows simultaneous updates of the 16-bit Timer reload value. In COMPARE Mode, the Timer Reload High and Low Byte registers store the 16-bit compare value.

Table 53. Timer 0–1 Reload High Byte Register (TxRH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | TRH | | | | | | | |
| RESET | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F02H, F0AH | | | | | | | |

Table 54. Timer 0–1 Reload Low Byte Register (TxRL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | TRL | | | | | | | |
| RESET | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F03H, F0BH | | | | | | | |

| Bit | Description |
|-----|-------------|
| [7] | |
| [6] | |
| [5] | |
| [4] | |
| [3] | |
| [2] | |
| [1] | |
| [0] | |

TRH and TRL—Timer Reload Register High and Low
These two bytes form the 16-bit reload value, {TRH[7:0], TRL[7:0]}. This value sets the maximum count value which initiates a timer reload to 0001H. In COMPARE Mode, these two bytes form the 16-bit compare value.

Timer 0–1 PWM High and Low Byte Registers

The Timer 0–1 PWM High and Low Byte (TxPWMH and TxPWML) registers (Table 55 and Table 56) control pulse-width modulator (PWM) operations. These registers also store the capture values for the CAPTURE and CAPTURE/COMPARE modes.

Table 55. Timer 0–1 PWM High Byte Register (TxPWMH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | PWMH | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F04H, F0CH | | | | | | | |

Table 56. Timer 0–1 PWM Low Byte Register (TxPWML)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|-----|-----|-----|-----|-----|-----|-----|
| Field | PWML | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F05H, F0DH | | | | | | | |

| Bit | Description |
|------------------------|---|
| [7:0] PWMH, PWML | Pulse-Width Modulator High and Low Bytes These two bytes, {PWMH[7:0], PWML[7:0]}, form a 16-bit value that is compared to the current 16-bit timer count. When a match occurs, the PWM output changes state. The PWM output value is set by the TPOL bit in the Timer Control Register (TxCTL1) register. |

These TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in CAPTURE or CAPTURE/COMPARE modes.

Timer 0–1 Control Registers

The Timer Control registers are 8-bit read/write registers that control the operation of their associated counter/timers.

Timer 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and Timer Control Register 1 (TxCTL1) determine the timer operating mode. It also includes a programmable PWM deadband delay,

two bits to configure timer interrupt definition, and a status bit to identify if the most recent timer interrupt is caused by an input capture event.

Table 57. Timer 0–1 Control Register 0 (TxCTL0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|----------|-----|----------|------|-----|-----|--------|
| Field | TMODEHI | TICONFIG | | Reserved | PWMD | | | INPCAP |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F06H, F0EH | | | | | | | |

| Bit | Description |
|-------------------|---|
| [7] TMODEHI | Timer Mode High Bit This bit along with the TMODE field in TxCTL1 Register determines the operating mode of the timer. This is the most-significant bit of the Timer mode selection value. |
| [6:5] TICONFIG | Timer Interrupt Configuration This field configures timer interrupt definition. 0x = Timer Interrupt occurs on all defined reload, compare and input events. 10 = Timer Interrupt only on defined input capture/deassertion events. 11 = Timer Interrupt only on defined reload/compare events. |
| [4] | Reserved This bit is reserved and must be programmed to 0. |
| [3:1] PWMD | PWMD—PWM Delay value This field is a programmable delay to control the number of system clock cycles delay before the Timer Output and the Timer Output Complement are forced to their active state. 000 = No delay. 001 = 2 cycles delay. 010 = 4 cycles delay. 011 = 8 cycles delay. 100 = 16 cycles delay. 101 = 32 cycles delay. 110 = 64 cycles delay. 111 = 128 cycles delay. |
| [0] INPCAP | Input Capture Event This bit indicates if the most recent timer interrupt is caused by a Timer Input capture event. 0 = Previous timer interrupt is not a result of Timer Input capture event. 1 = Previous timer interrupt is a result of Timer Input capture event. |

Timer 0–1 Control Register 1

The Timer 0–1 Control (TxCTL1) registers enable/disable the timers, set the prescaler value, and determine the timer operating mode.

Table 58. Timer 0–1 Control Register 1 (TxCTL1)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------------|------|------|-----|-----|-------|-----|-----|
| Field | TEN | TPOL | PRES | | | TMODE | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F07H, F0FH | | | | | | | |

| Bit | Description |
|-------------|---|
| [7] TEN | Timer Enable 0 = Timer is disabled. 1 = Timer enabled to count. |
| [6] TPOL | Timer Input/Output Polarity Operation of this bit is a function of the current operating mode of the timer. ONE-SHOT Mode When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload. CONTINUOUS Mode When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload. |

| Bit | Description (Continued) |
|--------------------------|---|
| [6] TPOl (cont'd.) | <p>COUNTER Mode If the timer is enabled the Timer Output signal is complemented after timer reload. 0 = Count occurs on the rising edge of the Timer Input signal. 1 = Count occurs on the falling edge of the Timer Input signal.</p> <p>PWM SINGLE OUTPUT Mode 0 = Timer Output is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon reload. 1 = Timer Output is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon reload.</p> <p>CAPTURE Mode 0 = Count is captured on the rising edge of the Timer Input signal. 1 = Count is captured on the falling edge of the Timer Input signal.</p> <p>COMPARE Mode When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload.</p> <p>GATED Mode 0 = Timer counts when the Timer Input signal is High (1) and interrupts are generated on the falling edge of the Timer Input. 1 = Timer counts when the Timer Input signal is Low (0) and interrupts are generated on the rising edge of the Timer Input.</p> <p>CAPTURE/COMPARE Mode 0 = Counting is started on the first rising edge of the Timer Input signal. The current count is captured on subsequent rising edges of the Timer Input signal. 1 = Counting is started on the first falling edge of the Timer Input signal. The current count is captured on subsequent falling edges of the Timer Input signal.</p> <p>PWM DUAL OUTPUT Mode 0 = Timer Output is forced Low (0) and Timer Output Complement is forced High (1) when the timer is disabled. When enabled, the Timer Output is forced High (1) upon PWM count match and forced Low (0) upon reload. When enabled, the Timer Output Complement is forced Low (0) upon PWM count match and forced High (1) upon reload. The PWMD field in TxCTL0 register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to High (1). 1 = Timer Output is forced High (1) and Timer Output Complement is forced Low (0) when the timer is disabled. When enabled, the Timer Output is forced Low (0) upon PWM count match and forced High (1) upon reload. When enabled, the Timer Output Complement is forced High (1) upon PWM count match and forced Low (0) upon reload. The PWMD field in TxCTL0 register is a programmable delay to control the number of cycles time delay before the Timer Output and the Timer Output Complement is forced to Low (0).</p> <p>CAPTURE RESTART Mode 0 = Count is captured on the rising edge of the Timer Input signal. 1 = Count is captured on the falling edge of the Timer Input signal.</p> |

| Bit | Description (Continued) |
|--------------------------|---|
| [6] TPOL (cont'd.) | <p>COMPARATOR COUNTER Mode</p> <p>When the timer is disabled, the Timer Output signal is set to the value of this bit. When the timer is enabled, the Timer Output signal is complemented upon timer reload.</p> <p>Caution: When the Timer Output alternate function TxOUT on a GPIO port pin is enabled, TxOUT changes to whatever state the TPOL bit is in. The timer does not need to be enabled for that to happen. Also, the port data direction sub register is not needed to be set to output on TxOUT. Changing the TPOL bit with the timer enabled and running does not immediately change the TxOUT.</p> |
| [5:3] PRES | <p>Prescale Value</p> <p>The timer input clock is divided by 2^{PRES}, where PRES can be set from 0 to 7. The prescaler is reset each time the timer is disabled. This reset ensures proper clock division each time the timer is restarted.</p> <p>000 = Divide by 1. 001 = Divide by 2. 010 = Divide by 4. 011 = Divide by 8. 100 = Divide by 16. 101 = Divide by 32. 110 = Divide by 64. 111 = Divide by 128.</p> |
| [2:0] TMODE | <p>Timer Mode</p> <p>This field, along with the TMODEHI bit in TxCTL0 Register, determines the operating mode of the timer. TMODEHI is the most significant bit of the timer mode selection value.</p> <p>0000 = ONE-SHOT Mode. 0001 = CONTINUOUS Mode. 0010 = COUNTER Mode. 0011 = PWM SINGLE OUTPUT Mode. 0100 = CAPTURE Mode. 0101 = COMPARE Mode. 0110 = GATED Mode. 0111 = CAPTURE/COMPARE Mode. 1000 = PWM DUAL OUTPUT Mode. 1001 = CAPTURE RESTART Mode. 1010 = COMPARATOR COUNTER Mode.</p> |

Watchdog Timer

The Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults, and other system-level problems which can place Z8 Encore! XP F0823 Series devices into unsuitable operating states. The features of Watchdog Timer include:

- On-chip RC oscillator
- A selectable time-out response: reset or interrupt
- 24-bit programmable time-out value

Operation

The WDT is a retriggerable one-shot timer that resets or interrupts F0823 Series devices when the WDT reaches its terminal count. The Watchdog Timer uses a dedicated on-chip RC oscillator as its clock source. The Watchdog Timer operates in only two modes: ON and OFF. Once enabled, it always counts and must be refreshed to prevent a time-out. Perform an enable by executing the WDT instruction or by setting the WDT_AO Flash Option Bit. The WDT_AO bit forces the Watchdog Timer to operate immediately upon reset, even if a WDT instruction has not been executed.

The Watchdog Timer is a 24-bit reloadable down counter that uses three 8-bit registers in the eZ8 CPU register space to set the reload value. The nominal WDT time-out period is described by the following equation:

$$\text{WDT Time-out Period (ms)} = \frac{\text{WDT Reload Value}}{10}$$

where the WDT reload value is the decimal value of the 24-bit value given by {WDTU[7:0], WDTM[7:0], WDTL[7:0]} and the typical Watchdog Timer RC oscillator frequency is 10kHz. The Watchdog Timer cannot be refreshed after it reaches 000002H. The WDT Reload Value must not be set to values below 000004H. Table 59 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

Table 59. Watchdog Timer Approximate Time-Out Delays

| WDT Reload Value (Hex) | WDT Reload Value (Decimal) | Approximate Time-Out Delay (with 10kHz typical WDT oscillator frequency) | |
|---------------------------|-------------------------------|---|------------------------|
| | | Typical | Description |
| 000004 | 4 | 400 μs | Minimum time-out delay |
| FFFFFF | 16,777,215 | 28 minutes | Maximum time-out delay |

Watchdog Timer Refresh

When first enabled, the WDT is loaded with the value in the Watchdog Timer Reload registers. The Watchdog Timer counts down to 000000H unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the down counter to be reloaded with the WDT reload value stored in the Watchdog Timer Reload registers. Counting resumes following the reload operation.

When Z8 Encore! XP F0823 Series devices are operating in DEBUG Mode (using the OCD), the Watchdog Timer is continuously refreshed to prevent any Watchdog Timer time-outs.

Watchdog Timer Time-Out Response

The Watchdog Timer times out when the counter reaches 000000H. A time-out of the Watchdog Timer generates either an interrupt or a system reset. The WDT_RES Flash Option Bit determines the time-out response of the Watchdog Timer. For information about programming of the WDT_RES Flash Option Bit, see [the Flash Option Bits](#) chapter on page 146.

WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the Watchdog Timer issues an interrupt request to the interrupt controller and sets the WDT status bit in the Watchdog Timer Control Register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executing code from the vector address. After time-out and interrupt generation, the Watchdog Timer counter rolls over to its maximum value of FFFFFH and continues counting. The Watchdog Timer counter is not automatically returned to its Reload Value.

The Reset Status Register (see the [Reset Status Register](#) section on page 28) must be read before clearing the WDT interrupt. This read clears the WDT time-out Flag and prevents further WDT interrupts for immediately occurring.

WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and F0823 Series are in STOP Mode, the Watchdog Timer automatically initiates a Stop Mode Recovery and generates an interrupt request. Both the WDT status bit and the STOP bit in the Watchdog Timer Control Register are set to 1 following a WDT time-out in STOP Mode. For more information about Stop Mode Recovery, see [the Reset and Stop Mode Recovery](#) chapter on page 21.

If interrupts are enabled, following completion of the Stop Mode Recovery the eZ8 CPU responds to the interrupt request by fetching the Watchdog Timer interrupt vector and executing code from the vector address.

WDT Reset in NORMAL Operation

If configured to generate a Reset when a time-out occurs, the Watchdog Timer forces the device into the System Reset state. The WDT status bit in the Watchdog Timer Control Register is set to 1. For more information about System Reset, see [the Reset and Stop Mode Recovery](#) chapter on page 21.

WDT Reset in STOP Mode

If configured to generate a Reset when a time-out occurs and the device is in STOP Mode, the Watchdog Timer initiates a Stop Mode Recovery. Both the WDT status bit and the STOP bit in the Watchdog Timer Control Register are set to 1 following WDT time-out in STOP Mode. For more information, see [the Reset and Stop Mode Recovery](#) chapter on page 21.

Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer Control Register (WDTCTL) address unlocks the three Watchdog Timer Reload Byte Registers (WDTU, WDTL, and WDTM) to allow changes to the time-out period. These write operations to the WDTCTL Register address produce no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the Reload registers. The following sequence is required to unlock the Watchdog Timer Reload Byte Registers (WDTU, WDTL, and WDTM) for write access.

1. Write 55H to the Watchdog Timer Control Register (WDTCTL).
2. Write AAH to the Watchdog Timer Control Register (WDTCTL).
3. Write the Watchdog Timer Reload Upper Byte register (WDTU).
4. Write the Watchdog Timer Reload High Byte register (WDTL).
5. Write the Watchdog Timer Reload Low Byte register (WDTM).

All three Watchdog Timer Reload registers must be written in the order just listed. There must be no other register writes between each of these operations. If a register write occurs, the lock state machine resets and no further writes can occur unless the sequence is restarted. The value in the Watchdog Timer Reload registers is loaded into the counter when the Watchdog Timer is first enabled and every time a WDT instruction is executed.

Watchdog Timer Control Register Definitions

This section defines the features of the following Watchdog Timer Control registers.

[Watchdog Timer Control Register \(WDTCTL\)](#): see page 94

[Watchdog Timer Reload Upper Byte Register \(WDTU\)](#): see page 95

[Watchdog Timer Reload High Byte Register \(WDTH\)](#): see page 95

[Watchdog Timer Reload Low Byte Register \(WDTL\)](#): see page 95

Watchdog Timer Control Register

The Watchdog Timer Control (WDTCTL) register is a write-only control register. Writing the 55H, AAH unlock sequence to the WDTCTL Register address unlocks the three Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) to allow changes to the time-out period. These write operations to the WDTCTL Register address produce no effect on the bits in the WDTCTL Register. The locking mechanism prevents spurious writes to the Reload registers.

This register address is shared with the read-only Reset Status Register.

Table 60. Watchdog Timer Control Register (WDTCTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------|---|---|---|---|---|---|---|
| Field | WDTUNLK | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | W | W | W | W | W | W | W | W |
| Address | FF0H | | | | | | | |

| Bit | Description |
|------------------|---|
| [7:0] WDTUNLK | Watchdog Timer Unlock The software must write the correct unlocking sequence to this register before it is allowed to modify the contents of the Watchdog Timer reload registers. |

Watchdog Timer Reload Upper, High and Low Byte Registers

The Watchdog Timer Reload Upper, High and Low Byte (WDTU, WDTH, WDTL) registers, shown in Tables 61 through 63, form the 24-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes. The 24-bit reload value ranges across bits [23:0] to encompass the three bytes {WDTU[7:0], WDTH[7:0], WDTL[7:0]}. Writing to these registers sets the appropriate Reload Value. Reading from these registers returns the current Watchdog Timer count value.



Caution: The 24-bit WDT Reload Value must not be set to a value less than 000004H.

Table 61. Watchdog Timer Reload Upper Byte Register (WDTU)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|------|------|------|------|------|------|------|------|
| Field | WDTU | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Address | FF1H | | | | | | | |
| Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value. | | | | | | | | |

| Bit | Description |
|---------------|---|
| [7:0] WDTU | WDT Reload Upper Byte Most significant byte (MSB), Bits[23:16], of the 24-bit WDT reload value. |

Table 62. Watchdog Timer Reload High Byte Register (WDTH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|------|------|------|------|------|------|------|------|
| Field | WDTH | | | | | | | |
| RESET | 04H | | | | | | | |
| R/W | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Address | FF2H | | | | | | | |
| Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value. | | | | | | | | |

| Bit | Description |
|---------------|---|
| [7:0] WDTH | WDT Reload High Byte Middle byte, Bits[15:8], of the 24-bit WDT reload value. |

Table 63. Watchdog Timer Reload Low Byte Register (WDTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|------|------|------|------|------|------|------|------|
| Field | WDTL | | | | | | | |
| RESET | 00H | | | | | | | |
| R/W | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* | R/W* |
| Address | FF3H | | | | | | | |
| Note: R/W*—Read returns the current WDT count value. Write sets the appropriate Reload Value. | | | | | | | | |

| Bit | Description |
|------------|--|
| [7:0] | WDT Reload Low |
| WDTL | Least significant byte (LSB), Bits[7:0], of the 24-bit WDT reload value. |

Universal Asynchronous Receiver/ Transmitter

The universal asynchronous receiver/transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The UART uses a single 8-bit data mode with selectable parity. The features of UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- Option of one or two STOP bits
- Separate transmit and receive interrupts
- Framing, parity, overrun, and break detection
- Separate transmit and receive enables
- 16-bit baud rate generator (BRG)
- Selectable MULTIPROCESSOR (9-bit) Mode with three configurable interrupt schemes
- BRG can be configured and used as a basic 16-bit timer
- Driver Enable output for external bus transceivers

Architecture

The UART consists of three primary functional blocks: transmitter, receiver, and baud rate generator. The UART's transmitter and receiver function independently, but employ the same baud rate and data format. Figure 10 displays the UART architecture.



Figure 10. UART Block Diagram

Operation

The UART always transmits and receives data in an 8-bit data format, least-significant bit (lsb) first. An even or odd parity bit can be added to the data stream. Each character begins with an active Low Start bit and ends with either 1 or 2 active High Stop bits. Figure 11 and Figure 12 display the asynchronous data format employed by the UART without parity and with parity, respectively.

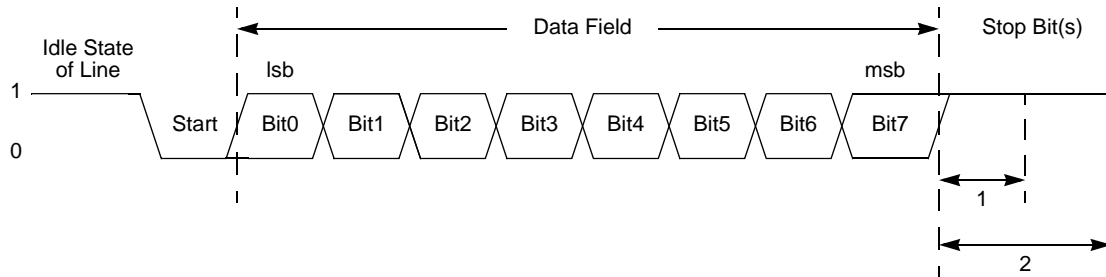


Figure 11. UART Asynchronous Data Format without Parity



Figure 12. UART Asynchronous Data Format with Parity

Transmitting Data Using the Polled Method

Observe the following steps to transmit data using the polled method of operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the required baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Write to the UART Control 1 Register, if MULTIPROCESSOR Mode is appropriate, to enable MULTIPROCESSOR (9-bit) Mode functions.
4. Set the Multiprocessor Mode Select (MPEN) bit to enable MULTIPROCESSOR Mode.
5. Write to the UART Control 0 Register to:
 - Set the transmit enable bit (TEN) to enable the UART for data transmission
 - Set the parity enable bit (PEN), if parity is appropriate and MULTIPROCESSOR Mode is not enabled, and select either even or odd parity (PSEL)

- Set or clear the CTSE bit to enable or disable control from the remote receiver using the CTS pin
6. Check the TDRE bit in the UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If empty, continue to [Step 7](#). If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.
 7. Write the UART Control 1 Register to select the outgoing address bit.
 8. Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
 9. Write the data byte to the UART Transmit Data Register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.
 10. Make any changes to the Multiprocessor Bit Transmitter (MPBT) value, if appropriate and MULTIPROCESSOR Mode is enabled,.
 11. To transmit additional bytes, return to [Step 5](#).

Transmitting Data Using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Observe the following steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the appropriate baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Transmitter interrupt and set the acceptable priority.
5. Write to the UART Control 1 Register to enable MULTIPROCESSOR (9-bit) Mode functions, if MULTIPROCESSOR Mode is appropriate.
6. Set the MULTIPROCESSOR Mode Select (MPEN) to Enable MULTIPROCESSOR Mode.
7. Write to the UART Control 0 Register to:
 - Set the transmit enable bit (TEN) to enable the UART for data transmission.
 - Enable parity, if appropriate and if MULTIPROCESSOR Mode is not enabled, and select either even or odd parity.

- Set or clear `CTSE` to enable or disable control from the remote receiver using the `CTS` pin.
8. Execute an `EI` instruction to enable interrupts.

The UART is now configured for interrupt-driven data transmission. Because the UART Transmit Data Register is empty, an interrupt is generated immediately. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Write the UART Control 1 Register to select the multiprocessor bit for the byte to be transmitted:
Set the Multiprocessor Bit Transmitter (`MPBT`) if sending an address byte, clear it if sending a data byte.
2. Write the data byte to the UART Transmit Data Register. The transmitter automatically transfers the data to the Transmit Shift register and transmits the data.
3. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
4. Execute the `IRET` instruction to return from the interrupt-service routine and wait for the Transmit Data Register to again become empty.

Receiving Data Using the Polled Method

Observe the following steps to configure the UART for polled data reception:

1. Write to the UART Baud Rate High and Low Byte registers to set an acceptable baud rate for the incoming data stream.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Write to the UART Control 1 Register to enable `MULTIPROCESSOR` Mode functions, if appropriate.
4. Write to the UART Control 0 Register to:
 - Set the receive enable bit (`REN`) to enable the UART for data reception
 - Enable parity, if appropriate and if Multiprocessor mode is not enabled, and select either even or odd parity
5. Check the `RDA` bit in the UART Status 0 Register to determine if the Receive Data Register contains a valid data byte (indicated by a 1). If `RDA` is set to 1 to indicate available data, continue to [Step 6](#). If the Receive Data Register is empty (indicated by a 0), continue to monitor the `RDA` bit awaiting reception of the valid data.
6. Read data from the UART Receive Data Register. If operating in `MULTIPROCESSOR (9-bit) Mode`, further actions may be required depending on the `MULTIPROCESSOR Mode` bits `MPMD[1:0]`.

7. Return to [Step 4](#) to receive additional data.

Receiving Data Using the Interrupt-Driven Method

The UART Receiver interrupt indicates the availability of new data (as well as error conditions). Observe the following steps to configure the UART receiver for interrupt-driven operation:

1. Write to the UART Baud Rate High and Low Byte registers to set the acceptable baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a `DI` instruction to disable interrupts.
4. Write to the Interrupt control registers to enable the UART Receiver interrupt and set the acceptable priority.
5. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
6. Write to the UART Control 1 Register to enable Multiprocessor (9-bit) mode functions, if appropriate.
 - Set the Multiprocessor Mode Select (`MPEN`) to Enable `MULTIPROCESSOR Mode`
 - Set the Multiprocessor Mode Bits, `MPMD[1:0]`, to select the acceptable address matching scheme
 - Configure the UART to interrupt on received data and errors or errors only (interrupt on errors only is unlikely to be useful for Z8 Encore! XP devices without a DMA block)
7. Write the device address to the Address Compare Register (automatic `MULTIPROCESSOR` modes only).
8. Write to the UART Control 0 Register to:
 - Set the receive enable bit (`REN`) to enable the UART for data reception
 - Enable parity, if appropriate and if multiprocessor mode is not enabled, and select either even or odd parity
9. Execute an `EI` instruction to enable interrupts.

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated interrupt service routine (ISR) performs the following:

1. Checks the UART Status 0 Register to determine the source of the interrupt - error, break, or received data.
2. Reads the data from the UART Receive Data Register if the interrupt was because of data available. If operating in MULTIPROCESSOR (9-bit) Mode, further actions may be required depending on the MULTIPROCESSOR Mode bits MPMD[1:0].
3. Clears the UART Receiver interrupt in the applicable Interrupt Request register.
4. Executes the IRET instruction to return from the interrupt-service routine and await more data.

Clear To Send ($\overline{\text{CTS}}$) Operation

The CTS pin, if enabled by the CTSE bit of the UART Control 0 Register, performs flow control on the outgoing transmit datastream. The Clear To Send ($\overline{\text{CTS}}$) input pin is sampled one system clock before beginning any new character transmission. To delay transmission of the next data character, an external receiver must deassert $\overline{\text{CTS}}$ at least one system clock cycle before a new data transmission begins. For multiple character transmissions, this action is typically performed during Stop Bit transmission. If $\overline{\text{CTS}}$ deasserts in the middle of a character transmission, the current character is sent completely.

MULTIPROCESSOR (9-Bit) Mode

The UART has a MULTIPROCESSOR (9-bit) Mode that uses an extra (9th) bit for selective communication when a number of processors share a common UART bus. In MULTIPROCESSOR Mode (also referred to as 9-bit mode), the multiprocessor bit (MP) is transmitted immediately following the 8-bits of data and immediately preceding the Stop bit(s) as displayed in Figure 13. The character format is given below:



Figure 13. UART Asynchronous MULTIPROCESSOR Mode Data Format

In MULTIPROCESSOR (9-bit) Mode, the parity bit location (9th bit) becomes the Multiprocessor control bit. The UART Control 1 and Status 1 registers provide MULTIPROCESSOR (9-bit) Mode control and status information. If an automatic address matching

scheme is enabled, the UART Address Compare register holds the network address of the device.

MULTIPROCESSOR (9-bit) Mode Receive Interrupts

When MULTIPROCESSOR Mode is enabled, the UART only processes frames addressed to it. The determination of whether a frame of data is addressed to the UART can be made in hardware, software or some combination of the two, depending on the multiprocessor configuration bits. In general, the address compare feature reduces the load on the CPU, because it does not require access to the UART when it receives data directed to other devices on the multi-node network. The following three MULTIPROCESSOR modes are available in hardware:

- Interrupt on all address bytes
- Interrupt on matched address bytes and correctly framed data bytes
- Interrupt only on correctly framed data bytes

These modes are selected with $MPMD[1:0]$ in the UART Control 1 Register. For all multiprocessor modes, bit $MPEN$ of the UART Control 1 Register must be set to 1.

The first scheme is enabled by writing $01b$ to $MPMD[1:0]$. In this mode, all incoming address bytes cause an interrupt, while data bytes never cause an interrupt. The interrupt service routine must manually check the address byte that caused triggered the interrupt. If it matches the UART address, the software clears $MPMD[0]$. Each new incoming byte interrupts the CPU. The software is responsible for determining the end of the frame. It checks for the end-of-frame by reading the $MPRX$ bit of the UART Status 1 Register for each incoming byte. If $MPRX=1$, a new frame has begun. If the address of this new frame is different from the UART's address, $MPMD[0]$ must be set to 1 causing the UART interrupts to go inactive until the next address byte. If the new frame's address matches the UART's, the data in the new frame is processed as well.

The second scheme requires the following: set $MPMD[1:0]$ to $10B$ and write the UART's address into the UART Address Compare register. This mode introduces additional hardware control, interrupting only on frames that match the UART's address. When an incoming address byte does not match the UART's address, it is ignored. All successive data bytes in this frame are also ignored. When a matching address byte occurs, an interrupt is issued and further interrupts now occur on each successive data byte. When the first data byte in the frame is read, the $NEWFRM$ bit of the UART Status 1 Register is asserted. All successive data bytes have $NEWFRM=0$. When the next address byte occurs, the hardware compares it to the UART's address. If there is a match, the interrupts continues and the $NEWFRM$ bit is set for the first byte of the new frame. If there is no match, the UART ignores all incoming bytes until the next address match.

The third scheme is enabled by setting $MPMD[1:0]$ to $11b$ and by writing the UART's address into the UART Address Compare Register. This mode is identical to the second

scheme, except that there are no interrupts on address bytes. The first data byte of each frame remains accompanied by a NEWFRM assertion.

External Driver Enable

The UART provides a Driver Enable (DE) signal for off-chip bus transceivers. This feature reduces the software overhead associated with using a GPIO pin to control the transceiver when communicating on a multi-transceiver bus, such as RS-485.

Driver Enable is an active High signal that envelopes the entire transmitted data frame including parity and Stop bits as displayed in Figure 14. The Driver Enable signal asserts when a byte is written to the UART Transmit Data Register. The Driver Enable signal asserts at least one UART bit period and no greater than two UART bit periods before the Start bit is transmitted. This allows a setup time to enable the transceiver. The Driver Enable signal deasserts one system clock period after the final Stop bit is transmitted. This one system clock delay allows both time for data to clear the transceiver before disabling it, as well as the ability to determine if another character follows the current character. In the event of back to back characters (new data must be written to the Transmit Data Register before the previous character is completely transmitted) the DE signal is not deasserted between characters. The DEPOL bit in the UART Control Register 1 sets the polarity of the Driver Enable signal.



Figure 14. UART Driver Enable Signal Timing (shown with 1 Stop Bit and Parity)

The Driver Enable to Start bit setup time is calculated as follows:

UART Interrupts

The UART features separate interrupts for the transmitter and the receiver. In addition, when the UART primary functionality is disabled, the Baud Rate Generator can also function as a basic timer with interrupt capability.

$$\left(\frac{1}{\text{Baud Rate (Hz)}}\right) \leq \text{DE to Start Bit Setup Time (s)} \leq \left(\frac{2}{\text{Baud Rate (Hz)}}\right)$$

Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Data Register Empty bit (TDRE) is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The TDRE interrupt occurs after the Transmit shift register has shifted the first bit of data out. The Transmit Data Register can now be written with the next character to send. This action provides 7 bit periods of latency to load the Transmit Data Register before the Transmit shift register completes shifting the current character. Writing to the UART Transmit Data Register clears the TDRE bit to 0.

Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- A data byte is received and is available in the UART Receive Data Register. This interrupt can be disabled independently of the other receiver interrupt sources. The received data interrupt occurs after the receive character has been received and placed in the Receive Data Register. To avoid an overrun error, software must respond to this received data available condition before the next character is completely received.

► **Note:** In MULTIPROCESSOR Mode (MPEN = 1), the receive data interrupts are dependent on the multiprocessor configuration and the most recent address byte.

- A break is received
- An overrun is detected
- A data framing error is detected

UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data Register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the UART Status 0 Register is updated to indicate the overrun condition (and Break Detect, if applicable). The RDA bit is set to 1 to indicate that the Receive Data Register contains a data byte. However, because the overrun error

occurred, this byte cannot contain valid data and must be ignored. The BRKD bit indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data Register must be read again to clear the error bits in the UART Status 0 Register. Updates to the Receive Data Register occur only when the next data word is received.

UART Data and Error Handling Procedure

Figure 15 displays the recommended procedure for use in UART receiver interrupt service routines.



Figure 15. UART Receiver Interrupt Service Routine Flow

Baud Rate Generator Interrupts

If the Baud Rate Generator (BRG) interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This condition allows the Baud Rate Generator to function as an additional counter if the UART functionality is not employed.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

When the UART is disabled, the Baud Rate Generator functions as a basic 16-bit timer with interrupt on time-out. Observe the following steps to configure the Baud Rate Generator as a timer with interrupt on time-out:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 Register to 0.
2. Load the acceptable 16-bit count value into the UART Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BIRQ bit in the UART Control 1 Register to 1.

When configured as a general purpose timer, the interrupt interval is calculated using the following equation:

$$\text{Interrupt Interval (s)} = \text{System Clock Period (s)} \times \text{BRG}[15:0]$$

UART Control Register Definitions

The UART control registers support the UART and the associated infrared encoder/decoders. For more information about the infrared operation, see the [Infrared Encoder/Decoder](#) chapter on page 117.

UART Transmit Data Register

Data bytes written to the UART Transmit Data Register (Table 64) are shifted out on the TXD_x pin. The Write-only UART Transmit Data Register shares a Register File address with the read-only UART Receive Data Register.

Table 64. UART Transmit Data Register (U0TXD)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|---|---|---|---|---|---|---|
| Field | TXD | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | W | W | W | W | W | W | W | W |
| Address | F40H | | | | | | | |

| Bit | Description |
|--------------|--|
| [7:0] TXD | Transmit Data UART transmitter data byte to be shifted out through the TXD _x pin. |

UART Receive Data Register

Data bytes received through the RXD_x pin are stored in the UART Receive Data Register (Table 65). The read-only UART Receive Data Register shares a Register File address with the Write-only UART Transmit Data Register.

Table 65. UART Receive Data Register (U0RXD)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|---|---|---|---|---|---|---|
| Field | RXD | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R | R | R | R | R | R | R | R |
| Address | F40H | | | | | | | |

| Bit | Description |
|--------------|---|
| [7:0] RXD | Receive Data UART receiver data byte from the RXD _x pin. |

UART Status 0 Register

The UART Status 0 and Status 1 registers (Table 66 and Table 67) identify the current UART operating configuration and status.

Table 66. UART Status 0 Register (U0STAT0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|----|----|----|------|------|-----|-----|
| Field | RDA | PE | OE | FE | BRKD | TDRE | TXE | CTS |
| RESET | 0 | 0 | 0 | 0 | 0 | 1 | 1 | X |
| R/W | R | R | R | R | R | R | R | R |
| Address | F41H | | | | | | | |

| Bit | Description |
|-------------|--|
| [7] RDA | Receive Data Available This bit indicates that the UART Receive Data Register has received data. Reading the UART Receive Data Register clears this bit. 0 = The UART Receive Data Register is empty. 1 = There is a byte in the UART Receive Data Register. |
| [6] PE | Parity Error This bit indicates that a parity error has occurred. Reading the UART Receive Data register clears this bit. 0 = No parity error has occurred. 1 = A parity error has occurred. |
| [5] OE | Overrun Error This bit indicates that an overrun error has occurred. An overrun occurs when new data is received and the UART Receive Data Register has not been read. If the RDA bit is reset to 0, reading the UART Receive Data Register clears this bit. 0 = No overrun error occurred. 1 = An overrun error occurred. |
| [4] FE | Framing Error This bit indicates that a framing error (no Stop bit following data reception) was detected. Reading the UART Receive Data Register clears this bit. 0 = No framing error occurred. 1 = A framing error occurred. |
| [3] BRKD | Break Detect This bit indicates that a break occurred. If the data bits, parity/multiprocessor bit, and Stop bit(s) are all 0s this bit is set to 1. Reading the UART Receive Data Register clears this bit. 0 = No break occurred. 1 = A break occurred. |

| Bit | Description (Continued) |
|-------------|--|
| [2] TDRE | Transmitter Data Register Empty This bit indicates that the UART Transmit Data Register is empty and ready for additional data. Writing to the UART Transmit Data Register resets this bit. 0 = Do not write to the UART Transmit Data Register. 1 = The UART Transmit Data Register is ready to receive an additional byte to be transmitted. |
| [1] TXE | Transmitter Empty This bit indicates that the transmit shift register is empty and character transmission is finished. 0 = Data is currently transmitting. 1 = Transmission is complete. |
| [0] CTS | CTS Signal When this bit is read, it returns the level of the $\overline{\text{CTS}}$ signal. This signal is active Low. |

UART Status 1 Register

This register contains multiprocessor control and status bits.

Table 67. UART Status 1 Register (U0STAT1)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|---|---|---|-----|-----|--------|------|
| Field | Reserved | | | | | | NEWFRM | MPRX |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R | R |
| Address | F44H | | | | | | | |

| Bit | Description |
|---------------|--|
| [7:2] | Reserved These bits are reserved; R/W bits must be programmed to 000000 during writes and 000000 when read. |
| [1] NEWFRM | New Frame A status bit denoting the start of a new frame. Reading the UART Receive Data Register resets this bit to 0. 0 = The current byte is not the first data byte of a new frame. 1 = The current byte is the first data byte of a new frame. |
| [0] MPRX | Multiprocessor Receive Returns the value of the most recent multiprocessor bit received. Reading from the UART Receive Data Register resets this bit to 0. |

UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (Table 68 and Table 69) configure the properties of the UART's transmit and receive operations. The UART Control registers must not be written while the UART is enabled.

Table 68. UART Control 0 Register (U0CTL0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|-----|------|-----|------|------|------|------|
| Field | TEN | REN | CTSE | PEN | PSEL | SBRK | STOP | LBEN |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F42H | | | | | | | |

| Bit | Description |
|-------------|---|
| [7] TEN | Transmit Enable This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled. 0 = Transmitter disabled. 1 = Transmitter enabled. |
| [6] REN | Receive Enable This bit enables or disables the receiver. 0 = Receiver disabled. 1 = Receiver enabled. |
| [5] CTSE | CTSE—CTS Enable 0 = The CTS signal has no effect on the transmitter. 1 = The UART recognizes the CTS signal as an enable control from the transmitter. |
| [4] PEN | Parity Enable This bit enables or disables parity. Even or odd is determined by the PSEL bit. 0 = Parity is disabled. 1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit . |
| [3] PSEL | Parity Select 0 = Even parity is transmitted and expected on all received data. 1 = Odd parity is transmitted and expected on all received data. |
| [2] SBRK | Send Break This bit pauses or breaks data transmission. Sending a break interrupts any transmission in progress, so ensure that the transmitter has finished sending data before setting this bit. 0 = No break is sent. 1 = Forces a break condition by setting the output of the transmitter to zero. |

| Bit | Description (Continued) |
|-------------|---|
| [1] STOP | Stop Bit Select 0 = The transmitter sends one stop bit. 1 = The transmitter sends two stop bits. |
| [0] LBEN | Loop Back Enable 0 = Normal operation. 1 = All transmitted data is looped back to the receiver. |

Table 69. UART Control 1 Register (U0CTL1)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------------|---------|------|---------|------|-------|--------|--------|------|
| Field | MPMD[1] | MPEN | MPMD[0] | MPBT | DEPOL | BRGCTL | RDAIRQ | IREN |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F43H | | | | | | | |

| Bit | Description |
|--------------------|--|
| [7,5] MPMD[1:0] | MULTIPROCESSOR Mode If MULTIPROCESSOR (9-bit) Mode is enabled. 00 = The UART generates an interrupt request on all received bytes (data and address). 01 = The UART generates an interrupt request only on received address bytes. 10 = The UART generates an interrupt request when a received address byte matches the value stored in the Address Compare Register and on all successive data bytes until an address mismatch occurs. 11 = The UART generates an interrupt request on all received data bytes for which the most recent address byte matched the value in the Address Compare Register. |
| [6] MPEN | MULTIPROCESSOR (9-bit) Enable This bit is used to enable MULTIPROCESSOR (9-bit) Mode. 0 = Disable MULTIPROCESSOR (9-bit) Mode. 1 = Enable MULTIPROCESSOR (9-bit) Mode. |
| [4] MPBT | Multiprocessor Bit Transmit This bit is applicable only when MULTIPROCESSOR (9-bit) Mode is enabled. The 9th bit is used by the receiving device to determine if the data byte contains address or data information. 0 = Send a 0 in the multiprocessor bit location of the data stream (data byte). 1 = Send a 1 in the multiprocessor bit location of the data stream (address byte). |
| [3] DEPOL | Driver Enable Polarity 0 = DE signal is Active High. 1 = DE signal is Active Low. |

| Bit | Description (Continued) |
|---------------|--|
| [2] BRGCTL | <p>Baud Rate Control</p> <p>This bit causes an alternate UART behavior depending on the value of the REN bit in the UART Control 0 Register.</p> <p>When the UART receiver is not enabled (REN=0), this bit determines whether the Baud Rate Generator issues interrupts.</p> <p>0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value. 1 = The Baud Rate Generator generates a receive interrupt when it counts down to 0.</p> <p>Reads from the Baud Rate High and Low Byte registers return the current BRG count value. When the UART receiver is enabled (REN=1), this bit allows reads from the Baud Rate Registers to return the BRG count value instead of the Reload Value.</p> <p>0 = Reads from the Baud Rate High and Low Byte registers return the BRG Reload Value. 1 = Reads from the Baud Rate High and Low Byte registers return the current BRG count value. Unlike the Timers, there is no mechanism to latch the Low Byte when the High Byte is read.</p> |
| [1] RDAIRQ | <p>Receive Data Interrupt Enable</p> <p>0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller.</p> <p>1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.</p> |
| [0] IREN | <p>Infrared Encoder/Decoder Enable</p> <p>0 = Infrared encoder/decoder is disabled. UART operates normally.</p> <p>1 = Infrared encoder/decoder is enabled. The UART transmits and receives data through the infrared encoder/decoder.</p> |

UART Address Compare Register

The UART Address Compare Register stores the multinode network address of the UART. When the MPMD[1] bit of UART Control Register 0 is set, all incoming address bytes are compared to the value stored in the Address Compare Register. Receive interrupts and RDA assertions only occur in the event of a match.

Table 70. UART Address Compare Register (U0ADDR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-----------|-----|-----|-----|-----|-----|-----|-----|
| Field | COMP_ADDR | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F45H | | | | | | | |

| Bit | Description |
|-----------|---|
| [7:0] | Compare Address |
| COMP_ADDR | This 8-bit value is compared to incoming address bytes. |

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High and Low Byte registers (Table 71 and Table 72) combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART.

Table 71. UART Baud Rate High Byte Register (U0BRH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|-----|-----|-----|-----|-----|-----|-----|
| Field | BRH | | | | | | | |
| RESET | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F46H | | | | | | | |

Table 72. UART Baud Rate Low Byte Register (U0BRL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|-----|-----|-----|-----|-----|-----|-----|
| Field | BRL | | | | | | | |
| RESET | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F47H | | | | | | | |

The UART data rate is calculated using the following equation:

$$\text{UART Baud Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

For a given UART data rate, calculate the integer baud rate divisor value using the following equation:

$$\text{UART Baud Rate Divisor Value (BRG)} = \text{Round}\left(\frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Data Rate (bits/s)}}\right)$$

The baud rate error relative to the acceptable baud rate is calculated using the following equation:

$$\text{UART Baud Rate Error (\%)} = 100 \times \left(\frac{\text{Actual Data Rate} - \text{Desired Data Rate}}{\text{Desired Data Rate}}\right)$$

For reliable communication, the UART baud rate error must never exceed five percent. Table 73 provides information about data rate errors for a 5.5296MHz System Clock.

Table 73. UART Baud Rates

| 5.5296MHz System Clock | | | |
|------------------------|-----------------------|-------------------|-----------|
| Acceptable Rate (kHz) | BRG Divisor (Decimal) | Actual Rate (kHz) | Error (%) |
| 1250.0 | N/A | N/A | N/A |
| 625.0 | N/A | N/A | N/A |
| 250.0 | 1 | 345.6 | 38.24 |
| 115.2 | 3 | 115.2 | 0.00 |
| 57.6 | 6 | 57.6 | 0.00 |
| 38.4 | 9 | 38.4 | 0.00 |
| 19.2 | 18 | 19.2 | 0.00 |
| 9.60 | 36 | 9.60 | 0.00 |
| 4.80 | 72 | 4.80 | 0.00 |
| 2.40 | 144 | 2.40 | 0.00 |
| 1.20 | 288 | 1.20 | 0.00 |
| 0.60 | 576 | 0.60 | 0.00 |
| 0.30 | 1152 | 0.30 | 0.00 |

Infrared Encoder/Decoder

Z8 Encore! XP F0823 Series products contain a fully-functional, high-performance UART with an infrared encoder/decoder (endec). The infrared endec is integrated with an on-chip UART to allow easy communication between the Z8 Encore! XP and IrDA Physical Layer Specification, Version 1.3-compliant infrared transceivers. Infrared communication provides secure, reliable, low-cost, point-to-point communication between PCs, PDAs, cell phones, printers and other infrared enabled devices.

Architecture

Figure 16 displays the architecture of the infrared endec.

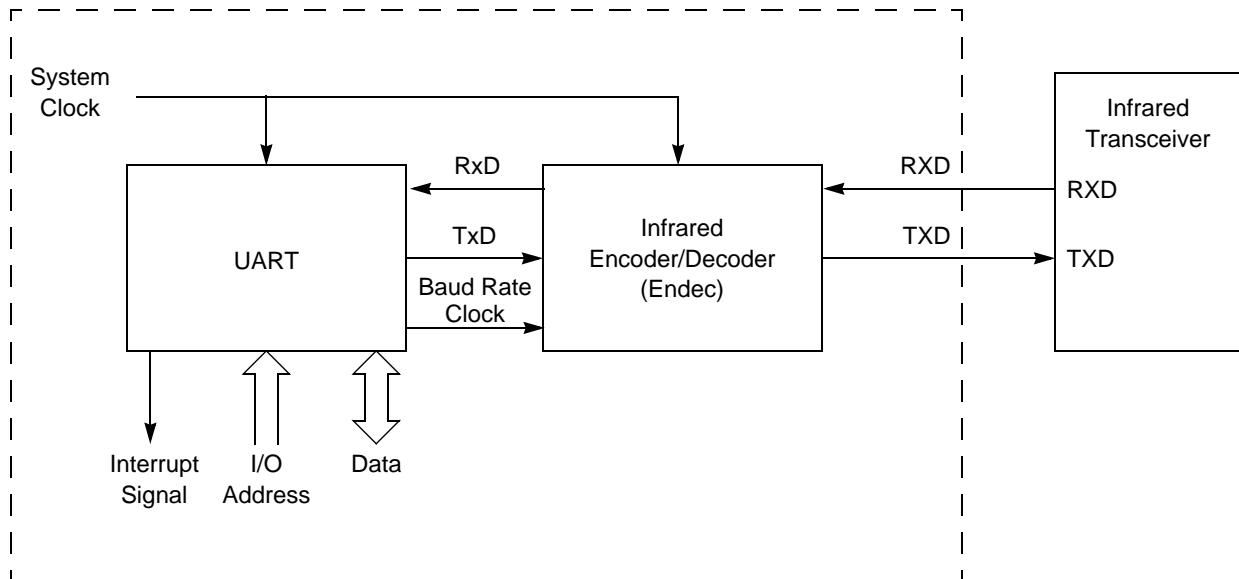


Figure 16. Infrared Data Communication System Block Diagram

Operation

When the infrared endec is enabled, the transmit data from the associated on-chip UART is encoded as digital signals in accordance with the IrDA standard and output to the infrared transceiver through the TXD pin. Similarly, data received from the infrared transceiver is passed to the infrared endec through the RXD pin, decoded by the infrared endec, and

passed to the UART. Communication is half-duplex, which means simultaneous data transmission and reception is not allowed.

The baud rate is set by the UART's baud rate generator and supports IrDA standard baud rates from 9600 baud to 115.2 kbaud. Higher baud rates are possible, but do not meet IrDA specifications. The UART must be enabled to use the infrared endec. The infrared endec data rate is calculated using the following equation:

$$\text{Infrared Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value}}$$

Transmitting IrDA Data

The data to be transmitted using the infrared transceiver is first sent to the UART. The UART's transmit signal (TXD) and baud rate clock are used by the IrDA to generate the modulation signal (IR_TXD) that drives the infrared transceiver. Each UART/Infrared data bit is 16 clocks wide. If the data to be transmitted is 1, the IR_TXD signal remains low for the full 16 clock period. If the data to be transmitted is 0, the transmitter first outputs a 7 clock low period, followed by a 3 clock high pulse. Finally, a 6 clock low pulse is output to complete the full 16 clock data period. Figure 17 displays IrDA data transmission. When the infrared endec is enabled, the UART's TXD signal is internal to Z8 Encore! XP F0823 Series products while the IR_TXD signal is output through the TXD pin.



Figure 17. Infrared Data Transmission

Receiving IrDA Data

Data received from the infrared transceiver using the IR_RXD signal through the RXD pin is decoded by the infrared endec and passed to the UART. The UART's baud rate clock is used by the infrared endec to generate the demodulated signal (RXD) that drives the UART. Each UART/Infrared data bit is 16-clocks wide. Figure 18 displays data reception. When the infrared endec is enabled, the UART's RXD signal is internal to the Z8 Encore! XP F0823 Series products while the IR_RXD signal is received through the RXD pin.



Figure 18. IrDA Data Reception

Infrared Data Reception



Caution: The system clock frequency must be at least 1.0MHz to ensure proper reception of the 1.4μs minimum width pulses allowed by the IrDA standard.

Endec Receiver Synchronization

The IrDA receiver uses a local baud rate clock counter (0 to 15 clock periods) to generate an input stream for the UART and to create a sampling window for detection of incoming pulses. The generated UART input (UART RXD) is delayed by 8 baud rate clock periods with respect to the incoming IrDA data stream. When a falling edge in the input data stream is detected, the endec counter is reset. When the count reaches a value of 8, the UART RXD value is updated to reflect the value of the decoded data. When the count reaches 12 baud clock periods, the sampling window for the next incoming pulse opens.

The window remains open until the count again reaches 8 (that is, 24 baud clock periods since the previous pulse was detected), giving the endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal, allowing the endec to tolerate jitter and baud rate errors in the incoming datastream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a Start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART control registers as defined in the [Universal Asynchronous Receiver/Transmitter](#) chapter on page 97.



Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the endec before enabling the GPIO port alternate function for the corresponding pin.

Analog-to-Digital Converter

The Analog-to-Digital Converter (ADC) converts an analog input signal to its digital representation. The features of this sigma-delta ADC include:

- 10-bit resolution
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports
- Interrupt upon conversion complete
- Bandgap generated internal voltage reference generator with two selectable levels
- Factory offset and gain calibration

Architecture

Figure 19 displays the major functional blocks of the ADC. An analog multiplexer network selects the ADC input from the available analog pins, ANA0 through ANA7.



Figure 19. Analog-to-Digital Converter Block Diagram

Operation

The output of the ADC is an 11-bit, signed, two's-complement digital value. The output generally ranges from 0 to +1023, but offset errors can cause small negative values.

The ADC registers return 13 bits of data, but the two LSBs are intended for compensation use only. When the compensation routine is performed on the 13 bit raw ADC value, two bits of resolution are lost because of a rounding error. As a result, the final value is an 11-bit number.

Automatic Powerdown

If the ADC is idle (no conversions in progress) for 160 consecutive system clock cycles, portions of the ADC are automatically powered down. From this powerdown state, the ADC requires 40 system clock cycles to powerup. The ADC powers up when a conversion is requested by the ADC Control Register.

Single-Shot Conversion

When configured for single-shot conversion, the ADC performs a single analog-to-digital conversion on the selected analog input channel. After completion of the conversion, the ADC shuts down. Observe the following steps for setting up the ADC and initiating a single-shot conversion:

1. Enable the acceptable analog inputs by configuring the general-purpose I/O pins for alternate function. This configuration disables the digital input and output drivers.
2. Write the ADC Control/Status Register 1 to configure the ADC
 - Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.
3. Write to the ADC Control Register 0 to configure the ADC and begin the conversion. The bit fields in the ADC Control Register can be written simultaneously:
 - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
 - Clear CONT to 0 to select a single-shot conversion.
 - If the internal voltage reference must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
 - Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in the ADC Control Register 0.
 - Set CEN to 1 to start the conversion.
4. CEN remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power-up before beginning the 5129 cycle conversion.
5. When the conversion is complete, the ADC control logic performs the following operations:
 - 11-bit two's-complement result written to {ADCD_H[7:0], ADCD_L[7:5]}

- CEN resets to 0 to indicate the conversion is complete
6. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

Continuous Conversion

When configured for continuous conversion, the ADC continuously performs an analog-to-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated after each conversion.



Caution: In CONTINUOUS Mode, ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not detected at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.

Observe the following steps for setting up the ADC and initiating continuous conversion:

1. Enable the acceptable analog input by configuring the general-purpose I/O pins for alternate function. This action disables the digital input and output driver.
2. Write the ADC Control/Status Register 1 to configure the ADC:
 - Write the REFSELH bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELH bit is contained in the ADC Control/Status Register 1.
3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control Register can be written simultaneously:
 - Write to the ANAIN[3:0] field to select from the available analog input sources (different input pins available depending on the device).
 - Set CONT to 1 to select continuous conversion.
 - If the internal VREF must be output to a pin, set the REFEXT bit to 1. The internal voltage reference must be enabled in this case.
 - Write the REFSELL bit of the pair {REFSELH, REFSELL} to select the internal voltage reference level or to disable the internal reference. The REFSELL bit is contained in ADC Control Register 0.
 - Set CEN to 1 to start the conversions.

4. When the first conversion in continuous operation is complete (after 5129 system clock cycles, plus the 40 cycles for power-up, if necessary), the ADC control logic performs the following operations:
 - CEN resets to 0 to indicate the first conversion is complete. CEN remains 0 for all subsequent conversions in continuous operation
 - An interrupt request is sent to the Interrupt Controller to indicate the conversion is complete

5. The ADC writes a new data result every 256 system clock cycles. For each completed conversion, the ADC control logic performs the following operations:
 - Writes the 11-bit two's complement result to {ADCD_H[7:0], ADCD_L[7:5]}
 - An interrupt request to the Interrupt Controller denoting conversion complete

6. To disable continuous conversion, clear the CONT bit in the ADC Control Register to 0.

Interrupts

The ADC is able to interrupt the CPU whenever a conversion has been completed and the ADC is enabled.

When the ADC is disabled, an interrupt is not asserted; however, an interrupt pending when the ADC is disabled is not cleared.

Calibration and Compensation

Z8 Encore! XP F0823 Series ADC can be factory calibrated for offset error and gain error, with the compensation data stored in Flash memory. Alternatively, user code can perform its own calibration, storing the values into Flash themselves.

Factory Calibration

Devices that have been factory calibrated contain nine bytes of calibration data in the Flash option bit space. This data consists of three bytes for each reference type. For a list of input modes for which calibration data exists, see the [Zilog Calibration Data](#) section on page 152. There is 1 byte for offset, and there are 2 bytes for gain correction.

User Calibration

If you have precision references available, its own external calibration can be performed, storing the values into Flash themselves.

Software Compensation Procedure

The value read from the ADC high and low byte registers are uncompensated. The user mode software must apply gain and offset correction to this uncompensated value for maximum accuracy. The following formula yields the compensated value:

$$ADC_{comp} = (ADC_{uncomp} - OFFCAL) + ((ADC_{uncomp} - OFFCAL) * GAINCAL) / 2^7$$

where GAINCAL is the gain calibration byte, OFFCAL is the offset calibration byte and ADC_{uncomp} is the uncompensated value read from the ADC. The OFFCAL value is in two's complement format, as are the compensated and uncompensated ADC values.

► **Note:** The offset compensation is performed first, followed by the gain compensation. One bit of resolution is lost because of rounding on both the offset and gain computations. As a result the ADC registers read back 13 bits: 1 sign bit, two calibration bits lost to rounding and 10 data bits. Also note that in the second term, the multiplication must be performed before the division by 2^{16} . Otherwise, the second term evaluates to zero incorrectly.

 **Caution:** Although the ADC can be used without the gain and offset compensation, it does exhibit non-unity gain. Designing the ADC with sub-unity gain reduces noise across the ADC range but requires the ADC results to be scaled by a factor of 8/7.

ADC Control Register Definitions

The following sections define the ADC Control registers.

ADC Control Register 0

The ADC Control Register selects the analog input channel and initiates the analog-to-digital conversion.

Table 74. ADC Control Register 0 (ADCCTL0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|---------|--------|------|------------|-----|-----|-----|
| Field | CEN | REFSELL | REFEXT | CONT | ANAIN[3:0] | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F70H | | | | | | | |

| Bit | Description |
|----------------|---|
| [7] CEN | <p>Conversion Enable</p> <p>0 = Conversion is complete. Writing a 0 produces no effect. The ADC automatically clears this bit to 0 when a conversion is complete.</p> <p>1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.</p> |
| [6] REFSELL | <p>Voltage Reference Level Select Low Bit</p> <p>In conjunction with the High bit (REFSELH) in ADC Control/Status Register 1, this determines the level of the internal voltage reference; the following details the effects of {REFSELH, REFSELL}. This reference is independent of the Comparator reference.</p> <p>00 = Internal Reference Disabled, reference comes from external pin.</p> <p>01 = Internal Reference set to 1.0V.</p> <p>10 = Internal Reference set to 2.0V (default).</p> |
| [5] REFEXT | <p>External Reference Select</p> <p>0 = External reference buffer is disabled; V_{REF} pin is available for GPIO functions.</p> <p>1 = The internal ADC reference is buffered and connected to the V_{REF} pin.</p> |
| [4] CONT | <p>Continuous Conversion</p> <p>0 = Single-shot conversion. ADC data is output once at completion of the 5129 system clock cycles.</p> <p>1 = Continuous conversion. ADC data updated every 256 system clock cycles.</p> |

| Bit | Description (Continued) |
|----------------|--|
| [3:0] ANAIN | <p>Analog Input Select</p> <p>These bits select the analog input for conversion. Not all port pins in this list are available in all packages for Z8 Encore! XP F0823 Series. For information about the port pins available with each package style, see the Pin Description section on page 7. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1.</p> <p>For the reserved values, all input switches are disabled to avoid leakage or other undesirable operation. ADC samples taken with reserved bit settings are undefined.</p> <p>Single-Ended:</p> <p>0000 = ANA0. 0001 = ANA1. 0010 = ANA2. 0011 = ANA3. 0100 = ANA4. 0101 = ANA5. 0110 = ANA6. 0111 = ANA7. 1000 = Reserved. 1001 = Reserved. 1010 = Reserved. 1011 = Reserved. 1100 = Reserved. 1101 = Reserved. 1110 = Reserved. 1111 = Reserved.</p> |

ADC Control/Status Register 1

The second ADC Control Register contains the voltage reference level selection bit.

Table 75. ADC Control/Status Register 1 (ADCCTL1)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------|----------|-----|-----|-----|-----|-----|-----|
| Field | REFSELH | Reserved | | | | | | |
| RESET | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F71H | | | | | | | |

| Bit | Description |
|----------------|--|
| [7] REFSELH | <p>Voltage Reference Level Select High Bit</p> <p>In conjunction with the Low bit (REFSELL) in ADC Control Register 0, this bit determines the level of the internal voltage reference; the following details the effects of {REFSELH, REFSELL}; this reference is independent of the Comparator reference.</p> <p>00 = Internal Reference Disabled, reference comes from external pin. 01 = Internal Reference set to 1.0V. 10 = Internal Reference set to 2.0V (default).</p> |
| [6:0] | <p>Reserved</p> <p>These bits are reserved and must be programmed to 0000000.</p> |

ADC Data High Byte Register

The ADC Data High Byte Register contains the upper eight bits of the ADC output. The output is an 11-bit two's complement value. During a single-shot conversion, this value is invalid. Access to the ADC Data High Byte register is read-only. Reading the ADC Data High Byte Register latches data in the ADC Low Bits Register.

Table 76. ADC Data High Byte Register (ADCD_H)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------|---|---|---|---|---|---|---|
| Field | ADCDH | | | | | | | |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R | R | R | R | R | R | R | R |
| Address | F72H | | | | | | | |

| Bit | Description |
|-------|--|
| [7:0] | ADC Data High Byte |
| ADCDH | This byte contains the upper eight bits of the ADC output. These bits are not valid during a single-shot conversion. During a continuous conversion, the most recent conversion output is held in this register. These bits are undefined after a Reset. |

ADC Data Low Bits Register

The ADC Data Low Byte register contains the lower bits of the ADC output as well as an overflow status bit. The output is a 11-bit two's complement value. During a single-shot conversion, this value is invalid. Access to the ADC Data Low Byte register is read-only. Reading the ADC Data High Byte register latches data in the ADC Low Bits Register.

Table 77. ADC Data Low Bits Register (ADCD_L)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------|---|---|----------|---|---|---|-----|
| Field | ADCDL | | | Reserved | | | | OVF |
| RESET | X | X | X | X | X | X | X | X |
| R/W | R | R | R | R | R | R | R | R |
| Address | F73H | | | | | | | |

| Bit | Description |
|----------------|--|
| [7:5] ADCDL | ADC Data Low Bits These bits are the least significant three bits of the 11-bits of the ADC output. These bits are undefined after a Reset. |
| [4:1] | Reserved These bits are reserved and are undefined when read. |
| [0] OVF | Overflow Status 0 = An overflow did not occur in the digital filter for the current sample. 1 = An overflow did occur in the digital filter for the current sample. |

Comparator

Z8 Encore! XP F0823 Series devices feature a general purpose comparator that compares two analog input signals. A GPIO (CINP) pin provides the positive comparator input. The negative input (CINN) can be taken from either an external GPIO pin or an internal reference. The output is available as an interrupt source or can be routed to an external pin using the GPIO multiplex.

The features of the comparator include:

- Two inputs which can be connected up using the GPIO multiplex (MUX)
- One input can be connected to a programmable internal reference
- One input can be connected to the on-chip temperature sensor
- Output can be either an interrupt source or an output to an external pin

Operation

One of the comparator inputs can be connected to an internal reference which is a user selectable reference that is user programmable with 200 mV resolution.

The comparator can be powered down to save on supply current. For details, see the [Power Control Register 0](#) section on page 31.



Caution: Because of the propagation delay of the comparator, Zilog does not recommend enabling or reconfiguring the comparator without first disabling the interrupts and waiting for the comparator output to settle. Doing so can result in spurious interrupts.

The following example shows how to safely enable the comparator:

```
di
ld cmp0
nop
nop      ; wait for output to settle
clr irq0 ; clear any spurious interrupts pending
ei
```

Comparator Control Register Definition

The Comparator Control Register (CMPCTL) configures the comparator inputs and sets the value of the internal voltage reference.

Table 78. Comparator Control Register (CMP0)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------|--------|--------|-----|-----|-----|----------|-----|
| Field | INPSEL | INNSEL | REFLVL | | | | Reserved | |
| RESET | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F90H | | | | | | | |

| Bit | Description |
|-----------------|--|
| [7] INPSEL | Signal Select for Positive Input 0 = GPIO pin used as positive comparator input. 1 = temperature sensor used as positive comparator input. |
| [6] INNSEL | Signal Select for Negative Input 0 = internal reference disabled, GPIO pin used as negative comparator input. 1 = internal reference enabled as negative comparator input. |
| [5:2] REFLVL | Internal Reference Voltage Level 0000 = 0.0V. 0001 = 0.2V. 0010 = 0.4V. 0011 = 0.6V. 0100 = 0.8V. 0101 = 1.0V (Default). 0110 = 1.2V. 0111 = 1.4V. 1000 = 1.6V. 1001 = 1.8V. 1010–1111 = Reserved. Note: This reference is independent of the ADC voltage reference. |
| [1:0] | Reserved These bits are reserved; R/W bits must be programmed to 00 during writes and to 00 when read. |

Flash Memory

The products in Z8 Encore! XP F0823 Series features either 8KB (8192), 4KB (4096), 2KB (2048) or 1KB (1024) of nonvolatile Flash memory with read/write/erase capability. Flash Memory can be programmed and erased in-circuit by either user code or through the On-Chip Debugger.

The Flash Memory array is arranged in pages with 512 bytes per page. The 512-byte page is the minimum Flash block size that can be erased. Each page is divided into 8 rows of 64 bytes.

For program/data protection, the Flash memory is also divided into sectors. In the Z8 Encore! XP F0823 Series, these sectors are either 1024 bytes (in the 8KB devices) or 512 bytes in size (all other memory sizes); each sector maps to a page. Page and sector sizes are not generally equal.

The first two bytes of the Flash program memory are used as Flash Option bits. For more information about their operation, see the [Flash Option Bits](#) chapter on page 146.

Table 79 describes the Flash memory configuration for each device in the Z8 Encore! XP F0823 Series. Figure 20 displays the Flash memory arrangement.

Table 79. Z8 Encore! XP F0823 Series Flash Memory Configurations

| Part Number | Flash Size KB (Bytes) | Flash Pages | Program Memory Addresses | Flash Sector Size (bytes) |
|-------------|--------------------------|-------------|--------------------------------|------------------------------|
| Z8F08x3 | 8 (8192) | 16 | 0000H–1FFFH | 1024 |
| Z8F04x3 | 4 (4096) | 8 | 0000H–0FFFH | 512 |
| Z8F02x3 | 2 (2048) | 4 | 0000H–07FFH | 512 |
| Z8F01x3 | 1 (1024) | 2 | 0000H–03FFH | 512 |

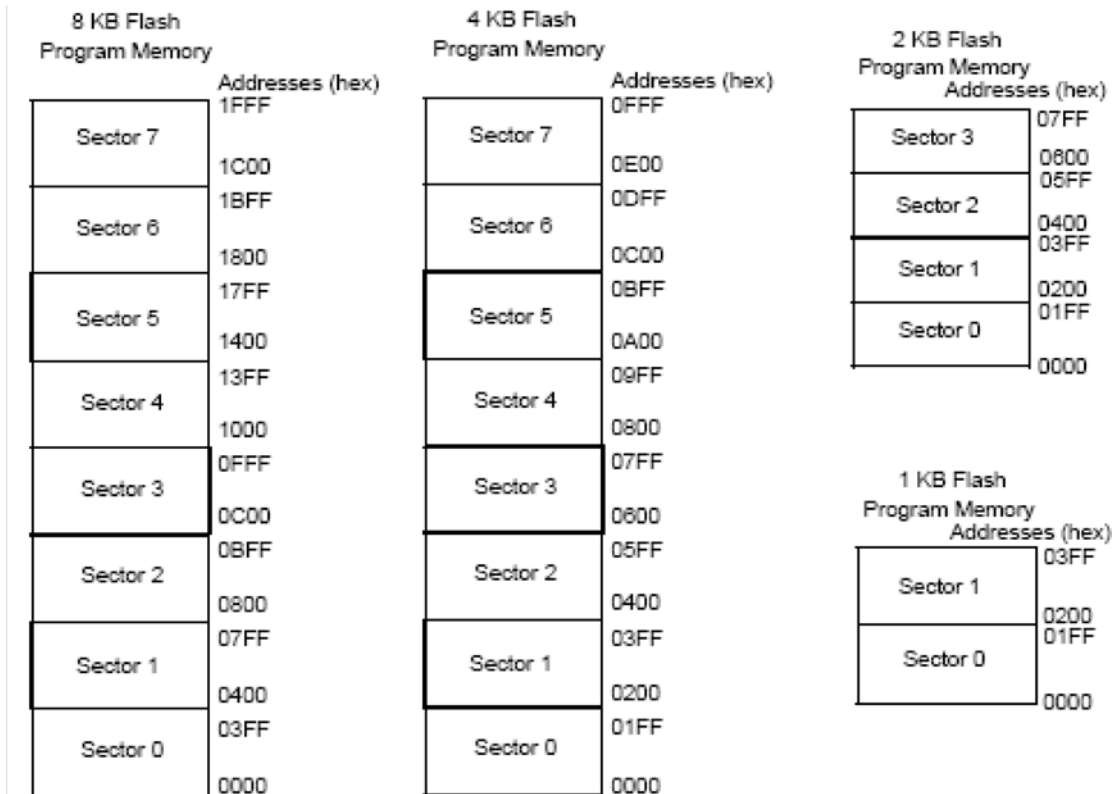


Figure 20. Flash Memory Arrangement

Flash Information Area

The Flash information area is separate from program memory and is mapped to the address range FE00H to FFFFH. Not all these addresses are accessible. Factory trim values for the analog peripherals are stored here. Factory calibration data for the ADC is also stored here.

Operation

The Flash Controller programs and erases Flash memory. The Flash Controller provides the proper Flash controls and timing for Byte Programming, Page Erase, and Mass Erase of Flash memory.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels.

Figure 21 displays a basic Flash Controller flow. The following subsections provide details about the various operations (Lock, Unlock, Byte Programming, Page Protect, Page Unprotect, Page Select Page Erase, and Mass Erase) displayed in Figure 21.



Figure 21. Flash Controller Operation Flowchart

Flash Operation Timing Using the Flash Frequency Registers

Before performing either a program or erase operation on Flash memory, you must first configure the Flash Frequency High and Low Byte registers. The Flash Frequency registers allow programming and erasing of the Flash with system clock frequencies ranging from 32kHz (32768Hz) through 20MHz.

The Flash Frequency High and Low Byte registers combine to form a 16-bit value, `FFREQ`, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz). This value is calculated using the following equation:

$$\text{FFREQ}[15:0] = \frac{\text{System Clock Frequency (Hz)}}{1000}$$



Caution: Flash programming and erasure are not supported for system clock frequencies below 32kHz (32768 Hz) or above 20MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure operation of Z8 Encore! XP F0823 Series devices.

Flash Code Protection Against External Access

The user code contained within the Flash memory can be protected against external access with the On-Chip Debugger. Programming the FRP Flash Option Bit prevents reading of the user code with the On-Chip Debugger. For more information, see the [Flash Option Bits](#) section on page 146 and the [On-Chip Debugger](#) chapter on page 156.

Flash Code Protection Against Accidental Program and Erasure

F0823 Series provides several levels of protection against accidental program and erasure of the Flash memory contents. This protection is provided by a combination of the Flash Option bits, the register locking mechanism, the page select redundancy and the sector level protection control of the Flash Controller.

Flash Code Protection Using the Flash Option Bits

The FRP and FWP Flash Option Bits combine to provide three levels of Flash Program Memory protection as listed in Table 80. For more information, see the [Flash Option Bits](#) section on page 146.

Table 80. Flash Code Protection Using the Flash Option Bits

| FWP | Flash Code Protection Description |
|-----|--|
| 0 | Programming and erasing disabled for all of Flash Program Memory. In user code programming, Page Erase, and Mass Erase are all disabled. Mass Erase is available through the On-Chip Debugger. |
| 1 | Programming, Page Erase, and Mass Erase are enabled for all of Flash Program Memory. |

Flash Code Protection Using the Flash Controller

At Reset, the Flash Controller locks to prevent accidental program or erasure of the Flash memory. To program or erase the Flash memory, first write the Page Select Register with the target page. Unlock the Flash Controller by making two consecutive writes to the Flash Control Register with the values 73H and 8CH, sequentially. The Page Select Register must be rewritten with the same page previously stored there. If the two Page Select writes do not match, the controller reverts to a locked state. If the two writes match, the selected page becomes active. For more details, see Figure 21.

After unlocking a specific page, you can enable either Page Program or Erase. Writing the value 95H causes a Page Erase only if the active page resides in a sector that is not protected. Any other value written to the Flash Control Register locks the Flash Controller. Mass Erase is not allowed in the user code but only in through the Debug Port.

After unlocking a specific page, you can also write to any byte on that page. After a byte is written, the page remains unlocked, allowing for subsequent writes to other bytes on the same page. Further writes to the Flash Control Register cause the active page to revert to a locked state.

Sector-Based Flash Protection

The final protection mechanism is implemented on a per-sector basis. The Flash memories of Z8 Encore! XP devices are divided into maximum number of 8 sectors. A sector is 1/8 of the total Flash memory size unless this value is smaller than the page size – in which case, the sector and page sizes are equal. On Z8 Encore! F0823 Series devices, the sector size is varied according to the Flash memory configuration shown in [Table 79](#) on page 134.

The Flash Sector Protect Register can be configured to prevent sectors from being programmed or erased. After a sector is protected, it cannot be unprotected by user code. The Flash Sector Protect Register is cleared after reset, and any previously-written protection values are lost. User code must write this register in their initialization routine if they prefer to enable sector protection.

The Flash Sector Protect Register shares its Register File address with the Page Select Register. The Flash Sector Protect Register is accessed by writing the Flash Control Register with 5EH. After the Flash Sector Protect Register is selected, it can be accessed at the Page Select Register address. When user code writes the Flash Sector Protect Register,

bits can only be set to 1. Thus, sectors can be protected, but not unprotected, via register write operations. Writing a value other than 5EH to the Flash Control Register deselects the Flash Sector Protect Register and reenables access to the Page Select Register.

Observe the following procedure to setup the Flash Sector Protect Register from user code:

1. Write 00H to the Flash Control Register to reset the Flash Controller.
2. Write 5EH to the Flash Control Register to select the Flash Sector Protect Register.
3. Read and/or write the Flash Sector Protect Register which is now at Register File address FF9H.
4. Write 00H to the Flash Control Register to return the Flash Controller to its reset state.

The Sector Protect Register is initialized to 0 on reset, putting each sector into an unprotected state. When a bit in the Sector Protect Register is written to 1, the corresponding sector can no longer be written or erased by the CPU. External Flash programming through the OCD or via the Flash Controller Bypass mode are unaffected. After a bit of the Sector Protect Register has been set, it cannot be cleared except by powering down the device.

Byte Programming

The Flash Memory is enabled for byte programming after unlocking the Flash Controller and successfully enabling either Mass Erase or Page Erase. When the Flash Controller is unlocked and Mass Erase is successfully completed, all Program Memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and Page Erase is successfully enabled, only the locations of the selected page are available for byte programming. An erased Flash byte contains all 1's (FFH). The programming operation can only be used to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the Page Erase or Mass Erase commands.

Byte Programming is accomplished using the On-Chip Debugger's Write Memory command or eZ8 CPU execution of the LDC or LDCI instructions. For a description of the LDC and LDCI instructions, refer to the [eZ8 CPU Core User Manual \(UM0128\)](#), available for download at www.zilog.com. While the Flash Controller programs the Flash memory, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. To exit programming mode and lock the Flash, write any value to the Flash Control Register, except the Mass Erase or Page Erase commands.



Caution: The byte at each address of the Flash memory cannot be programmed (any bits written to 0) more than twice before an erase cycle occurs. Doing so may result in corrupted data at the target byte.

Page Erase

The Flash memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Flash Page Select register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95h to the Flash Control Register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. The eZ8 CPU resumes operation after the Page Erase operation completes. If the Page Erase operation is performed using the On-Chip Debugger, poll the Flash Status Register to determine when the Page Erase operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state.

Mass Erase

The Flash memory can also be Mass Erased using the Flash Controller, but only by using the On-Chip Debugger. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked and the Mass Erase successfully enabled, writing the value 63H to the Flash Control Register initiates the Mass Erase operation. While the Flash Controller executes the Mass Erase operation, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Using the On-Chip Debugger, poll the Flash Status Register to determine when the Mass Erase operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state.

Flash Controller Bypass

The Flash Controller can be bypassed and the control signals for the Flash memory brought out to the GPIO pins. Bypassing the Flash Controller allows faster Row Programming algorithms by controlling the Flash programming signals directly.

Row programming is recommended for gang programming applications and large volume customers who do not require in-circuit initial programming of the Flash memory. Page Erase operations are also supported when the Flash Controller is bypassed.

For more information about bypassing the Flash Controller, refer to the Zilog application note titled, [Third-Party Flash Programming Support for Z8 Encore! MCUs \(AN0117\)](#), available for download at www.zilog.com.

Flash Controller Behavior in DEBUG Mode

The following changes in behavior of the Flash Controller occur when the Flash Controller is accessed using the On-Chip Debugger:

- The Flash Write Protect option bit is ignored
- The Flash Sector Protect register is ignored for programming and erase operations

- Programming operations are not limited to the page selected in the Page Select register
- Bits in the Flash Sector Protect register can be written to one or zero
- The second write of the Page Select register to unlock the Flash Controller is not necessary
- The Page Select register can be written when the Flash Controller is unlocked
- The Mass Erase command is enabled through the Flash Control Register



Caution: For security reasons, the Flash Controller allows only a single page to be opened for write/erase. When writing multiple Flash pages, the Flash controller must repeat the unlock sequence to select another page.

Flash Control Register Definitions

This section defines the features of the following Flash Control registers.

[Flash Control Register](#): see page 141

[Flash Status Register](#): see page 143

[Flash Page Select Register](#): see page 143

[Flash Sector Protect Register](#): see page 145

[Flash Frequency High and Low Byte Registers](#): see page 145

Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FTCTL) Register before programming or erasing the Flash memory. Writing the sequence 73H 8CH, sequentially, to the Flash Control Register unlocks the Flash Controller. When the Flash Controller is unlocked, the Flash memory can be enabled for Mass Erase or Page Erase by writing the appropriate enable command to the FCTL. Page Erase applies only to the active page selected in Flash Page Select register. Mass Erase is enabled only through the On-Chip Debugger. Writing an invalid value or an invalid sequence returns the Flash Controller to its locked state. The Write-only Flash Control Register shares its Register File address with the read-only Flash Status Register.

Table 81. Flash Control Register (FCTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|------|---|---|---|---|---|---|---|
| Field | FCMD | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W |
| Address | FF8H | | | | | | | |

| Bit | Description |
|-------|---|
| [7:0] | Flash Command |
| FCMD | 73H = First unlock command. 8CH = Second unlock command. 95H = Page Erase command (must be third command in sequence to initiate Page Erase). 63H = Mass Erase command (must be third command in sequence to initiate Mass Erase). 5EH = Enable Flash Sector Protect Register Access. |

Flash Status Register

The Flash Status Register indicates the current state of the Flash Controller. This register can be read at any time. The read-only Flash Status Register shares its Register File address with the write-only Flash Control Register.

Table 82. Flash Status Register (FSTAT)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------|---|-------|---|---|---|---|---|
| Field | Reserved | | FSTAT | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |
| Address | FF8H | | | | | | | |

| Bit | Description |
|----------------|--|
| [7:6] | Reserved These bits are reserved and must be programmed to 0 when read. |
| [5:0] FSTAT | Flash Controller Status 000000 = Flash Controller locked. 000001 = First unlock command received (73H written). 000010 = Second unlock command received (8CH written). 000011 = Flash Controller unlocked. 000100 = Sector protect register selected. 001xxx = Program operation in progress. 010xxx = Page erase operation in progress. 100xxx = Mass erase operation in progress. |

Flash Page Select Register

The Flash Page Select (FPS) register shares address space with the Flash Sector Protect Register. Unless the Flash controller is unlocked and written with 5EH, writes to this address target the Flash Page Select Register.

The register is used to select one of the eight available Flash memory pages to be programmed or erased. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory having addresses with the most significant 7-bits given by FPS[6:0] are chosen for program/erase operation.

Table 83. Flash Page Select Register (FPS)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------|------|-----|-----|-----|-----|-----|-----|
| Field | INFO_EN | PAGE | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FF9H | | | | | | | |

| Bit | Description |
|----------------|--|
| [7] INFO_EN | <p>Information Area Enable</p> <p>0 = Information Area us not selected. 1 = Information Area is selected. The Information Area is mapped into the Program Memory address space at addresses FE00H through FFFFH.</p> |
| [6:0] PAGE | <p>Page Select</p> <p>This 7-bit field identifies the Flash memory page for Page Erase and page unlocking.</p> <ul style="list-style-type: none"> • Program Memory Address[15:9] = PAGE[6:0]. • For Z8F04x3 devices, the upper 4 bits must always be 0. • For Z8F02x3 devices, the upper 5 bits must always be 0. • For Z8F01x3 devices, the upper 6 bits must always be 0. |

Flash Sector Protect Register

The Flash Sector Protect (FPROT) Register is shared with the Flash Page Select Register. When the Flash Control Register is written with 5EH, the next write to this address targets the Flash Sector Protect Register. In all other cases, it targets the Flash Page Select Register.

This register selects one of the 8 available Flash memory sectors to be protected. The reset state of each Sector Protect bit is an unprotected state. After a sector is protected by setting its corresponding register bit, it cannot be unprotected (the register bit cannot be cleared) without powering down the device.

Table 84. Flash Sector Protect Register (FPROT)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| Field | SPROT7 | SPROT6 | SPROT5 | SPROT4 | SPROT3 | SPROT2 | SPROT1 | SPROT0 |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FF9H | | | | | | | |

| Bit | Description |
|-----------|---|
| [7] | Sector Protection |
| SPROT n | Each bit corresponds to a 1024-byte Flash sector on devices in the 8K range, while the remaining devices correspond to a 512-byte Flash sector. To determine the appropriate Flash memory sector address range and sector number for your Z8F0823 Series product, please refer to Table 79 on page 134 and to Figure 20, which follows the table. <ul style="list-style-type: none"> For Z8F08x3 and Z8F04x3 devices, all bits are used. For Z8F02x3 devices, the upper 4 bits are unused. For Z8F01x3 devices, the upper 6 bits are unused. |

Note: n indicates the specific Flash sector (7–0).

Flash Frequency High and Low Byte Registers

The Flash Frequency High (FFREQH) and Low Byte (FFREQL) registers combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain the system clock frequency (in kHz) and is calculated using the following equation:

$$FFREQ[15:0] = \{FFREQH[7:0], FFREQL[7:0]\} = \frac{\text{System Clock Frequency}}{1000}$$



Caution: The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device. Also, Flash programming and erasure is not supported for system clock frequencies below 20kHz or above 20MHz.

Table 85. Flash Frequency High Byte Register (FFREQH)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------|-----|-----|-----|-----|-----|-----|-----|
| Field | FFREQH | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FFAH | | | | | | | |

| Bit | Description |
|-----|-------------|
|-----|-------------|

| | |
|-------|---|
| [7:0] | Flash Frequency High Byte FFREQH High byte of the 16-bit Flash Frequency value. |
|-------|---|

Table 86. Flash Frequency Low Byte Register (FFREQL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|--------|---|---|---|---|---|---|---|
| Field | FFREQL | | | | | | | |
| RESET | 0 | | | | | | | |
| R/W | R/W | | | | | | | |
| Address | FFBH | | | | | | | |

| Bit | Description |
|-----|-------------|
|-----|-------------|

| | |
|-------|---|
| [7:0] | Flash Frequency Low Byte FFREQL Low byte of the 16-bit Flash Frequency value. |
|-------|---|

Option Bit Types

This section describes the five types of Flash option bits offered in the F083A Series.

User Option Bits

The user option bits are contained in the first two bytes of program memory. Access to these bits has been provided because these locations contain application-specific device configurations. The information contained here is lost when page 0 in program memory is erased.

Trim Option Bits

The trim option bits are contained in a Flash memory information page. These bits are factory programmed values required to optimize the operation of onboard analog circuitry and cannot be permanently altered. Program memory may be erased without endangering these values. It is possible to alter working values of these bits by accessing the Trim Bit Address and Data Registers, but these working values are lost after a power loss or any other reset event.

There are 32 bytes of trim data. To modify one of these values the user code must first write a value between 00H and 1FH into the Trim Bit Address Register. The next write to the Trim Bit Data Register changes the working value of the target trim data byte.

Reading the trim data requires the user code to write a value between 00H and 1FH into the Trim Bit Address Register. The next read from the Trim Bit Data Register returns the working value of the target trim data byte.

► **Note:** The trim address range is from information address 20–3F only. The remainder of the information page is not accessible through the trim bit address and data registers.

Calibration Option Bits

The calibration option bits are also contained in the information page. These bits are factory programmed values intended for use in software correcting the device's analog performance. To read these values, the user code must employ the LDC instruction to access the information area of the address space as defined in the [Flash Information Area](#) section on page 15.

Serialization Bits

As an optional feature, Zilog is able to provide factory-programmed serialization. For serialized products, the individual devices are programmed with unique serial numbers. These serial numbers are binary values, four bytes in length. The numbers increase in size with each device, but gaps in the serial sequence may exist.

These serial numbers are stored in the Flash information page (for more details, see the [Reading the Flash Information Page](#) section on page 148 and the [Serialization Data](#) section on page 154) and are unaffected by mass erasure of the device’s Flash memory.

Randomized Lot Identification Bits

As an optional feature, Zilog is able to provide a factory-programmed random lot identifier. With this feature, all devices in a given production lot are programmed with the same random number. This random number is uniquely regenerated for each successive production lot and is not likely to be repeated.

The randomized lot identifier is a 32-byte binary value, stored in the flash information page (for more details, see the [Reading the Flash Information Page](#) section on page 148 and the [Randomized Lot Identifier](#) section on page 154) and is unaffected by mass erasure of the device’s flash memory.

Reading the Flash Information Page

The following code example shows how to read data from the Flash Information Area.

```

; get value at info address 60 (FE60h)
ldx FPS, #80 ; enable access to flash info page
ld R0, #FE
ld R1, #60
ldc R2, @RR0 ; R2 now contains the calibration value
    
```

Flash Option Bit Control Register Definitions

This section briefly describes the features of the Trim Bit Address and Data registers.

Trim Bit Address Register

The Trim Bit Address (TRMADR) Register contains the target address for an access to the trim option bits.

Table 87. Trim Bit Address Register (TRMADR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---------------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | TRMADR: Trim Bit Address (00H to 1FH) | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FF6H | | | | | | | |

Trim Bit Data Register

The Trim Bid Data (TRMDR) register contains the read or write data for access to the trim option bits.

Table 88. Trim Bit Data Register (TRMDR)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | TRMDR: Trim Bit Data | | | | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | FF7H | | | | | | | |

Flash Option Bit Address Space

The first two bytes of Flash program memory at addresses 0000H and 0001H are reserved for the user-programmable Flash option bits.

Table 89. Flash Option Bits at Program Memory Address 0000H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------------------|--------|----------|-----|--------|-----|----------|-----|
| Field | WDT_RES | WDT_AO | Reserved | | VBO_AO | FRP | Reserved | FWP |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Program Memory 0000H | | | | | | | |

Note: U = Unchanged by Reset. R/W = Read/Write.

| Bit | Description |
|----------------|---|
| [7] WDT_RES | Watchdog Timer Reset 0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request. 1 = Watchdog Timer time-out causes a system reset. This setting is the default for unprogrammed (erased) Flash. |
| [6] WDT_AO | Watchdog Timer Always ON 0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled. 1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash. |
| [5:4] | Reserved These bits are reserved and must be programmed to 11 during writes, and to 11 when read. |

| Bit | Description (Continued) |
|---------------|--|
| [3] VBO_AO | <p>Voltage Brown-Out Protection Always ON</p> <p>0 = Voltage Brown-Out Protection can be disabled in STOP Mode to reduce total power consumption. For the block to be disabled, the power control register bit must also be written (see the Power Control Register 0 section on page 31).</p> <p>1 = Voltage Brown-Out Protection is always enabled including during STOP Mode. This setting is the default for unprogrammed (erased) Flash.</p> |
| [2] FRP | <p>Flash Read Protect</p> <p>0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger.</p> <p>1 = User program code is accessible. All On-Chip Debugger commands are enabled. This setting is the default for unprogrammed (erased) Flash.</p> |
| [1] | <p>Reserved</p> <p>This bit is reserved and must be programmed to 1.</p> |
| [0] FWP | <p>Flash Write Protect</p> <p>This Option Bit provides Flash Program Memory protection:</p> <p>0 = Programming and erasure disabled for all of Flash Program Memory. Programming, Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available using the On-Chip Debugger.</p> <p>1 = Programming, Page Erase, and Mass Erase are enabled for all of Flash program memory.</p> |

Table 90. Flash Options Bits at Program Memory Address 0001H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|----------------------|-----|-----|--------|----------|-----|-----|-----|
| Field | Reserved | | | XTLDIS | Reserved | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Program Memory 0001H | | | | | | | |

Note: U = Unchanged by Reset. R/W = Read/Write.

| Bit | Description |
|---------------|---|
| [7:5] | Reserved These bits are reserved and must be programmed to 111 during writes and to 111 when read. |
| [4] XTLDIS | State of Crystal Oscillator at Reset This bit only enables the crystal oscillator. Its selection as a system clock must be performed manually. 0 = The crystal oscillator is enabled during reset, resulting in longer reset timing. 1 = The crystal oscillator is disabled during reset, resulting in shorter reset timing. Caution: Programming the XTLDIS bit to zero on 8-pin versions of F0823 Series devices prevents any further communication via the debug pin due to the X _{IN} and DBG functions being shared on pin 2 of the 8-pin package. Do not program this bit to zero on 8-pin devices unless no further debugging or Flash programming is required. |
| [3:0] | Reserved These bits are reserved and must be programmed to 1111 during writes and to 1111 when read. |

Trim Bit Address Space

All available trim bit addresses and their functions are listed in Tables 91 through 93.

Table 91. Trim Options Bits at Address 0000H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|-------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | Reserved | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Information Page Memory 0020H | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|-------|--|
| [7:0] | Reserved These bits are reserved. Altering this register may result in incorrect device operation. |

Table 92. Trim Option Bits at 0001H

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|-------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | Reserved | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Information Page Memory 0021H | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|-------|--|
| [7:0] | Reserved These bits are reserved. Altering this register may result in incorrect device operation. |

Table 93. Trim Option Bits at 0002H (TIPO)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|-------------------------------|---|---|---|---|---|---|---|
| Field | IPO_TRIM | | | | | | | |
| RESET | U | | | | | | | |
| R/W | R/W | | | | | | | |
| Address | Information Page Memory 0022H | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|----------|---|
| [7:0] | Internal Precision Oscillator Trim Byte |
| IPO_TRIM | Contains trimming bits for the Internal Precision Oscillator. |

Zilog Calibration Data

This section briefly describes the features of the following Flash Option Bit calibration registers.

[ADC Calibration Data](#): see page 153

[Serialization Data](#): see page 154

[Randomized Lot Identifier](#): see page 154

ADC Calibration Data

Table 94. ADC Calibration Bits

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | ADC_CAL | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Information Page Memory 0060H–007DH | | | | | | | |

Note: U = Unchanged by Reset. R/W = Read/Write.

| Bit | Description |
|------------------|---|
| [7:0] ADC_CAL | Analog-to-Digital Converter Calibration Values Contains factory-calibrated values for ADC gain and offset compensation. Each of the ten supported modes has one byte of offset calibration and two bytes of gain calibration. These values are read by the software to compensate ADC measurements as detailed in the Software Compensation Procedure section on page 126. The location of each calibration byte is provided in Table 95. |

Table 95. ADC Calibration Data Location

| Info Page Address | Memory Address | Compensation Usage | ADC Mode | Reference Type |
|-------------------|----------------|--------------------|-------------------------|----------------|
| 60 | FE60 | Offset | Single-Ended Unbuffered | Internal 2.0V |
| 08 | FE08 | Gain High Byte | Single-Ended Unbuffered | Internal 2.0V |
| 09 | FE09 | Gain Low Byte | Single-Ended Unbuffered | Internal 2.0V |
| 63 | FE63 | Offset | Single-Ended Unbuffered | Internal 1.0V |
| 0A | FE0A | Gain High Byte | Single-Ended Unbuffered | Internal 1.0V |
| 0B | FE0B | Gain Low Byte | Single-Ended Unbuffered | Internal 1.0V |
| 66 | FE66 | Offset | Single-Ended Unbuffered | External 2.0V |
| 0C | FE0C | Gain High Byte | Single-Ended Unbuffered | External 2.0V |
| 0D | FE0D | Gain Low Byte | Single-Ended Unbuffered | External 2.0V |

Serialization Data

Table 96. Serial Number at 001C–001F (S_NUM)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|-----------------------------------|-----|-----|-----|-----|-----|-----|-----|
| Field | S_NUM | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Information Page Memory 001C–001F | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|----------------|--|
| [7:0] S_NUM | Serial Number Byte The serial number is a unique four-byte binary value; see Table 97. |

Table 97. Serialization Data Locations

| Info Page Address | Memory Address | Usage |
|-------------------|----------------|---|
| 1C | FE1C | Serial Number Byte 3 (most significant). |
| 1D | FE1D | Serial Number Byte 2. |
| 1E | FE1E | Serial Number Byte 1. |
| 1F | FE1F | Serial Number Byte 0 (least significant). |

Randomized Lot Identifier

Table 98. Lot Identification Number (RAND_LOT)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|-----|-----|-----|-----|-----|-----|-----|
| Field | RAND_LOT | | | | | | | |
| RESET | U | U | U | U | U | U | U | U |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | Interspersed throughout Information Page Memory | | | | | | | |
| Note: U = Unchanged by Reset. R/W = Read/Write. | | | | | | | | |

| Bit | Description |
|-----------------|---|
| [7] RAND_LOT | Randomized Lot ID The randomized lot ID is a 32-byte binary value that changes for each production lot; see Table 99. |

Table 99. Randomized Lot ID Locations

| Info Page Address | Memory Address | Usage |
|--------------------------|-----------------------|--|
| 3C | FE3C | Randomized Lot ID Byte 31 (most significant) |
| 3D | FE3D | Randomized Lot ID Byte 30 |
| 3E | FE3E | Randomized Lot ID Byte 29 |
| 3F | FE3F | Randomized Lot ID Byte 28 |
| 58 | FE58 | Randomized Lot ID Byte 27 |
| 59 | FE59 | Randomized Lot ID Byte 26 |
| 5A | FE5A | Randomized Lot ID Byte 25 |
| 5B | FE5B | Randomized Lot ID Byte 24 |
| 5C | FE5C | Randomized Lot ID Byte 23 |
| 5D | FE5D | Randomized Lot ID Byte 22 |
| 5E | FE5E | Randomized Lot ID Byte 21 |
| 5F | FE5F | Randomized Lot ID Byte 20 |
| 61 | FE61 | Randomized Lot ID Byte 19 |
| 62 | FE62 | Randomized Lot ID Byte 18 |
| 64 | FE64 | Randomized Lot ID Byte 17 |
| 65 | FE65 | Randomized Lot ID Byte 16 |
| 67 | FE67 | Randomized Lot ID Byte 15 |
| 68 | FE68 | Randomized Lot ID Byte 14 |
| 6A | FE6A | Randomized Lot ID Byte 13 |
| 6B | FE6B | Randomized Lot ID Byte 12 |
| 6D | FE6D | Randomized Lot ID Byte 11 |
| 6E | FE6E | Randomized Lot ID Byte 10 |
| 70 | FE70 | Randomized Lot ID Byte 9 |
| 71 | FE71 | Randomized Lot ID Byte 8 |
| 73 | FE73 | Randomized Lot ID Byte 7 |
| 74 | FE74 | Randomized Lot ID Byte 6 |
| 76 | FE76 | Randomized Lot ID Byte 5 |
| 77 | FE77 | Randomized Lot ID Byte 4 |
| 79 | FE79 | Randomized Lot ID Byte 3 |
| 7A | FE7A | Randomized Lot ID Byte 2 |
| 7C | FE7C | Randomized Lot ID Byte 1 |
| 7D | FE7D | Randomized Lot ID Byte 0 (least significant) |

On-Chip Debugger

Z8 Encore! XP F0823 Series devices contain an integrated On-Chip Debugger (OCD) which provides advanced debugging features that include:

- Single pin interface
- Reading and writing of the register file
- Reading and writing of program and data memory
- Setting of breakpoints and watchpoints
- Executing eZ8 CPU instructions
- Debug pin sharing with general-purpose input-output function to maximize the pins available

Architecture

The on-chip debugger consists of four primary functional blocks: transmitter, receiver, auto-baud detector/generator, and debug controller. Figure 22 displays the architecture of the OCD.



Figure 22. On-Chip Debugger Block Diagram

Operation

The following section describes the operation of the OCD.

OCD Interface

The OCD uses the DBG pin for communication with an external host. This one-pin interface is a bidirectional open-drain interface that transmits and receives data. Data transmission is half-duplex, in that transmit and receive cannot occur simultaneously. The serial data on the DBG pin is sent using the standard asynchronous data format defined in RS-232. This pin creates an interface from the F0823 Series products to the serial port of a host PC using minimal external hardware. Two different methods for connecting the DBG pin to an RS-232 interface are displayed in Figure 23 and Figure 24. The recommended method is the buffered implementation depicted in Figure 24. The DBG pin has an internal pull-up resistor which is sufficient for some applications (for more details about the pull-up current, see the [Electrical Characteristics](#) chapter on page 196). For OCD operation at higher data rates or in noisy systems, Zilog recommends an external pull-up resistor.



Caution: For operation of the OCD, all power pins (V_{DD} and AV_{DD}) must be supplied with power, and all ground pins (V_{SS} and AV_{SS}) must be properly grounded. The DBG pin is open-drain and may require an external pull-up resistor to ensure proper operation.



Figure 23. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 1 of 2



Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface, # 2 of 2

DEBUG Mode

The operating characteristics of the devices in DEBUG Mode are:

- The eZ8 CPU fetch unit stops, idling the eZ8 CPU, unless directed by the OCD to execute specific instructions
- The system clock operates unless in STOP Mode
- All enabled on-chip peripherals operate unless in STOP Mode
- Automatically exits HALT Mode
- Constantly refreshes the Watchdog Timer, if enabled.

Entering DEBUG Mode

The device enters DEBUG Mode following the operations below:

- The device enters DEBUG Mode after the eZ8 CPU executes a BRK (breakpoint) instruction
- If the DBG pin is held Low during the most recent clock cycle of System Reset, the part enters DEBUG Mode upon exiting System Reset

► **Note:** Holding the DBG pin Low for an additional 5000 (minimum) clock cycles after reset (making sure to account for any specified frequency error if using an internal oscillator) prevents a false interpretation of an autobaud sequence (see [the OCD Autobaud Detector/Generator section on page 159](#)).

- If the PA2/ $\overline{\text{RESET}}$ pin is held Low while a 32-bit key sequence is issued to the PA0/DBG pin, the DBG feature is unlocked. After releasing PA2/ $\overline{\text{RESET}}$, it is pulled high. At this point, the PA0/DBG pin can be used to autobaud and cause the device to enter DEBUG Mode. For more details, see the [OCD Unlock Sequence \(8-Pin Devices Only\)](#) section on page 161.

Exiting DEBUG Mode

The device exits DEBUG Mode following any of these operations:

- Clearing the DBGMODE bit in the OCD Control Register to 0
- Power-On Reset
- Voltage Brown-Out reset
- Watchdog Timer reset
- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a Reset
- Driving the DBG pin Low while the device is in STOP Mode initiates a system reset

OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 Start bit, 8 data bits (least-significant bit first), and 1 Stop bit as displayed in Figure 25.



Figure 25. OCD Data Format

-
- **Note:** When responding to a request for data, the OCD may commence transmitting immediately after receiving the stop bit of an incoming frame. Therefore, when sending the stop bit, the host must not actively drive the DBG pin High for more than 0.5 bit times. Zilog recommends that, if possible, the host drives the DBG pin using an open-drain output.
-

OCD Autobaud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the OCD contains an auto-baud detector/generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the host is the character 80H. The character 80H has eight continuous bits Low (one Start bit plus 7 data

bits), framed between High bits. The auto-baud detector measures this period and sets the OCD baud rate generator accordingly.

The auto-baud detector/generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. For optimal operation with asynchronous datastreams, the maximum recommended baud rate is the system clock frequency divided by eight. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by four, but this theoretical maximum is possible only for low noise designs with clean signals. Table 100 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 100. OCD Baud-Rate Limits

| System Clock Frequency (MHz) | Recommended Maximum Baud Rate (kbps) | Recommended Standard PC Baud Rate (bps) | Minimum Baud Rate (kbps) |
|------------------------------|--------------------------------------|---|--------------------------|
| 5.5296 | 1382.4 | 691,200 | 1.08 |
| 0.032768 (32kHz) | 4.096 | 2400 | 0.064 |

If the OCD receives a Serial Break (nine or more continuous bits Low) the auto-baud detector/generator resets. Reconfigure the auto-baud detector/generator by sending 80H.

OCD Serial Errors

The OCD detects any of the following error conditions on the DBG pin:

- Serial Break (a minimum of nine continuous bits Low)
- Framing Error (received Stop bit is Low)
- Transmit Collision (OCD and host simultaneous transmission detected by the OCD)

When the OCD detects one of these errors, it aborts any command currently in progress, transmits a four character long Serial Break back to the host, and resets the auto-baud detector/generator. A Framing Error or Transmit Collision may be caused by the host sending a Serial Break to the OCD. Because of the open-drain nature of the interface, returning a Serial Break back to the host only extends the length of the Serial Break if the host releases the Serial Break early.

The host transmits a Serial Break on the DBG pin when first connecting to the F0823 Series devices or when recovering from an error. A Serial Break from the host resets the auto-baud generator/detector but does not reset the OCD Control Register. A Serial Break leaves the device in DEBUG Mode if that is the current mode. The OCD is held in Reset until the end of the Serial Break when the DBG pin returns High. Because of the open-drain nature of the DBG pin, the host sends a Serial Break to the OCD even if the OCD is transmitting a character.

OCD Unlock Sequence (8-Pin Devices Only)

Because of pin-sharing on the 8-pin device, an unlock sequence must be performed to access the DBG pin. If this sequence is not completed during a system reset, then the PA0/DBG pin functions only as a GPIO pin.

The following sequence unlocks the DBG pin:

1. Hold PA2/ $\overline{\text{RESET}}$ Low.
2. Wait 5 ms for the internal reset sequence to complete.
3. Send the following bytes serially to the debug pin:

```
DBG ← 80H (autobaud)
DBG ← EBH
DBG ← 5AH
DBG ← 70H
DBG ← CDH (32-bit unlock key)
```

4. Release PA2/ $\overline{\text{RESET}}$. The PA0/DBG pin is now identical in function to that of the DBG pin on the 20- or 28-pin device. To enter DEBUG Mode, reautobaud and write 80H to the OCD Control Register (see the [On-Chip Debugger Commands](#) section on page 162).

Breakpoints

Execution breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the OCD. If breakpoints are enabled, the OCD enters DEBUG Mode and idles the eZ8 CPU. If breakpoints are not enabled, the OCD ignores the BRK signal and the BRK instruction operates as an NOP instruction.

Breakpoints in Flash Memory

The BRK instruction is opcode 00H, which corresponds to the fully programmed state of a byte in Flash memory. To implement a breakpoint, write 00H to the required break address, overwriting the current instruction. To remove a breakpoint, the corresponding page of Flash memory must be erased and reprogrammed with the original data.

Runtime Counter

The OCD contains a 16-bit Runtime Counter. It counts system clock cycles between breakpoints. The counter starts counting when the OCD leaves DEBUG Mode and stops counting when it enters DEBUG Mode again or when it reaches the maximum count of FFFFH.

On-Chip Debugger Commands

The host communicates to the OCD by sending OCD commands using the DBG interface. During normal operation, only a subset of the OCD commands are available. In DEBUG Mode, all OCD commands become available unless the user code and control registers are protected by programming the Flash Read Protect Option bit (FRP). The Flash Read Protect Option bit prevents the code in memory from being read out of Z8 Encore! XP F0823 Series products. When this option is enabled, several of the OCD commands are disabled.

Table 101 is a summary of the OCD commands. Each OCD command is described in further detail in the pages that follow this table. [Table 102](#) on page 167 also indicates those commands that operate when the device is not in DEBUG Mode (normal operation) and those commands that are disabled by programming the Flash Read Protect Option bit.

Table 101. OCD Commands

| Debug Command | Command Byte | Enabled when not in DEBUG Mode? | Disabled by Flash Read Protect Option Bit |
|----------------------------|--------------|---------------------------------|--|
| Read OCD Revision | 00H | Yes | – |
| Reserved | 01H | – | – |
| Read OCD Status Register | 02H | Yes | – |
| Read Runtime Counter | 03H | – | – |
| Write OCD Control Register | 04H | Yes | Cannot clear DBGMODE bit. |
| Read OCD Control Register | 05H | Yes | – |
| Write Program Counter | 06H | – | Disabled. |
| Read Program Counter | 07H | – | Disabled. |
| Write Register | 08H | – | Only writes of the Flash Memory Control registers are allowed. Additionally, only the Mass Erase command is allowed to be written to the Flash Control Register. |
| Read Register | 09H | – | Disabled. |
| Write Program Memory | 0AH | – | Disabled. |
| Read Program Memory | 0BH | – | Disabled. |
| Write Data Memory | 0CH | – | Yes. |
| Read Data Memory | 0DH | – | – |
| Read Program Memory CRC | 0EH | – | – |
| Reserved | 0FH | – | – |
| Step Instruction | 10H | – | Disabled. |
| Stuff Instruction | 11H | – | Disabled. |
| Execute Instruction | 12H | – | Disabled. |
| Reserved | 13H–FFH | – | – |

In the following list of OCD Commands, data and commands sent from the host to the OCD are identified by 'DBG ← Command/Data'. Data sent from the OCD back to the host is identified by 'DBG → Data'.

Read OCD Revision (00H). The Read OCD Revision command determines the version of the OCD. If OCD commands are added, removed, or changed, this revision number changes.

```
DBG ← 00H
DBG → OCDRev[15:8] (Major revision number)
DBG → OCDRev[7:0] (Minor revision number)
```

Read OCD Status Register (02H). The Read OCD Status Register command reads the OCDSTAT Register.

```
DBG ← 02H
DBG → OCDSTAT[7:0]
```

Read Runtime Counter (03H). The Runtime Counter counts system clock cycles in between breakpoints. The 16-bit Runtime Counter counts up from 0000H and stops at the maximum count of FFFFH. The Runtime Counter is overwritten during the Write Memory, Read Memory, Write Register, Read Register, Read Memory CRC, Step Instruction, Stuff Instruction, and Execute Instruction commands.

```
DBG ← 03H
DBG → RuntimeCounter[15:8]
DBG → RuntimeCounter[7:0]
```

Write OCD Control Register (04H). The Write OCD Control Register command writes the data that follows to the OCDCTL register. When the Flash Read Protect Option Bit is enabled, the DBGMODE bit (OCDCTL[7]) can only be set to 1, it cannot be cleared to 0 and the only method of returning the device to normal operating mode is to reset the device.

```
DBG ← 04H
DBG ← OCDCTL[7:0]
```

Read OCD Control Register (05H). The Read OCD Control Register command reads the value of the OCDCTL register.

```
DBG ← 05H
DBG → OCDCTL[7:0]
```

Write Program Counter (06H). The Write Program Counter command writes the data that follows to the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG Mode or if the Flash Read Protect Option bit is enabled, the Program Counter (PC) values are discarded.

```
DBG ← 06H
DBG ← ProgramCounter[15:8]
DBG ← ProgramCounter[7:0]
```

Read Program Counter (07H). The Read Program Counter command reads the value in the eZ8 CPU's Program Counter (PC). If the device is not in DEBUG Mode or if the Flash Read Protect Option bit is enabled, this command returns FFFFH.

```
DBG ← 07H
DBG → ProgramCounter[15:8]
DBG → ProgramCounter[7:0]
```

Write Register (08H). The Write Register command writes data to the Register File. Data can be written 1–256 bytes at a time (256 bytes can be written by setting size to 0). If the device is not in DEBUG Mode, the address and data values are discarded. If the Flash Read Protect Option bit is enabled, only writes to the Flash Control Registers are allowed and all other register write data values are discarded.

```
DBG ← 08H
DBG ← {4'h0, Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG ← 1-256 data bytes
```

Read Register (09H). The Read Register command reads data from the Register File. Data can be read 1–256 bytes at a time (256 bytes can be read by setting size to 0). If the device is not in DEBUG Mode or if the Flash Read Protect Option bit is enabled, this command returns FFH for all the data values.

```
DBG ← 09H
DBG ← {4'h0, Register Address[11:8]}
DBG ← Register Address[7:0]
DBG ← Size[7:0]
DBG → 1-256 data bytes
```

Write Program Memory (0AH). The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCl instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the device is not in DEBUG Mode or if the Flash Read Protect Option bit is enabled, the data is discarded.

```
DBG ← 0AH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Program Memory (0BH). The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCl instructions. Data can be read 1–65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG Mode or if the Flash Read Protect Option Bit is enabled, this command returns FFH for the data.

```
DBG ← 0BH
DBG ← Program Memory Address[15:8]
DBG ← Program Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

Write Data Memory (0CH). The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1–65536 bytes at a time (65536 bytes can be written by setting size to 0). If the device is not in DEBUG Mode or if the Flash Read Protect Option Bit is enabled, the data is discarded.

```
DBG ← 0CH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG ← 1-65536 data bytes
```

Read Data Memory (0DH). The Read Data Memory command reads from Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be read 1 to 65536 bytes at a time (65536 bytes can be read by setting size to 0). If the device is not in DEBUG Mode, this command returns FFH for the data.

```
DBG ← 0DH
DBG ← Data Memory Address[15:8]
DBG ← Data Memory Address[7:0]
DBG ← Size[15:8]
DBG ← Size[7:0]
DBG → 1-65536 data bytes
```

Read Program Memory CRC (0EH). The Read Program Memory Cyclic Redundancy Check (CRC) command computes and returns the CRC of Program Memory using the 16-bit CRC-CCITT polynomial. If the device is not in DEBUG Mode, this command returns FFFFH for the CRC value. Unlike most other OCD Read commands, there is a delay from issuing of the command until the OCD returns the data. The OCD reads the Program Memory, calculates the CRC value, and returns the result. The delay is a function of the Program Memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Program Memory.

```
DBG ← 0EH
DBG → CRC[15:8]
DBG → CRC[7:0]
```

Step Instruction (10H). The Step Instruction steps one assembly instruction at the current Program Counter (PC) location. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 10H
```


Stuff Instruction (11H). The Stuff command steps one assembly instruction and allows specification of the first byte of the instruction. The remaining 0–4 bytes of the instruction are read from Program Memory. This command is useful for stepping over instructions where the first byte of the instruction has been overwritten by a Breakpoint. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, the OCD ignores this command.

```
DBG ← 11H  
DBG ← opcode[7:0]
```

Execute Instruction (12H). The Execute command allows sending an entire instruction to be executed to the eZ8 CPU. This command can also step over breakpoints. The number of bytes to send for the instruction depends on the opcode. If the device is not in DEBUG Mode or the Flash Read Protect Option bit is enabled, this command reads and discards one byte.

```
DBG ← 12H  
DBG ← 1-5 byte opcode
```

On-Chip Debugger Control Register Definitions

This section describes the features of the On-Chip Debugger Control and Status registers.

OCD Control Register

The OCD Control Register controls the state of the OCD. This register is used to enter or exit DEBUG Mode and to enable the BRK instruction. It also resets Z8 Encore! XP F0823 Series device.

A reset and stop function can be achieved by writing 81H to this register. A reset and go function can be achieved by writing 41H to this register. If the device is in DEBUG Mode, a run function can be implemented by writing 40H to this register.

Table 102. OCD Control Register (OCDCTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|---------|-------|--------|----------|---|---|---|-----|
| Field | DBGMODE | BRKEN | DBGACK | Reserved | | | | RST |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R/W |

| Bit | Description |
|----------------|---|
| [7] DBGMODE | <p>DEBUG Mode</p> <p>The device enters DEBUG Mode when this bit is 1. When in DEBUG Mode, the eZ8 CPU stops fetching new instructions. Clearing this bit causes the eZ8 CPU to restart. This bit is automatically set when a BRK instruction is decoded and breakpoints are enabled. If the Flash Read Protect Option Bit is enabled, this bit can only be cleared by resetting the device. It cannot be written to 0.</p> <p>0 = F0823 Series device is operating in NORMAL Mode. 1 = F0823 Series device is in DEBUG Mode.</p> |
| [6] BRKEN | <p>Breakpoint Enable</p> <p>This bit controls the behavior of the BRK instruction (opcode 00H). By default, breakpoints are disabled and the BRK instruction behaves similar to an NOP instruction. If this bit is 1, when a BRK instruction is decoded, the DBGMODE bit of the OCDCTL register is automatically set to 1.</p> <p>0 = Breakpoints are disabled. 1 = Breakpoints are enabled.</p> |
| [5] DBGACK | <p>Debug Acknowledge</p> <p>This bit enables the debug acknowledge feature. If this bit is set to 1, the OCD sends a Debug Acknowledge character (FFH) to the host when a Breakpoint occurs.</p> <p>0 = Debug Acknowledge is disabled. 1 = Debug Acknowledge is enabled.</p> |
| [4:1] | <p>Reserved</p> <p>These bits are reserved and must be 00000 when read.</p> |
| [0] RST | <p>Reset</p> <p>Setting this bit to 1 resets the Z8F04xA family device. The device goes through a normal Power-On Reset sequence with the exception that the OCD is not reset. This bit is automatically cleared to 0 at the end of reset.</p> <p>0 = No effect. 1 = Reset the Flash Read Protect Option Bit device.</p> |

OCD Status Register

The OCD Status Register reports status information about the current state of the debugger and the system.

Table 103. OCD Status Register (OCDSTAT)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|------|--------|----------|---|---|---|---|
| Field | DBG | HALT | FRPENB | Reserved | | | | |
| RESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | Description |
|---------------|---|
| [7] DBG | Debug Status 0 = NORMAL Mode. 1 = DEBUG Mode. |
| [6] HALT | HALT Mode 0 = Not in HALT Mode. 1 = In HALT Mode. |
| [5] FRPENB | Flash Read Protect Option Bit Enable 0 = FRP bit enabled to allow disabling of many OCD commands. 1 = FRP bit has no effect. |
| [4:0] | Reserved These bits are reserved and must be 00000 when read. |

Oscillator Control

Z8 Encore! XP F0823 Series devices uses three possible clocking schemes, each user-selectable. These three schemes are:

- On-chip precision trimmed RC oscillator
- External clock drive
- On-chip low power Watchdog Timer oscillator

In addition, F0823 Series devices contain clock failure detection and recovery circuitry, which allow continued operation despite a failure of the primary oscillator.

Operation

This chapter discusses the logic used to select the system clock and handle primary oscillator failures. A description of the specific operation of each oscillator is outlined elsewhere in this document.

System Clock Selection

The oscillator control block selects from the available clocks. Table 104 details each clock source and its usage.

Table 104. Oscillator Configuration and Selection

| Clock Source | Characteristics | Required Setup |
|------------------------------------|--|---|
| Internal Precision RC Oscillator | <ul style="list-style-type: none"> • 32.8kHz or 5.53MHz • $\pm 4\%$ accuracy when trimmed • No external components required | <ul style="list-style-type: none"> • Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 5.53MHz or 32.8kHz |
| External Clock Drive | <ul style="list-style-type: none"> • 0 to 20MHz • Accuracy dependent on external clock source | <ul style="list-style-type: none"> • Write GPIO registers to configure PB3 pin for external clock function • Unlock and write OSCCTL to select external system clock • Apply external clock signal to GPIO |
| Internal Watchdog Timer Oscillator | <ul style="list-style-type: none"> • 10kHz nominal • $\pm 40\%$ accuracy; no external components required • Very Low power consumption | <ul style="list-style-type: none"> • Enable WDT if not enabled and wait until WDT Oscillator is operating. • Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator |



Caution: Unintentional accesses to the Oscillator Control Register can actually stop the chip by switching to a non-functioning oscillator. To prevent this condition, the oscillator control block employs a register unlocking/locking scheme.

OSC Control Register Unlocking/Locking

To write to the Oscillator Control Register, unlock it by making two writes to the OSCCTL Register with the values `E7H` followed by `18H`. A third write to the OSCCTL Register changes the value of the actual register and returns the register to a locked state. Any other sequence of Oscillator Control Register writes has no effect. The values written to unlock the register must be ordered correctly, but are not necessarily consecutive. It is possible to write to or read from other registers within the unlocking/locking operation.

When selecting a new clock source, the primary oscillator failure detection circuitry and the Watchdog Timer oscillator failure circuitry must be disabled. If `POFEN` and `WOFEN` are not disabled prior to a clock switch-over, it is possible to generate an interrupt for a failure of either oscillator. The Failure detection circuitry can be enabled anytime after a successful write of `OSCSEL` in the Oscillator Control Register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it is appropriate to disable the IPO for power savings. Disabling the IPO does not occur automatically.

Clock Failure Detection and Recovery

Should an oscillator or timer fail, there are methods of recovery, as this section describes.

Primary Oscillator Failure

Z8 Encore! XP F0823 Series devices can generate non-maskable interrupt-like events when the primary oscillator fails. To maintain system function in this situation, the clock failure recovery circuitry automatically forces the Watchdog Timer oscillator to drive the system clock. The Watchdog Timer oscillator must be enabled to allow the recovery. Although this oscillator runs at a much slower speed than the original system clock, the CPU continues to operate, allowing execution of a clock failure vector and software routines that either remedy the oscillator failure or issue a failure alert. This automatic switch-over is not available if the Watchdog Timer is the primary oscillator. It is also unavailable if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function outlined in the [Watchdog Timer](#) section on page 91.

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below $1\text{ kHz} \pm 50\%$. If an external signal is selected as the system oscillator, it is possible that a very slow but non-failing clock can generate a failure condition. Under these

conditions, do not enable the clock failure circuitry (POFEN must be deasserted in the OSCCTL Register).

Watchdog Timer Failure

In the event of a Watchdog Timer oscillator failure, a similar non-maskable interrupt-like event is issued. This event does not trigger an attendant clock switch-over, but alerts the CPU of the failure. After a Watchdog Timer failure, it is no longer possible to detect a primary oscillator failure. The failure detection circuitry does not function if the Watchdog Timer is used as the primary oscillator or if the Watchdog Timer oscillator has been disabled. For either of these cases, it is necessary to disable the detection circuitry by deasserting the WDFEN bit of the OSCCTL Register.

The Watchdog Timer oscillator failure detection circuit counts system clocks while searching for a Watchdog Timer clock. The logic counts 8004 system clock cycles before determining that a failure has occurred. The system clock rate determines the speed at which the Watchdog Timer failure can be detected. A very slow system clock results in very slow detection times.



Caution: It is possible to disable the clock failure detection circuitry as well as all functioning clock sources. In this case, the Z8 Encore! XP F0823 Series device ceases functioning and can only be recovered by Power-On Reset.

Oscillator Control Register Definitions

The following section provides the bit definitions for the Oscillator Control Register.

Oscillator Control Register

The Oscillator Control Register (OSCCTL) enables/disables the various oscillator circuits, enables/disables the failure detection/recovery circuitry and selects the primary oscillator, which becomes the system clock.

The Oscillator Control Register must be unlocked before writing. Writing the two step sequence E7H followed by 18H to the Oscillator Control Register unlocks it. The register is locked at successful completion of a register write to the OSCCTL.

Table 105. Oscillator Control Register (OSCCTL)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------|----------|-------|-------|-------|--------|-----|-----|
| Field | INTEN | Reserved | WDTEN | POFEN | WDFEN | SCKSEL | | |
| RESET | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Address | F86H | | | | | | | |

| Bit | Description |
|-----------------|--|
| [7] INTEN | Internal Precision Oscillator Enable 1 = Internal precision oscillator is enabled. 0 = Internal precision oscillator is disabled. |
| [6] | Reserved This bit is reserved and must be programmed to 0 during writes and to 0 when read. |
| [5] WDTEN | Watchdog Timer Oscillator Enable 1 = Watchdog Timer oscillator is enabled. 0 = Watchdog Timer oscillator is disabled. |
| [4] POFEN | Primary Oscillator Failure Detection Enable 1 = Failure detection and recovery of primary oscillator is enabled. 0 = Failure detection and recovery of primary oscillator is disabled. |
| [3] WDFEN | Watchdog Timer Oscillator Failure Detection Enable 1 = Failure detection of Watchdog Timer oscillator is enabled. 0 = Failure detection of Watchdog Timer oscillator is disabled. |
| [2:0] SCKSEL | System Clock Oscillator Select 000 = Internal precision oscillator functions as system clock at 5.53MHz. 001 = Internal precision oscillator functions as system clock at 32kHz. 010 = Reserved. 011 = Watchdog Timer oscillator functions as system clock. 100 = External clock signal on PB3 functions as system clock. 101 = Reserved. 110 = Reserved. 111 = Reserved. |

Internal Precision Oscillator

The internal precision oscillator (IPO) is designed for use without external components. You can either manually trim the oscillator for a non-standard frequency or use the automatic factory-trimmed version to achieve a 5.53MHz frequency. The features of IPO include:

- On-chip RC oscillator that does not require external components
- Output frequency of either 5.53MHz or 32.8kHz (contains both a fast and a slow mode)
- Trimming possible through Flash option bits with user override
- Elimination of crystals or ceramic resonators in applications where high timing accuracy is not required

Operation

An 8-bit trimming register, incorporated into the design, compensates for absolute variation of oscillator frequency. Once trimmed the oscillator frequency is stable and does not require subsequent calibration. Trimming is performed during manufacturing and is not necessary for you to repeat unless a frequency other than 5.53MHz (fast mode) or 32.8kHz (slow mode) is required. This trimming is done at +30°C and a supply voltage of 3.3 V, so accuracy of this operating point is optimal.

Power down this block for minimum system power. By default, the oscillator is configured through the Flash Option bits. However, the user code can override these trim values, as described in [the Trim Bit Address Space](#) section on page 151.

Select one of the two frequencies for the oscillator: 5.53MHz and 32.8kHz, using the OSCSEL bits in the [Oscillator Control](#) chapter on page 169.

eZ8 CPU Instruction Set

This chapter describes the following features of the eZ8 CPU instruction set:

[Assembly Language Programming Introduction](#): see page 174

[Assembly Language Syntax](#): see page 175

[eZ8 CPU Instruction Notation](#): see page 176

[eZ8 CPU Instruction Classes](#): see page 178

[eZ8 CPU Instruction Summary](#): see page 182

Assembly Language Programming Introduction

The eZ8 CPU assembly language provides a means for writing an application program without concern for actual memory addresses or machine instruction formats. A program written in assembly language is called a source program. Assembly language allows the use of symbolic addresses to identify memory locations. It also allows mnemonic codes (opcodes and operands) to represent the instructions themselves. The opcodes identify the instruction while the operands represent memory locations, registers, or immediate data values.

Each assembly language program consists of a series of symbolic commands called *statements*. Each statement can contain labels, operations, operands, and comments.

Labels are assigned to a particular instruction step in a source program. The label identifies that step in the program as an entry point for use by other instructions.

The assembly language also includes assembler directives that supplement the machine instruction. The assembler directives, or pseudo-ops, are not translated into a machine instruction. Rather, the pseudo-ops are interpreted as directives that control or assist the assembly process.

The source program is processed (assembled) by the assembler to obtain a machine language program called the object code. The object code is executed by the eZ8 CPU. An example segment of an assembly language program is detailed in the following example.

Assembly Language Source Program Example

```

JP START          ; Everything after the semicolon is a comment.
START:           ; A label called 'START'. The first instruction (JP START) in this
                 ; example causes program execution to jump to the point within the
                 ; program where the START label occurs.

LD R4, R7        ; A Load (LD) instruction with two operands. The first operand,
                 ; Working Register R4, is the destination. The second operand,
                 ; Working Register R7, is the source. The contents of R7 is
                 ; written into R4.

LD 234H, #%01    ; Another Load (LD) instruction with two operands.
                 ; The first operand, Extended Mode Register Address 234H,
                 ; identifies the destination. The second operand, Immediate Data
                 ; value 01H, is the source. The value 01H is written into the
                 ; Register at address 234H.
    
```

Assembly Language Syntax

For proper instruction execution, eZ8 CPU assembly language syntax requires that the operands be written as 'destination, source'. After assembly, the object code usually has the operands in the order 'source, destination', but ordering is opcode-dependent. The following instruction examples illustrate the format of some basic assembly instructions and the resulting object code produced by the assembler. You must follow this binary format if you prefer manual program coding or intend to implement your own assembler.

Example 1

If the contents of registers 43H and 08H are added and the result is stored in 43H, the assembly syntax and resulting object code is shown in Table 106.

Table 106. Assembly Language Syntax Example 1

| | | | | |
|-------------------------------|-----|------|-----|----------------|
| Assembly Language Code | ADD | 43H, | 08H | (ADD dst, src) |
| Object Code | 04 | 08 | 43 | (OPC src, dst) |

Example 2

In general, when an instruction format requires an 8-bit register address, that address can specify any register location in the range 0–255 or, using Escaped Mode Addressing, a Working Register R0–R15. If the contents of Register 43H and Working Register R8 are added and the result is stored in 43H, the assembly syntax and resulting object code is shown in Table 107.

Table 107. Assembly Language Syntax Example 2

| | | | | |
|-------------------------------|-----|------|----|----------------|
| Assembly Language Code | ADD | 43H, | R8 | (ADD dst, src) |
| Object Code | 04 | E8 | 43 | (OPC src, dst) |

See the device-specific Z8 Encore! XP Product Specification to determine the exact register file range available. The register file size varies, depending on the device type.

eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary and Description sections, the operands, condition codes, status flags, and address modes are represented by a notational shorthand that is noted in Table 108.

Table 108. Notational Shorthand

| Notation | Description | Operand | Range |
|----------|--------------------------------|---------|---|
| b | Bit | b | b represents a value from 0 to 7 (000B to 111B). |
| cc | Condition Code | — | See the Condition Codes overview in the eZ8 CPU Core User Manual (UM0128) . |
| DA | Direct Address | Addr | Addr represents a number in the range of 0000H to FFFFH. |
| ER | Extended Addressing Register | Reg | Reg represents a number in the range of 000H to FFFH. |
| IM | Immediate Data | #Data | Data is a number between 00H to FFH. |
| Ir | Indirect Working Register | @Rn | n = 0–15. |
| IR | Indirect Register | @Reg | Reg. represents a number in the range of 00H to FFH. |
| Irr | Indirect Working Register Pair | @RRp | p = 0, 2, 4, 6, 8, 10, 12, or 14. |
| IRR | Indirect Register Pair | @Reg | Reg represents an even number in the range 00H to FEH |
| p | Polarity | p | Polarity is a single bit binary value of either 0B or 1B. |
| r | Working Register | Rn | n = 0–15. |
| R | Register | Reg | Reg. represents a number in the range of 00H to FFH. |

Table 108. Notational Shorthand (Continued)

| Notation | Description | Operand | Range |
|----------|-----------------------|---------|---|
| RA | Relative Address | X | X represents an index in the range of +127 to -128 which is an offset relative to the address of the next instruction |
| rr | Working Register Pair | RRp | p = 0, 2, 4, 6, 8, 10, 12, or 14. |
| RR | Register Pair | Reg | Reg. represents an even number in the range of 00H to FEH. |
| Vector | Vector Address | Vector | Vector represents a number in the range of 00H to FFH. |
| X | Indexed | #Index | The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to -128 range. |

Table 109 lists additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 109. Additional Symbols

| Symbol | Definition |
|--------|---------------------------|
| dst | Destination Operand |
| src | Source Operand |
| @ | Indirect Address Prefix |
| SP | Stack Pointer |
| PC | Program Counter |
| FLAGS | Flags Register |
| RP | Register Pointer |
| # | Immediate Operand Prefix |
| B | Binary Number Suffix |
| % | Hexadecimal Number Prefix |
| H | Hexadecimal Number Suffix |

Assignment of a value is indicated by an arrow, as shown in the following example.

dst ← dst + src

This example indicates that the source data is added to the destination data; the result is stored in the destination location.

eZ8 CPU Instruction Classes

eZ8 CPU instructions are divided functionally into the following groups:

- Arithmetic
- Bit Manipulation
- Block Transfer
- CPU Control
- Load
- Logical
- Program Control
- Rotate and Shift

Tables 110 through 117 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instruction can be considered as a subset of more than one category. Within these tables, the source operand is identified as 'src', the destination operand is 'dst' and a condition code is 'cc'.

Table 110. Arithmetic Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|--|
| ADC | dst, src | Add with Carry |
| ADCX | dst, src | Add with Carry using Extended Addressing |
| ADD | dst, src | Add |
| ADDX | dst, src | Add using Extended Addressing |
| CP | dst, src | Compare |
| CPC | dst, src | Compare with Carry |
| CPCX | dst, src | Compare with Carry using Extended Addressing |
| CPX | dst, src | Compare using Extended Addressing |
| DA | dst | Decimal Adjust |
| DEC | dst | Decrement |
| DECW | dst | Decrement Word |
| INC | dst | Increment |
| INCW | dst | Increment Word |

Table 110. Arithmetic Instructions (Continued)

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---|
| MULT | dst | Multiply |
| SBC | dst, src | Subtract with Carry |
| SBCX | dst, src | Subtract with Carry using Extended Addressing |
| SUB | dst, src | Subtract |
| SUBX | dst, src | Subtract using Extended Addressing |

Table 111. Bit Manipulation Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|--|
| BCLR | bit, dst | Bit Clear |
| BIT | p, bit, dst | Bit Set or Clear |
| BSET | bit, dst | Bit Set |
| BSWAP | dst | Bit Swap |
| CCF | — | Complement Carry Flag |
| RCF | — | Reset Carry Flag |
| SCF | — | Set Carry Flag |
| TCM | dst, src | Test Complement Under Mask |
| TCMX | dst, src | Test Complement Under Mask using Extended Addressing |
| TM | dst, src | Test Under Mask |
| TMX | dst, src | Test Under Mask using Extended Addressing |

Table 112. Block Transfer Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---|
| LDCI | dst, src | Load Constant to/from Program Memory and Auto-Increment Addresses |
| LDEI | dst, src | Load External Data to/from Data Memory and Auto-Increment Addresses |

Table 113. CPU Control Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|------------------------|
| ATM | — | Atomic Execution |
| CCF | — | Complement Carry Flag |
| DI | — | Disable Interrupts |
| EI | — | Enable Interrupts |
| HALT | — | HALT Mode |
| NOP | — | No Operation |
| RCF | — | Reset Carry Flag |
| SCF | — | Set Carry Flag |
| SRP | src | Set Register Pointer |
| STOP | — | STOP Mode |
| WDT | — | Watchdog Timer Refresh |

Table 114. Load Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---|
| CLR | dst | Clear |
| LD | dst, src | Load |
| LDC | dst, src | Load Constant to/from Program Memory |
| LDCI | dst, src | Load Constant to/from Program Memory and Auto-Increment Addresses |
| LDE | dst, src | Load External Data to/from Data Memory |
| LDEI | dst, src | Load External Data to/from Data Memory and Auto-Increment Addresses |
| LDWX | dst, src | Load Word using Extended Addressing |
| LDX | dst, src | Load using Extended Addressing |
| LEA | dst, X(src) | Load Effective Address |
| POP | dst | Pop |
| POPX | dst | Pop using Extended Addressing |
| PUSH | src | Push |
| PUSHX | src | Push using Extended Addressing |

Table 115. Logical Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|--|
| AND | dst, src | Logical AND |
| ANDX | dst, src | Logical AND using Extended Addressing |
| COM | dst | Complement |
| OR | dst, src | Logical OR |
| ORX | dst, src | Logical OR using Extended Addressing |
| XOR | dst, src | Logical Exclusive OR |
| XORX | dst, src | Logical Exclusive OR using Extended Addressing |

Table 116. Program Control Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|-------------------------------|
| BRK | — | On-Chip Debugger Break |
| BTJ | p, bit, src, DA | Bit Test and Jump |
| BTJNZ | bit, src, DA | Bit Test and Jump if Non-Zero |
| BTJZ | bit, src, DA | Bit Test and Jump if Zero |
| CALL | dst | Call Procedure |
| DJNZ | dst, src, RA | Decrement and Jump Non-Zero |
| IRET | — | Interrupt Return |
| JP | dst | Jump |
| JP cc | dst | Jump Conditional |
| JR | DA | Jump Relative |
| JR cc | DA | Jump Relative Conditional |
| RET | — | Return |
| TRAP | vector | Software Trap |

Table 117. Rotate and Shift Instructions

| Mnemonic | Operands | Instruction |
|-----------------|-----------------|---------------------------|
| BSWAP | dst | Bit Swap |
| RL | dst | Rotate Left |
| RLC | dst | Rotate Left through Carry |

Table 117. Rotate and Shift Instructions (Continued)

| Mnemonic | Operands | Instruction |
|----------|----------|----------------------------|
| RR | dst | Rotate Right |
| RRC | dst | Rotate Right through Carry |
| SRA | dst | Shift Right Arithmetic |
| SRL | dst | Shift Right Logical |
| SWAP | dst | Swap Nibbles |

eZ8 CPU Instruction Summary

Table 118 summarizes the eZ8 CPU instruction set. The table identifies the addressing modes employed by the instruction, the effect upon the Flags Register, the number of CPU clock cycles required for the instruction fetch, and the number of CPU clock cycles required for the instruction execution.

Table 118. eZ8 CPU Instruction Summary

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | Fetch Cycles | Instr. Cycles | |
|-------------------|---------------------|--------------|-----|--------------------|-------|---|---|---|---|--------------|---------------|---|
| | | dst | src | | C | Z | S | V | D | | | H |
| ADC dst, src | dst ← dst + src + C | r | r | 12 | * | * | * | * | 0 | * | 2 | 3 |
| | | r | lr | 13 | | | | | | | 2 | 4 |
| | | R | R | 14 | | | | | | | 3 | 3 |
| | | R | IR | 15 | | | | | | | 3 | 4 |
| | | R | IM | 16 | | | | | | | 3 | 3 |
| | | IR | IM | 17 | | | | | | | 3 | 4 |
| ADCX dst, src | dst ← dst + src + C | ER | ER | 18 | * | * | * | * | 0 | * | 4 | 3 |
| | | ER | IM | 19 | | | | | | | 4 | 3 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|----------------------|--|--------------|-----|-----------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| ADD dst, src | dst ← dst + src | r | r | 02 | * | * | * | * | 0 | * | 2 | 3 |
| | | r | lr | 03 | | | | | | | 2 | 4 |
| | | R | R | 04 | | | | | | | 3 | 3 |
| | | R | IR | 05 | | | | | | | 3 | 4 |
| | | R | IM | 06 | | | | | | | 3 | 3 |
| | | IR | IM | 07 | | | | | | | 3 | 4 |
| ADDX dst, src | dst ← dst + src | ER | ER | 08 | * | * | * | * | 0 | * | 4 | 3 |
| | | ER | IM | 09 | | | | | | | 4 | 3 |
| AND dst, src | dst ← dst AND src | r | r | 52 | – | * | * | 0 | – | – | 2 | 3 |
| | | r | lr | 53 | | | | | | | 2 | 4 |
| | | R | R | 54 | | | | | | | 3 | 3 |
| | | R | IR | 55 | | | | | | | 3 | 4 |
| | | R | IM | 56 | | | | | | | 3 | 3 |
| | | IR | IM | 57 | | | | | | | 3 | 4 |
| ANDX dst, src | dst ← dst AND src | ER | ER | 58 | – | * | * | 0 | – | – | 4 | 3 |
| | | ER | IM | 59 | | | | | | | 4 | 3 |
| ATM | Block all interrupt and DMA requests during execution of the next 3 instructions | | | 2F | – | – | – | – | – | – | 1 | 2 |
| BCLR bit, dst | dst[bit] ← 0 | r | | E2 | – | – | – | – | – | – | 2 | 2 |
| BIT p, bit, dst | dst[bit] ← p | r | | E2 | – | – | – | 0 | – | – | 2 | 2 |
| BRK | Debugger Break | | | 00 | – | – | – | – | – | – | 1 | 1 |
| BSET bit, dst | dst[bit] ← 1 | r | | E2 | – | – | – | 0 | – | – | 2 | 2 |
| BSWAP dst | dst[7:0] ← dst[0:7] | R | | D5 | X | * | * | 0 | – | – | 2 | 2 |
| BTJ p, bit, src, dst | if src[bit] = p PC ← PC + X | | r | F6 | – | – | – | – | – | – | 3 | 3 |
| | | | lr | F7 | | | | | | | 3 | 4 |

Note: Flags Notation:

- * = Value is a function of the result of the operation.
- = Unaffected.
- X = Undefined.
- 0 = Reset to 0.
- 1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|------------------------|-------------------------------------|--------------|-----|--------------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| BTJNZ bit, src, dst | if src[bit] = 1 PC ← PC + X | | r | F6 | - | - | - | - | - | - | 3 | 3 |
| | | | lr | F7 | | | | | | | 3 | 4 |
| BTJZ bit, src, dst | if src[bit] = 0 PC ← PC + X | | r | F6 | - | - | - | - | - | - | 3 | 3 |
| | | | lr | F7 | | | | | | | 3 | 4 |
| CALL dst | SP ← SP - 2 @SP ← PC PC ← dst | IRR | | D4 | - | - | - | - | - | - | 2 | 6 |
| | | DA | | D6 | | | | | | | 3 | 3 |
| CCF | C ← ~C | | | EF | * | - | - | - | - | - | 1 | 2 |
| CLR dst | dst ← 00H | R | | B0 | - | - | - | - | - | - | 2 | 2 |
| | | IR | | B1 | | | | | | | 2 | 3 |
| COM dst | dst ← ~dst | R | | 60 | - | * | * | 0 | - | - | 2 | 2 |
| | | IR | | 61 | | | | | | | 2 | 3 |
| CP dst, src | dst - src | r | r | A2 | * | * | * | * | - | - | 2 | 3 |
| | | r | lr | A3 | | | | | | | 2 | 4 |
| | | R | R | A4 | | | | | | | 3 | 3 |
| | | R | IR | A5 | | | | | | | 3 | 4 |
| | | R | IM | A6 | | | | | | | 3 | 3 |
| | | IR | IM | A7 | | | | | | | 3 | 4 |
| CPC dst, src | dst - src - C | r | r | 1F A2 | * | * | * | * | - | - | 3 | 3 |
| | | r | lr | 1F A3 | | | | | | | 3 | 4 |
| | | R | R | 1F A4 | | | | | | | 4 | 3 |
| | | R | IR | 1F A5 | | | | | | | 4 | 4 |
| | | R | IM | 1F A6 | | | | | | | 4 | 3 |
| | | IR | IM | 1F A7 | | | | | | | 4 | 4 |
| CPCX dst, src | dst - src - C | ER | ER | 1F A8 | * | * | * | * | - | - | 5 | 3 |
| | | ER | IM | 1F A9 | | | | | | | 5 | 3 |
| CPX dst, src | dst - src | ER | ER | A8 | * | * | * | * | - | - | 4 | 3 |
| | | ER | IM | A9 | | | | | | | 4 | 3 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|-------------------|--|--------------|-----|--------------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| DA dst | dst ← DA(dst) | R | | 40 | * | * | * | X | - | - | 2 | 2 |
| | | IR | | 41 | | | | | | | 2 | 3 |
| DEC dst | dst ← dst - 1 | R | | 30 | - | * | * | * | - | - | 2 | 2 |
| | | IR | | 31 | | | | | | | 2 | 3 |
| DECW dst | dst ← dst - 1 | RR | | 80 | - | * | * | * | - | - | 2 | 5 |
| | | IRR | | 81 | | | | | | | 2 | 6 |
| DI | IRQCTL[7] ← 0 | | | 8F | - | - | - | - | - | - | 1 | 2 |
| DJNZ dst, RA | dst ← dst - 1 if dst ≠ 0 PC ← PC + X | r | | 0A-FA | - | - | - | - | - | - | 2 | 3 |
| EI | IRQCTL[7] ← 1 | | | 9F | - | - | - | - | - | - | 1 | 2 |
| HALT | HALT Mode | | | 7F | - | - | - | - | - | - | 1 | 2 |
| INC dst | dst ← dst + 1 | R | | 20 | - | * | * | - | - | - | 2 | 2 |
| | | IR | | 21 | | | | | | | 2 | 3 |
| | | r | | 0E-FE | | | | | | | 1 | 2 |
| INCW dst | dst ← dst + 1 | RR | | A0 | - | * | * | * | - | - | 2 | 5 |
| | | IRR | | A1 | | | | | | | 2 | 6 |
| IRET | FLAGS ← @SP SP ← SP + 1 PC ← @SP SP ← SP + 2 IRQCTL[7] ← 1 | | | BF | * | * | * | * | * | * | 1 | 5 |
| JP dst | PC ← dst | DA | | 8D | - | - | - | - | - | - | 3 | 2 |
| | | IRR | | C4 | | | | | | | 2 | 3 |
| JP cc, dst | if cc is true PC ← dst | DA | | 0D-FD | - | - | - | - | - | - | 3 | 2 |
| JR dst | PC ← PC + X | DA | | 8B | - | - | - | - | - | - | 2 | 2 |
| JR cc, dst | if cc is true PC ← PC + X | DA | | 0B-FB | - | - | - | - | - | - | 2 | 2 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|-------------------|---------------------------------------|--------------|------|-----------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| LD dst, rc | dst ← src | r | IM | 0C-FC | - | - | - | - | - | - | 2 | 2 |
| | | r | X(r) | C7 | | | | | | | 3 | 3 |
| | | X(r) | r | D7 | | | | | | | 3 | 4 |
| | | r | lr | E3 | | | | | | | 2 | 3 |
| | | R | R | E4 | | | | | | | 3 | 2 |
| | | R | IR | E5 | | | | | | | 3 | 4 |
| | | R | IM | E6 | | | | | | | 3 | 2 |
| | | IR | IM | E7 | | | | | | | 3 | 3 |
| | | lr | r | F3 | | | | | | | 2 | 3 |
| LDC dst, src | dst ← src | r | lrr | C2 | - | - | - | - | - | - | 2 | 5 |
| | | lr | lrr | C5 | | | | | | | 2 | 9 |
| | | lrr | r | D2 | | | | | | | 2 | 5 |
| LDCI dst, src | dst ← src r ← r + 1 rr ← rr + 1 | lr | lrr | C3 | - | - | - | - | - | - | 2 | 9 |
| | | lrr | lr | D3 | | | | | | | 2 | 9 |
| LDE dst, src | dst ← src | r | lrr | 82 | - | - | - | - | - | - | 2 | 5 |
| | | lrr | r | 92 | | | | | | | 2 | 5 |
| LDEI dst, src | dst ← src r ← r + 1 rr ← rr + 1 | lr | lrr | 83 | - | - | - | - | - | - | 2 | 9 |
| | | lrr | lr | 93 | | | | | | | 2 | 9 |
| LDWX dst, src | dst ← src | ER | ER | 1FE8 | - | - | - | - | - | - | 5 | 4 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|-------------------|-------------------------------------|--------------|-------|--------------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| LDX dst, src | dst ← src | r | ER | 84 | - | - | - | - | - | - | 3 | 2 |
| | | lr | ER | 85 | | | | | | | 3 | 3 |
| | | R | IRR | 86 | | | | | | | 3 | 4 |
| | | IR | IRR | 87 | | | | | | | 3 | 5 |
| | | r | X(rr) | 88 | | | | | | | 3 | 4 |
| | | X(rr) | r | 89 | | | | | | | 3 | 4 |
| | | ER | r | 94 | | | | | | | 3 | 2 |
| | | ER | lr | 95 | | | | | | | 3 | 3 |
| | | IRR | R | 96 | | | | | | | 3 | 4 |
| | | IRR | IR | 97 | | | | | | | 3 | 5 |
| | | ER | ER | E8 | | | | | | | 4 | 2 |
| ER | IM | E9 | | | | | | | 4 | 2 | | |
| LEA dst, X(src) | dst ← src + X | r | X(r) | 98 | - | - | - | - | - | - | 3 | 3 |
| | | rr | X(rr) | 99 | | | | | | | 3 | 5 |
| MULT dst | dst[15:0] ← dst[15:8] * dst[7:0] | RR | | F4 | - | - | - | - | - | - | 2 | 8 |
| NOP | No operation | | | 0F | - | - | - | - | - | - | 1 | 2 |
| OR dst, src | dst ← dst OR src | r | r | 42 | - | * | * | 0 | - | - | 2 | 3 |
| | | r | lr | 43 | | | | | | | 2 | 4 |
| | | R | R | 44 | | | | | | | 3 | 3 |
| | | R | IR | 45 | | | | | | | 3 | 4 |
| | | R | IM | 46 | | | | | | | 3 | 3 |
| | | IR | IM | 47 | | | | | | | 3 | 4 |
| ORX dst, src | dst ← dst OR src | ER | ER | 48 | - | * | * | 0 | - | - | 4 | 3 |
| | | ER | IM | 49 | | | | | | | 4 | 3 |
| POP dst | dst ← @SP SP ← SP + 1 | R | | 50 | - | - | - | - | - | - | 2 | 2 |
| | | IR | | 51 | | | | | | | 2 | 3 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | Fetch Cycles | Instr. Cycles | |
|-------------------|---|--------------|-----|--------------------|-------|---|---|---|---|--------------|---------------|---|
| | | dst | src | | C | Z | S | V | D | | | H |
| POPX dst | dst ← @SP SP ← SP + 1 | ER | | D8 | - | - | - | - | - | - | 3 | 2 |
| PUSH src | SP ← SP - 1 @SP ← src | R | | 70 | - | - | - | - | - | - | 2 | 2 |
| | | IR | | 71 | | | | | | | 2 | 3 |
| | | IM | | IF70 | | | | | | | 3 | 2 |
| PUSHX src | SP ← SP - 1 @SP ← src | ER | | C8 | - | - | - | - | - | - | 3 | 2 |
| RCF | C ← 0 | | | CF | 0 | - | - | - | - | - | 1 | 2 |
| RET | PC ← @SP SP ← SP + 2 | | | AF | - | - | - | - | - | - | 1 | 4 |
| RL dst |  | R | | 90 | * | * | * | * | - | - | 2 | 2 |
| | | IR | | 91 | | | | | | | 2 | 3 |
| RLC dst |  | R | | 10 | * | * | * | * | - | - | 2 | 2 |
| | | IR | | 11 | | | | | | | 2 | 3 |
| RR dst |  | R | | E0 | * | * | * | * | - | - | 2 | 2 |
| | | IR | | E1 | | | | | | | 2 | 3 |
| RRC dst |  | R | | C0 | * | * | * | * | - | - | 2 | 2 |
| | | IR | | C1 | | | | | | | 2 | 3 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|-------------------|---|--------------|-----|-----------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| SBC dst, src | $dst \leftarrow dst - src - C$ | r | r | 32 | * | * | * | * | 1 | * | 2 | 3 |
| | | r | lr | 33 | | | | | | | 2 | 4 |
| | | R | R | 34 | | | | | | | 3 | 3 |
| | | R | IR | 35 | | | | | | | 3 | 4 |
| | | R | IM | 36 | | | | | | | 3 | 3 |
| | | IR | IM | 37 | | | | | | | 3 | 4 |
| SBCX dst, src | $dst \leftarrow dst - src - C$ | ER | ER | 38 | * | * | * | * | 1 | * | 4 | 3 |
| | | ER | IM | 39 | | | | | | | 4 | 3 |
| SCF | $C \leftarrow 1$ | | | DF | 1 | - | - | - | - | - | 1 | 2 |
| SRA dst |  | R | | D0 | * | * | * | 0 | - | - | 2 | 2 |
| | | IR | | D1 | | | | | | | 2 | 3 |
| SRL dst |  | R | | 1F C0 | * | * | 0 | * | - | - | 3 | 2 |
| | | IR | | 1F C1 | | | | | | | 3 | 3 |
| SRP src | $RP \leftarrow src$ | | IM | 01 | - | - | - | - | - | - | 2 | 2 |
| STOP | STOP Mode | | | 6F | - | - | - | - | - | - | 1 | 2 |
| SUB dst, src | $dst \leftarrow dst - src$ | r | r | 22 | * | * | * | * | 1 | * | 2 | 3 |
| | | r | lr | 23 | | | | | | | 2 | 4 |
| | | R | R | 24 | | | | | | | 3 | 3 |
| | | R | IR | 25 | | | | | | | 3 | 4 |
| | | R | IM | 26 | | | | | | | 3 | 3 |
| | | IR | IM | 27 | | | | | | | 3 | 4 |
| SUBX dst, src | $dst \leftarrow dst - src$ | ER | ER | 28 | * | * | * | * | 1 | * | 4 | 3 |
| | | ER | IM | 29 | | | | | | | 4 | 3 |
| SWAP dst | $dst[7:4] \leftrightarrow dst[3:0]$ | R | | F0 | X | * | * | X | - | - | 2 | 2 |
| | | IR | | F1 | | | | | | | 2 | 3 |

Note: Flags Notation:

- * = Value is a function of the result of the operation.
- = Unaffected.
- X = Undefined.
- 0 = Reset to 0.
- 1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|-------------------|---|--------------|--------|--------------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| TCM dst, src | (NOT dst) AND src | r | r | 62 | - | * | * | 0 | - | - | 2 | 3 |
| | | r | lr | 63 | | | | | | | 2 | 4 |
| | | R | R | 64 | | | | | | | 3 | 3 |
| | | R | IR | 65 | | | | | | | 3 | 4 |
| | | R | IM | 66 | | | | | | | 3 | 3 |
| | | IR | IM | 67 | | | | | | | 3 | 4 |
| TCMX dst, src | (NOT dst) AND src | ER | ER | 68 | - | * | * | 0 | - | - | 4 | 3 |
| | | ER | IM | 69 | | | | | | | 4 | 3 |
| TM dst, src | dst AND src | r | r | 72 | - | * | * | 0 | - | - | 2 | 3 |
| | | r | lr | 73 | | | | | | | 2 | 4 |
| | | R | R | 74 | | | | | | | 3 | 3 |
| | | R | IR | 75 | | | | | | | 3 | 4 |
| | | R | IM | 76 | | | | | | | 3 | 3 |
| | | IR | IM | 77 | | | | | | | 3 | 4 |
| TMX dst, src | dst AND src | ER | ER | 78 | - | * | * | 0 | - | - | 4 | 3 |
| | | ER | IM | 79 | | | | | | | 4 | 3 |
| TRAP Vector | SP ← SP - 2 @SP ← PC SP ← SP - 1 @SP ← FLAGS PC ← @Vector | | Vector | F2 | - | - | - | - | - | - | 2 | 6 |
| WDT | | | | 5F | - | - | - | - | - | - | 1 | 2 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

- = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Table 118. eZ8 CPU Instruction Summary (Continued)

| Assembly Mnemonic | Symbolic Operation | Address Mode | | Opcode(s) (Hex) | Flags | | | | | | Fetch Cycles | Instr. Cycles |
|-------------------|--------------------|--------------|-----|--------------------|-------|---|---|---|---|---|--------------|---------------|
| | | dst | src | | C | Z | S | V | D | H | | |
| XOR dst, src | dst ← dst XOR src | r | r | B2 | – | * | * | 0 | – | – | 2 | 3 |
| | | r | lr | B3 | | | | | | | 2 | 4 |
| | | R | R | B4 | | | | | | | 3 | 3 |
| | | R | IR | B5 | | | | | | | 3 | 4 |
| | | R | IM | B6 | | | | | | | 3 | 3 |
| | | IR | IM | B7 | | | | | | | 3 | 4 |
| XORX dst, src | dst ← dst XOR src | ER | ER | B8 | – | * | * | 0 | – | – | 4 | 3 |
| | | ER | IM | B9 | | | | | | | 4 | 3 |

Note: Flags Notation:

* = Value is a function of the result of the operation.

– = Unaffected.

X = Undefined.

0 = Reset to 0.

1 = Set to 1.

Opcode Maps

A description of the opcode map data and the abbreviations are provided in Figure 26. Figures 27 and 28 provide information about each of the eZ8 CPU instructions. Table 119 lists Opcode Map abbreviations.



Figure 26. Opcode Map Cell Description

Table 119. Opcode Map Abbreviations

| Abbreviation | Description | Abbreviation | Description |
|---------------------|------------------------------------|---|------------------------|
| b | Bit position | IRR | Indirect Register Pair |
| cc | Condition code | p | Polarity (0 or 1) |
| X | 8-bit signed index or displacement | r | 4-bit Working Register |
| DA | Destination address | R | 8-bit register |
| ER | Extended Addressing register | r1, R1, Ir1, Irr1, IR1, rr1, RR1, IRR1, ER1 | Destination address |
| IM | Immediate data value | r2, R2, Ir2, Irr2, IR2, rr2, RR2, IRR2, ER2 | Source address |
| Ir | Indirect Working Register | RA | Relative |
| IR | Indirect register | rr | Working Register Pair |
| Irr | Indirect Working Register Pair | RR | Register Pair |

| | | Lower Nibble (Hex) | | | | | | | | | | | | | | | |
|--------------------|---|---------------------------|----------------------------|------------------------------|--------------------------------|-----------------------------|-------------------------------|-------------------------------|---------------------------------|-------------------------------|--------------------------------|----------------------------|--------------------------|---------------------------|---------------------------|-------------------------|--------------------------|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Upper Nibble (Hex) | 0 | 1.1 BRK | 2.2 SRP IM | 2.3 ADD r1,r2 | 2.4 ADD r1,lr2 | 3.3 ADD R2,R1 | 3.4 ADD IR2,R1 | 3.3 ADD R1,IM | 3.4 ADD IR1,IM | 4.3 ADDX ER2,ER1 | 4.3 ADDX IM,ER1 | 2.3 DJNZ r1,X | 2.2 JR cc,X | 2.2 LD r1,IM | 3.2 JP cc,DA | 1.2 INC r1 | 1.2 NOP |
| | 1 | 2.2 RLC R1 | 2.3 RLC IR1 | 2.3 ADC r1,r2 | 2.4 ADC r1,lr2 | 3.3 ADC R2,R1 | 3.4 ADC IR2,R1 | 3.3 ADC R1,IM | 3.4 ADC IR1,IM | 4.3 ADCX ER2,ER1 | 4.3 ADCX IM,ER1 | | | | | | See 2nd Opcode Map |
| | 2 | 2.2 INC R1 | 2.3 INC IR1 | 2.3 SUB r1,r2 | 2.4 SUB r1,lr2 | 3.3 SUB R2,R1 | 3.4 SUB IR2,R1 | 3.3 SUB R1,IM | 3.4 SUB IR1,IM | 4.3 SUBX ER2,ER1 | 4.3 SUBX IM,ER1 | | | | | | 1 |
| | 3 | 2.2 DEC R1 | 2.3 DEC IR1 | 2.3 SBC r1,r2 | 2.4 SBC r1,lr2 | 3.3 SBC R2,R1 | 3.4 SBC IR2,R1 | 3.3 SBC R1,IM | 3.4 SBC IR1,IM | 4.3 SBCX ER2,ER1 | 4.3 SBCX IM,ER1 | | | | | | |
| | 4 | 2.2 DA R1 | 2.3 DA IR1 | 2.3 OR r1,r2 | 2.4 OR r1,lr2 | 3.3 OR R2,R1 | 3.4 OR IR2,R1 | 3.3 OR R1,IM | 3.4 OR IR1,IM | 4.3 ORX ER2,ER1 | 4.3 ORX IM,ER1 | | | | | | |
| | 5 | 2.2 POP R1 | 2.3 POP IR1 | 2.3 AND r1,r2 | 2.4 AND r1,lr2 | 3.3 AND R2,R1 | 3.4 AND IR2,R1 | 3.3 AND R1,IM | 3.4 AND IR1,IM | 4.3 ANDX ER2,ER1 | 4.3 ANDX IM,ER1 | | | | | | 1.2 WDT |
| | 6 | 2.2 COM R1 | 2.3 COM IR1 | 2.3 TCM r1,r2 | 2.4 TCM r1,lr2 | 3.3 TCM R2,R1 | 3.4 TCM IR2,R1 | 3.3 TCM R1,IM | 3.4 TCM IR1,IM | 4.3 TCMX ER2,ER1 | 4.3 TCMX IM,ER1 | | | | | | 1.2 STOP |
| | 7 | 2.2 PUSH R2 | 2.3 PUSH IR2 | 2.3 TM r1,r2 | 2.4 TM r1,lr2 | 3.3 TM R2,R1 | 3.4 TM IR2,R1 | 3.3 TM R1,IM | 3.4 TM IR1,IM | 4.3 TMX ER2,ER1 | 4.3 TMX IM,ER1 | | | | | | 1.2 HALT |
| | 8 | 2.5 DECW RR1 | 2.6 DECW IRR1 | 2.5 LDE r1,lrr2 | 2.9 LDEI lr1,lrr2 | 3.2 LDX r1,ER2 | 3.3 LDX lr1,ER2 | 3.4 LDX IRR2,R1 | 3.4 LDX IRR2,IR1 | 3.5 LDX r1,rr2,X | 3.4 LDX rr1,rr2,X | | | | | | 1.2 DI |
| | 9 | 2.2 RL R1 | 2.3 RL IR1 | 2.5 LDE r2,lrr1 | 2.9 LDEI lr2,lrr1 | 3.2 LDX r2,ER1 | 3.3 LDX lr2,ER1 | 3.4 LDX R2,IRR1 | 3.5 LDX IR2,IRR1 | 3.3 LEA r1,rr2,X | 3.5 LEA rr1,rr2,X | | | | | | 1.2 EI |
| | A | 2.5 INCW RR1 | 2.6 INCW IRR1 | 2.3 CP r1,r2 | 2.4 CP r1,lr2 | 3.3 CP R2,R1 | 3.4 CP IR2,R1 | 3.3 CP R1,IM | 3.4 CP IR1,IM | 4.3 CPX ER2,ER1 | 4.3 CPX IM,ER1 | | | | | | 1.4 RET |
| | B | 2.2 CLR R1 | 2.3 CLR IR1 | 2.3 XOR r1,r2 | 2.4 XOR r1,lr2 | 3.3 XOR R2,R1 | 3.4 XOR IR2,R1 | 3.3 XOR R1,IM | 3.4 XOR IR1,IM | 4.3 XORX ER2,ER1 | 4.3 XORX IM,ER1 | | | | | | 1.5 IRET |
| | C | 2.2 RRC R1 | 2.3 RRC IR1 | 2.5 LDC r1,lrr2 | 2.9 LDCI lr1,lrr2 | 2.3 JP IRR1 | 2.9 LDC lr1,lrr2 | | 3.4 LD r1,rr2,X | 3.2 PUSHX ER2 | | | | | | | 1.2 RCF |
| | D | 2.2 SRA R1 | 2.3 SRA IR1 | 2.5 LDC r2,lrr1 | 2.9 LDCI lr2,lrr1 | 2.6 CALL IRR1 | 2.2 BSWAP R1 | 3.3 CALL DA | 3.4 LD r2,r1,X | 3.2 POPX ER1 | | | | | | | 1.2 SCF |
| | E | 2.2 RR R1 | 2.3 RR IR1 | 2.2 BIT p,b,r1 | 2.3 LD r1,lr2 | 3.2 LD R2,R1 | 3.3 LD IR2,R1 | 3.2 LD R1,IM | 3.3 LD IR1,IM | 4.2 LDX ER2,ER1 | 4.2 LDX IM,ER1 | | | | | | 1.2 CCF |
| | F | 2.2 SWAP R1 | 2.3 SWAP IR1 | 2.6 TRAP Vector | 2.3 LD lr1,r2 | 2.8 MULT RR1 | 3.3 LD R2,IR1 | 3.3 BTJ p,b,r1,X | 3.4 BTJ p,b,lrr1,X | | | | | | | | |

Figure 27. First Opcode Map

| | | Lower Nibble (Hex) | | | | | | | | | | | | | | | |
|--------------------|---|--------------------|-------------------|---------------------|---------------------|---------------------|----------------------|---------------------|----------------------|-------------------------|-----------------------|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| Upper Nibble (Hex) | 0 | | | | | | | | | | | | | | | | |
| | 1 | | | | | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | | | | | | |
| | 3 | | | | | | | | | | | | | | | | |
| | 4 | | | | | | | | | | | | | | | | |
| | 5 | | | | | | | | | | | | | | | | |
| | 6 | | | | | | | | | | | | | | | | |
| | 7 | 3 , | | | | | | | | | | | | | | | |
| | 8 | | | | | | | | | | | | | | | | |
| | 9 | | | | | | | | | | | | | | | | |
| | A | | | 3.3 CPC r1,r2 | 3.4 CPC r1,r2 | 4.3 CPC R2,R1 | 4.4 CPC IR2,R1 | 4.3 CPC R1,IM | 4.4 CPC IR1,IM | 5.3 CPCX ER2,ER1 | 5.3 CPCX IM,ER1 | | | | | | |
| | B | | | | | | | | | | | | | | | | |
| | C | 3.2 SRL R1 | 3.3 SRL IR1 | | | | | | | | | | | | | | |
| | D | | | | | | | | | | | | | | | | |
| | E | | | | | | | | | 5, 4 LDWX ER2,ER1 | | | | | | | |
| | F | | | | | | | | | | | | | | | | |

Figure 28. Second Opcode Map after 1FH

Electrical Characteristics

The data in this chapter represents all known data prior to qualification and characterization of the F0823 Series of products, and is therefore subject to change. Additional electrical characteristics may be found in the individual chapters of this document.

Absolute Maximum Ratings

Stresses greater than those listed in Table 120 may cause permanent damage to the device. These ratings are stress ratings only. Operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, tie unused inputs to one of the supply voltages (V_{DD} or V_{SS}).

Table 120. Absolute Maximum Ratings

| Parameter | Minimum | Maximum | Units | Notes |
|---|---------|---------|-------|-------|
| Ambient temperature under bias | -40 | +105 | °C | |
| Storage temperature | -65 | +150 | °C | |
| Voltage on any pin with respect to V_{SS} | -0.3 | +5.5 | V | 1 |
| | -0.3 | +3.9 | V | 2 |
| Voltage on V_{DD} pin with respect to V_{SS} | -0.3 | +3.6 | V | |
| Maximum current on input and/or inactive output pin | -5 | +5 | μA | |
| Maximum output current from active output pin | -25 | +25 | mA | |
| 8-pin Packages Maximum Ratings at 0°C to 70°C | | | | |
| Total power dissipation | | 220 | mW | |
| Maximum current into V_{DD} or out of V_{SS} | | 60 | mA | |
| 20-pin Packages Maximum Ratings at 0°C to 70°C | | | | |
| Total power dissipation | | 430 | mW | |
| Maximum current into V_{DD} or out of V_{SS} | | 120 | mA | |
| 28-pin Packages Maximum Ratings at 0°C to 70°C | | | | |
| Total power dissipation | | 450 | mW | |
| Maximum current into V_{DD} or out of V_{SS} | | 125 | mA | |

Notes: Operating temperature is specified in DC Characteristics.

1. This voltage applies to all pins except the following: V_{DD} , AV_{DD} , pins supporting analog input (Port B[5:0], Port C[2:0]) and pins supporting the crystal oscillator (PA0 and PA1). On the 8-pin packages, this applies to all pins but V_{DD} .
2. This voltage applies to pins on the 20/28 pin packages supporting analog input (Port B[5:0], Port C[2:0]) and pins supporting the crystal oscillator (PA0 and PA1).

DC Characteristics

Table 121 lists the DC characteristics of the Z8 Encore! XP F0823 Series products. All voltages are referenced to V_{SS} , the primary system ground.

Table 121. DC Characteristics

| $T_A = -40^{\circ}\text{C to } +105^{\circ}\text{C}$ (unless otherwise specified) | | | | | | |
|--|---------------------------|--------------------|-------------|--------------------|---------------|--|
| Symbol | Parameter | Minimum | Typical | Maximum | Units | Conditions |
| V_{DD} | Supply Voltage | 2.7 | – | 3.6 | V | |
| V_{IL1} | Low Level Input Voltage | –0.3 | – | $0.3 \cdot V_{DD}$ | V | |
| V_{IH1} | High Level Input Voltage | $0.7 \cdot V_{DD}$ | – | 5.5 | V | For all input pins without analog or oscillator function. For all signal pins on the 8-pin devices. Programmable pull-ups must also be disabled. |
| V_{IH2} | High Level Input Voltage | $0.7 \cdot V_{DD}$ | – | $V_{DD} + 0.3$ | V | For those pins with analog or oscillator function (20-/28-pin devices only), or when programmable pull-ups are enabled. |
| V_{OL1} | Low Level Output Voltage | – | – | 0.4 | V | $I_{OL} = 2\text{mA}$; $V_{DD} = 3.0\text{V}$ High Output Drive disabled. |
| V_{OH1} | High Level Output Voltage | 2.4 | – | – | V | $I_{OH} = -2\text{mA}$; $V_{DD} = 3.0\text{V}$ High Output Drive disabled. |
| V_{OL2} | Low Level Output Voltage | – | – | 0.6 | V | $I_{OL} = 20\text{mA}$; $V_{DD} = 3.3\text{V}$ High Output Drive enabled. |
| V_{OH2} | High Level Output Voltage | 2.4 | – | – | V | $I_{OH} = -20\text{mA}$; $V_{DD} = 3.3\text{V}$ High Output Drive enabled. |
| I_{IH} | Input Leakage Current | – | ± 0.002 | ± 5 | μA | $V_{IN} = V_{DD}$ $V_{DD} = 3.3\text{V}$ |
| I_{IL} | Input Leakage Current | – | ± 0.007 | ± 5 | μA | $V_{IN} = V_{SS}$ $V_{DD} = 3.3\text{V}$ |
| I_{TL} | Tristate Leakage Current | – | – | ± 5 | μA | |

Notes:

1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.

Table 121. DC Characteristics (Continued)

| Symbol | Parameter | $T_A = -40^{\circ}\text{C to } +105^{\circ}\text{C}$ (unless otherwise specified) | | | Units | Conditions |
|------------|----------------------------|--|---------|---------|---------------|---|
| | | Minimum | Typical | Maximum | | |
| I_{LED} | Controlled Current Drive | 1.8 | 3 | 4.5 | mA | {AFS2,AFS1} = {0,0}. |
| | | 2.8 | 7 | 10.5 | mA | {AFS2,AFS1} = {0,1}. |
| | | 7.8 | 13 | 19.5 | mA | {AFS2,AFS1} = {1,0}. |
| | | 12 | 20 | 30 | mA | {AFS2,AFS1} = {1,1}. |
| C_{PAD} | GPIO Port Pad Capacitance | – | 8.0^2 | – | pF | |
| C_{XIN} | X_{IN} Pad Capacitance | – | 8.0^2 | – | pF | |
| C_{XOUT} | X_{OUT} Pad Capacitance | – | 9.5^2 | – | pF | |
| I_{PU} | Weak Pull-up Current | 30 | 100 | 350 | μA | $V_{DD} = 3.0\text{V}–3.6\text{V}$. |
| V_{RAM} | RAM Data Retention Voltage | TBD | | | V | Voltage at which RAM retains static values; no reading or writing is allowed. |

Notes:

1. This condition excludes all pins that have on-chip pull-ups, when driven Low.
2. These values are provided for design guidance only and are not tested in production.

Table 122. Power Consumption

| Symbol | Parameter | $V_{DD} = 2.7V \text{ to } 3.6V$ | | | Units | Conditions |
|-----------------|--|----------------------------------|----------------------------------|----------------------------------|---------|--|
| | | Typical ¹ | Maximum ² Std Temp | Maximum ³ Ext Temp | | |
| I_{DD} Stop | Supply Current in STOP Mode | 0.1 | 2 | 7.5 | μA | No peripherals enabled. All pins driven to V_{DD} or V_{SS} . |
| I_{DD} Halt | Supply Current in HALT Mode (with all peripherals disabled) | 35 | 55 | 65 | μA | 32kHz. |
| | | 520 | 630 | 700 | μA | 5.5MHz. |
| I_{DD} | Supply Current in ACTIVE Mode (with all peripherals disabled) | 2.8 | 4.5 | 4.8 | mA | 32kHz. |
| | | 4.5 | 5.2 | 5.2 | mA | 5.5MHz. |
| I_{DD} WDT | Watchdog Timer Supply Current | 0.9 | 1.0 | 1.1 | μA | |
| I_{DD} IPO | Internal Precision Oscillator Supply Current | 350 | 500 | 550 | μA | |
| I_{DD} VBO | Voltage Brown-Out Supply Current | 50 | | | μA | For 20-/28-pin devices (VBO only). ⁴ For 8-pin devices. ⁴ |
| I_{DD} ADC | Analog-to-Digital Converter Supply Current (with External Reference) | 2.8 | 3.1 | 3.2 | mA | 32kHz. |
| | | 3.1 | 3.6 | 3.7 | mA | 5.5MHz. |
| | | 3.3 | 3.7 | 3.8 | mA | 10MHz. |
| | | 3.7 | 4.2 | 4.3 | mA | 20MHz. |
| I_{DD} ADCRef | ADC Internal Reference Supply Current | 0 | | | μA | See Note 4. |
| I_{DD} CMP | Comparator supply Current | 150 | 180 | 190 | μA | See Note 4. |
| I_{DD} BG | Band Gap Supply Current | 320 | 480 | 500 | μA | For 20-/28-pin devices. For 8-pin devices. |

Notes:

1. Typical conditions are defined as $V_{DD} = 3.3 V$ and $+30^{\circ}C$.
2. Standard temperature is defined as $T_A = 0^{\circ}C$ to $+70^{\circ}C$; these values not tested in production for worst case behavior, but are derived from product characterization and provided for design guidance only.
3. Extended temperature is defined as $T_A = -40^{\circ}C$ to $+105^{\circ}C$; these values not tested in production for worst case behavior, but are derived from product characterization and provided for design guidance only.
4. For this block to operate, the bandgap circuit is automatically turned on and must be added to the total supply current. This bandgap current is only added once, regardless of how many peripherals are using it.

AC Characteristics

The section provides information about the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs.

Table 123. AC Characteristics

| | | V_{DD} = 2.7V to 3.6V T_A = -40°C to +105°C (unless otherwise stated) | | | |
|---------------------|------------------------|--|-------------------|--------------|--|
| Symbol | Parameter | Minimum | Maximum | Units | Conditions |
| F _{SYSCLK} | System Clock Frequency | – | 20.0* | MHz | Read-only from Flash memory. |
| | | 0.032768 | 20.0 ¹ | MHz | Program or erasure of the Flash memory. |
| T _{XIN} | System Clock Period | 50 | – | ns | T _{CLK} = 1/F _{SYSCLK} . |
| T _{XINH} | System Clock High Time | 20 | 30 | ns | T _{CLK} = 50ns. |
| T _{XINL} | System Clock Low Time | 20 | 30 | ns | T _{CLK} = 50ns. |
| T _{XINR} | System Clock Rise Time | – | 3 | ns | T _{CLK} = 50ns. |
| T _{XINF} | System Clock Fall Time | – | 3 | ns | T _{CLK} = 50ns. |

Note: *System Clock Frequency is limited by the Internal Precision Oscillator on the Z8 Encore! XP F0823 Series. See [Table 124](#) on page 200.

Table 124. Internal Precision Oscillator Electrical Characteristics

| | | V_{DD} = 2.7V to 3.6V T_A = -40°C to +105°C (unless otherwise stated) | | | | |
|--------------------|--|--|----------------|----------------|--------------|---|
| Symbol | Parameter | Minimum | Typical | Maximum | Units | Conditions |
| F _{IPO} | Internal Precision Oscillator Frequency (High Speed) | | 5.53 | | MHz | V _{DD} = 3.3V T _A = 30°C |
| F _{IPO} | Internal Precision Oscillator Frequency (Low Speed) | | 32.7 | | kHz | V _{DD} = 3.3V T _A = 30°C |
| F _{IPO} | Internal Precision Oscillator Error | | ±1 | ±4 | % | |
| T _{IPOST} | Internal Precision Oscillator Startup Time | | 3 | | µs | |

On-Chip Peripheral AC and DC Electrical Characteristics

Table 125 tabulates the electrical characteristics of the POR and VBO blocks.

Table 125. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

| Symbol | Parameter | $T_A = -40^{\circ}\text{C to } +105^{\circ}\text{C}$ | | | Units | Conditions |
|-------------------|---|--|-----------------|---------|---------------|---|
| | | Minimum | Typical* | Maximum | | |
| V_{POR} | Power-On Reset Voltage Threshold | 2.20 | 2.45 | 2.70 | V | $V_{\text{DD}} = V_{\text{POR}}$ |
| V_{VBO} | Voltage Brown-Out Reset Voltage Threshold | 2.15 | 2.40 | 2.65 | V | $V_{\text{DD}} = V_{\text{VBO}}$ |
| | V_{POR} to V_{VBO} hysteresis | | 50 | 75 | mV | |
| | Starting V_{DD} voltage to ensure valid Power-On Reset. | – | V_{SS} | – | V | |
| T_{ANA} | Power-On Reset Analog Delay | – | 70 | – | μs | $V_{\text{DD}} > V_{\text{POR}}$; T_{POR} Digital Reset delay follows T_{ANA} |
| T_{POR} | Power-On Reset Digital Delay | | 16 | | μs | 66 Internal Precision Oscillator cycles + IPO startup time (T_{IPOST}) |
| T_{SMR} | Stop Mode Recovery | | 16 | | μs | 66 Internal Precision Oscillator cycles |
| T_{VBO} | Voltage Brown-Out Pulse Rejection Period | – | 10 | – | μs | Period of time in which $V_{\text{DD}} < V_{\text{VBO}}$ without generating a Reset. |
| T_{RAMP} | Time for V_{DD} to transition from V_{SS} to V_{POR} to ensure valid Reset | 0.10 | – | 100 | ms | |
| T_{SMP} | Stop Mode Recovery pin pulse rejection period | | 20 | | ns | For any SMR pin or for the Reset pin when it is asserted in STOP Mode. |

Note: *Data in the typical column is from characterization at 3.3 V and 30°C. These values are provided for design guidance only and are not tested in production.

Table 126. Flash Memory Electrical Characteristics and Timing

| $V_{DD} = 2.7V$ to $3.6V$ $T_A = -40^{\circ}C$ to $+105^{\circ}C$ (unless otherwise stated) | | | | | |
|---|---------|---------|---------|---------|--|
| Parameter | Minimum | Typical | Maximum | Units | Notes |
| Flash Byte Read Time | 100 | – | – | ns | |
| Flash Byte Program Time | 20 | – | 40 | μs | |
| Flash Page Erase Time | 10 | – | – | ms | |
| Flash Mass Erase Time | 200 | – | – | ms | |
| Writes to Single Address Before Next Erase | – | – | 2 | | |
| Flash Row Program Time | – | – | 8 | ms | Cumulative program time for single row cannot exceed limit before next erase. This parameter is only an issue when bypassing the Flash Controller. |
| Data Retention | 100 | – | – | years | $25^{\circ}C$ |
| Endurance | 10,000 | – | – | cycles | Program/erase cycles |

Table 127. Watchdog Timer Electrical Characteristics and Timing

| $V_{DD} = 2.7V$ to $3.6V$ $T_A = -40^{\circ}C$ to $+105^{\circ}C$ (unless otherwise stated) | | | | | | |
|---|--------------------------|---------|---------|----------|-------|--|
| Symbol | Parameter | Minimum | Typical | Maximum | Units | Conditions |
| F_{WDT} | WDT Oscillator Frequency | | 10 | | kHz | |
| F_{WDT} | WDT Oscillator Error | | | ± 50 | % | |
| $T_{WDT-CAL}$ | WDT Calibrated Timeout | 0.98 | 1 | 1.02 | s | $V_{DD} = 3.3V$; $T_A = 30^{\circ}C$ |
| | | 0.70 | 1 | 1.30 | s | $V_{DD} = 2.7V$ to $3.6V$ $T_A = 0^{\circ}C$ to $70^{\circ}C$ |
| | | 0.50 | 1 | 1.50 | s | $V_{DD} = 2.7V$ to $3.6V$ $T_A = -40^{\circ}C$ to $+105^{\circ}C$ |

Table 128. Analog-to-Digital Converter Electrical Characteristics and Timing

| $V_{DD} = 3.0V$ to $3.6V$ $T_A = 0^{\circ}C$ to $+70^{\circ}C$ (unless otherwise stated) | | | | | | |
|--|--|------------|------------|------------|---------------------|---|
| Symbol | Parameter | Minimum | Typical | Maximum | Units | Conditions |
| | Resolution | 10 | | – | bits | |
| | Differential Nonlinearity (DNL) | –1.0 | – | 1.0 | LSB ³ | External $V_{REF} = 2.0V$; $R_S \leftarrow 3.0\ k\Omega$ |
| | Integral Nonlinearity (INL) | –3.0 | – | 3.0 | LSB ³ | External $V_{REF} = 2.0V$; $R_S \leftarrow 3.0\ k\Omega$ |
| | Offset Error with Calibration | | ± 1 | | LSB ³ | |
| | Absolute Accuracy with Calibration | | ± 3 | | LSB ³ | |
| V_{REF} | Internal Reference Voltage | 1.0 2.0 | 1.1 2.2 | 1.2 2.4 | V | REFSEL=01 REFSEL=10 |
| V_{REF} | Internal Reference Variation with Temperature | | ± 1.0 | | % | Temperature variation with $V_{DD} = 3.0$ |
| V_{REF} | Internal Reference Voltage Variation with V_{DD} | | ± 0.5 | | % | Supply voltage variation with $T_A = 30^{\circ}C$ |
| R_{REFOUT} | Reference Buffer Output Impedance | | 850 | | W | When the internal reference is buffered and driven out to the VREF pin (REFOUT = 1) |
| | Single-Shot Conversion Time | – | 5129 | – | System clock cycles | All measurements but temperature sensor measurement |
| | | | 10258 | | | |
| | Continuous Conversion Time | – | 256 | – | System clock cycles | All measurements but temperature sensor measurement |
| | | | 512 | | | |

Notes:

1. Analog source impedance affects the ADC offset voltage (because of pin leakage) and input settling time.
2. Devices are factory calibrated at $V_{DD} = 3.3\ V$ and $T_A = +30^{\circ}C$, so the ADC is maximally accurate under these conditions.
3. LSBs are defined assuming 10-bit resolution.
4. This is the maximum recommended resistance seen by the ADC input pin.
5. The input impedance is inversely proportional to the system clock frequency.

Table 128. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)

| $V_{DD} = 3.0V$ to $3.6V$ $T_A = 0^{\circ}C$ to $+70^{\circ}C$ (unless otherwise stated) | | | | | | |
|--|--------------------------------------|---------|---------|----------|------------|--|
| Symbol | Parameter | Minimum | Typical | Maximum | Units | Conditions |
| | Signal Input Bandwidth | – | 10 | | kHz | As defined by –3 dB point |
| R_S | Analog Source Impedance ⁴ | – | – | 10 | k Ω | In unbuffered mode |
| Z_{in} | Input Impedance | – | 150 | | k Ω | In unbuffered mode at 20MHz ⁵ |
| V_{in} | Input Voltage Range | 0 | | V_{DD} | V | Unbuffered Mode |

Notes:

1. Analog source impedance affects the ADC offset voltage (because of pin leakage) and input settling time.
2. Devices are factory calibrated at $V_{DD} = 3.3 V$ and $T_A = +30^{\circ}C$, so the ADC is maximally accurate under these conditions.
3. LSBs are defined assuming 10-bit resolution.
4. This is the maximum recommended resistance seen by the ADC input pin.
5. The input impedance is inversely proportional to the system clock frequency.

Table 129. Comparator Electrical Characteristics

| $V_{DD} = 2.7V$ to $3.6V$ $T_A = -40^{\circ}C$ to $+105^{\circ}C$ | | | | | | |
|--|---|----------|---------|------------|-------|--------------------|
| Symbol | Parameter | Minimum | Typical | Maximum | Units | Conditions |
| V_{OS} | Input DC Offset | | 5 | | mV | |
| V_{CREF} | Programmable Internal Reference Voltage | | ± 5 | | % | 20-/28-pin devices |
| | | | ± 3 | | % | 8-pin devices |
| T_{PROP} | Propagation Delay | | 200 | | ns | |
| V_{HYS} | Input Hysteresis | | 4 | | mV | |
| V_{IN} | Input Voltage Range | V_{SS} | | $V_{DD}-1$ | V | |

General Purpose I/O Port Input Data Sample Timing

Figure 29 displays a timing sequence for the GPIO port input sampling. The input value on a GPIO port pin is sampled on the rising edge of the system clock. The port value is

available to the eZ8 CPU on the second rising clock edge following the change of the port value.



Figure 29. Port Input Sample Timing

Table 130. GPIO Port Input Timing

| Parameter | Abbreviation | Delay (ns) | |
|---------------------|--|------------|---------|
| | | Minimum | Maximum |
| T _{S_PORT} | Port Input Transition to X _{IN} Rise Setup Time (Not pictured) | 5 | – |
| T _{H_PORT} | X _{IN} Rise to Port Input Transition Hold Time (Not pictured) | 0 | – |
| T _{SMR} | GPIO Port Pin Pulse Width to ensure Stop Mode Recovery (for GPIO Port Pins enabled as SMR sources) | 1 μs | |

General Purpose I/O Port Output Timing

Figure 30 and Table 131 provide timing information for GPIO Port pins.

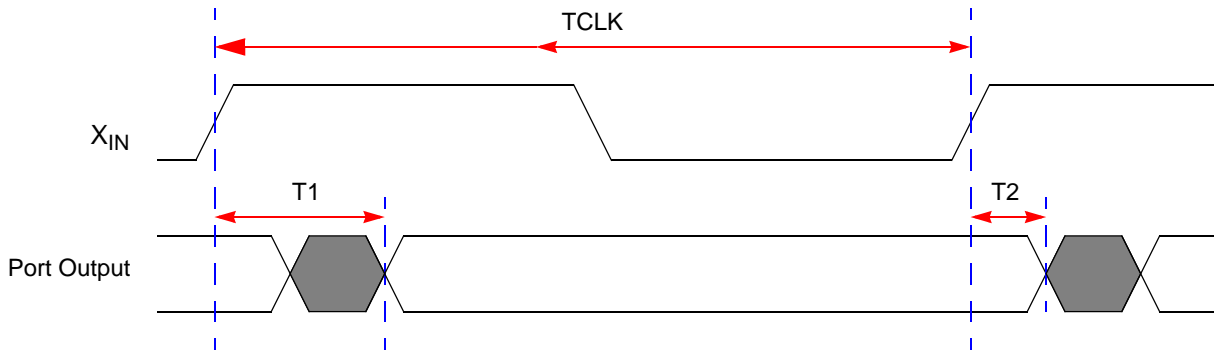


Figure 30. GPIO Port Output Timing

Table 131. GPIO Port Output Timing

| Parameter | Abbreviation | Delay (ns) | |
|-----------------------|---|------------|---------|
| | | Minimum | Maximum |
| GPIO Port pins | | | |
| T ₁ | X _{IN} Rise to Port Output Valid Delay | – | 15 |
| T ₂ | X _{IN} Rise to Port Output Hold Time | 2 | – |

On-Chip Debugger Timing

Figure 31 and Table 132 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 ns maximum rise and fall time.



Figure 31. On-Chip Debugger Timing

Table 132. On-Chip Debugger Timing

| Parameter | Abbreviation | Delay (ns) | |
|------------|---------------------------------------|------------|---------|
| | | Minimum | Maximum |
| DBG | | | |
| T_1 | X_{IN} Rise to DBG Valid Delay | – | 15 |
| T_2 | X_{IN} Rise to DBG Output Hold Time | 2 | – |
| T_3 | DBG to X_{IN} Rise Input Setup Time | 5 | – |
| T_4 | DBG to X_{IN} Rise Input Hold Time | 5 | – |

UART Timing

Figure 32 and Table 133 provide timing information for UART pins for the case where CTS is used for flow control. The CTS to DE assertion delay (T₁) assumes the transmit data register has been loaded with data prior to CTS assertion.



Figure 32. UART Timing With CTS

Table 133. UART Timing With CTS

| Parameter | Abbreviation | Delay (ns) | |
|----------------|--|----------------------------|---|
| | | Minimum | Maximum |
| UART | | | |
| T ₁ | CTS Fall to DE output delay | 2 * X _{IN} period | 2 * X _{IN} period + 1 bit time |
| T ₂ | DE assertion to TXD falling edge (start bit) delay | | ± 5 |
| T ₃ | End of Stop Bit(s) to DE deassertion delay | | ± 5 |

Figure 33 and Table 134 provide timing information for UART pins for the case where CTS is not used for flow control. DE asserts after the transmit data register has been written. DE remains asserted for multiple characters as long as the transmit data register is written with the next character before the current character has completed.



Figure 33. UART Timing Without CTS

Table 134. UART Timing Without CTS

| Parameter | Abbreviation | Delay (ns) | |
|----------------|--|----------------------------|------------|
| | | Minimum | Maximum |
| UART | | | |
| T ₁ | DE assertion to TXD falling edge (start bit) delay | 1 * X _{IN} period | 1 bit time |
| T ₂ | End of Stop Bit(s) to DE deassertion delay (Tx data register is empty) | ± 5 | |

Packaging

Zilog's F0823 Series of MCUs includes the Z8F0113, Z8F0123, Z8F0213, Z8F0223, Z8F0413, Z8F0423, Z8F0813 and Z8F0823 devices, which are available in the following packages:

- 8-pin Plastic Dual Inline Package (PDIP)
- 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S¹
- 20-pin Plastic Dual-Inline Package (PDIP)
- 20-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Small Shrink Outline Package (SSOP)
- 28-pin Plastic Dual-Inline Package (PDIP)
- 28-pin Small Outline Integrated Circuit Package (SOIC)
- 28-pin Small Shrink Outline Package (SSOP)

Current diagrams for each of these packages are published in Zilog's [Packaging Product Specification \(PS0072\)](#), which is available free for download from the Zilog website.

1. The footprint of the QFN)/MLF-S package is identical to that of the 8-pin SOIC package, but with a lower profile.

Ordering Information

Order your F0823 Series products from Zilog using the part numbers shown in Table 135. For more information about ordering, please consult your local Zilog sales office. The [Sales Location page](#) on the Zilog website lists all regional offices.

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|------|-----------|------------|---------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 8 KB Flash, 10-Bit Analog-to-Digital Converter | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0823PB005SG | 8 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0823QB005SG | 8 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0823SB005SG | 8 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0823SH005SG | 8 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0823HH005SG | 8 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0823PH005SG | 8 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0823SJ005SG | 8 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0823HJ005SG | 8 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0823PJ005SG | 8 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0823PB005EG | 8 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0823QB005EG | 8 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0823SB005EG | 8 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0823SH005EG | 8 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0823HH005EG | 8 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0823PH005EG | 8 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0823SJ005EG | 8 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0823HJ005EG | 8 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0823PJ005EG | 8 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 8 KB Flash | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0813PB005SG | 8 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0813QB005SG | 8 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0813SB005SG | 8 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0813SH005SG | 8 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0813HH005SG | 8 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0813PH005SG | 8 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0813SJ005SG | 8 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0813HJ005SG | 8 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0813PJ005SG | 8 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0813PB005EG | 8 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0813QB005EG | 8 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0813SB005EG | 8 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0813SH005EG | 8 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0813HH005EG | 8 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0813PH005EG | 8 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0813SJ005EG | 8 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0813HJ005EG | 8 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0813PJ005EG | 8 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 4 KB Flash, 10-Bit Analog-to-Digital Converter | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0423PB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0423QB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0423SB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0423SH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0423HH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0423PH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0423SJ005SG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0423HJ005SG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0423PJ005SG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0423PB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0423QB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0423SB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0423SH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0423HH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0423PH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0423SJ005EG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0423HJ005EG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0423PJ005EG | 4 KB | 1 KB | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 4 KB Flash | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0413PB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0413QB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0413SB005SG | 4 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0413SH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0413HH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0413PH005SG | 4 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0413SJ005SG | 4 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0413HJ005SG | 4 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0413PJ005SG | 4 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0413PB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0413QB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0413SB005EG | 4 KB | 1 KB | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0413SH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0413HH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0413PH005EG | 4 KB | 1 KB | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0413SJ005EG | 4 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0413HJ005EG | 4 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0413PJ005EG | 4 KB | 1 KB | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|-------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 2 KB Flash, 10-Bit Analog-to-Digital Converter | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0223PB005SG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0223QB005SG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0223SB005SG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0223SH005SG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0223HH005SG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0223PH005SG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0223SJ005SG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0223HJ005SG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0223PJ005SG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0223PB005EG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0223QB005EG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0223SB005EG | 2 KB | 512 B | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0223SH005EG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0223HH005EG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0223PH005EG | 2 KB | 512 B | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0223SJ005EG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0223HJ005EG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0223PJ005EG | 2 KB | 512 B | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|-------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 2 KB Flash | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0213PB005SG | 2 KB | 512 B | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0213QB005SG | 2 KB | 512 B | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0213SB005SG | 2 KB | 512 B | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0213SH005SG | 2 KB | 512 B | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0213HH005SG | 2 KB | 512 B | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0213PH005SG | 2 KB | 512 B | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0213SJ005SG | 2 KB | 512 B | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0213HJ005SG | 2 KB | 512 B | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0213PJ005SG | 2 KB | 512 B | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0213PB005EG | 2 KB | 512 B | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0213QB005EG | 2 KB | 512 B | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0213SB005EG | 2 KB | 512 B | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0213SH005EG | 2 KB | 512 B | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0213HH005EG | 2 KB | 512 B | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0213PH005EG | 2 KB | 512 B | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0213SJ005EG | 2 KB | 512 B | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0213HJ005EG | 2 KB | 512 B | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0213PJ005EG | 2 KB | 512 B | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|-------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 1 KB Flash, 10-Bit Analog-to-Digital Converter | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0123PB005SG | 1 KB | 256 B | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0123QB005SG | 1 KB | 256 B | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0123SB005SG | 1 KB | 256 B | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0123SH005SG | 1 KB | 256 B | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0123HH005SG | 1 KB | 256 B | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0123PH005SG | 1 KB | 256 B | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0123SJ005SG | 1 KB | 256 B | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0123HJ005SG | 1 KB | 256 B | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0123PJ005SG | 1 KB | 256 B | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0123PB005EG | 1 KB | 256 B | 6 | 12 | 2 | 4 | 1 | PDIP 8-pin package |
| Z8F0123QB005EG | 1 KB | 256 B | 6 | 12 | 2 | 4 | 1 | QFN 8-pin package |
| Z8F0123SB005EG | 1 KB | 256 B | 6 | 12 | 2 | 4 | 1 | SOIC 8-pin package |
| Z8F0123SH005EG | 1 KB | 256 B | 16 | 18 | 2 | 7 | 1 | SOIC 20-pin package |
| Z8F0123HH005EG | 1 KB | 256 B | 16 | 18 | 2 | 7 | 1 | SSOP 20-pin package |
| Z8F0123PH005EG | 1 KB | 256 B | 16 | 18 | 2 | 7 | 1 | PDIP 20-pin package |
| Z8F0123SJ005EG | 1 KB | 256 B | 22 | 18 | 2 | 8 | 1 | SOIC 28-pin package |
| Z8F0123HJ005EG | 1 KB | 256 B | 22 | 18 | 2 | 8 | 1 | SSOP 28-pin package |
| Z8F0123PJ005EG | 1 KB | 256 B | 22 | 18 | 2 | 8 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|-------|-----------|------------|------------------------|---------------------|----------------|---------------------|
| Z8 Encore! XP F0823 Series with 1 KB Flash | | | | | | | | |
| Standard Temperature: 0°C to 70°C | | | | | | | | |
| Z8F0113PB005SG | 1 KB | 256 B | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0113QB005SG | 1 KB | 256 B | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0113SB005SG | 1 KB | 256 B | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0113SH005SG | 1 KB | 256 B | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0113HH005SG | 1 KB | 256 B | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0113PH005SG | 1 KB | 256 B | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0113SJ005SG | 1 KB | 256 B | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0113HJ005SG | 1 KB | 256 B | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0113PJ005SG | 1 KB | 256 B | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |
| Extended Temperature: -40°C to 105°C | | | | | | | | |
| Z8F0113PB005EG | 1 KB | 256 B | 6 | 12 | 2 | 0 | 1 | PDIP 8-pin package |
| Z8F0113QB005EG | 1 KB | 256 B | 6 | 12 | 2 | 0 | 1 | QFN 8-pin package |
| Z8F0113SB005EG | 1 KB | 256 B | 6 | 12 | 2 | 0 | 1 | SOIC 8-pin package |
| Z8F0113SH005EG | 1 KB | 256 B | 16 | 18 | 2 | 0 | 1 | SOIC 20-pin package |
| Z8F0113HH005EG | 1 KB | 256 B | 16 | 18 | 2 | 0 | 1 | SSOP 20-pin package |
| Z8F0113PH005EG | 1 KB | 256 B | 16 | 18 | 2 | 0 | 1 | PDIP 20-pin package |
| Z8F0113SJ005EG | 1 KB | 256 B | 24 | 18 | 2 | 0 | 1 | SOIC 28-pin package |
| Z8F0113HJ005EG | 1 KB | 256 B | 24 | 18 | 2 | 0 | 1 | SSOP 28-pin package |
| Z8F0113PJ005EG | 1 KB | 256 B | 24 | 18 | 2 | 0 | 1 | PDIP 28-pin package |

Table 135. Z8 Encore! XP F0823 Series Ordering Matrix (Continued)

| Part Number | Flash | RAM | I/O Lines | Interrupts | 16-Bit Timers w/PWM | 10-Bit A/D Channels | UART with IrDA | Description |
|---|-------|-----|-----------|------------|------------------------|---------------------|----------------|---|
| Z8 Encore! XP F0823 Series Development Kit | | | | | | | | |
| Z8F08A28100KITG | | | | | | | | Z8 Encore! XP F082A Series Development Kit (20- and 28-Pin) |
| Z8F04A28100KITG | | | | | | | | Z8 Encore! XP F042A Series Development Kit (20- and 28-Pin) |
| Z8F04A08100KITG | | | | | | | | Z8 Encore! XP F042A Series Development Kit (8-Pin) |
| ZUSBSC00100ZACG | | | | | | | | USB Smart Cable Accessory Kit |
| ZUSBOPTSC01ZACG | | | | | | | | Opto-Isolated USB Smart Cable Accessory Kit |
| ZENETSC0100ZACG | | | | | | | | Ethernet Smart Cable Accessory Kit |

Part Number Suffix Designations

Zilog part numbers consist of a number of components, as indicated in the following example.

Example. Part number Z8F0423SH005SG is an 8-bit 20MHz Flash MCU with 4KB of Program Memory and equipped with 6–22 I/O lines and 4–8 ADC channels in a 20-pin SOIC package, operating within a 0°C to +70°C temperature range and built using lead-free solder.



Index

Numerics

10-bit ADC 4

A

absolute maximum ratings 196
AC characteristics 200
ADC 178
 architecture 121
 block diagram 122
 continuous conversion 124
 control register 126, 129
 control register definitions 126
 data high byte register 130
 data low bits register 131
 electrical characteristics and timing 203
 operation 122
 single-shot conversion 123
ADCCTL register 126, 129
ADCDH register 130
ADCDL register 131
ADCX 178
ADD 178
add - extended addressing 178
add with carry 178
add with carry - extended addressing 178
additional symbols 177
address space 13
ADDX 178
analog signals 10
analog-to-digital converter (ADC) 121
AND 181
ANDX 181
arithmetic instructions 178
assembly language programming 174
assembly language syntax 175

B

B 177

b 176
baud rate generator, UART 108
BCLR 179
binary number suffix 177
BIT 179
bit 176
 clear 179
 manipulation instructions 179
 set 179
 set or clear 179
 swap 179
 test and jump 181
 test and jump if non-zero 181
 test and jump if zero 181
bit jump and test if non-zero 181
bit swap 181
block diagram 3
block transfer instructions 179
BRK 181
BSET 179
BSWAP 179, 181
BTJ 181
BTJNZ 181
BTJZ 181

C

CALL procedure 181
CAPTURE mode 89
CAPTURE/COMPARE mode 89
cc 176
CCF 180
characteristics, electrical 196
clear 180
CLR 180
COM 181
COMPARE 89
compare - extended addressing 178
COMPARE mode 89
compare with carry 178

- compare with carry - extended addressing 178
- complement 181
- complement carry flag 179, 180
- condition code 176
- continuous conversion (ADC) 124
- CONTINUOUS mode 88
- control register definition, UART 108
- Control Registers 13, 16
- COUNTER modes 89
- CP 178
- CPC 178
- CPCX 178
- CPU and peripheral overview 4
- CPU control instructions 180
- CPX 178
- Customer Support 230

D

- DA 176, 178
- data memory 15
- DC characteristics 197
- debugger, on-chip 156
- DEC 178
- decimal adjust 178
- decrement 178
- decrement and jump non-zero 181
- decrement word 178
- DECW 178
- destination operand 177
- device, port availability 33
- DI 180
- direct address 176
- disable interrupts 180
- DJNZ 181
- dst 177

E

- EI 180
- electrical characteristics 196
 - ADC 203
 - flash memory and timing 202
 - GPIO input data sample timing 204

- Watchdog Timer 202, 204
- enable interrupt 180
- ER 176
- extended addressing register 176
- external pin reset 25
- eZ8 CPU features 4
- eZ8 CPU instruction classes 178
- eZ8 CPU instruction notation 176
- eZ8 CPU instruction set 174
- eZ8 CPU instruction summary 182

F

- FCTL register 141, 148, 149
- features, Z8 Encore! 1
- first opcode map 194
- FLAGS 177
- flags register 177
- flash
 - controller 4
 - option bit address space 149
 - option bit configuration - reset 146
 - program memory address 0000H 149
 - program memory address 0001H 150
- flash memory 134
 - arrangement 135
 - byte programming 139
 - code protection 137
 - configurations 134
 - control register definitions 141, 148
 - controller bypass 140
 - electrical characteristics and timing 202
 - flash control register 141, 148, 149
 - flash option bits 138
 - flash status register 142
 - flow chart 136
 - frequency high and low byte registers 144
 - mass erase 139
 - operation 135
 - operation timing 137
 - page erase 139
 - page select register 142, 144
- FPS register 142, 144
- FSTAT register 142

G

- GATED mode 89
- general-purpose I/O 33
- GPIO 4, 33
 - alternate functions 34
 - architecture 34
 - control register definitions 40
 - input data sample timing 204
 - interrupts 40
 - port A-C pull-up enable sub-registers 47, 48, 49
 - port A-H address registers 41
 - port A-H alternate function sub-registers 43
 - port A-H control registers 42
 - port A-H data direction sub-registers 43
 - port A-H high drive enable sub-registers 45
 - port A-H input data registers 50
 - port A-H output control sub-registers 44
 - port A-H output data registers 51
 - port A-H stop mode recovery sub-registers 46
 - port availability by device 33
 - port input timing 205
 - port output timing 206

H

- H 177
- HALT 180
- halt mode 31, 180
- hexadecimal number prefix/suffix 177

I

- I2C 4
- IM 176
- immediate data 176
- immediate operand prefix 177
- INC 178
- increment 178
- increment word 178
- INCW 178
- indexed 177
- indirect address prefix 177
- indirect register 176

- indirect register pair 176
- indirect working register 176
- indirect working register pair 176
- infrared encoder/decoder (IrDA) 117
- Instruction Set 174
- instruction set, eZ8 CPU 174
- instructions
 - ADC 178
 - ADCX 178
 - ADD 178
 - ADDX 178
 - AND 181
 - ANDX 181
 - arithmetic 178
 - BCLR 179
 - BIT 179
 - bit manipulation 179
 - block transfer 179
 - BRK 181
 - BSET 179
 - BSWAP 179, 181
 - BTJ 181
 - BTJNZ 181
 - BTJZ 181
 - CALL 181
 - CCF 179, 180
 - CLR 180
 - COM 181
 - CP 178
 - CPC 178
 - CPCX 178
 - CPU control 180
 - CPX 178
 - DA 178
 - DEC 178
 - DECW 178
 - DI 180
 - DJNZ 181
 - EI 180
 - HALT 180
 - INC 178
 - INCW 178
 - IRET 181
 - JP 181

- LD 180
- LDC 180
- LDCI 179, 180
- LDE 180
- LDEI 179
- LDX 180
- LEA 180
- load 180
- logical 181
- MULT 179
- NOP 180
- OR 181
- ORX 181
- POP 180
- POPX 180
- program control 181
- PUSH 180
- PUSHX 180
- RCF 179, 180
- RET 181
- RL 181
- RLC 181
- rotate and shift 181
- RR 182
- RRC 182
- SBC 179
- SCF 179, 180
- SRA 182
- SRL 182
- SRP 180
- STOP 180
- SUB 179
- SUBX 179
- SWAP 182
- TCM 179
- TCMX 179
- TM 179
- TMX 179
- TRAP 181
- Watchdog Timer refresh 180
- XOR 181
- XORX 181
- instructions, eZ8 classes of 178
- interrupt control register 68
- Interrupt Controller 54
- interrupt controller
 - architecture 54
 - interrupt assertion types 57
 - interrupt vectors and priority 57
 - operation 56
 - register definitions 59
 - software interrupt assertion 58
- interrupt edge select register 66
- interrupt request 0 register 59
- interrupt request 1 register 60
- interrupt request 2 register 61
- interrupt return 181
- interrupt vector listing 54
- interrupts
 - UART 105
- IR 176
- Ir 176
- IrDA
 - architecture 117
 - block diagram 117
 - control register definitions 120
 - operation 117
 - receiving data 119
 - transmitting data 118
- IRET 181
- IRQ0 enable high and low bit registers 61
- IRQ1 enable high and low bit registers 63
- IRQ2 enable high and low bit registers 65
- IRR 176
- Irr 176

- J**
- JP 181
- jump, conditional, relative, and relative conditional 181

- L**
- LD 180
- LDC 180
- LDCI 179, 180
- LDE 180

LDEI 179, 180
LDX 180
LEA 180
load 180
load constant 179
load constant to/from program memory 180
load constant with auto-increment addresses 180
load effective address 180
load external data 180
load external data to/from data memory and auto-increment addresses 179
load external to/from data memory and auto-increment addresses 180
load instructions 180
load using extended addressing 180
logical AND 181
logical AND/extended addressing 181
logical exclusive OR 181
logical exclusive OR/extended addressing 181
logical instructions 181
logical OR 181
logical OR/extended addressing 181
low power modes 30

M

master interrupt enable 56
memory
 data 15
 program 13
mode
 CAPTURE 89
 CAPTURE/COMPARE 89
 CONTINUOUS 88
 COUNTER 89
 GATED 89
 ONE-SHOT 88
 PWM 89
modes 89
MULT 179
multiply 179
MULTIPROCESSOR mode, UART 103

N

NOP (no operation) 180
notation
 b 176
 cc 176
 DA 176
 ER 176
 IM 176
 IR 176
 Ir 176
 IRR 176
 Irr 176
 p 176
 R 176
 r 176
 RA 177
 RR 177
 rr 177
 vector 177
 X 177
notational shorthand 176

O

OCD
 architecture 156
 auto-baud detector/generator 159
 baud rate limits 160
 block diagram 156
 breakpoints 161
 commands 162
 control register 166
 data format 159
 DBG pin to RS-232 Interface 157
 DEBUG mode 158
 debugger break 181
 interface 157
 serial errors 160
 status register 168
 timing 207
OCD commands
 execute instruction (12H) 166
 read data memory (0DH) 165
 read OCD control register (05H) 163

read OCD revision (00H) 163
 read OCD status register (02H) 163
 read program counter (07H) 164
 read program memory (0BH) 164
 read program memory CRC (0EH) 165
 read register (09H) 164
 read runtime counter (03H) 163
 step instruction (10H) 165
 stuff instruction (11H) 166
 write data memory (0CH) 165
 write OCD control register (04H) 163
 write program counter (06H) 163
 write program memory (0AH) 164
 write register (08H) 164
 on-chip debugger (OCD) 156
 on-chip debugger signals 10
 ONE-SHOT mode 88
 opcode map
 abbreviations 193
 cell description 192
 first 194
 second after 1FH 195
 Operational Description 21, 30, 33, 69, 91, 97, 117,
 121, 132, 134, 146, 156, 169, 173
 OR 181
 ordering information 211
 ORX 181

P

p 176
 Packaging 210
 part selection guide 2
 PC 177
 peripheral AC and DC electrical characteristics 201
 pin characteristics 11
 Pin Descriptions 7
 polarity 176
 POP 180
 pop using extended addressing 180
 POPX 180
 port availability, device 33
 port input timing (GPIO) 205
 port output timing, GPIO 206

power supply signals 10
 Power-on and Voltage Brownout electrical charac-
 teristics and timing 201
 Power-On Reset (POR) 23
 program control instructions 181
 program counter 177
 program memory 13
 PUSH 180
 push using extended addressing 180
 PUSHX 180
 PWM mode 89
 PxADDR register 41
 PxCTL register 42

R

R 176
 r 176
 RA
 register address 177
 RCF 179, 180
 receive
 IrDA data 119
 receiving UART data-interrupt-driven method 102
 receiving UART data-pollled method 101
 register 176
 ADC control (ADCCTL) 126, 129
 ADC data high byte (ADCDH) 130
 ADC data low bits (ADCDL) 131
 flash control (FCTL) 141, 148, 149
 flash high and low byte (FFREQH and FRE-
 EQL) 144
 flash page select (FPS) 142, 144
 flash status (FSTAT) 142
 GPIO port A-H address (PxADDR) 41
 GPIO port A-H alternate function sub-registers
 44
 GPIO port A-H control address (PxCTL) 42
 GPIO port A-H data direction sub-registers 43
 OCD control 166
 OCD status 168
 UARTx baud rate high byte (UxBRH) 115
 UARTx baud rate low byte (UxBRL) 115
 UARTx Control 0 (UxCTL0) 112, 115

- UARTx control 1 (UxCTL1) 113
 - UARTx receive data (UxRXD) 109
 - UARTx status 0 (UxSTAT0) 110
 - UARTx status 1 (UxSTAT1) 111
 - UARTx transmit data (UxTXD) 109
 - Watchdog Timer control (WDTCTL) 94, 133
 - watch-dog timer control (WDTCTL) 172
 - Watchdog Timer reload high byte (WDTH) 95
 - Watchdog Timer reload low byte (WDTL) 95
 - Watchdog Timer reload upper byte (WDTU) 95
 - register file 13
 - register pair 177
 - register pointer 177
 - reset
 - and stop mode characteristics 21
 - and stop mode recovery 21
 - carry flag 179
 - sources 23
 - RET 181
 - return 181
 - RL 181
 - RLC 181
 - rotate and shift instructions 181
 - rotate left 181
 - rotate left through carry 181
 - rotate right 182
 - rotate right through carry 182
 - RP 177
 - RR 177, 182
 - rr 177
 - RRC 182
- S**
- SBC 179
 - SCF 179, 180
 - second opcode map after 1FH 195
 - set carry flag 179, 180
 - set register pointer 180
 - shift right arithmetic 182
 - shift right logical 182
 - signal descriptions 9
 - single-sho conversion (ADC) 123
 - software trap 181
 - source operand 177
 - SP 177
 - SRA 182
 - src 177
 - SRL 182
 - SRP 180
 - stack pointer 177
 - STOP 180
 - STOP mode 30, 180
 - Stop Mode Recovery
 - sources 26
 - using a GPIO port pin transition 27, 28
 - using Watchdog Timer time-out 27
 - SUB 179
 - subtract 179
 - subtract - extended addressing 179
 - subtract with carry 179
 - subtract with carry - extended addressing 179
 - SUBX 179
 - SWAP 182
 - swap nibbles 182
 - symbols, additional 177
- T**
- TCM 179
 - TCMX 179
 - test complement under mask 179
 - test complement under mask - extended addressing 179
 - test under mask 179
 - test under mask - extended addressing 179
 - timer signals 9
 - timers 69
 - architecture 70
 - block diagram 70
 - CAPTURE mode 78, 79, 89
 - CAPTURE/COMPARE mode 82, 89
 - COMPARE mode 80, 89
 - CONTINUOUS mode 71, 88
 - COUNTER mode 72, 73
 - COUNTER modes 89
 - GATED mode 81, 89

- ONE-SHOT mode 70, 88
- operating mode 70
- PWM mode 75, 76, 89
- reading the timer count values 83
- reload high and low byte registers 84
- timer control register definitions 83
- timer output signal operation 83
- timers 0-3
 - control registers 86, 87
 - high and low byte registers 83, 86
- TM 179
- TMX 179
- tools, hardware and software 220
- transmit
 - IrDA data 118
- transmitting UART data-pollled method 99
- transmitting UART dat-interrupt-driven method 100
- TRAP 181

U

- UART 4
 - architecture 97
 - baud rate generator 108
 - control register definitions 108
 - controller signals 9
 - interrupts 105
 - MULTIPROCESSOR mode 103
 - receiving data using interrupt-driven method 102
 - receiving data using the polled method 101
 - transmitting data using the interrupt-driven method 100
 - transmitting data using the polled method 99
 - x baud rate high and low registers 115
 - x control 0 and control 1 registers 112
 - x status 0 and status 1 registers 110, 111
- UxBRH register 115
- UxBRL register 115
- UxCTL0 register 112, 115
- UxCTL1 register 113
- UxRXD register 109
- UxSTAT0 register 110

- UxSTAT1 register 111
- UxTXD register 109

V

- vector 177
- Voltage Brownout reset (VBR) 24

W

- Watchdog Timer
 - approximate time-out delay 91
 - CNTL 24
 - control register 94, 171
 - electrical characteristics and timing 202, 204
 - interrupt in normal operation 92
 - interrupt in STOP mode 92
 - refresh 92, 180
 - reload unlock sequence 93
 - reload upper, high and low registers 94
 - reset 25
 - reset in normal operation 93
 - reset in STOP mode 93
 - time-out response 92
- Watchdog Timer Control Register (WDTCTL) 94
- WDTCTL register 94, 133, 172
- WDTH register 95
- WDTL register 95
- WDTU register 95
- working register 176
- working register pair 177

X

- X 177
- XOR 181
- XORX 181

Z

- Z8 Encore!
 - block diagram 3
 - features 1

part selection guide 2

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at <http://support.zilog.com>.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at <http://zilog.com/kb> or consider participating in the Zilog Forum at <http://zilog.com/forum>.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at <http://www.zilog.com>.



Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный)

Факс: 8 (812) 320-02-42

Электронная почта: org@eplast1.ru

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.