THE DATASHEET OF

Z8F022ASB020SG

www.win-source.net 0086-755-83957316

zilog
Embedded in Life
An IXYS Company

High-Performance 8-Bit Microcontrollers

Z8 Encore! XP® F082A Series

Product Specification

PS022829-0814

Copyright ©2014 Zilog®, Inc. All rights reserved.
www.zilog.com

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

©2014 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concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded.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.
Aug 2014	29	Direct LED Drive Features Alternative Function Register Port Alternate Function Mapping	Clarified the Enabling through the LED senence. Corrected Port C enabling sentence. Added LED Drive to the Alternate function description in table 14.	38, 40, 53
Apr 2013	28	Timer Pin Signal Operation	Clarified use/availabity of the T0OUT and T1OUT timer functions by mode.	84
Dec 2012	27	Port Alternate Function Mapping (Non 8-Pin Parts), Port Alternate Function Mapping (8-Pin Parts)	Added missing Port D data to Table 15; corrected active Low status (set overlines) for PA0 ($\overline{\text{T0OUT}}$), PA2 ($\overline{\text{RESET}}$) and PA5 ($\overline{\text{T1OUT}}$) in Table 16.	40, 43
Sep 2011	26	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.	53, 157, 245
Sep 2008	25	Overview, Address Space, Register Map, General-Purpose Input/Output, Available Packages, Ordering Information	Added references to F042A Series back in Table 1, Table 5, Table 7 and Table 14.	2, 8, 16, 18, 36, 246
May 2008	24	Overview, Address Space, Register Map, General-Purpose Input/Output, Available Packages, Ordering Information	Changed title to Z8 Encore! XP F082A Series and removed references to F042A Series in Table 1, Table 5, Table 7 and Table 14.	2, 8, 16, 18, 36, 246
Dec 2007	23	Pin Description, General-Purpose Input/Output, Watchdog Timer	Updated Figure 3, Table 15, Tables 60 through 62.	9, 40, 97
Jul 2007	22	Electrical Characteristics	Updated Tables 16 and 132; power consumption data.	43, 229
Jun 2007	21	n/a	Revision number update.	All

Table of Contents

Revision History iii

List of Figures xi

List of Tables xiii

Overview 1
- Features 1
- Part Selection Guide 2
- Block Diagram 3
- CPU and Peripheral Overview 4
  - 10-Bit Analog-to-Digital Converter 4
  - Low-Power Operational Amplifier 4
  - Internal Precision Oscillator 5
  - Temperature Sensor 5
  - Analog Comparator 5
  - External Crystal Oscillator 5
  - Low Voltage Detector 5
  - On-Chip Debugger 5
  - Universal Asynchronous Receiver/Transmitter 5
  - Timers 5
  - General-Purpose Input/Output 6
  - Direct LED Drive 6
  - Flash Controller 6
  - Non-Volatile Data Storage 6
  - Interrupt Controller 6
  - Reset Controller 6

Pin Description 8
- Available Packages 8
- Pin Configurations 8
- Signal Descriptions 10
- Pin Characteristics 12

Address Space 15
- Register File 15
- Program Memory 15
- Data Memory 17
- Flash Information Area 17

Register Map 18

Reset, Stop Mode Recovery and Low Voltage Detection 22
Reset Types 22
Reset Sources 24
Power-On Reset 24
Voltage Brown-Out Reset 25
Watchdog Timer Reset 26
External Reset Input 26
External Reset Indicator 27
On-Chip Debugger Initiated Reset 27
Stop Mode Recovery 27
Stop Mode Recovery Using Watchdog Timer Time-Out 28
Stop Mode Recovery Using a GPIO Port Pin Transition 28
Stop Mode Recovery Using the External RESET Pin 29
Low Voltage Detection 29
Reset Register Definitions 29

Low-Power Modes 32
Stop Mode 32
Halt Mode 33
Peripheral-Level Power Control 33
Power Control Register Definitions 33

General-Purpose Input/Output 36
GPIO Port Availability By Device 36
Architecture 37
GPIO Alternate Functions 37
Direct LED Drive 38
Shared Reset Pin 38
Shared Debug Pin 39
Crystal Oscillator Override 39
5V Tolerance 39
External Clock Setup 39
GPIO Interrupts 44
GPIO Control Register Definitions 44
Port A–D Address Registers 45
Port A–D Control Registers 46
Port A–D Data Direction Subregisters 46
Port A–D Alternate Function Subregisters 47
Port A–C Input Data Registers 52
Port A–D Output Data Register 52
LED Drive Enable Register 53
LED Drive Level High Register 53

LED Drive Level Low Register 54

GPIO Mode Interrupt Controller 55
Interrupt Vector Listing 55
Architecture 57
Operation 57
Master Interrupt Enable 57
Interrupt Vectors and Priority 58
Interrupt Assertion 58
Software Interrupt Assertion 59
Watchdog Timer Interrupt Assertion 59
Interrupt Control Register Definitions 60
Interrupt Request 0 Register 60
Interrupt Request 1 Register 61
Interrupt Request 2 Register 62
IRQ0 Enable High and Low Bit Registers 62
IRQ1 Enable High and Low Bit Registers 64
IRQ2 Enable High and Low Bit Registers 65
Interrupt Edge Select Register 67
Shared Interrupt Select Register 68
Interrupt Control Register 69

Timers 70
Architecture 70
Operation 71
Timer Operating Modes 71
Reading the Timer Count Values 84
Timer Pin Signal Operation 84
Timer Control Register Definitions 85
Timer 0–1 Control Registers 85
Timer 0–1 High and Low Byte Registers 89
Timer Reload High and Low Byte Registers 91
Timer 0–1 PWM High and Low Byte Registers 92

Watchdog Timer 93
Operation 93
Watchdog Timer Refresh 94
Watchdog Timer Time-Out Response 94
Watchdog Timer Reload Unlock Sequence 95
Watchdog Timer Calibration 95
Watchdog Timer Control Register Definitions 96
Watchdog Timer Control Register 96
Watchdog Timer Reload Upper, High and Low Byte Registers 97

Universal Asynchronous Receiver/Transmitter 99
Architecture 99
Operation 100
Transmitting Data using the Polled Method 101
Transmitting Data using the Interrupt-Driven Method 102
Receiving Data using the Polled Method 103
Receiving Data using the Interrupt-Driven Method 104
Clear To Send (CTS) Operation 105
MULTIPROCESSOR (9-bit) Mode 105
External Driver Enable 107
UART Interrupts 108
UART Baud Rate Generator 110
UART Control Register Definitions 110
UART Control 0 and Control 1 Registers 110
UART Status 0 Register 114
UART Status 1 Register 115
UART Transmit Data Register 115
UART Receive Data Register 116
UART Address Compare Register 116
UART Baud Rate High and Low Byte Registers 117

Infrared Encoder/Decoder 120
Architecture 120
Operation 120
Transmitting IrDA Data 121
Receiving IrDA Data 122
Infrared Encoder/Decoder Control Register Definitions 123

Analog-to-Digital Converter 124
Architecture 124
Operation 125
Hardware Overflow 126
Automatic Powerdown 126
Single-Shot Conversion 126
Continuous Conversion 127
Interrupts 128
Calibration and Compensation 129
ADC Compensation Details 130
Input Buffer Stage 133
ADC Control Register Definitions 133
ADC Control Register 0 134
ADC Control/Status Register 1 135

ADC Data High Byte Register 136
ADC Data Low Byte Register 137

Low Power Operational Amplifier 139
Operation 139

Comparator 140
Operation 140
Comparator Control Register Definition 141

Temperature Sensor 144
Temperature Sensor Operation 144

Flash Memory 146
Architecture 146
Flash Information Area 147
Operation 147
Flash Operation Timing Using the Flash Frequency Registers 149
Flash Code Protection Against External Access 149
Flash Code Protection Against Accidental Program and Erasure 149
Byte Programming 151
Page Erase 152
Mass Erase 152
Flash Controller Bypass 152
Flash Controller Behavior in Debug Mode 153
Flash Control Register Definitions 153
Flash Control Register 153
Flash Status Register 155
Flash Page Select Register 156
Flash Sector Protect Register 157
Flash Frequency High and Low Byte Registers 157

Flash Option Bits 159
Operation 159
Option Bit Configuration By Reset 159
Option Bit Types 160
Reading the Flash Information Page 161
Flash Option Bit Control Register Definitions 161
Trim Bit Address Register 161
Trim Bit Data Register 162
Flash Option Bit Address Space 162
Flash Program Memory Address 0000H 162
Flash Program Memory Address 0001H 164
Trim Bit Address Space 165

Trim Bit Address 0000H 165
Trim Bit Address 0001H 165
Trim Bit Address 0002H 166
Trim Bit Address 0003H 166
Trim Bit Address 0004H 168
Zilog Calibration Data 168
ADC Calibration Data 169
Temperature Sensor Calibration Data 171
Watchdog Timer Calibration Data 172
Serialization Data 173
Randomized Lot Identifier 174

Nonvolatile Data Storage 176
Operation 176
NVDS Code Interface 176
Byte Write 177
Byte Read 178
Power Failure Protection 178
Optimizing NVDS Memory Usage for Execution Speed 178

On-Chip Debugger 180
Architecture 180
Operation 181
OCD Interface 181
Debug Mode 182
OCD Data Format 183
OCD Auto-Baud Detector/Generator 183
OCD Serial Errors 184
OCD Unlock Sequence (8-Pin Devices Only) 185
Breakpoints 185
Runtime Counter 186
On-Chip Debugger Commands 186
On-Chip Debugger Control Register Definitions 191
OCD Control Register 191
OCD Status Register 192

Oscillator Control 193
Operation 193
System Clock Selection 193
Clock Failure Detection and Recovery 195
Oscillator Control Register Definitions 196

Crystal Oscillator 198

Operating Modes 198
Crystal Oscillator Operation 198
Oscillator Operation with an External RC Network 201

Internal Precision Oscillator 203
Operation 203

eZ8 CPU Instruction Set 204
Assembly Language Programming Introduction 204
Assembly Language Syntax 205
eZ8 CPU Instruction Notation 206
eZ8 CPU Instruction Classes 207
eZ8 CPU Instruction Summary 212

Opcode Maps 222

Electrical Characteristics 226
Absolute Maximum Ratings 226
DC Characteristics 227
AC Characteristics 232
On-Chip Peripheral AC and DC Electrical Characteristics 233
General Purpose I/O Port Input Data Sample Timing 240
General Purpose I/O Port Output Timing 241
On-Chip Debugger Timing 242
UART Timing 243

Packaging 245

Ordering Information 246
Part Number Suffix Designations 255

Index 256

Customer Support 265

List of Figures

Figure 1. Z8 Encore! XP F082A Series Block Diagram 3
Figure 2. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 8-Pin SOIC, QFN/MLF-S, or PDIP Package 9
Figure 3. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 20-Pin SOIC, SSOP or PDIP Package 9
Figure 4. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 28-Pin SOIC, SSOP or PDIP Package 9
Figure 5. Power-On Reset Operation 25
Figure 6. Voltage Brown-Out Reset Operation 26
Figure 7. GPIO Port Pin Block Diagram 37
Figure 8. Interrupt Controller Block Diagram 57
Figure 9. Timer Block Diagram 71
Figure 10. UART Block Diagram 100
Figure 11. UART Asynchronous Data Format without Parity 101
Figure 12. UART Asynchronous Data Format with Parity 101
Figure 13. UART Asynchronous MULTIPROCESSOR Mode Data Format 105
Figure 14. UART Driver Enable Signal Timing (shown with 1 Stop Bit and Parity) 107
Figure 15. UART Receiver Interrupt Service Routine Flow 109
Figure 16. Infrared Data Communication System Block Diagram 120
Figure 17. Infrared Data Transmission 121
Figure 18. IrDA Data Reception 122
Figure 19. Analog-to-Digital Converter Block Diagram 125
Figure 20. Comparator Block Diagram 140
Figure 21. Flash Memory Arrangement 147
Figure 22. Flash Controller Operation Flow Chart 148
Figure 23. On-Chip Debugger Block Diagram 180
Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #1 of 2 181

Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2 182
Figure 26. OCD Data Format 183
Figure 27. Recommended 20 MHz Crystal Oscillator Configuration 199
Figure 28. Connecting the On-Chip Oscillator to an External RC Network 201
Figure 29. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45 kΩ Resistor 202
Figure 30. Opcode Map Cell Description 222
Figure 31. First Opcode Map 224
Figure 32. Second Opcode Map after 1FH 225
Figure 33. Typical Active Mode IDD Versus System Clock Frequency 231
Figure 34. Port Input Sample Timing 240
Figure 35. GPIO Port Output Timing 241
Figure 36. On-Chip Debugger Timing 242
Figure 37. UART Timing With CTS 243
Figure 38. UART Timing Without CTS 244

List of Tables

Table 1. Z8 Encore! XP F082A Series Family Part Selection Guide 2
Table 2. Signal Descriptions 10
Table 3. Pin Characteristics (20- and 28-pin Devices) 13
Table 4. Pin Characteristics (8-Pin Devices) 14
Table 5. Z8 Encore! XP F082A Series Program Memory Maps 16
Table 6. Z8 Encore! XP F082A Series Flash Memory Information Area Map 17
Table 7. Register File Address Map 18
Table 8. Reset and Stop Mode Recovery Characteristics and Latency 23
Table 9. Reset Sources and Resulting Reset Type 24
Table 10. Stop Mode Recovery Sources and Resulting Action 28
Table 11. Reset Status Register (RSTSTAT) 30
Table 12. Reset and Stop Mode Recovery Bit Descriptions 31
Table 13. Power Control Register 0 (PWRCTL0) 34
Table 14. Port Availability by Device and Package Type 36
Table 15. Port Alternate Function Mapping (Non 8-Pin Parts) 40
Table 16. Port Alternate Function Mapping (8-Pin Parts) 43
Table 17. GPIO Port Registers and Subregisters 44
Table 18. Port A–D GPIO Address Registers (PxADDR) 45
Table 19. Port A–D GPIO Address Registers by Bit Description 45
Table 20. Port A–D Control Registers (PxCTL) 46
Table 21. Port A–D Data Direction Subregisters (PxDD) 46
Table 22. Port A–D Alternate Function Subregisters (PxAF) 47
Table 23. Port A–D Output Control Subregisters (PxOC) 48
Table 24. Port A–D High Drive Enable Subregisters (PxHDE) 48
Table 25. Port A–D Stop Mode Recovery Source Enable Subregisters (PxSMRE) . . 49
Table 26. Port A–D Pull-Up Enable Subregisters (PxPUE) 50
Table 27. Port A–D Alternate Function Set 2 Subregisters (PxAFS2) 51
Table 28. Port A–D Alternate Function Set 1 Subregisters (PxAFS1) 51

Table 29. Port A–C Input Data Registers (PxIN) 52
Table 30. Port A–D Output Data Register (PxOUT) 52
Table 31. LED Drive Enable (LEDEN) 53
Table 32. LED Drive Level High Register (LEDLVLH) 53
Table 33. LED Drive Level Low Register (LEDLVLL) 54
Table 34. Trap and Interrupt Vectors in Order of Priority 56
Table 35. Interrupt Request 0 Register (IRQ0) 60
Table 36. Interrupt Request 1 Register (IRQ1) 61
Table 37. Interrupt Request 2 Register (IRQ2) 62
Table 38. IRQ0 Enable and Priority Encoding 62
Table 39. IRQ0 Enable High Bit Register (IRQ0ENH) 63
Table 40. IRQ0 Enable Low Bit Register (IRQ0ENL) 63
Table 41. IRQ1 Enable and Priority Encoding 64
Table 42. IRQ1 Enable Low Bit Register (IRQ1ENL) 65
Table 43. IRQ1 Enable High Bit Register (IRQ1ENH) 65
Table 44. IRQ2 Enable and Priority Encoding 66
Table 45. IRQ2 Enable High Bit Register (IRQ2ENH) 66
Table 46. Interrupt Edge Select Register (IRQES) 67
Table 47. IRQ2 Enable Low Bit Register (IRQ2ENL) 67
Table 48. Shared Interrupt Select Register (IRQSS) 68
Table 49. Interrupt Control Register (IRQCTL) 69
Table 50. Timer 0–1 Control Register 0 (TxCTL0) 85
Table 51. Timer 0–1 Control Register 1 (TxCTL1) 86
Table 52. Timer 0–1 High Byte Register (TxH) 90
Table 53. Timer 0–1 Low Byte Register (TxL) 90
Table 54. Timer 0–1 Reload High Byte Register (TxRH) 91
Table 55. Timer 0–1 Reload Low Byte Register (TxRL) 91
Table 56. Timer 0–1 PWM High Byte Register (TxPWMH) 92
Table 57. Timer 0–1 PWM Low Byte Register (TxPWML) 92
Table 58. Watchdog Timer Approximate Time-Out Delays 93

Table 59. Watchdog Timer Control Register (WDTCTL) 96
Table 60. Watchdog Timer Reload Upper Byte Register (WDTU) 97
Table 61. Watchdog Timer Reload High Byte Register (WDTH) 97
Table 62. Watchdog Timer Reload Low Byte Register (WDTL) 98
Table 63. UART Control 0 Register (U0CTL0) 111
Table 64. UART Control 1 Register (U0CTL1) 112
Table 65. UART Status 0 Register (U0STAT0) 114
Table 66. UART Status 1 Register (U0STAT1) 115
Table 67. UART Transmit Data Register (U0TXD) 116
Table 68. UART Receive Data Register (U0RXD) 116
Table 69. UART Address Compare Register (U0ADDR) 117
Table 70. UART Baud Rate High Byte Register (U0BRH) 117
Table 71. UART Baud Rate Low Byte Register (U0BRL) 117
Table 72. UART Baud Rates 118
Table 73. ADC Control Register 0 (ADCCTL0) 134
Table 74. ADC Control/Status Register 1 (ADCCTL1) 136
Table 75. ADC Data High Byte Register (ADCD_H) 137
Table 76. ADC Data Low Byte Register (ADCD_L) 137
Table 77. Comparator Control Register (CMP0) 141
Table 78. Z8 Encore! XP F082A Series Flash Memory Configurations 146
Table 79. Flash Code Protection Using the Flash Option Bits 150
Table 80. Flash Status Register (FSTAT) 155
Table 81. Flash Control Register (FCTL) 155
Table 82. Flash Page Select Register (FPS) 156
Table 83. Flash Sector Protect Register (FPROT) 157
Table 84. Flash Frequency High Byte Register (FFREQH) 158
Table 85. Flash Frequency Low Byte Register (FFREQL) 158
Table 86. Trim Bit Address Register (TRMADR) 161
Table 87. Trim Bit Data Register (TRMDR) 162
Table 88. Flash Option Bits at Program Memory Address 0000H 162

Table 89. Flash Options Bits at Program Memory Address 0001H 164
Table 90. Trim Options Bits at Address 0000H 165
Table 91. Trim Option Bits at 0001H 165
Table 92. Trim Option Bits at 0002H (TIPO) 166
Table 93. Trim Option Bits at Address 0003H (TLVD) 166
Table 94. LVD Trim Values 167
Table 95. Trim Option Bits at 0004H 168
Table 96. ADC Calibration Bits 169
Table 97. ADC Calibration Data Location 169
Table 98. Temperature Sensor Calibration High Byte at 003A (TSCALH) 171
Table 99. Temperature Sensor Calibration Low Byte at 003B (TSCALL) 171
Table 100. Watchdog Calibration High Byte at 007EH (WDTCALH) 172
Table 101. Serial Number at 001C - 001F (S_NUM) 173
Table 102. Serialization Data Locations 173
Table 103. Watchdog Calibration Low Byte at 007FH (WDTCALL) 173
Table 104. Lot Identification Number (RAND_LOT) 174
Table 105. Randomized Lot ID Locations 174
Table 106. Write Status Byte 177
Table 107. NVDS Read Time 179
Table 108. OCD Baud-Rate Limits 184
Table 109. Debug Command Enable/Disable 186
Table 110. OCD Control Register (OCDCTL) 191
Table 111. OCD Status Register (OCDSTAT) 192
Table 112. Oscillator Configuration and Selection 194
Table 113. Oscillator Control Register (OSCCTL) 196
Table 114. Recommended Crystal Oscillator Specifications 200
Table 115. Transconductance Values for Low, Medium and High Gain Operating Modes 200
Table 116. Assembly Language Syntax Example 1 205
Table 117. Assembly Language Syntax Example 2 205
Table 118. Notational Shorthand 206

Table 119. Additional Symbols 207
Table 120. Arithmetic Instructions 208
Table 121. Bit Manipulation Instructions 209
Table 122. Block Transfer Instructions 209
Table 123. CPU Control Instructions 209
Table 124. Load Instructions 210
Table 125. Logical Instructions 210
Table 126. Program Control Instructions 211
Table 127. Rotate and Shift Instructions 211
Table 128. eZ8 CPU Instruction Summary 212
Table 129. Opcode Map Abbreviations 223
Table 130. Absolute Maximum Ratings 226
Table 131. DC Characteristics 227
Table 132. Power Consumption 229
Table 133. AC Characteristics 232
Table 134. Internal Precision Oscillator Electrical Characteristics 232
Table 135. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing 233
Table 136. Flash Memory Electrical Characteristics and Timing 234
Table 137. Watchdog Timer Electrical Characteristics and Timing 235
Table 138. Non-Volatile Data Storage 235
Table 139. Analog-to-Digital Converter Electrical Characteristics and Timing 236
Table 140. Low Power Operational Amplifier Electrical Characteristics 238
Table 141. Comparator Electrical Characteristics 238
Table 142. Temperature Sensor Electrical Characteristics 239
Table 143. GPIO Port Input Timing 240
Table 144. GPIO Port Output Timing 241
Table 145. On-Chip Debugger Timing 242
Table 146. UART Timing With CTS 243
Table 147. UART Timing Without CTS 244
Table 148. Z8 Encore! XP F082A Series Ordering Matrix 246

Overview

Zilog's Z8 Encore! MCU family of products are the first in a line of Zilog microcontroller products based upon the 8-bit eZ8 CPU. Zilog's Z8 Encore! XP F082A 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 the Z8 Encore! XP F082A 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 F082A Series products include:

- 20MHz eZ8 CPU
- 1KB, 2KB, 4KB, or 8KB Flash memory with in-circuit programming capability
- 256B, 512B, or 1KB register RAM
- Up to 128B nonvolatile data storage (NVDS)
- Internal precision oscillator trimmed to ±1% accuracy
- External crystal oscillator, operating up to 20MHz
- Optional 8-channel, 10-bit analog-to-digital converter (ADC)
- Optional on-chip temperature sensor
- On-chip analog comparator
- Optional on-chip low-power operational amplifier (LPO)
- Full-duplex UART
- The 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 the UART
- Two enhanced 16-bit timers with capture, compare and PWM capability
- Watchdog Timer (WDT) with dedicated internal RC oscillator
- Up to 20 vectored interrupts
- 6 to 25 I/O pins depending upon package
- Up to thirteen 5 V-tolerant input pins

- Up to 8 ports capable of direct LED drive with no current limit resistor required
- On-Chip Debugger (OCD)
- Voltage Brown-Out (VBO) protection
- Programmable low battery detection (LVD) (8-pin devices only)
- Bandgap generated precision voltage references available for the ADC, comparator, VBO and LVD
- Power-On Reset (POR)
- 2.7V to 3.6V operating voltage
- 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 identifies the basic features and package styles available for each device within the Z8 Encore! XP F082A Series product line.

Table 1. Z8 Encore! XP F082A Series Family Part Selection Guide

Part Number	Flash (KB)	RAM (B)	NVDS[1] (B)	I/O	Comparator	Advanced Analog[2]	ADC Inputs	Packages
Z8F082A	8	1024	0	6–23	Yes	Yes	4–8	8-, 20- and 28-pin
Z8F081A	8	1024	0	6–25	Yes	No	0	8-, 20- and 28-pin
Z8F042A	4	1024	128	6–23	Yes	Yes	4–8	8-, 20- and 28-pin
Z8F041A	4	1024	128	6–25	Yes	No	0	8-, 20- and 28-pin
Z8F022A	2	512	64	6–23	Yes	Yes	4–8	8-, 20- and 28-pin
Z8F021A	2	512	64	6–25	Yes	No	0	8-, 20- and 28-pin
Z8F012A	1	256	16	6–23	Yes	Yes	4–8	8-, 20- and 28-pin
Z8F011A	1	256	16	6–25	Yes	No	0	8-, 20- and 28-pin

Notes:
1. Non-volatile data storage.
2. Advanced Analog includes ADC, temperature sensor and low-power operational amplifier.

Block Diagram

Figure 1 displays the block diagram of the architecture of the Z8 Encore! XP F082A Series devices.

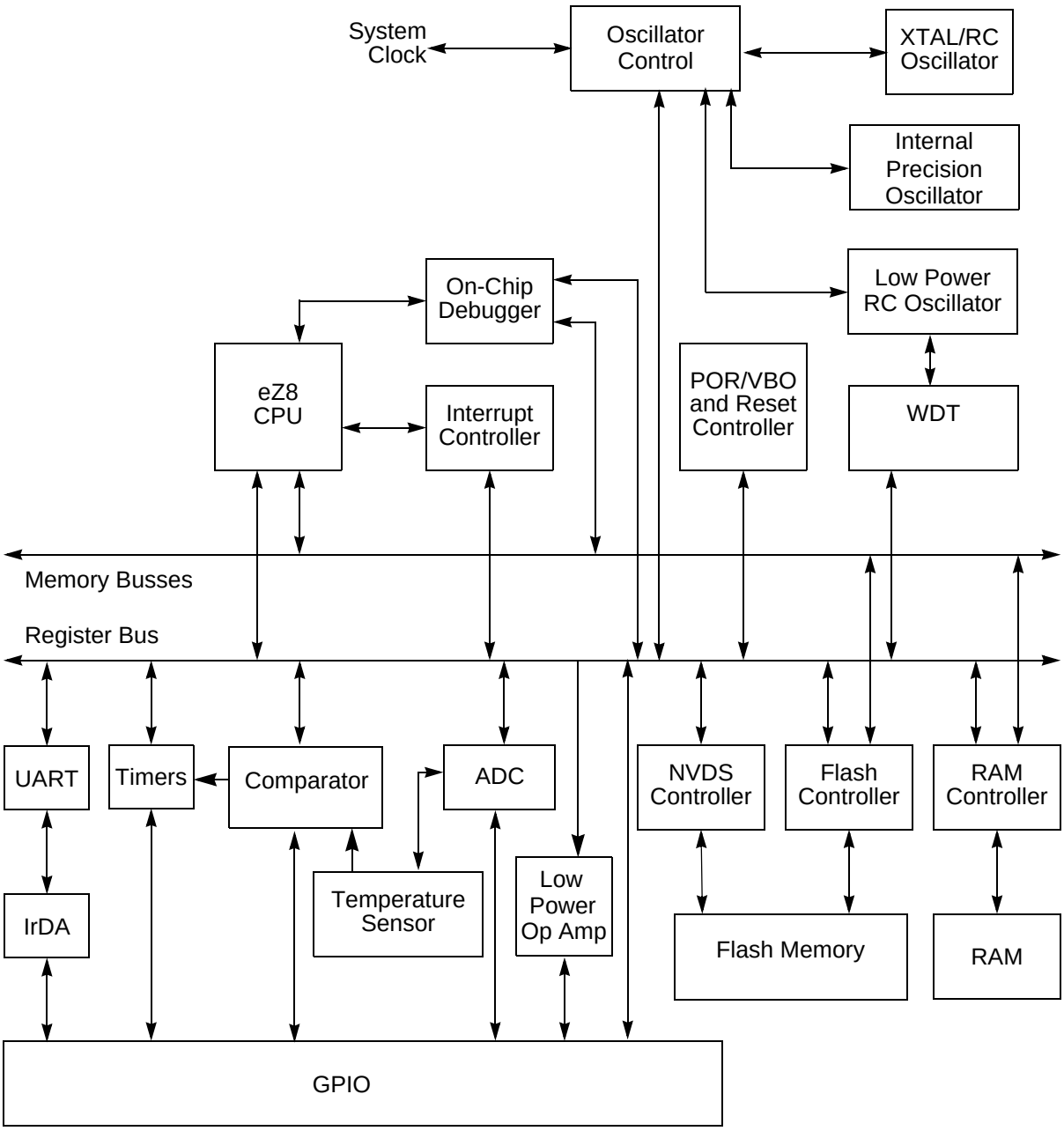


Figure 1. Z8 Encore! XP F082A Series Block Diagram

CPU and Peripheral Overview

The eZ8 CPU, Zilog's latest 8-bit Central Processing Unit (CPU), meets the continuing demand for faster and more code-efficient microcontrollers. The eZ8 CPU executes a superset of the original Z8 instruction set. The features of eZ8 CPU 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 eZ8 CPU, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com.

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. The ADC also features a unity gain buffer when high input impedance is required.

Low-Power Operational Amplifier

The optional low-power operational amplifier (LPO) is a general-purpose amplifier primarily targeted for current sense applications. The LPO output may be routed internally to the ADC or externally to a pin.

Internal Precision Oscillator

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

Temperature Sensor

The optional temperature sensor produces an analog output proportional to the device temperature. This signal can be sent to either the ADC or the analog comparator.

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.

External Crystal Oscillator

The crystal oscillator circuit provides highly accurate clock frequencies with the use of an external crystal, ceramic resonator or RC network.

Low Voltage Detector

The low voltage detector (LVD) is able to generate an interrupt when the supply voltage drops below a user-programmable level. The LVD is available on 8-pin devices only.

On-Chip Debugger

The Z8 Encore! XP F082A Series products feature an integrated on-chip debugger (OCD) accessed via a single-pin interface. The OCD provides a rich-set of debugging capabilities, such as reading and writing registers, programming Flash memory, setting breakpoints and executing code.

Universal Asynchronous Receiver/Transmitter

The full-duplex universal asynchronous receiver/transmitter (UART) is included in all Z8 Encore! XP package types. 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 (BRG) 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.

General-Purpose Input/Output

The Product Line MCUs feature 6 to 25 port pins (Ports A–D) for general- purpose input/output (GPIO). The number of GPIO pins available is a function of package and each pin is individually programmable. 5 V tolerant input pins are available on all I/Os on 8-pin devices and most I/Os on other package types.

Direct LED Drive

The 20- and 28-pin devices support controlled current sinking output pins capable of driving LEDs without the need for a current limiting resistor. These LED drivers are independently programmable to four different intensity levels.

Flash Controller

The Flash Controller programs and erases Flash memory. The Flash Controller supports several protection mechanisms against accidental program and erasure, plus factory serialization and read protection.

Non-Volatile Data Storage

The nonvolatile data storage (NVDS) uses a hybrid hardware/software scheme to implement a byte programmable data memory and is capable of over 100,000 write cycles.

Note: Devices with 8KB of Flash memory do not include the NVDS feature.

Interrupt Controller

The Z8 Encore! XP F082A Series products support up to 20 interrupts. These interrupts consist of 8 internal peripheral interrupts and 12 general-purpose I/O pin interrupt sources. The interrupts have three levels of programmable interrupt priority.

Reset Controller

The Z8 Encore! XP F082A Series products can be reset using the $\overline{\text{RESET}}$ pin, Power-On Reset, Watchdog Timer (WDT) time-out, Stop Mode exit, or Voltage Brown-Out (VBO) warning signal. The $\overline{\text{RESET}}$ pin is bidirectional, that is, it functions as reset source and as a reset indicator.

Pin Description

The Z8 Encore! XP F082A Series products are available in a variety of packages styles and pin configurations. This chapter describes the signals and available pin configurations for each of the package styles. For information about physical package specifications, see the Packaging chapter on page 245.

Available Packages

The following package styles are available for each device in the Z8 Encore! XP F082A Series product line:

- SOIC: 8-, 20- and 28-pin
- PDIP: 8-, 20- and 28-pin
- SSOP: 20- and 28- pin
- QFN 8-pin (MLF-S, a QFN-style package with an 8-pin SOIC footprint)

In addition, the Z8 Encore! XP F082A Series devices are available both with and without advanced analog capability (ADC, temperature sensor and op amp). Devices Z8F082A, Z8F042A, Z8F022A and Z8F012A contain the advanced analog, while devices Z8F081A, Z8F041A, Z8F021A and Z8F011A do not have the advanced analog capability.

Pin Configurations

Figure 2 through Figure 4 display the pin configurations for all the packages available in the Z8 Encore! XP F082A Series. See Table 2 on page 10 for a description of the signals. The analog input alternate functions (ANA*x*) are not available on the Z8F081A, Z8F041A, Z8F021A and Z8F011A 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 Port A, B and C pins 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. At powerup, the PD0 pin defaults to the $\overline{\text{RESET}}$ alternate function.

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

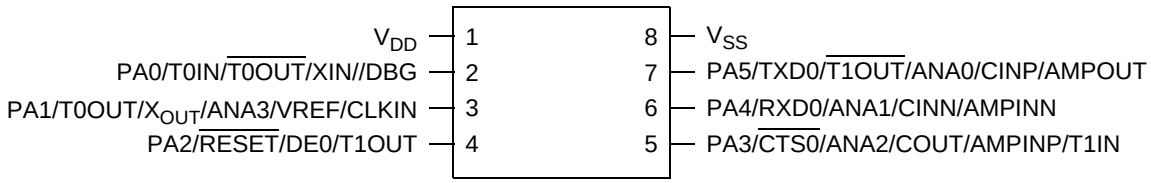


Figure 2. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 8-Pin SOIC, QFN/MLF-S, or PDIP Package

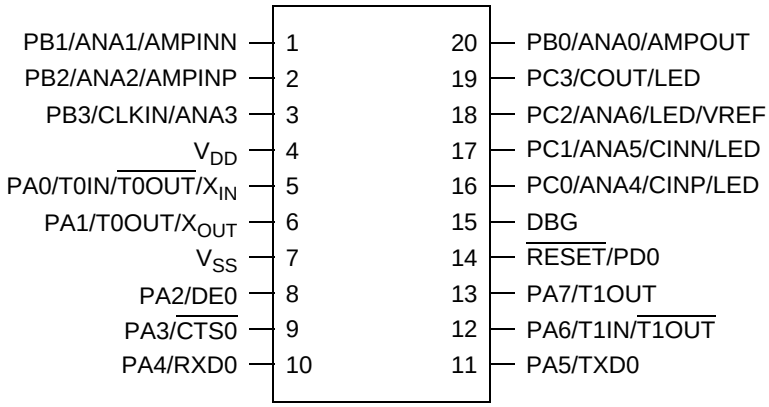


Figure 3. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 20-Pin SOIC, SSOP or PDIP Package

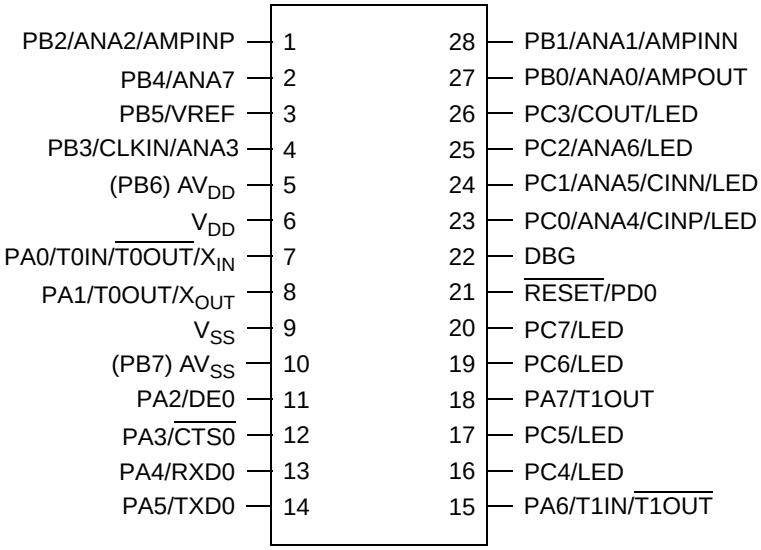


Figure 4. Z8F08xA, Z8F04xA, Z8F02xA and Z8F01xA in 28-Pin SOIC, SSOP or PDIP Package

Signal Descriptions

Table 2 describes the Z8 Encore! XP F082A Series signals. See the Pin Configurations section on page 8 to determine the signals available for the specific package styles.

Table 2. 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]	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.
PD[0]	I/O	Port D. This pin is used for general-purpose output only.
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 may be used to ensure the external RS-485 driver is enabled when data is transmitted by the UART.
Timers		
T0OUT/T1OUT	O	Timer Output 0–1. These signals are outputs from the timers.
$\overline{\text{T0OUT}}$/$\overline{\text{T1OUT}}$	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.
Comparator		
CINP/CINN	I	Comparator Inputs. These signals are the positive and negative inputs to the comparator.
COUT	O	Comparator Output.

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 2. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
Analog		
ANA[7:0]	I	Analog Port. These signals are used as inputs to the analog-to-digital converter (ADC).
V_{REF}	I/O	Analog-to-digital converter reference voltage input, or buffered output for internal reference.
Low-Power Operational Amplifier (LPO)		
AMPINP/AMPINN	I	LPO inputs. If enabled, these pins drive the positive and negative amplifier inputs respectively.
AMPOUT	O	LPO output. If enabled, this pin is driven by the on-chip LPO.
Oscillators		
X_{IN}	I	External Crystal Input. This is the input pin to the crystal oscillator. A crystal can be connected between it and the X_{OUT} pin to form the oscillator. In addition, this pin is used with external RC networks or external clock drivers to provide the system clock.
X_{OUT}	O	External Crystal Output. This pin is the output of the crystal oscillator. A crystal can be connected between it and the X_{IN} pin to form the oscillator.
Clock Input		
CLKIN	I	Clock Input Signal. This pin may 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 On-Chip Debugger. **Caution:** The DBG pin is open-drain and requires a pull-up resistor to ensure proper operation.

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 2. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
Reset		
$\overline{\text{RESET}}$	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}	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 3 describes the characteristics for each pin available on the Z8 Encore! XP F082A Series 20- and 28-pin devices. Data in Table 3 is sorted alphabetically by the pin symbol mnemonic.

Table 4 on page 14 provides detailed information about the characteristics for each pin available on the Z8 Encore! XP F082A Series 8-pin devices.

Note: All six I/O pins on the 8-pin packages are 5 V-tolerant (unless the pull-up devices are enabled). The column in Table 3 below describes 5 V-tolerance for the 20- and 28-pin packages only.

Table 3. 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	5 V Tolerance
AV_{DD}	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	Yes	Yes	Yes	No
PA[7:0]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	PA[7:2] unless pullups enabled
PB[7:0]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	PB[7:6] unless pullups enabled
PC[7:0]	I/O	I	N/A	Yes	Programmable Pull-up	Yes	Yes, Programmable	PC[7:3] unless pullups enabled
RESET/ PD0	I/O	I/O (defaults to RESET)	Low (in Reset mode)	Yes (PD0 only)	Programmable for PD0; always on for RESET	Yes	Programmable for PD0; always on for RESET	Yes, unless pullups enabled
V_{DD}	N/A	N/A	N/A	N/A			N/A	N/A
V_{SS}	N/A	N/A	N/A	N/A			N/A	N/A

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

Table 4. 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
$\overline{\text{RESET}}$/ PA2	I/O	I/O (defaults to $\overline{\text{RESET}}$)	Low (in Reset mode)	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
V_{DD}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
V_{SS}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Address Space

The eZ8 CPU can access the following 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 information about eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), which is available for download on www.zilog.com.

Register File

The Register File address space in the Z8 Encore! MCU is 4 KB (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 4 KB 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. The Z8 Encore! XP™ F082A Series devices contain 256 B to 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 64 KB of Program Memory address space. The Z8 Encore! XP F082A Series devices contain 1 KB to 8KB of on-chip Flash memory in the Program Memory address space, depending on the device. Reading from Program Memory

addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 5 describes the Program Memory Maps for the Z8 Encore! XP F082A Series products.

Table 5. Z8 Encore! XP F082A Series Program Memory Maps

Program Memory Address (Hex)	Function
Z8F082A and Z8F081A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–0039	Reserved
003A–003D	Oscillator Fail Trap Vectors
003E–1FFF	Program Memory
Z8F042A and Z8F041A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–0039	Reserved
003A–003D	Oscillator Fail Trap Vectors
003E–0FFF	Program Memory
Z8F022A and Z8F021A Products	
0000–0001	Flash Option Bits
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–0039	Reserved
003A–003D	Oscillator Fail Trap Vectors
003E–07FF	Program Memory
Z8F012A and Z8F011A Products	
0000–0001	Flash Option Bits

Note: *See Table 32 on page 56 for a list of the interrupt vectors.

Table 5. Z8 Encore! XP F082A Series Program Memory Maps (Continued)

Program Memory Address (Hex)	Function
0002–0003	Reset Vector
0004–0005	WDT Interrupt Vector
0006–0007	Illegal Instruction Trap
0008–0037	Interrupt Vectors*
0038–0039	Reserved
003A–003D	Oscillator Fail Trap Vectors
003E–03FF	Program Memory

Note: *See Table 32 on page 56 for a list of the interrupt vectors.

Data Memory

The Z8 Encore! XP F082A Series does not use the eZ8 CPU's 64 KB Data Memory address space.

Flash Information Area

Table 6 describes the Z8 Encore! XP F082A 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 6. Z8 Encore! XP F082A Series Flash Memory Information Area Map

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

Register Map

Table 7 provides the address map for the Register File of the Z8 Encore! XP F082A Series devices. Not all devices and package styles in the Z8 Encore! XP F082A Series support the ADC, or all of the GPIO Ports. Consider registers for unimplemented peripherals as Reserved.

Table 7. Register File Address Map

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
General-Purpose RAM				
Z8F082A/Z8F081A Devices				
000–3FF	General-Purpose Register File RAM	—	XX	
400–EFF	Reserved	—	XX	
Z8F042A/Z8F041A Devices				
000–3FF	General-Purpose Register File RAM	—	XX	
400–EFF	Reserved	—	XX	
Z8F022A/Z8F021A Devices				
000–1FF	General-Purpose Register File RAM	—	XX	
200–EFF	Reserved	—	XX	
Z8F012A/Z8F011A Devices				
000–0FF	General-Purpose Register File RAM	—	XX	
100–EFF	Reserved	—	XX	
Timer 0				
F00	Timer 0 High Byte	T0H	00	89
F01	Timer 0 Low Byte	T0L	01	89
F02	Timer 0 Reload High Byte	T0RH	FF	90
F03	Timer 0 Reload Low Byte	T0RL	FF	90
F04	Timer 0 PWM High Byte	T0PWMH	00	91
F05	Timer 0 PWM Low Byte	T0PWML	00	91
F06	Timer 0 Control 0	T0CTL0	00	85
F07	Timer 0 Control 1	T0CTL1	00	86

Notes:
1. XX = Undefined.
2. Refer to the eZ8 CPU Core User Manual (UM0128).

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
Timer 1				
F08	Timer 1 High Byte	T1H	00	89
F09	Timer 1 Low Byte	T1L	01	89
F0A	Timer 1 Reload High Byte	T1RH	FF	90
Timer 1 (cont'd)				
F0B	Timer 1 Reload Low Byte	T1RL	FF	90
F0C	Timer 1 PWM High Byte	T1PWMH	00	91
F0D	Timer 1 PWM Low Byte	T1PWML	00	91
F0E	Timer 1 Control 0	T1CTL0	00	85
F0F	Timer 1 Control 1	T1CTL1	00	86
F10–F6F	Reserved	—	XX	
UART				
F40	UART Transmit/Receive Data registers	TXD, RXD	XX	115
F41	UART Status 0 Register	U0STAT0	00	114
F42	UART Control 0 Register	U0CTL0	00	110
F43	UART Control 1 Register	U0CTL1	00	110
F44	UART Status 1 Register	U0STAT1	00	115
F45	UART Address Compare Register	U0ADDR	00	116
F46	UART Baud Rate High Byte Register	U0BRH	FF	117
F47	UART Baud Rate Low Byte Register	U0BRL	FF	117
Analog-to-Digital Converter (ADC)				
F70	ADC Control 0	ADCCTL0	00	134
F71	ADC Control 1	ADCCTL1	80	136
F72	ADC Data High Byte	ADCD_H	XX	137
F73	ADC Data Low Byte	ADCD_L	XX	137
F74–F7F	Reserved	—	XX	
Low Power Control				
F80	Power Control 0	PWRCTL0	80	34
F81	Reserved	—	XX	
LED Controller				
F82	LED Drive Enable	LEDEN	00	53
F83	LED Drive Level High Byte	LEDLVLH	00	53
F84	LED Drive Level Low Byte	LEDLVLL	00	54

Notes:
1. XX = Undefined.
2. Refer to the eZ8 CPU Core User Manual (UM0128).

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
F85	Reserved	—	XX	
Oscillator Control				
F86	Oscillator Control	OSCCTL	A0	196
F87–F8F	Reserved	—	XX	
Comparator 0				
F90	Comparator 0 Control	CMP0	14	141
F91–FBF	Reserved	—	XX	
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	60
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	63
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	63
FC3	Interrupt Request 1	IRQ1	00	61
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	65
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	65
FC6	Interrupt Request 2	IRQ2	00	62
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	66
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	67
FC9–FCC	Reserved	—	XX	
FCD	Interrupt Edge Select	IRQES	00	68
FCE	Shared Interrupt Select	IRQSS	00	68
FCF	Interrupt Control	IRQCTL	00	69
GPIO Port A				
FD0	Port A Address	PAADDR	00	44
FD1	Port A Control	PACTL	00	46
FD2	Port A Input Data	PAIN	XX	46
FD3	Port A Output Data	PAOUT	00	46
GPIO Port B				
FD4	Port B Address	PBADDR	00	44
FD5	Port B Control	PBCTL	00	46
FD6	Port B Input Data	PBIN	XX	46
FD7	Port B Output Data	PBOUT	00	46
GPIO Port C				
FD8	Port C Address	PCADDR	00	44

Notes:
1. XX = Undefined.
2. Refer to the eZ8 CPU Core User Manual (UM0128).

Table 7. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page
FD9	Port C Control	PCCTL	00	46
FDA	Port C Input Data	PCIN	XX	46
FDB	Port C Output Data	PCOUT	00	46
GPIO Port D				
FDC	Port D Address	PDADDR	00	44
FDD	Port D Control	PDCTL	00	46
FDE	Reserved	—	XX	
FDF	Port D Output Data	PDOUT	00	46
FE0–FEF	Reserved	—	XX	
Watchdog Timer (WDT)				
FF0	Reset Status (Read-only)	RSTSTAT	X0	29
	Watchdog Timer Control (Write-only)	WDTCTL	N/A	96
FF1	Watchdog Timer Reload Upper Byte	WDTU	00	97
FF2	Watchdog Timer Reload High Byte	WDTH	04	97
FF3	Watchdog Timer Reload Low Byte	WDTL	00	98
FF4–FF5	Reserved	—	XX	
Trim Bit Control				
FF6	Trim Bit Address	TRMADR	00	161
FF7	Trim Bit Data	TRMDR	00	162
Flash Memory Controller				
FF8	Flash Control	FCTL	00	155
FF8	Flash Status	FSTAT	00	155
FF9	Flash Page Select	FPS	00	156
	Flash Sector Protect	FPROT	00	157
FFA	Flash Programming Frequency High Byte	FFREQH	00	158
FFB	Flash Programming Frequency Low Byte	FFREQL	00	158
eZ8 CPU				
FFC	Flags	—	XX	See footnote 2.
FFD	Register Pointer	RP	XX	
FFE	Stack Pointer High Byte	SPH	XX	
FFF	Stack Pointer Low Byte	SPL	XX	

Notes:
1. XX = Undefined.
2. Refer to the eZ8 CPU Core User Manual (UM0128).

Reset, Stop Mode Recovery and Low Voltage Detection

The Reset Controller within the Z8 Encore! XP F082A 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 occurrences:

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

The low voltage detection circuitry on the device (available on the 8-pin product versions only) performs the following functions:

- Generates the VBO reset when the supply voltage drops below a minimum safe level.
- Generates an interrupt when the supply voltage drops below a user-defined level (8-pin devices only).

Reset Types

The Z8 Encore! XP F082A Series provides several different types of Reset operation. Stop Mode Recovery is considered as a form of Reset. Table 8 lists the types of Reset and their operating characteristics. The System Reset is longer if the external crystal oscillator is enabled by the Flash option bits, allowing additional time for oscillator start-up.

Table 8. 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
System Reset with Crystal Oscillator Enabled	Reset (as applicable)	Reset	5000 Internal Precision Oscillator Cycles
Stop Mode Recovery	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	66 Internal Precision Oscillator Cycles + IPO startup time
Stop Mode Recovery with Crystal Oscillator Enabled	Unaffected, except WDT_CTL and OSC_CTL registers	Reset	5000 Internal Precision Oscillator Cycles

During a System Reset or Stop Mode Recovery, the Internal Precision Oscillator requires 4 µs to start up. Then the Z8 Encore! XP F082A 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 (POR), 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, except PD0 (or PA2 on 8-pin devices) which is shared with the reset pin. On reset, the PD0 is configured as a bidirectional open-drain reset. The pin is internally driven low during port reset, after which the user code may reconfigure this pin as a general purpose output.

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.

As the control registers are reinitialized 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 9 lists the possible sources of a system reset.

Table 9. 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.
	$\overline{\text{RESET}}$ pin assertion	All reset pulses less than three system clocks in width are ignored.
	On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)	System Reset, except the On-Chip Debugger is unaffected by the reset.
Stop Mode	Power-On Reset/Voltage Brown-Out	Reset delay begins after supply voltage exceeds POR level.
	$\overline{\text{RESET}}$ pin assertion	All reset pulses less than the specified analog delay are ignored. See Table 131 on page 229.
	DBG pin driven Low	None.

Power-On Reset

Z8 Encore! XP F082A Series devices contain an internal Power-On Reset 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 F082A Series device exits the Power-On Reset state, the eZ8 CPU fetches the Reset vector. Following Power-On Reset, the POR status bit in the Reset Status (RSTSTAT) Register is set to 1.

Figure 5 displays Power-On Reset operation. See Electrical Characteristics on page 221 for the POR threshold voltage (V_{POR}).

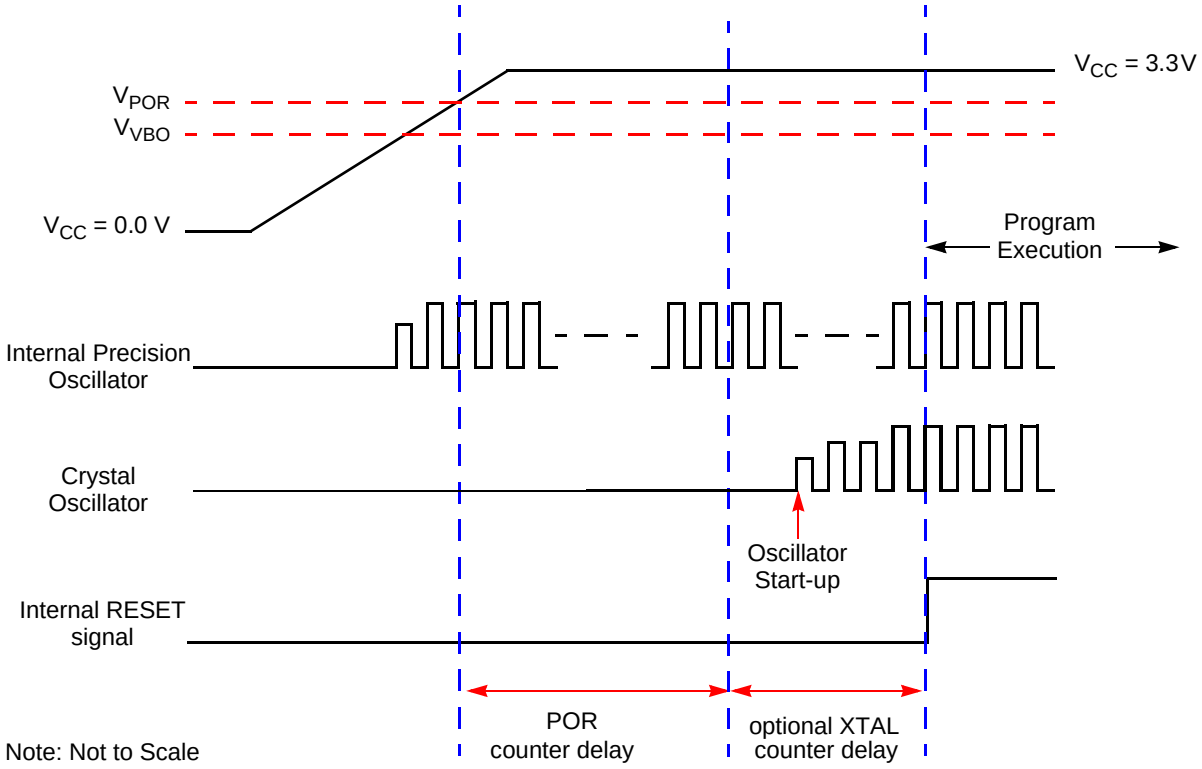


Figure 5. Power-On Reset Operation

Voltage Brown-Out Reset

The devices in the Z8 Encore! XP F082A Series provide low Voltage Brown-Out (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 Power-On Reset 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. Following Power-On Reset, the POR status bit in the Reset Status (RSTSTAT) Register is set to 1. Figure 6 displays Voltage Brown-Out operation. See the Electrical Characteristics chapter on page 226 for the VBO and POR threshold voltages (V_{VBO} and V_{POR}).

The Voltage Brown-Out circuit can be either enabled or disabled during Stop Mode. Operation during Stop Mode is set by the VBO_AO Flash option bit. See the Flash Option Bits chapter on page 159 for information about configuring VBO_AO.

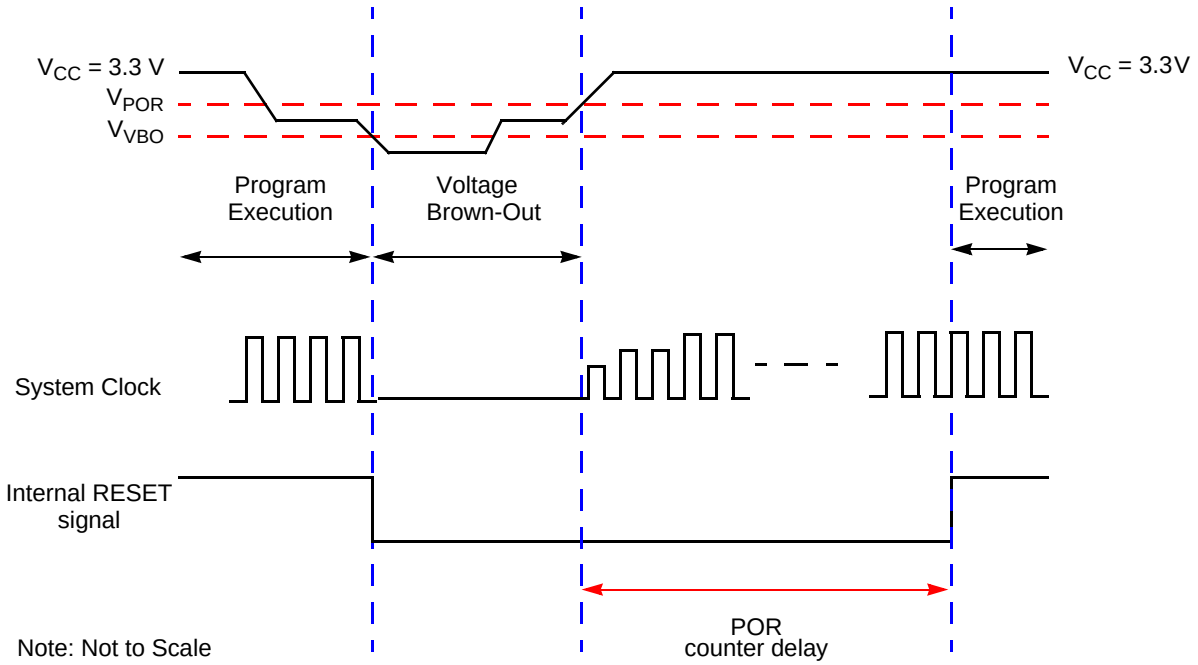


Figure 6. Voltage Brown-Out Reset Operation

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

Watchdog Timer Reset

If the device is operating in Normal or Halt Mode, the Watchdog Timer can initiate a System Reset at time-out if the WDT_RES Flash option bit is programmed to 1, i.e., 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 bit in the Reset Status (RSTSTAT) 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 may 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 F082A 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 Reset Status (RSTSTAT) Register is set to 1.

External Reset Indicator

During System Reset or when enabled by the GPIO logic (see Table 20 on page 46), 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 a Z8 Encore! XP F082A 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 8 has elapsed.

On-Chip Debugger Initiated Reset

A Power-On Reset can be initiated using the On-Chip Debugger by setting the RST bit in the OCD Control Register. The On-Chip Debugger 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

Stop Mode is entered by execution of a Stop instruction by the eZ8 CPU. See the Low-Power Modes chapter on page 32 for detailed Stop Mode information. During Stop Mode Recovery (SMR), 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 (see Table 135 on page 233) T_{SMR}, also includes 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, 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 vec-

tor address. Following Stop Mode Recovery, the Stop bit in the Reset Status (RSTSTAT) Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions. The text following provides more detailed information about each of the Stop Mode Recovery sources.

Table 10. 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 $\overline{\text{RESET}}$ 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 Reset Status (RSTSTAT) Register, the WDT and Stop bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and the Z8 Encore! XP F082A 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 may 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: SMR pulses shorter than specified do not trigger a recovery (see Table 135 on page 233). In this instance, 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 the Z8 Encore! XP F082A 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. See the Electrical Characteristics chapter on page 226 for details.

Low Voltage Detection

In addition to the Voltage Brown-Out (VBO) Reset described above, it is also possible to generate an interrupt when the supply voltage drops below a user-selected value. For details about configuring the Low Voltage Detection (LVD) and the threshold levels available, see the Trim Option Bits at Address 0003H (TLVD) Register on page 166. The LVD function is available on the 8-pin product versions only.

When the supply voltage drops below the LVD threshold, the LVD bit of the Reset Status (RSTSTAT) Register is set to one. This bit remains one until the low-voltage condition goes away. Reading or writing this bit does not clear it. The LVD circuit can also generate an interrupt when so enabled, see the GPIO Mode Interrupt Controller chapter on page 55. The LVD bit is not latched; therefore, enabling the interrupt is the only way to guarantee detection of a transient low voltage event.

The LVD functionality depends on circuitry shared with the VBO block; therefore, disabling the VBO also disables the LVD.

Reset Register Definitions

The following sections define the Reset registers.

Reset Status Register

The read-only Reset Status (RSTSTAT) Register, shown in Table 11, 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 write-only Watchdog Timer Control Register.

Table 12 lists the bit settings for Reset and Stop Mode Recovery events.

Table 11. Reset Status Register (RSTSTAT)

Bit	7	6	5	4	3	2	1	0
Field	POR	STOP	WDT	EXT	Reserved			LVD
RESET	See descriptions below			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 occurs. 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.
[6] STOP	**Stop Mode Recovery Indicator** If this bit is set to 1, a Stop Mode Recovery occurs. If the Stop and WDT bits are both set to 1, the Stop Mode Recovery occurs 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 Power-On Reset or a WDT time-out that occurred while not in Stop Mode. Reading this register also resets this bit.
[5] WDT	**Watchdog Timer Time-Out Indicator** If this bit is set to 1, a WDT time-out occurs. 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.
[4] EXT	**External Reset Indicator** If this bit is set to 1, a Reset initiated by the external $\overline{\text{RESET}}$ pin occurs. 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.
[3:1]	**Reserved** These bits are reserved and must be programmed to 000.
[0] LVD	**Low Voltage Detection Indicator** If this bit is set to 1 the current state of the supply voltage is below the low voltage detection threshold. This value is not latched but is a real-time indicator of the supply voltage level.

Table 12. Reset and Stop Mode Recovery Bit Descriptions

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 Watchdog Timer time-out	0	0	1	0
Reset using the On-Chip Debugger (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 Watchdog Timer time-out	0	1	1	0

Low-Power Modes

The Z8 Encore! XP F082A 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, the Low-power Operational Amplifier and the Watchdog Timer. These three blocks may also be disabled for additional power savings. Specifically, 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
- Low-power operational amplifier continues to operate if enabled by the Power Control Register
- 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, Stop Mode Recovery and Low Voltage Detection chapter on page 22.

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 (PC) 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, if enabled

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 F082A Series devices. Disabling a given peripheral minimizes its power consumption.

Power Control Register Definitions

The following sections define 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. The default state of the low-power

operational amplifier (LPO) is OFF. To use the LPO, clear the LPO bit, turning it ON. Clearing this bit might interfere with normal ADC measurements on ANA0 (the LPO output). This bit enables the amplifier even in Stop Mode. If the amplifier is not required in Stop Mode, disable it. Failure to perform this results in Stop Mode currents greater than specified.

Note: This register is only reset during a POR sequence. Other system reset events do not affect it.

Table 13. Power Control Register 0 (PWRCTL0)

Bit	7	6	5	4	3	2	1	0
Field	LPO	Reserved		VBO	TEMP	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] LPO	**Low-Power Operational Amplifier Disable** 0 = LPO is enabled (this applies even in Stop Mode). 1 = LPO is disabled.
[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] TEMP	**Temperature Sensor Disable** 0 = Temperature Sensor enabled. 1 = Temperature Sensor disabled.
[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.

Note: Asserting any power control bit disables the targeted block regardless of any enable bits contained in the target block's control registers.

General-Purpose Input/Output

The Z8 Encore! XP F082A Series products support a maximum of 25 port pins (Ports A–D) 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 14 lists the port pins available with each device and package type.

Table 14. Port Availability by Device and Package Type

Devices	Package	ADC	Port A	Port B	Port C	Port D	Total I/O
Z8F082ASB, Z8F082APB, Z8F082AQB Z8F042ASB, Z8F042APB, Z8F042AQB Z8F022ASB, Z8F022APB, Z8F022AQB Z8F012ASB, Z8F012APB, Z8F012AQB	8-pin	Yes	[5:0]	No	No	No	6
Z8F081ASB, Z8F081APB, Z8F081AQB Z8F041ASB, Z8F041APB, Z8F041AQB Z8F021ASB, Z8F021APB, Z8F021AQB Z8F011ASB, Z8F011APB, Z8F011AQB	8-pin	No	[5:0]	No	No	No	6
Z8F082APH, Z8F082AHH, Z8F082ASH Z8F042APH, Z8F042AHH, Z8F042ASH Z8F022APH, Z8F022AHH, Z8F022ASH Z8F012APH, Z8F012AHH, Z8F012ASH	20-pin	Yes	[7:0]	[3:0]	[3:0]	[0]	17
Z8F081APH, Z8F081AHH, Z8F081ASH Z8F041APH, Z8F041AHH, Z8F041ASH Z8F021APH, Z8F021AHH, Z8F021ASH Z8F011APH, Z8F011AHH, Z8F011ASH	20-pin	No	[7:0]	[3:0]	[3:0]	[0]	17
Z8F082APJ, Z8F082ASJ, Z8F082AHJ Z8F042APJ, Z8F042ASJ, Z8F042AHJ Z8F022APJ, Z8F022ASJ, Z8F022AHJ Z8F012APJ, Z8F012ASJ, Z8F012AHJ	28-pin	Yes	[7:0]	[5:0]	[7:0]	[0]	23
Z8F081APJ, Z8F081ASJ, Z8F081AHJ Z8F041APJ, Z8F041ASJ, Z8F041AHJ Z8F021APJ, Z8F021ASJ, Z8F021AHJ Z8F011APJ, Z8F011ASJ, Z8F011AHJ	28-pin	No	[7:0]	[7:0]	[7:0]	[0]	25

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.

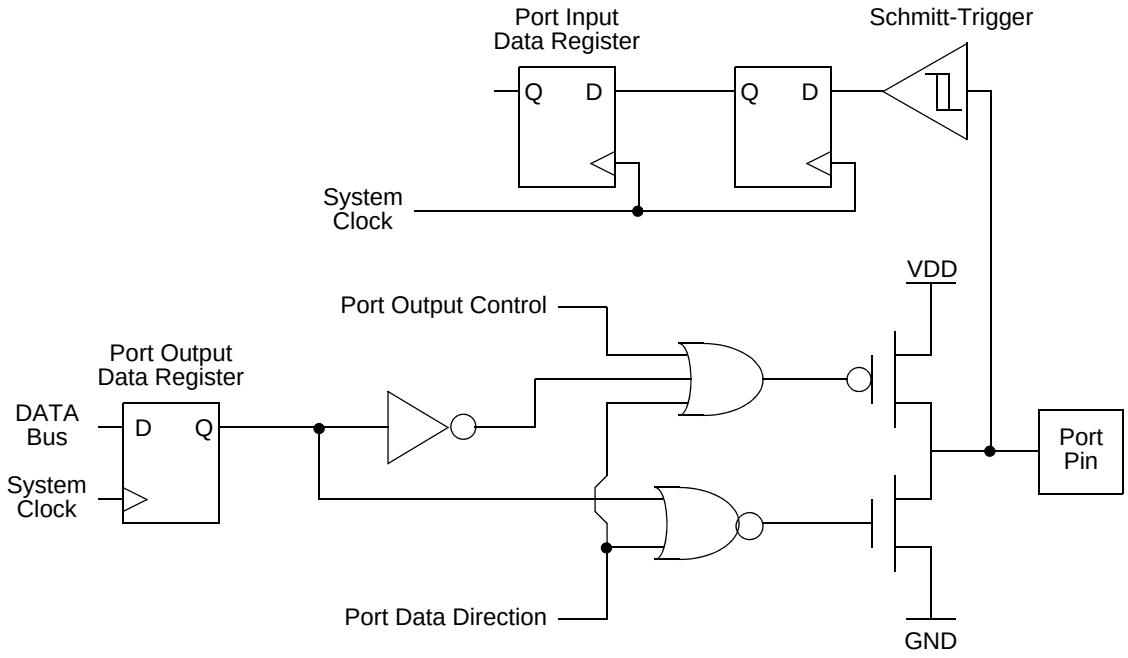


Figure 7. GPIO Port Pin Block Diagram

GPIO Alternate Functions

Many of the GPIO port pins can be 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 General-Purpose I/O 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. Table 15 on page 40 lists the alternate functions possible with each port pin. For those pins with more one alternate function, the alternate function 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. See the Timers chapter on page 70 for more details.

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. As a result, spurious transitions through unwanted alternate function modes will be prevented.

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 Alternate Function register and Alternate Function Subregister AFS1 and is programmable through the LED 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 function, the LED anode must be connected to V_{DD} and the cathode to the GPIO pin. Using all Port C pins in LED drive mode with maximum current may result in excessive total current. See the Electrical Characteristics chapter on page 226 for the maximum total current for the applicable package.

Shared Reset Pin

On the 20- and 28-pin devices, the PD0 pin shares function with a bidirectional reset pin. Unlike all other I/O pins, this pin does not default to GPIO function on power-up. This pin acts as a bidirectional input/open-drain output reset until the software reconfigures it. The PD0 pin is an output-only open drain when in GPIO mode. There are no pull-up, High Drive, or Stop Mode Recovery source features associated with the PD0 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. Since PA2 returns to its $\overline{\text{RESET}}$ 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 180.

Crystal Oscillator Override

For systems using a crystal oscillator, PA0 and PA1 are used to connect the crystal. When the crystal oscillator is enabled, the GPIO settings are overridden and PA0 and PA1 are disabled. See the Oscillator Control Register Definitions section on page 196 for details.

5V Tolerance

All six I/O pins on the 8-pin devices are 5V-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} except when the programmable pull-ups are 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 (OSCCTL) Register such that the external oscillator is selected as the system clock. See the Oscillator Control Register Definitions section on page 196 for details. For 8-pin devices, use PA1 instead of PB3.

Table 15. Port Alternate Function Mapping (Non 8-Pin Parts)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A[1,2]	PA0	T0IN/$\overline{\text{T0OUT}}$	Timer 0 Input/Timer 0 Output Complement	N/A
		Reserved		
	PA1	T0OUT	Timer 0 Output	
		Reserved		
	PA2	DE0	UART 0 Driver Enable	
		Reserved		
	PA3	$\overline{\text{CTS0}}$	$\overline{\text{UART 0 Clear to Send}}$	
		Reserved		
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data	
		Reserved		
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data	
		Reserved		
	PA6	T1IN/$\overline{\text{T1OUT}}$	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 automatically enables the associated alternate function. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
2. Whether PA0/PA6 takes on the timer input or timer output complement function depends on the timer configuration. See the Timer Pin Signal Operation section on page 84 for details.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
4. V_{REF} is available on PB5 in 28-pin products and on PC2 in 20-pin parts.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
6. Because there is only a single alternate function for the Port PD0 pin, the Alternate Function Set registers are not implemented for Port D. Enabling alternate function selections automatically enables the associated alternate function. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.

Table 15. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port B[3]	PB0	Reserved		AFS1[0]: 0
		ANA0/AMPOUT	ADC Analog Input/LPO Output	AFS1[0]: 1
	PB1	Reserved		AFS1[1]: 0
		ANA1/AMPINN	ADC Analog Input/LPO Input (N)	AFS1[1]: 1
	PB2	Reserved		AFS1[2]: 0
		ANA2/AMPINP	ADC Analog Input/LPO Input (P)	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 automatically enables the associated alternate function. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
2. Whether PA0/PA6 takes on the timer input or timer output complement function depends on the timer configuration. See the Timer Pin Signal Operation section on page 84 for details.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
4. V_{REF} is available on PB5 in 28-pin products and on PC2 in 20-pin parts.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
6. Because there is only a single alternate function for the Port PD0 pin, the Alternate Function Set registers are not implemented for Port D. Enabling alternate function selections automatically enables the associated alternate function. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.

Table 15. Port Alternate Function Mapping (Non 8-Pin Parts) (Continued)

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port C[5]	PC0	Reserved		AFS1[0]: 0
		ANA4/CINP/LED	ADC, Comparator Input, or LED Drive	AFS1[0]: 1
	PC1	Reserved		AFS1[1]: 0
		ANA5/CINN/LED	ADC, Comparator Input, or LED Drive	AFS1[1]: 1
	PC2	Reserved		AFS1[2]: 0
		ANA6/LED/V_{REF}[4]	ADC Analog Input, LED, or ADC Voltage Reference	AFS1[2]: 1
	PC3	COUT	Comparator Output	AFS1[3]: 0
		LED	LED drive	AFS1[3]: 1
	PC4	Reserved		AFS1[4]: 0
		LED	LED drive	AFS1[4]: 1
	PC5	Reserved		AFS1[5]: 0
		LED	LED drive	AFS1[5]: 1
	PC6	Reserved		AFS1[6]: 0
		LED	LED drive	AFS1[6]: 1
	PC7	Reserved		AFS1[7]: 0
		LED	LED drive	AFS1[7]: 1
Port D[6]	PD0	$\overline{\text{RESET}}$	External Reset	N/A

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 automatically enables the associated alternate function. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
2. Whether PA0/PA6 takes on the timer input or timer output complement function depends on the timer configuration. See the Timer Pin Signal Operation section on page 84 for details.
3. Because there are at most two choices of alternate function for any pin of Port B, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
4. V_{REF} is available on PB5 in 28-pin products and on PC2 in 20-pin parts.
5. Because there are at most two choices of alternate function for any pin of Port C, the Alternate Function Set Register AFS2 is not used to select the function. Alternate function selection must also be enabled. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.
6. Because there is only a single alternate function for the Port PD0 pin, the Alternate Function Set registers are not implemented for Port D. Enabling alternate function selections automatically enables the associated alternate function. See the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.

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
		$\overline{\text{T0OUT}}$	Timer 0 Output Complement	AFS1[0]: 1	AFS2[0]: 1
	PA1	T0OUT	Timer 0 Outp ut	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[1]	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	$\overline{\text{CTS0}}$	$\overline{\text{UART 0 Clear to Send}}$	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[2]	ADC Analog Input/LPO Input (P)	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[2]	ADC/Comparator Input (N)/LPO Input (N)	AFS1[4]: 1	AFS2[4]: 1
	PA5	TXD0	UART 0 Transmit Data	AFS1[5]: 0	AFS2[5]: 0
		$\overline{\text{T1OUT}}$	Timer 1 Output Complement	AFS1[5]: 0	AFS2[5]: 1
		Reserved		AFS1[5]: 1	AFS2[5]: 0
		Analog Functions[2]	ADC/Comparator Input (P) LPO Output	AFS1[5]: 1	AFS2[5]: 1

Notes:
1. Analog functions include ADC inputs, ADC reference, comparator inputs and LPO ports.
2. The alternate function selection must be enabled; see the Port A–D Alternate Function Subregisters (PxAF) section on page 47 for details.

GPIO Interrupts

Many of the GPIO port pins can be used as interrupt sources. Some port pins can be 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). See the GPIO Mode Interrupt Controller chapter on page 55 for more information about interrupts using the GPIO pins.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data and output data. Table 17 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 17. GPIO Port Registers and Subregisters

Port Register Mnemonic	Port Register Name
P*x*ADDR	Port A–D Address Register; selects subregisters.
P*x*CTL	Port A–D Control Register; provides access to subregisters.
P*x*IN	Port A–D Input Data Register.
P*x*OUT	Port A–D Output Data Register.
Port Subregister Mnemonic	**Port Register Name**
P*x*DD	Data Direction.
P*x*AF	Alternate Function.
P*x*OC	Output Control (Open-Drain).
P*x*HDE	High Drive Enable.
P*x*SMRE	Stop Mode Recovery Source Enable.
P*x*PUE	Pull-up Enable.
PxAFS1	Alternate Function Set 1.
PxAFS2	Alternate Function Set 2.

Port A–D Address Registers

The Port A–D Address registers select the GPIO port functionality accessible through the Port A–D Control registers. The Port A–D Address and Control registers combine to provide access to all GPIO port controls; see Tables 18 and 19.

Table 18. Port A–D GPIO Address Registers (P*x*ADDR)

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, FDCH							

Bit	Description
[7:0] PADDRx	**Port Address** The Port Address selects one of the subregisters accessible through the Port Control Register.

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

Table 19. Port A–D GPIO Address Registers by Bit Description

PADDR[7:0]	Port Control Subregister accessible using the Port A–D 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.
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–D Control Registers

The Port A–D Control registers set the GPIO port operation. The value in the corresponding Port A–D Address Register determines which subregister is read from or written to by a Port A–D Control Register transaction; see Table 20.

Table 20. Port A–D 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, FDDH							

Bit	Description
[7:0] PCTLx	**Port Control** The Port Control Register provides access to all subregisters that configure the GPIO port operation.

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

Port A–D Data Direction Subregisters

The Port A–D Data Direction subregister is accessed through the Port A–D Control Register by writing 01H to the Port A–D Address Register; see Table 21.

Table 21. Port A–D 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–D Address Register, accessible through the Port A–D Control Register							

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

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

Port A–D Alternate Function Subregisters

The Port A–D Alternate Function Subregister, shown in Table 22, is accessed through the Port A–D Control Register by writing 02H to the Port A–D Address Register. The Port A–D Alternate Function subregisters enable the alternate function selection on pins. If disabled, pins functions as GPIO. If enabled, select one of four alternate functions using alternate function set subregisters 1 and 2 as described in the the Port A–D Alternate Function Set 1 Subregisters section on page 50, the GPIO Alternate Functions section on page 37 and the Port A–D Alternate Function Set 2 Subregisters section on page 51. See the GPIO Alternate Functions section on page 37 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 22. Port A–D 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); 01H (Port D); 04H (Port A of 8-pin device)							
R/W	R/W							
Address	If 02H in Port A–D Address Register, accessible through the Port A–D 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–D 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–D Output Control Subregisters

The Port A–D Output Control Subregister, shown in Table 23, is accessed through the Port A–D Control Register by writing 03H to the Port A–D Address Register. Setting the bits in the Port A–D 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 23. Port A–D Output Control Subregisters (PxOC)

Bit	7	6	5	4	3	2	1	0
Field	POC7	POC6	POC5	POC4	POC3	POC2	POC1	POC0
RESET	00H (Ports A-C); 01H (Port D)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	If 03H in Port A–D Address Register, accessible through the Port A–D 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 source current is enabled for any output mode unless overridden by the alternate function (push-pull output). 1 = The source current for the associated pin is disabled (open-drain mode).

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

Port A–D High Drive Enable Subregisters

The Port A–D High Drive Enable Subregister, shown in Table 24, is accessed through the port A–D Control Register by writing 04H to the Port A–D Address Register. Setting the bits in the Port A–D High Drive Enable subregisters to 1 configures the specified port pins for high current output drive operation. The Port A–D High Drive Enable subregister affects the pins directly and, as a result, alternate functions are also affected.

Table 24. Port A–D High Drive Enable Subregisters (PxHDE)

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–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PHDEx	**Port High Drive Enabled** 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–D Stop Mode Recovery Source Enable Subregisters

The Port A–D Stop Mode Recovery Source Enable Subregister, shown in Table 25, is accessed through the Port A–D Control Register by writing 05H to the Port A–D Address Register. Setting the bits in the Port A–D 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 25. Port A–D Stop Mode Recovery Source Enable Subregisters (PxSMRE)

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–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PSMREx	**Port Stop Mode Recovery Source Enabled** 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–D Pull-up Enable Subregisters

The Port A–D Pull-up Enable Subregister, shown in Table 26, is accessed through the Port A–D Control Register by writing 06H to the Port A–D Address Register. Setting the bits in the Port A–D Pull-up Enable subregisters enables a weak internal resistive pull-up on the specified port pins.

Table 26. Port A–D Pull-Up Enable Subregisters (PxPUE)

Bit	7	6	5	4	3	2	1	0
Field	PPUE7	PPUE6	PPUE5	PPUE4	PPUE3	PPUE2	PPUE1	PPUE0
RESET	00H (Ports A-C); 01H (Port D); 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 06H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PPUEx	**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–D Alternate Function Set 1 Subregisters

The Port A–D Alternate Function Set1 Subregister, shown in Table 27, is accessed through the Port A–D Control Register by writing 07H to the Port A–D 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 the GPIO Alternate Functions section on page 37.

Note: Alternate function selection on port pins must also be enabled as described in the Port A–D Alternate Function Subregisters section on page 47.

Table 27. Port A–D Alternate Function Set 1 Subregisters (PxAFS1)

Bit	7	6	5	4	3	2	1	0
Field	PAFS17	PAFS16	PAFS15	PAFS14	PAFS13	PAFS12	PAFS11	PAFS10
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 07H in Port A–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7:0] PAFSx	**Port Alternate Function Set 1** 0 = Port Alternate Function selected, as defined in Tables 15 and 16 on page 43. 1 = Port Alternate Function selected, as defined in Tables 15 and 16 on page 43.

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

Port A–D Alternate Function Set 2 Subregisters

The Port A–D Alternate Function Set 2 Subregister, shown in Table 28, is accessed through the Port A–D Control Register by writing `08H` to the Port A–D 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 16 on page 43.

Note: Alternate function selection on the port pins must also be enabled. See the Port A–D Alternate Function Subregisters section on page 47 for details.

Table 28. Port A–D 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–D Address Register, accessible through the Port A–D Control Register							

Bit	Description
[7] PAFS2x	**Port Alternate Function Set 2** 0 = Port Alternate Function selected, as defined in Table 16. 1 = Port Alternate Function selected, as defined in Table 16.

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, shown in Table 29, return 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 29. 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							
X = Undefined.								

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–D Output Data Register

The Port A–D Output Data Register, shown in Table 30, controls the output data to the pins.

Table 30. Port A–D Output Data Register (P*x*OUT)

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, FDFH							

Bit	Description
[7:0] PxOUT	**Port Output Data** 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. 0 = Drive a logical 0 (Low). 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.
Note: x indicates the specific GPIO port pin number (7–0).	

LED Drive Enable Register

The LED Drive Enable Register, shown in Table 31, activates the controlled current drive. The Port C pin must first be enabled for the LED function by setting Alternate Function sub-register AFS1 and Alternate Function register.. LEDEN bits [7:0] correspond to Port C bits [7:0], respectively.

Table 31. 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] LEDENx	**LED Drive Enable** These bits determine which Port C pins are connected to an internal current sink. 0 = Tristate the Port C pin. 1 = Enable controlled current sink on the Port C pin. **Note:** *x* indicates the specific GPIO port pin number (7–0).

LED Drive Level High Register

The LED Drive Level registers contain two control bits for each Port C pin, as shown in Table 32. These two bits select between four programmable drive levels. Each pin is individually programmable.

Table 32. 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] LEDLVLHx	**LED Level High Bit** {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA 01 = 7mA 10 = 13mA 11 = 20mA

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

LED Drive Level Low 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 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] LEDLVLLx	**LED Level Low Bit** {LEDLVLH, LEDLVLL} select one of four programmable current drive levels for each Port C pin. 00 = 3mA 01 = 7mA 10 = 13mA 11 = 20mA

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

GPIO Mode Interrupt Controller

The interrupt controller on the Z8 Encore! XP F082A Series products prioritizes the interrupt requests from the on-chip peripherals and the GPIO port pins. The features of interrupt controller include:

- 20 possible interrupt sources with 18 unique interrupt vectors:
  - Twelve GPIO port pin interrupt sources (two interrupt vectors are shared)
  - Eight on-chip peripheral interrupt sources (two interrupt vectors 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 and LVD can be configured to generate an interrupt
- Supports vectored and polled interrupts

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), which is available for download on www.zilog.com.

Interrupt Vector Listing

Table 34 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 34. 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 Watchdog Timer)
	003AH	Primary Oscillator Fail Trap (not an interrupt)
	003CH	Watchdog 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 or LVD (see Reset, Stop Mode Recovery and Low Voltage Detection)
	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, selectable rising or falling input edge
	0022H	Port A 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
Lowest	0036H	Port C Pin 0, both input edges
	0038H	Reserved

Architecture

Figure 8 displays the interrupt controller block diagram.

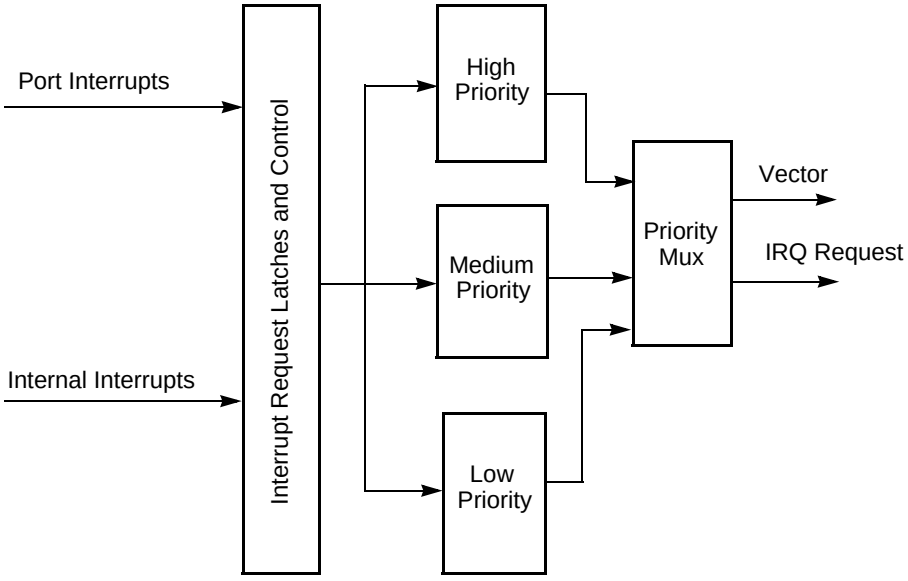


Figure 8. Interrupt Controller Block Diagram

Operation

This section describes the operational aspects of the following functions.

Master Interrupt Enable: see page 57

Interrupt Vectors and Priority: see page 58

Interrupt Assertion: see page 58

Software Interrupt Assertion: see page 59

Watchdog Timer Interrupt Assertion: see page 59

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 EI (Enable Interrupt) instruction
- Execution of an IRET (Return from Interrupt) 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 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 of the interrupts are enabled with identical interrupt priority (all as Level 2 interrupts, for example), the interrupt priority is assigned from highest to lowest as specified in Table 34 on page 56. 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 34, above. Reset, Watchdog Timer interrupt (if enabled), Primary Oscillator Fail Trap, Watchdog 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 can generate 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 time-out 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 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, shown in Table 35, 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 can read the Interrupt Request 0 Register to determine if any interrupt requests are pending.

Table 35. Interrupt Request 0 Register (IRQ0)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1I	T0I	U0RXI	U0TXI	Reserved	Reserved	ADCI
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	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] ADCI	**ADC Interrupt Request** 0 = No interrupt request is pending for the analog-to-digital Converter. 1 = An interrupt request from the Analog-to-Digital Converter is awaiting service.

Interrupt Request 1 Register

The Interrupt Request 1 (IRQ1) Register, shown in Table 36, 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 can read the Interrupt Request 1 Register to determine if any interrupt requests are pending.

Table 36. 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 A Pin 7 or LVD Interrupt Request** 0 = No interrupt request is pending for GPIO Port A or LVD. 1 = An interrupt request from GPIO Port A or LVD.
[6] PA6CI	**Port A Pin 6 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] PA5I	**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, shown in 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 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 37. 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 (0–3).

IRQ0 Enable High and Low Bit Registers

Table 38 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers, shown in Tables 39 and 40, form a priority-encoded enabling for interrupts in the Interrupt Request 0 Register.

Table 38. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	IRQ0ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Medium
1	1	Level 3	High

Note: x indicates register bits 0–7.

Table 39. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	Reserved	Reserved	ADCENH
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	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 40. IRQ0 Enable Low Bit Register (IRQ0ENL)

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

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1ENL	**Timer 1 Interrupt Request Enable Low Bit**
[5] T0ENL	**Timer 0 Interrupt Request Enable Low Bit**

Bit	Description (Continued)
[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 41 describes the priority control for IRQ1. The IRQ1 Enable High and Low Bit registers, shown in Tables 41 and 42, form a priority-encoded enabling for interrupts in the Interrupt Request 1 Register.

Table 41. IRQ1 Enable and Priority Encoding

IRQ1ENH[*x*]	IRQ1ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Medium
1	1	Level 3	High

Note: *x* indicates register bits 0–7.

Table 42. 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] or LVD 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**

See the Shared Interrupt Select Register (IRQSS) Register on page 68 for selection of either the LVD or the comparator as the interrupt source.

Table 43. 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] or LVD Interrupt Request Enable Low Bit**
[6] PA6CENL	**Port A Bit[6] or Comparator Interrupt Request Enable Low Bit**
[5:0] PAxENL	**Port A Bit[*x*] Interrupt Request Enable Low Bit**

IRQ2 Enable High and Low Bit Registers

Table 44 describes the priority control for IRQ2. The IRQ2 Enable High and Low Bit registers, shown in Tables 44 and 45, form a priority-encoded enabling for interrupts in the Interrupt Request 2 Register.

Table 44. IRQ2 Enable and Priority Encoding

IRQ2ENH[*x*]	IRQ2ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Medium
1	1	Level 3	High

Note: *x* indicates register bits 0–7.

Table 45. 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 46. 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 Low Bit**

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 47, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port A input pin.

Table 47. 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
[7:0] IESx	**Interrupt Edge Select *x*** 0 = An interrupt request is generated on the falling edge of the PA*x* input. 1 = An interrupt request is generated on the rising edge of the PA*x* input.

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

Shared Interrupt Select Register

The Shared Interrupt Select (IRQSS) Register, shown in Table 48, 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 48. Shared Interrupt Select Register (IRQSS)

Bit	7	6	5	4	3	2	1	0
Field	PA7VS	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] PA7VS	**PA7/LVD Selection** 0 = PA7 is used for the interrupt for PA7VS interrupt request. 1 = The LVD is used for the interrupt for PA7VS interrupt request.
[6] PA6CS	**PA6/Comparator Selection** 0 = PA6 is used for the interrupt for PA6CS interrupt request. 1 = The Comparator is used for the interrupt for PA6CS interrupt request.
[5:0]	**Reserved** These bits are reserved and must be programmed to 000000.

Interrupt Control Register

The Interrupt Control (IRQCTL) Register, shown in Table 49, contains the master enable bit for all interrupts.

Table 49. 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	**Interrupt Request Enable** This bit is set to 1 by executing an EI (Enable Interrupts) or IRET (Interrupt Return) 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. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	**Reserved** These bits are reserved and must be programmed to 0000000.

Timers

These Z8 Encore! XP F082A Series products contain two 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse-width modulated (PWM) signals. The timers' feature 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) may also provide basic timing functionality. For information about using the Baud Rate Generator as an additional timer, see the Universal Asynchronous Receiver/ Transmitter chapter on page 99.

Architecture

Figure 9 displays the architecture of the timers.

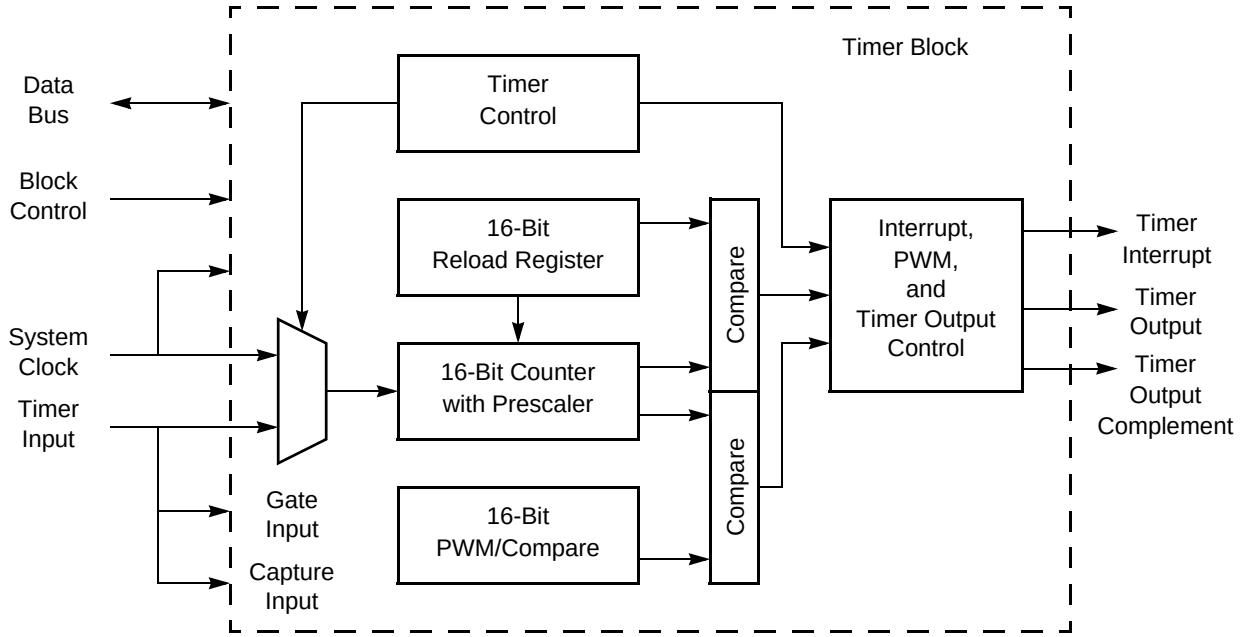


Figure 9. Timer Block Diagram

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 for configuring 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 for configuring a timer for Continuous Mode and initiating 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. Further, the high or low state of the input signal pulse must be no less than twice the system clock period. A shorter pulse may not be captured.

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 for configuring 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:

COUNTER Mode Timer Input Transitions = Current Count Value - 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. Further, the high or low state of the comparator output signal pulse must be no less than twice the system clock period. A shorter pulse may not be captured.

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 for configuring 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:

Comparator Output Transitions = Current Count Value – Start Value

PWM Single Output Mode

In PWM Single Output Mode, the timer outputs a Pulse-Width Modulator (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 for configuring 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 Single Output 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 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:

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

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

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

PWM Dual Output Mode

In PWM Dual Output Mode, the timer outputs a Pulse-Width Modulated (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 delay ensures a time gap between the deassertion of one PWM output to the assertion of its complement.

Observe the following steps for configuring 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 by writing the TMODE bits in the TxCTL1 Register and the TMODEHI bit in TxCTL0 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 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 TxCTL0 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 TxCTL0 Register clears indicating the timer interrupt is not because of an input capture event.

Observe the following steps for configuring 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 event or a reload. 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 TxCTL0 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 TxCTL0 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 TxCTL0 Register is cleared to indicate the timer interrupt is not caused by an input capture event.

Observe the following steps for configuring 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 by writing the TMODE bits in the TxCTL1 Register and the TMODEHI bit in TxCTL0 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. This allows the software to determine if interrupts were generated by either a capture event or a reload. 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 TxCTL0 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 for configuring a timer for Compare Mode and initiating 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 for configuring a timer for Gated Mode and initiating 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 TxCTL0 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 TxCTL0 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 TxCTL0 Register is cleared to indicate the timer interrupt is not because of an input capture event.

Observe the following steps for configuring 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 TxCTL0 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. 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

The timer output function 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 ($\overline{\text{TxOUT}}$). When selected by the GPIO Alternate Function registers, this pin functions as a timer input in all modes except for Dual PWM Output Mode. For this mode, there is no timer input available. For the 8-pin device, the $\overline{\text{T0OUT}}$ function is available for the various timer out functions. The $\overline{\text{T1OUT}}$ function is only available in Dual PWM Output Mode.

Timer Control Register Definitions

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

Timer 0–1 Control Registers: see page 84

Timer 0–1 High and Low Byte Registers: see page 88

Timer Reload High and Low Byte Registers: see page 90

Timer 0–1 PWM High and Low Byte Registers: see page 91

Timer 0–1 Control Registers

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

Time 0–1 Control Register 0

The Timer Control Register 0 (TxCTL0) and Timer Control Register 1 (TxCTL1), shown in Table 50, determine the timer operating mode. These registers each include 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 50. 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
Address	F06H, F0EH							

Bit	Description
[7] TMODEHI	**Timer Mode High Bit** This bit, along with the TMODE field in the TxCTL1 Register, determines the operating mode of the timer. This bit is the most significant bit of the timer mode selection value. See the description of the Timer 0–1 Control Register 1 (TxCTL1) for details about the full timer mode decoding.
[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	**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, shown in Table 51, enable and disable the timers, set the prescaler value and determine the timer operating mode.

Table 51. 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. **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. **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. **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. **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.

Bit	Description (Continued)
[6] TPOL (cont'd)	**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. **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. **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). **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. **Comparator Counter 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. Also: 0 = Count is captured on the rising edge of the comparator output. 1 = Count is captured on the falling edge of the comparator output. **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 Subregister is not required to be set to output on TxOUT. Changing the TPOL bit with the timer enabled and running does not immediately change the TxOUT.

Bit	Description (Continued)
[5:3] PRES	**Prescale value** 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. 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.
[2:0] TMODE	**Timer Mode** This field, along with the TMODEHI bit in the TxCTL0 Register, determines the operating mode of the timer. TMODEHI is the most significant bit of the timer mode selection value. The entire operating mode bits are expressed as {TMODEHI, TMODE[2:0]}. The TMODEHI is bit 7 of the TxCTL0 Register while TMODE[2:0] is the lower 3 bits of the TxCTL1 Register. 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.

Timer 0–1 High and Low Byte Registers

The Timer 0–1 High and Low Byte (TxH and TxL) registers, shown in Tables 52 and 53, 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 TxL read 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 52. 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 53. 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, shown in Tables 54 and 55, 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 54. 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 55. 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
[7:0] TRH, 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, shown in Tables 56 and 57, control Pulse-Width Modulator (PWM) operations. These registers also store the Capture values for the Capture and Capture/Compare modes.

Table 56. 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 57. 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.

The TxPWMH and TxPWML registers also store the 16-bit captured timer value when operating in Capture or Capture/Compare modes.

Watchdog Timer

The Watchdog Timer (WDT) protects against corrupt or unreliable software, power faults and other system-level problems which may place the Z8 Encore! XP F082A 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 Watchdog Timer is a one-shot timer that resets or interrupts the Z8 Encore! XP F082A 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 downcounter 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], WDTH[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 58 provides information about approximate time-out delays for the minimum and maximum WDT reload values.

Table 58. 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 Watchdog Timer 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 downcounter to be reloaded with the WDT reload value stored in the Watchdog Timer Reload registers. Counting resumes following the reload operation.

When the Z8 Encore! XP F082A Series devices are operating in Debug Mode (using the on-chip debugger), 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 the WDT_RES Flash option bit, see the Flash Option Bits chapter on page 159.

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 Reset Status (RSTSTAT) Register; see the Reset Status Register on page 29. 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 (RSTSTAT) Register must be read before clearing the WDT interrupt. This read clears the WDT time-out Flag and prevents further WDT interrupts from immediately occurring.

WDT Interrupt in Stop Mode

If configured to generate an interrupt when a time-out occurs and the Z8 Encore! XP F082A Series devices 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 Reset Status (RSTSTAT) Register are set to 1 following a WDT time-out in Stop Mode. For more information about Stop Mode Recovery, see the Reset, Stop Mode Recovery and Low Voltage Detection chapter on page 22.

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 Reset Status (RSTSTAT) Register is set to 1. For more information about system reset, see the Reset, Stop Mode Recovery and Low Voltage Detection chapter on page 22.

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 Reset Status (RSTSTAT) Register are set to 1 following WDT time-out in Stop Mode.

Watchdog Timer Reload Unlock Sequence

Writing the unlock sequence to the Watchdog Timer (WDTCTL) Control 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. Observe the following steps to unlock the Watchdog Timer Reload Byte registers (WDTU, WDTH and WDTL) 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) with the appropriate time-out value.
4. Write the Watchdog Timer Reload High Byte Register (WDTH) with the appropriate time-out value.
5. Write the Watchdog Timer Reload Low Byte Register (WDTL) with the appropriate time-out value.

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 Calibration

Due to its extremely low operating current, the Watchdog Timer oscillator is somewhat inaccurate. This variation can be corrected using the calibration data stored in the Flash Information Page; see Tables 100 and 101 on page 173 for details. Loading these values

into the Watchdog Timer Reload registers results in a one-second time-out at room temperature and 3.3V supply voltage. Time-outs other than one second may be obtained by scaling the calibration values up or down as required.

Note: The Watchdog Timer accuracy still degrades as temperature and supply voltage vary. See Table 137 on page 235 for details.

Watchdog Timer Control Register Definitions

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

Watchdog Timer Control Register (WDTCTL): see page 96

Watchdog Timer Reload Upper Byte Register (WDTU): see page 97

Watchdog Timer Reload High Byte Register (WDTH): see page 97

Watchdog Timer Reload Low Byte Register (WDTL): see page 98

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 59. 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							

Note: X = Undefined.

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 60 through 62, 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 60. Watchdog Timer Reload Upper Byte Register (WDTU)

Bit	7	6	5	4	3	2	1	0
Field	WDTU							
RESET	00H							
R/W	R/W*							
Address	FF1H							

Note: A read returns the current WDT count value; a 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 61. Watchdog Timer Reload High Byte Register (WDTH)

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
RESET	04H							
R/W	R/W*							
Address	FF2H							

Note: A read returns the current WDT count value; a 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 62. Watchdog Timer Reload Low Byte Register (WDTL)

Bit	7	6	5	4	3	2	1	0
Field	WDTL							
RESET	00H							
R/W	R/W*							
Address	FF3H							
Note: A read returns the current WDT count value; a write sets the appropriate reload value.								

Bit	Description
[7:0] WDTL	**WDT Reload Low** 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. Features of the 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
- Baud rate generator (BRG) can be configured and used as a basic 16-bit timer
- Driver enable (DE) 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.

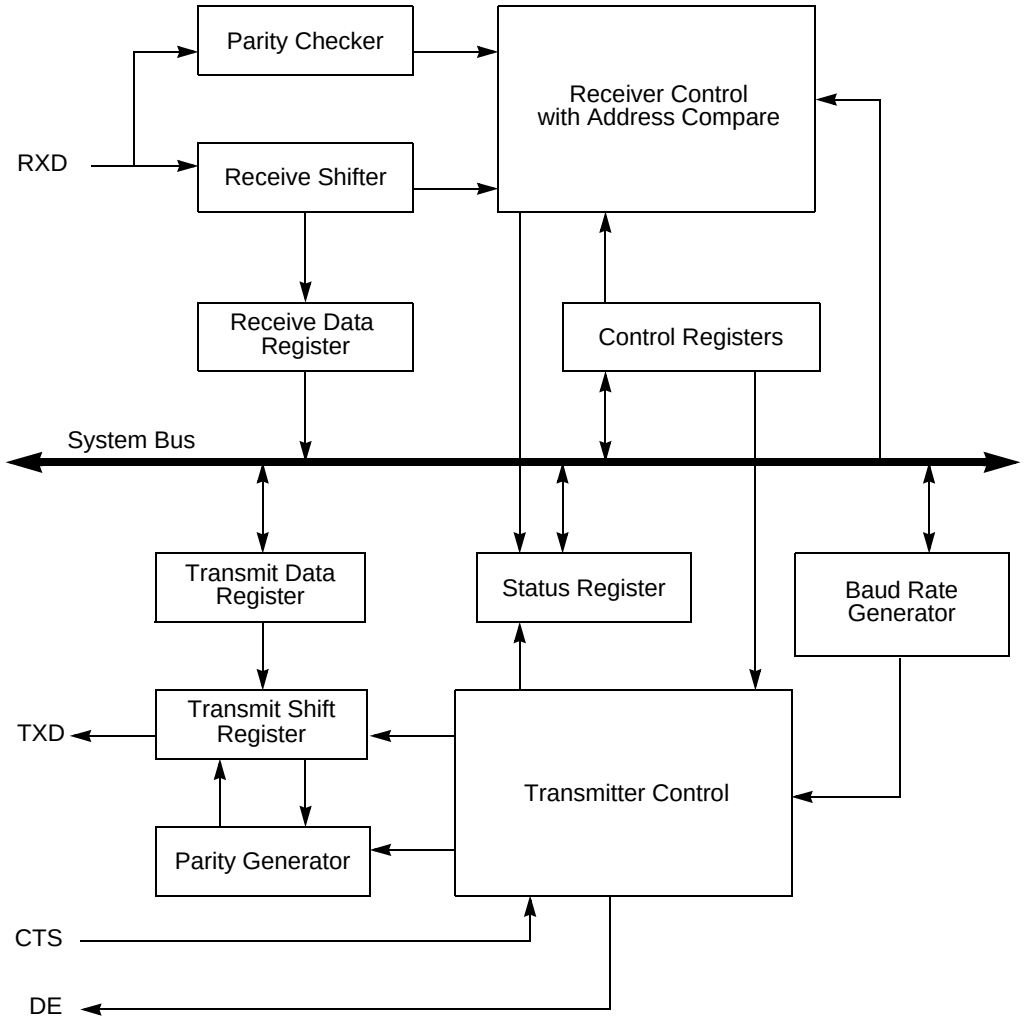


Figure 10. UART Block Diagram

Operation

The UART always transmits and receives data in an 8-bit data format, least-significant bit 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. Figures 11 and 12 display the asynchronous data format employed by the UART without parity and with parity, respectively.

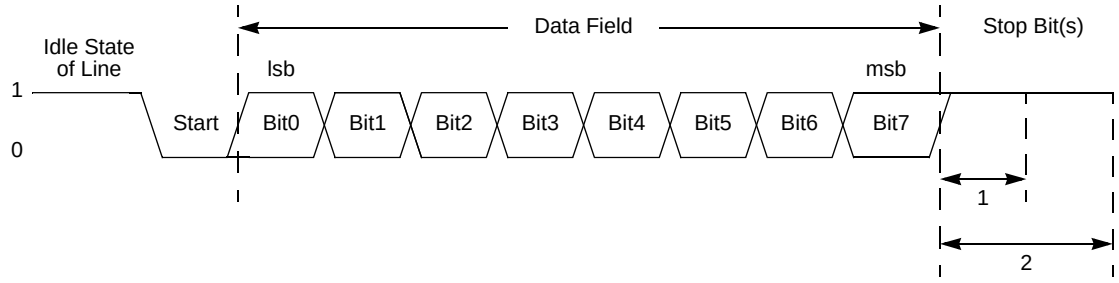


Figure 11. UART Asynchronous Data Format without Parity

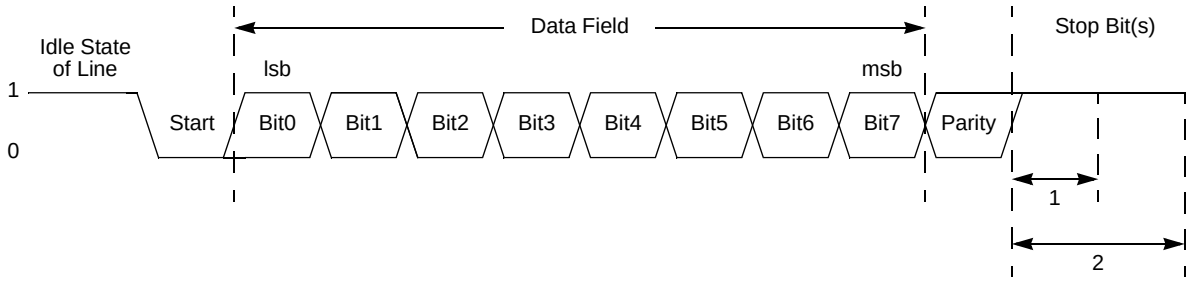


Figure 12. UART Asynchronous Data Format with Parity

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 $\overline{\text{CTS}}$ 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 $\overline{\text{CTS}}$ 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:
2. Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
3. 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.
4. Clear the UART Transmit interrupt bit in the applicable Interrupt Request Register.
5. 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 5. 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 (and 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! 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 features 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:

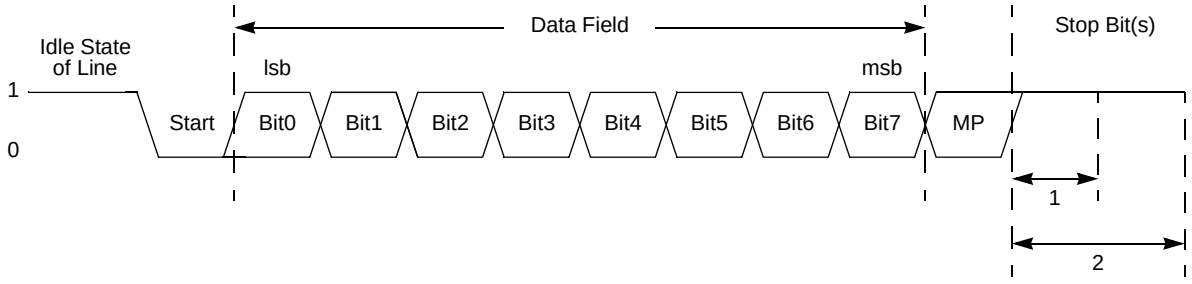


Figure 13. UART Asynchronous MULTIPROCESSOR Mode Data Format

In MULTIPROCESSOR (9-bit) Mode, the Parity (9th) bit location 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, plus 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.

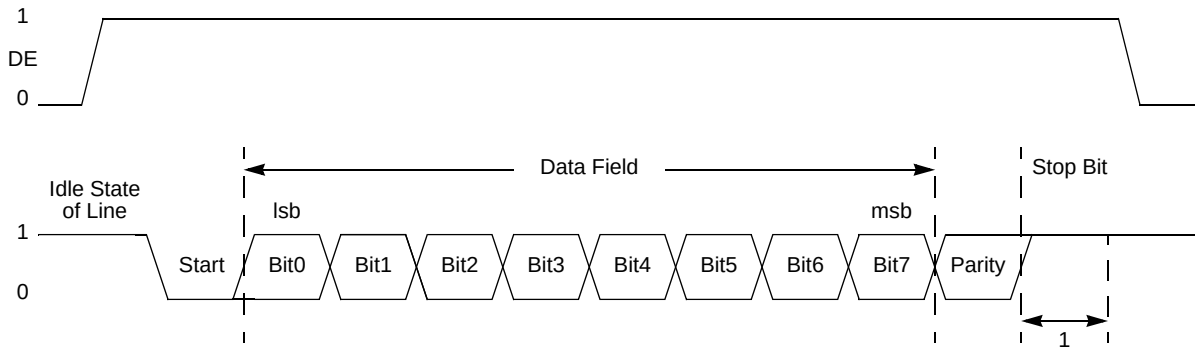


Figure 14. UART Driver Enable Signal Timing (shown with 1 Stop Bit and Parity)

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

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

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.

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 actions occur:

- 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 may not 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 is 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.

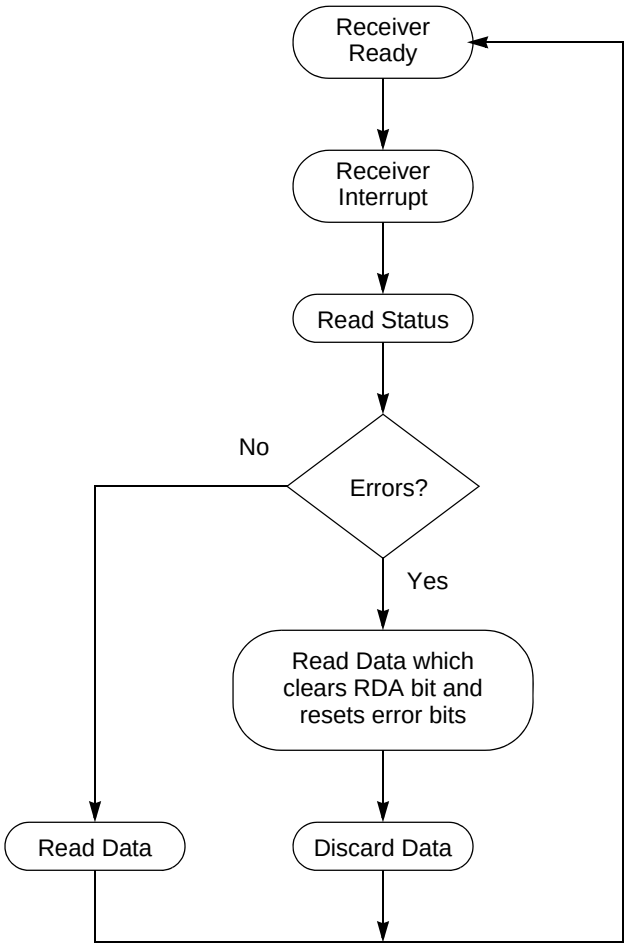


Figure 15. UART Receiver Interrupt Service Routine Flow

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 an interrupt upon time-out. Observe the following steps to configure the Baud Rate Generator as a timer with an interrupt upon 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 BRGCTL 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 infrared operation, see the Infrared Encoder/Decoder chapter on page 120.

UART Control 0 and Control 1 Registers

The UART Control 0 (UxCTL0) and Control 1 (UxCTL1) registers, shown in Tables 63 and 64, 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 63. 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 $\overline{CTS}$ signal and the CTSE bit. If the $\overline{CTS}$ 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	**CTS Enable** 0 = The $\overline{CTS}$ signal has no effect on the transmitter. 1 = The UART recognizes the $\overline{CTS}$ 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.
[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 64. 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	**Baud Rate Control** This bit causes an alternate UART behavior depending on the value of the REN bit in the UART Control 0 Register. When the UART receiver is not enabled (REN=0), this bit determines whether the Baud Rate Generator issues interrupts. 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. 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. 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.
[1] RDAIRQ	**Receive Data Interrupt Enable** 0 = Received data and receiver errors generates an interrupt request to the Interrupt Controller. 1 = Received data does not generate an interrupt request to the Interrupt Controller. Only receiver errors generate an interrupt request.
[0] IREN	**Infrared Encoder/Decoder Enable** 0 = Infrared Encoder/Decoder is disabled. UART operates normally. 1 = Infrared Encoder/Decoder is enabled. The UART transmits and receives data through the Infrared Encoder/Decoder.

UART Status 0 Register

The UART Status 0 (UxSTAT0) and Status 1(UxSTAT1) registers, shown in Tables 65 and 66, identify the current UART operating configuration and status.

Table 65. 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	**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	**$\overline{\text{CTS}}$ 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 66. 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 and must be programmed to 000000.
[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 Transmit Data Register

Data bytes written to the UART Transmit Data (UxTXD) Register, shown in Table 67, are shifted out on the TXDx pin. The Write-only UART Transmit Data Register shares a Register File address with the read-only UART Receive Data Register.

Table 67. 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							

Note: X = Undefined.

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 (UxRXD) Register, shown in Table 68. The read-only UART Receive Data Register shares a Register File address with the Write-only UART Transmit Data Register.

Table 68. 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							

Note: X = Undefined.

Bit	Description
[7:0] RXD	**Receive Data** UART receiver data byte from the RXD*x* pin.

UART Address Compare Register

The UART Address Compare (UxADDR) Register stores the multi-node network address of the UART (see Table 69). 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 69. 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] COMP_ADDR	**Compare Address** This 8-bit value is compared to incoming address bytes.

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High (UxBRH) and Low Byte (UxBRL) registers, shown in Tables 70 and 71, 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 70. 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							

Bit	Description
[7:0] BRH	**UART Baud Rate High Byte**

Table 71. 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							

Bit	Description
[7:0] BRL	**UART Baud Rate Low Byte**

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 5 percent. Table 72 provides information about the data rate errors for popular baud rates and commonly used crystal oscillator frequencies.

Table 72. UART Baud Rates

10.0 MHz System Clock				5.5296 MHz System Clock			
Acceptable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Acceptable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	1	625.0	0.00	625.0	N/A	N/A	N/A
250.0	3	208.33	–16.67	250.0	1	345.6	38.24
115.2	5	125.0	8.51	115.2	3	115.2	0.00
57.6	11	56.8	–1.36	57.6	6	57.6	0.00
38.4	16	39.1	1.73	38.4	9	38.4	0.00
19.2	33	18.9	0.16	19.2	18	19.2	0.00
9.60	65	9.62	0.16	9.60	36	9.60	0.00
4.80	130	4.81	0.16	4.80	72	4.80	0.00
2.40	260	2.40	–0.03	2.40	144	2.40	0.00
1.20	521	1.20	–0.03	1.20	288	1.20	0.00
0.60	1042	0.60	–0.03	0.60	576	0.60	0.00
0.30	2083	0.30	0.2	0.30	1152	0.30	0.00
3.579545 MHz System Clock				1.8432 MHz System Clock			

Table 72. UART Baud Rates (Continued)

Acceptable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)	Acceptable Rate (kHz)	BRG Divisor (Decimal)	Actual Rate (kHz)	Error (%)
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	N/A	N/A	N/A	625.0	N/A	N/A	N/A
250.0	1	223.72	–10.51	250.0	N/A	N/A	N/A
115.2	2	111.9	–2.90	115.2	1	115.2	0.00
57.6	4	55.9	–2.90	57.6	2	57.6	0.00
38.4	6	37.3	–2.90	38.4	3	38.4	0.00
19.2	12	18.6	–2.90	19.2	6	19.2	0.00
9.60	23	9.73	1.32	9.60	12	9.60	0.00
4.80	47	4.76	–0.83	4.80	24	4.80	0.00
2.40	93	2.41	0.23	2.40	48	2.40	0.00
1.20	186	1.20	0.23	1.20	96	1.20	0.00
0.60	373	0.60	–0.04	0.60	192	0.60	0.00
0.30	746	0.30	–0.04	0.30	384	0.30	0.00

Infrared Encoder/Decoder

Z8 Encore! XP F082A Series products contain a fully-functional, high-performance UART to Infrared Encoder/Decoder (endec). The infrared endec is integrated with an on-chip UART to allow easy communication between the Z8 Encore! XP MCU 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.

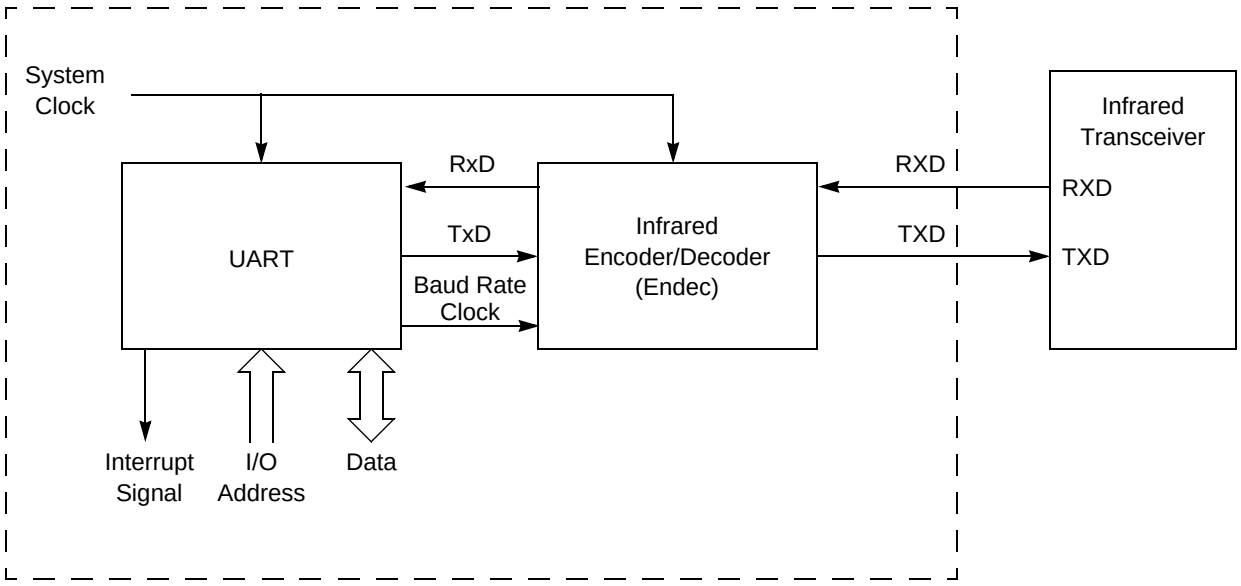


Figure 16. Infrared Data Communication System Block Diagram

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. Likewise, 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 the Z8 Encore! XP F082A Series products while the IR_TXD signal is output through the TXD pin.

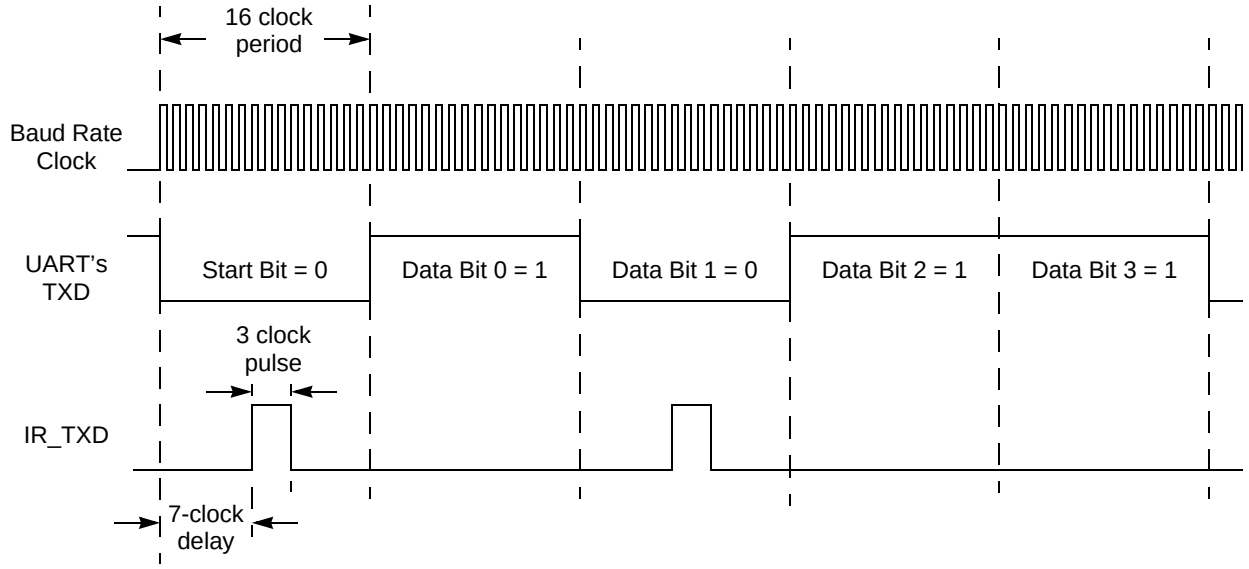


Figure 17. Infrared Data Transmission

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 F082A Series products while the IR_RXD signal is received through the RXD pin.

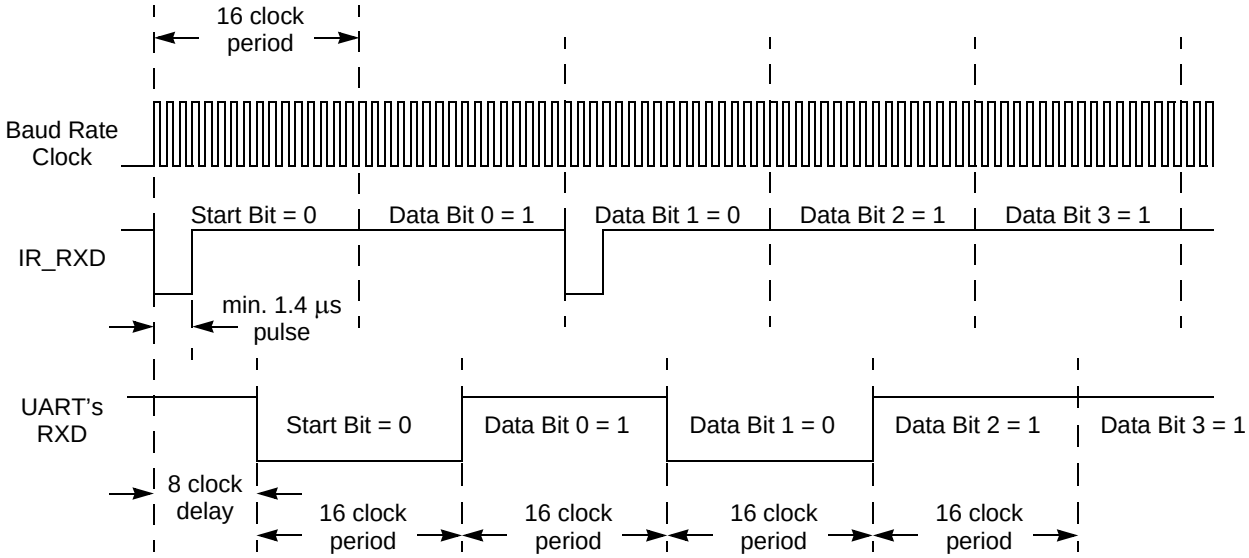


Figure 18. IrDA Data Reception

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 section on page 99.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UART Control 1 Register to 1 to enable the Infrared Encoder/Decoder 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:

- 11-bit resolution in DIFFERENTIAL Mode
- 10-bit resolution in SINGLE-ENDED Mode
- Eight single-ended analog input sources are multiplexed with general-purpose I/O ports
- 9th analog input obtained from temperature sensor peripheral
- 11 pairs of differential inputs also multiplexed with general-purpose I/O ports
- Low-power operational amplifier (LPO)
- Interrupt on conversion complete
- Bandgap generated internal voltage reference with two selectable levels
- Manual in-circuit calibration is possible employing user code (offset calibration)
- Factory calibrated for in-circuit error compensation

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.

The input stage of the ADC allows both differential gain and buffering. The following input options are available:

- Unbuffered input (SINGLE-ENDED and DIFFERENTIAL modes)
- Buffered input with unity gain (SINGLE-ENDED and DIFFERENTIAL modes)
- LPO output with full pin access to the feedback path

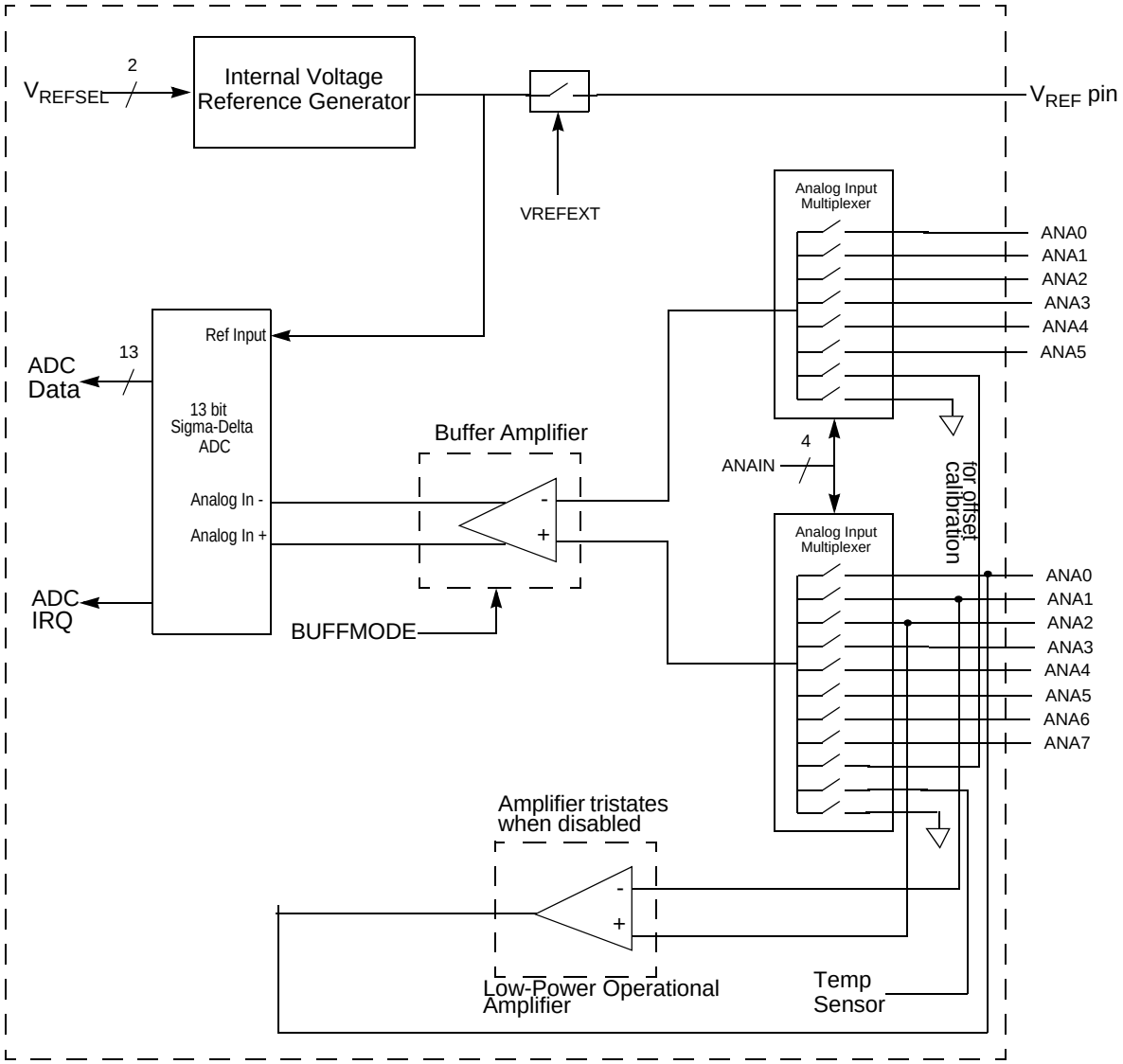


Figure 19. Analog-to-Digital Converter Block Diagram

Operation

In both SINGLE-ENDED and DIFFERENTIAL modes, the effective output of the ADC is an 11-bit, signed, two's complement digital value. In DIFFERENTIAL Mode, the ADC can output values across the entire 11-bit range, from –1024 to +1023. In SINGLE-ENDED Mode, the output generally ranges from 0 to +1023, but offset errors can cause small negative values.

The ADC registers actually return 13 bits of data, but the two LSBs are intended for compensation use only. When the software 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.

Hardware Overflow

When the hardware overflow bit (OVF) is set in ADC Data Low Byte (ADCD_L) Register, all other data bits are invalid. The hardware overflow bit is set for values greater than V_{REF} and less than $-V_{REF}$ (DIFFERENTIAL Mode).

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 power up. 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 appropriate analog inputs by configuring the general-purpose I/O pins for alternate analog function. This configuration disables the digital input and output drivers.
2. Write the ADC Control/Status Register 1 to configure the ADC.
   - Write to BUFMODE[2:0] to select SINGLE-ENDED or DIFFERENTIAL mode, plus unbuffered or buffered mode.
   - Write the REFSELH 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.
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 (the ADC can be configured and enabled with the same write instruction):
   - 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 REFSELH bit is contained in the ADC Control/Status Register 1.
- 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:
   - 13-bit two's-complement result written to {ADCD_H[7:0], ADCD_L[7:3]}
   - Sends an interrupt request to the Interrupt Controller denoting conversion complete
   - 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 overwrites 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 immediately 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 appropriate 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 to BUFMODE[2:0] to select SINGLE-ENDED or DIFFERENTIAL mode, plus unbuffered or buffered mode.
- Write the REFSELH 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.

3. Write to the ADC Control Register 0 to configure the ADC for continuous conversion. The bit fields in the ADC Control Register may 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 REFSELH bit is contained in ADC Control/Status Register 1.
   - 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 13-bit two's complement result to {ADCD_H[7:0], ADCD_L[7:3]}
   - Sends 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 when a conversion has been completed. When the ADC is disabled, no new interrupts are asserted; however, an interrupt pending when the ADC is disabled is not cleared.

Calibration and Compensation

The Z8 Encore! XP F082A Series ADC is factory calibrated for offset error and gain error, with the compensation data stored in Flash memory. Alternatively, you can perform your own calibration, storing the values into Flash themselves. Thirdly, the user code can perform a manual offset calibration during DIFFERENTIAL Mode operation.

Factory Calibration

Devices that have been factory calibrated contain 30 bytes of calibration data in the Flash option bit space. This data consists of 3 bytes for each input mode, one for offset and two for gain correction. For a list of input modes for which calibration data exists, see the Zilog Calibration Data section on page 168.

User Calibration

If you have precision references available, its own external calibration can be performed using any input modes. This calibration data takes into account buffer offset and nonlinearity; therefore Zilog recommends that this calibration be performed separately for each of the ADC input modes planned for use.

Manual Offset Calibration

When uncalibrated, the ADC has significant offset (see Table 139 on page 236). Subsequently, manual offset calibration capability is built into the block. When the ADC Control Register 0 sets the input mode (`ANAIN[2:0]`) to MANUAL OFFSET CALIBRATION Mode, the differential inputs to the ADC are shorted together by an internal switch. Reading the ADC value at this point produces 0 in an ideal system. The value actually read is the ADC offset. This value can be stored in nonvolatile memory (see the Nonvolatile Data Storage chapter on page 176) and accessed by user code to compensate for the input offset error. There is no provision for manual gain calibration.

Software Compensation Procedure Using Factory Calibration Data

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

$$ADC_{comp} = (ADC_{uncomp} - OFFCAL) + ((ADC_{uncomp} - OFFCAL) \times GAINCAL) / 2^{14}$$

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

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 incorrectly evaluates to zero.

Caution: Although the ADC can be used without the gain and offset compensation, it does exhibit nonunity 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 Compensation Details

High-efficiency assembly code that performs ADC compensation is available for download on www.zilog.com. This section offers a bit-specific description of the ADC compensation process used by this code.

The following data bit definitions are used:

0–9, a–f = bit indices in hexadecimal
s = sign bit
v = overflow bit
– = unused

Input Data

MSB	LSB		
s b a 9 8 7 6 5	4 3 2 1 0 – – v	(ADC)	ADC Output Word; if v = 1, the data is invalid
	s 6 5 4 3 2 1 0		Offset Correction Byte
s s s s s 7 6 5	4 3 2 1 0 0 0 0	(Offset)	Offset Byte shifted to align with ADC data
s e d c b a 9 8	7 6 5 4 3 2 1 0	(Gain)	Gain Correction Word

Compensation Steps:

1. Correct for Offset:

ADC MSB	ADC LSB

–

Offset MSB	Offset LSB

=

#1 MSB	#1 LSB

2. Compute the absolute value of the offset-corrected ADC value *if negative*; the gain correction factor is computed assuming positive numbers, with sign restoration afterward.

#2 MSB	#2 LSB

Also compute the absolute value of the gain correction word, if negative.

AGain MSB	AGain LSB

3. Multiply by the Gain Correction Word. If operating in DIFFERENTIAL Mode, there are two gain correction values: one for positive ADC values, another for negative ADC values. Use the appropriate Gain Correction Word based on the sign computed by byte #2.

#2 MSB	#2 LSB

*

AGain MSB	AGain LSB

=

#3	#3	#3	#3

4. Round the result and discard the least significant two bytes (equivalent to dividing by 2^{16}).

#3	#3	#3	#3

–

0x00	0x00	0x80	0x00

=

#4 MSB	#4 LSB

5. Determine the sign of the gain correction factor using the sign bits from Step 2. If the offset-corrected ADC value *and* the gain correction word both have the same sign, then the factor is positive and remains unchanged. If they have differing signs, then the factor is negative and must be multiplied by –1.

#5 MSB	#5 LSB

6. Add the gain correction factor to the original offset corrected value.

#5 MSB	#5 LSB

+

#1 MSB	#1 LSB

=

#6 MSB	#6 LSB

7. Shift the result to the right, using the sign bit determined in Step 1, to allow for the detection of computational overflow.

S →	#6 MSB	#6 LSB

Output Data

The output format of the corrected ADC value is shown below.

MSB								LSB							
s	v	b	a	9	8	7	6	5	4	3	2	1	0	–	–

The overflow bit in the corrected output indicates that the computed value was greater than the maximum logical value (+1023) or less than the minimum logical value (–1024). Unlike the hardware overflow bit, this is not a simple binary flag. For a normal (nonoverflow) sample, the sign and the overflow bit match. If the sign bit and overflow bit do not match, a computational overflow has occurred.

Input Buffer Stage

Many applications require the measurement of an input voltage source with a high output impedance. This ADC provides a buffered input for such situations. The drawback of the buffered input is a limitation of the input range. When using unity gain buffered mode, the input signal must be prevented from coming too close to either V_{SS} or V_{DD}. See Table 139 on page 236 for details.

This condition applies only to the input voltage level (with respect to ground) of each differential input signal. The actual differential input voltage magnitude may be less than 300 mV.

The input range of the unbuffered ADC swings from V_{SS} to V_{DD}. Input signals smaller than 300 mV must use the unbuffered input mode. If these signals do not contain low output impedances, they might require off-chip buffering.

Signals outside the allowable input range can be used without instability or device damage. Any ADC readings made outside the input range are subject to greater inaccuracy than specified.

ADC Control Register Definitions

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

ADC Control Register 0 (ADCCTL0): see page 134

ADC Control/Status Register 1 (ADCCTL1): see page 136

ADC Data High Byte Register (ADCD_H): see page 137

ADC Data Low Byte Register (ADCD_L): see page 137

ADC Control Register 0

The ADC Control Register 0 (ADCCTL0) selects the analog input channel and initiates the analog-to-digital conversion. It also selects the voltage reference configuration.

Table 73. ADC Control Register 0 (ADCCTL0)

Bit	7	6	5	4	3	2	1	0
Field	CEN	REFSELL	REFOUT	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	**Conversion Enable** 0 = Conversion is complete. Writing a 0 produces no effect. The ADC automatically clears this bit to 0 when a conversion is complete. 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.
[6] REFSELL	**Voltage Reference Level Select Low Bit** 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}; note that this reference is independent of the Comparator reference. 00 = Internal Reference Disabled, reference comes from external pin. 01 = Internal Reference set to 1.0 V. 10 = Internal Reference set to 2.0 V (default). 11 = Reserved.
[5] REFOUT	**Internal Reference Output Enable** 0 = Reference buffer is disabled; Vref pin is available for GPIO or analog functions. 1 = The internal ADC reference is buffered and driven out to the V_{REF} pin. **Caution:** When the ADC is used with an external reference ({REFSELH,REFSELL}=00), the REFOUT bit must be set to 0.
[4] CONT	**Conversion** 0 = Single-shot conversion. ADC data is output once at completion of the 5129 system clock cycles (measurements of the internal temperature sensor take twice as long). 1 = Continuous conversion. ADC data updated every 256 system clock cycles after an initial 5129 clock conversion (measurements of the internal temperature sensor take twice as long).
[3:0] ANAIN[3:0]	**Analog Input Select** These bits select the analog input for conversion. Not all Port pins in this list are available in all packages for the Z8 Encore! XP F082A Series. For information about port pins available with each package style, see the Pin Description chapter on page 8. Do not enable unavailable analog inputs. Usage of these bits changes depending on the buffer mode selected in ADC Control/Status Register 1.

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.

SINGLE-ENDED Mode:

0000 = ANA0 (transimpedance amp output when enabled)
0001 = ANA1 (transimpedance amp inverting input)
0010 = ANA2 (transimpedance amp noninverting input)
0011 = ANA3
0100 = ANA4
0101 = ANA5
0110 = ANA6
0111 = ANA7
1000 = Reserved
1001 = Reserved
1010 = Reserved
1011 = Reserved
1100 = Hold transimpedance input nodes (ANA1 and ANA2) to ground.
1101 = Reserved
1110 = Temperature Sensor.
1111 = Reserved.

DIFFERENTIAL Mode (noninverting input and inverting input respectively):

0000 = ANA0 and ANA1
0001 = ANA2 and ANA3
0010 = ANA4 and ANA5
0011 = ANA1 and ANA0
0100 = ANA3 and ANA2
0101 = ANA5 and ANA4
0110 = ANA6 and ANA5
0111 = ANA0 and ANA2
1000 = ANA0 and ANA3
1001 = ANA0 and ANA4
1010 = ANA0 and ANA5
1011 = Reserved
1100 = Reserved
1101 = Reserved
1110 = Reserved
1111 = Manual Offset Calibration Mode

ADC Control/Status Register 1

The ADC Control/Status Register 1 (ADCCTL1) configures the input buffer stage, enables the threshold interrupts and contains the status of both threshold triggers. It is also used to select the voltage reference configuration.

Table 74. ADC Control/Status Register 1 (ADCCTL1)

Bit	7	6	5	4	3	2	1	0
Field	REFSELH	Reserved				BUFMODE[2:0]		
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	**Voltage Reference Level Select High Bit** In conjunction with the Low bit (REFSELL) in ADC Control Register 0, 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. 00= Internal Reference Disabled, reference comes from external pin. 01= Internal Reference set to 1.0V. 10= Internal Reference set to 2.0V (default). 11= Reserved.
[6:3]	**Reserved** These bits are reserved and must be programmed to 0000.
[2:0] BUFMODE[2:0]	**Input Buffer Mode Select** 000 = Single-ended, unbuffered input. 001 = Single-ended, buffered input with unity gain. 010 = Reserved. 011 = Reserved. 100 = Differential, unbuffered input. 101 = Differential, buffered input with unity gain. 110 = Reserved. 111 = Reserved.

ADC Data High Byte Register

The ADC Data High Byte (ADCD_H) Register contains the upper eight bits of the ADC output. The output is an 13-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 75. 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							
X = Undefined.								

Bit	Description
[7:0] ADCDH	**ADC Data High Byte** 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 Byte Register

The ADC Data Low Byte (ADCD_L) Register contains the lower bits of the ADC output plus an overflow status bit. The output is a 13-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 76. ADC Data Low Byte 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							
X = Undefined.								

Bit	Description
[7:3] ADCDL	**ADC Data Low Bits** These bits are the least significant five bits of the 13-bits of the ADC output. These bits are undefined after a Reset.

Bit	Description (Continued)
[2:1]	**Reserved** These bits are reserved and must be undefined.
[0] OVF	**Overflow Status** 0 = A hardware overflow did not occur in the ADC for the current sample. 1= A hardware overflow did occur in the ADC for the current sample, therefore the current sample is invalid.

Low Power Operational Amplifier

The LPO is a general-purpose low power operational amplifier. Each of the three ports of the amplifier is accessible from the package pins. The LPO contains only one pin configuration: ANA0 is the output/feedback node, ANA1 is the inverting input and ANA2 is the noninverting input.

Operation

To use the LPO, it must be enabled in the Power Control Register 0 (PWRCTL0). The default state of the LPO is OFF. To use the LPO, the LPO bit must be cleared by turning it ON (for details, see the Power Control Register 0 section on page 33). When making normal ADC measurements on ANA0 (i.e., measurements not involving the LPO output), the LPO bit must be turned OFF. Turning the LPO bit ON interferes with normal ADC measurements.

Caution: The LPO bit enables the amplifier even in Stop Mode. If the amplifier is not required in Stop Mode, disable it. Failing to perform this results in Stop Mode currents higher than necessary.

As with other ADC measurements, any pins used for analog purposes must be configured as such in the GPIO registers. See the Port A–D Alternate Function Subregisters section on page 47 for details.

LPO output measurements are made on ANA0, as selected by the `ANAIN[3:0]` bits of ADC Control Register 0. It is also possible to make single-ended measurements on ANA1 and ANA2 while the amplifier is enabled, which is often useful for determining offset conditions. Differential measurements between ANA0 and ANA2 may be useful for noise cancellation purposes.

If the LPO output is routed to the ADC, then the `BUFFMODE`[2:0] bits of ADC Control/Status Register 1 must also be configured for unity-gain buffered operation. Sampling the LPO in an unbuffered mode is not recommended.

When either input is overdriven, the amplifier output saturates at the positive or negative supply voltage. No instability results.

Comparator

The Z8 Encore! XP F082A Series devices feature a general purpose comparator that compares two analog input signals. These analog signals may be external stimulus from a pin (CINP and/or CINN) or internally generated signals. Both a programmable voltage reference and the temperature sensor output voltage are available internally. The output is available as an interrupt source or can be routed to an external pin.

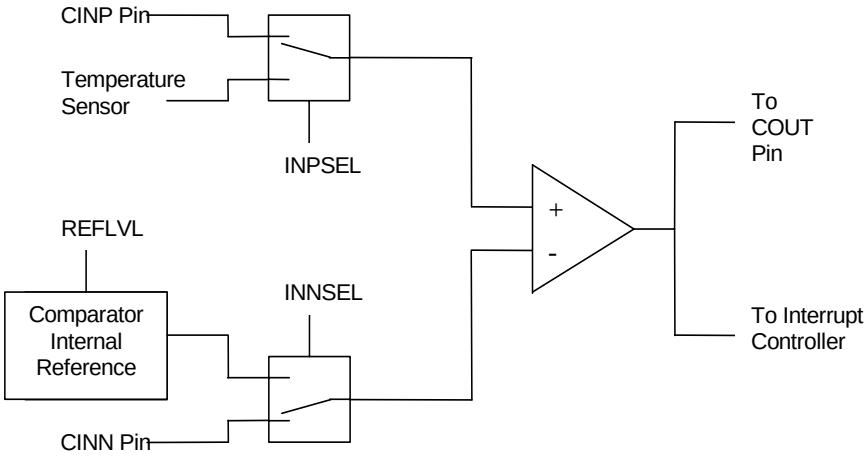


Figure 20. Comparator Block Diagram

Operation

When the positive comparator input exceeds the negative input by more than the specified hysteresis, the output is a logic High. When the negative input exceeds the positive by more than the hysteresis, the output is a logic Low. Otherwise, the comparator output retains its present value. See Table 141 on page 238 for details.

The comparator may be powered down to reduce supply current. See the Power Control Register 0 section on page 33 for details.

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 code example illustrates how to safely enable the comparator:

```
di
ld cmp0, r0 ; load some new configuration
nop
nop      ; wait for output to settle
clr irq0 ; clear any spurious interrupts pending
ei
```

Comparator Control Register Definition

The Comparator Control Register (CMP0) configures the comparator inputs and sets the value of the internal voltage reference.

Table 77. Comparator Control Register (CMP0)

Bit	7	6	5	4	3	2	1	0
Field	INPSEL	INNSEL	REFLVL				Reserved (20-/28-pin) REFLVL (8-pin)	
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.

Bit	Description (Continued)
[5:2] REFLVL	**Internal Reference Voltage Level** This reference is independent of the ADC voltage reference. **Note:** 8-pin devices contain two additional LSBs for increased resolution. For 20-/28-pin devices: 0000 = 0.0 V 0001 = 0.2 V 0010 = 0.4 V 0011 = 0.6 V 0100 = 0.8 V 0101 = 1.0 V (Default) 0110 = 1.2 V 0111 = 1.4 V 1000 = 1.6 V 1001 = 1.8 V 1010–1111 = Reserved

Bit	Description (Continued)
[1:0]	For 8-pin devices, the following voltages can be configured; for 20- and 28-pin devices, these bits are reserved. 000000 = 0.00 V 000001 = 0.05 V 000010 = 0.10 V 000011 = 0.15 V 000100 = 0.20 V 000101 = 0.25 V 000110 = 0.30 V 000111 = 0.35 V 001000 = 0.40 V 001001 = 0.45 V 001010 = 0.50 V 001011 = 0.55 V 001100 = 0.60 V 001101 = 0.65 V 001110 = 0.70 V 001111 = 0.75 V 010000 = 0.80 V 010001 = 0.85 V 010010 = 0.90 V 010011 = 0.95 V 010100 = 1.00 V (Default) 010101 = 1.05 V 010110 = 1.10 V 010111 = 1.15 V 011000 = 1.20 V 011001 = 1.25 V 011010 = 1.30 V 011011 = 1.35 V 011100 = 1.40 V 011101 = 1.45 V 011110 = 1.50 V 011111 = 1.55 V 100000 = 1.60 V 100001 = 1.65 V 100010 = 1.70 V 100011 = 1.75 V 100100 = 1.80 V

Temperature Sensor

The on-chip Temperature Sensor allows you to measure temperature on the die with either the on-board ADC or on-board comparator. This block is factory calibrated for in-circuit software correction. Uncalibrated accuracy is significantly worse, therefore the temperature sensor is not recommended for uncalibrated use.

Temperature Sensor Operation

The on-chip temperature sensor is a Proportional to Absolute Temperature (PTAT) topology. A pair of Flash option bytes contain the calibration data. The temperature sensor can be disabled by a bit in the Power Control Register 0 section on page 33 to reduce power consumption.

The temperature sensor can be directly read by the ADC to determine the absolute value of its output. The temperature sensor output is also available as an input to the comparator for threshold type measurement determination. The accuracy of the sensor when used with the comparator is substantially less than when measured by the ADC.

If the temperature sensor is routed to the ADC, the ADC must be configured in unity-gain buffered mode (for details, see the Input Buffer Stage section on page 133). The value read back from the ADC is a signed number, although it is always positive.

The sensor is factory-trimmed through the ADC using the external 2.0 V reference. Unless the sensor is retrimmed for use with a different reference, it is most accurate when used with the external 2.0 V reference.

Because this sensor is an on-chip sensor, Zilog recommends that the user account for the difference between ambient and die temperature when inferring ambient temperature conditions.

During normal operation, the die undergoes heating that causes a mismatch between the ambient temperature and that measured by the sensor. For best results, the Z8 Encore! XP device must be placed into Stop Mode for sufficient time such that the die and ambient temperatures converge (this time is dependent on the thermal design of the system). The temperature sensor measurement must then be made immediately after recovery from Stop Mode.

The following equation defines the transfer function between the temperature sensor output voltage and the die temperature. This is needed for comparator threshold measurements.

$$V = 0.01 \times T + 0.65$$

In the above equation, T is the temperature in °C; V is the sensor output in volts.

Assuming a compensated ADC measurement, the following equation defines the relationship between the ADC reading and the die temperature:

$$T = (25/128) \times (ADC - TSCAL[11:2]) + 30$$

In the above equation, T is the temperature in C; ADC is the 10-bit compensated ADC value; and TSCAL is the temperature sensor calibration value, ignoring the two least significant bits of the 12-bit value.

See the Temperature Sensor Calibration Data section on page 171 for the location of TSCAL.

Calibration

The temperature sensor undergoes calibration during the manufacturing process and is maximally accurate at 30°C. Accuracy decreases as measured temperatures move further from the calibration point.

Flash Memory

The products in the Z8 Encore! XP F082A Series feature a nonvolatile Flash memory of 8KB (8192), 4 KB (4096), 2 KB (2048 bytes), or 1KB (1024) with read/write/erase capability. The Flash Memory can be programmed and erased in-circuit by user code or through the On-Chip Debugger. The features include:

- User controlled read and write protect capability
- Sector-based write protection scheme
- Additional protection schemes against accidental program and erasure

Architecture

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 or data protection, the Flash memory is also divided into sectors. In the Z8 Encore! XP F082A Series, these sectors are either 1024 bytes (in the 8KB devices) or 512 bytes (all other memory sizes) in size. Page and sector sizes are not generally equal.

The first 2 bytes of Flash Program memory are used as Flash option bits. For more information about their operation, see the Flash Option Bits chapter on page 159.

Table 78 describes the Flash memory configuration for each device in the Z8 Encore! XP F082A Series. Figure 21 displays the Flash memory arrangement.

Table 78. Z8 Encore! XP F082A Series Flash Memory Configurations

Part Number	Flash Size KB (Bytes)	Flash Pages	Program Memory Addresses	Flash Sector Size (Bytes)
Z8F08xA	8 (8192)	16	0000H–1FFFH	1024
Z8F04xA	4 (4096)	8	0000H–0FFFH	512
Z8F02xA	2 (2048)	4	0000H–07FFH	512
Z8F01xA	1 (1024)	2	0000H–03FFH	512

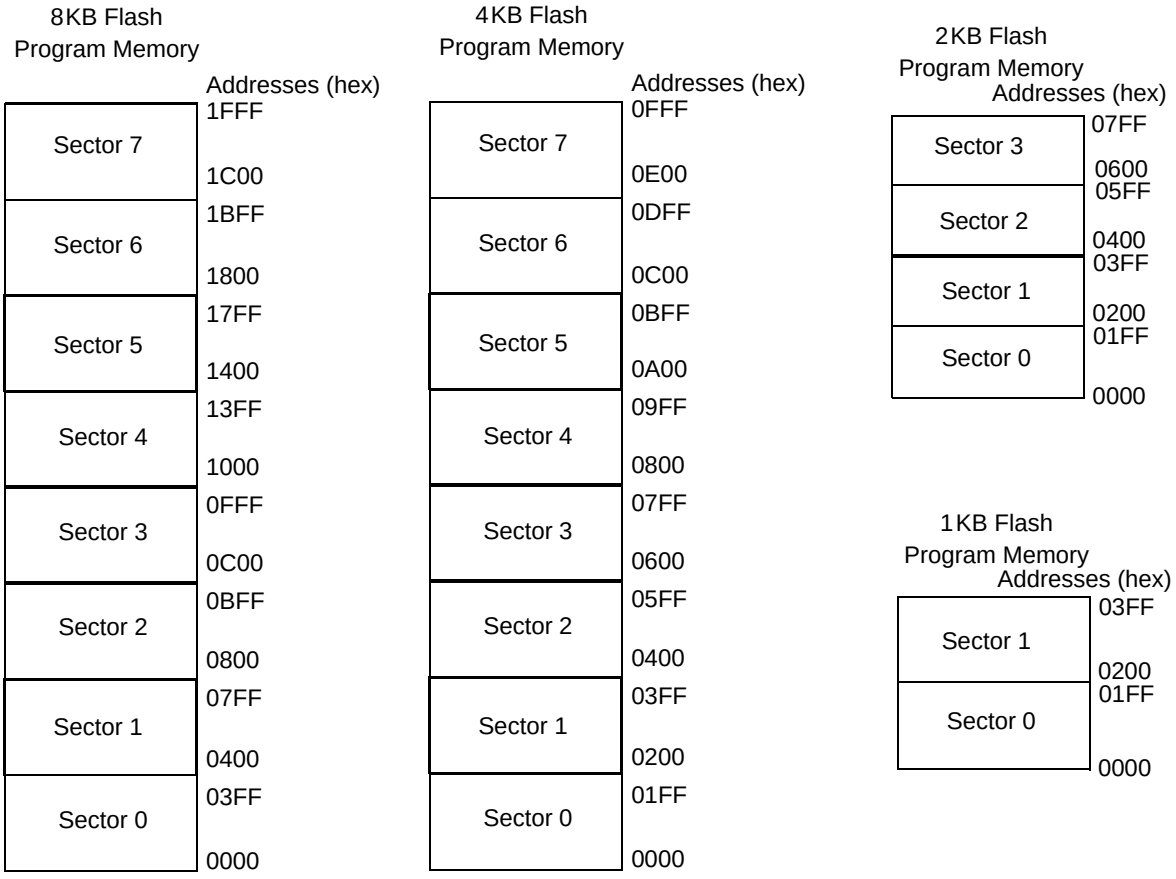


Figure 21. Flash Memory Arrangement

Flash Information Area

The Flash information area is separate from Program Memory and is mapped to the address range FE00H to FFFFH. This area is readable but cannot be erased or overwritten. 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 22 displays a basic Flash Controller flow. The following subsections provide details about the various operations displayed in Figure 22.

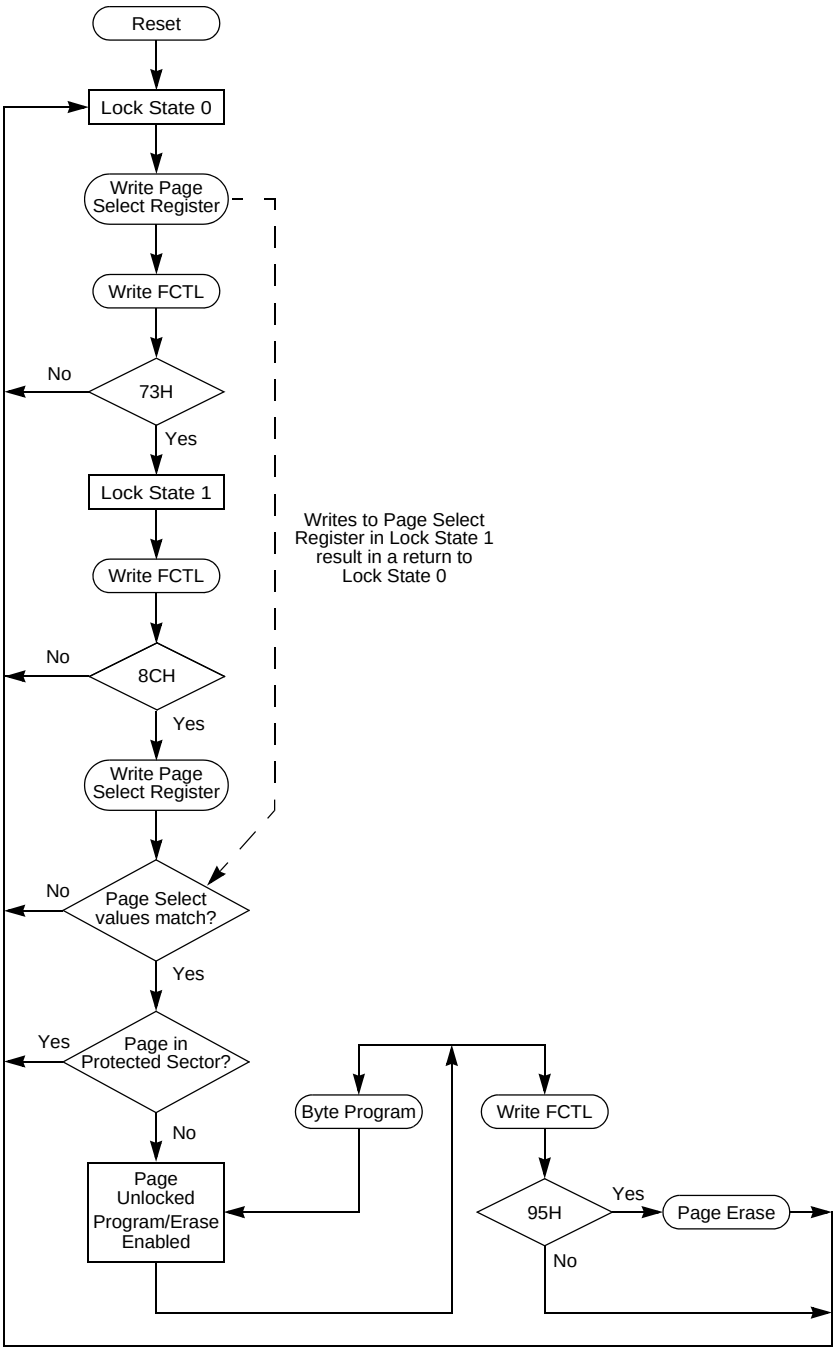


Figure 22. Flash Controller Operation Flow Chart

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 (32768Hz) or above 20MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure operation of the Z8 Encore! XP F082A Series devices.

Flash Code Protection Against External Access

The user code contained within the Flash memory can be protected against external access by the on-chip debugger. Programming the FRP Flash option bit prevents reading of the user code with the On-Chip Debugger. See the Flash Option Bits chapter on page 159 and the On-Chip Debugger chapter on page 180 for more information.

Flash Code Protection Against Accidental Program and Erasure

The Z8 Encore! XP F082A 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 shown in Table 79. See the Flash Option Bits chapter on page 159 for more information.

Table 79. 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 target page. 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. See Figure 22 on page 148 for details.

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! F082A Series devices, the sector size is varied according to the Flash memory configuration shown in Table 78 on page 146.

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 Regis-

ter 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 is no longer 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

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 completed, 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 can be accomplished using the On-Chip Debugger's Write Memory command or eZ8 CPU execution of the LDC or LDCI instructions. Refer to the eZ8 CPU Core User Manual (UM0128), available for download on www.zilog.com, for a description of the LDC and LDCI instructions. 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 Third-Party Flash Programming Support for Z8 Encore! MCUs Application Note (AN0117), which is available for download on 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 go through the unlock sequence again to select another page.

Flash Control Register Definitions

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

Flash Control Register: see page 153

Flash Status Register: see page 155

Flash Page Select Register: see page 156

Flash Sector Protect Register: see page 157

Flash Frequency High and Low Byte Registers: see page 157

Flash Control Register

The Flash Controller must be unlocked using the Flash Control (FCTL) 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 80. 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] FCMD	**Flash Command** 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 (FSTAT) 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 81. 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]	These bits are reserved and must be programmed to 00.
[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 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 82. 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	**Information Area Enable** 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.
[6:0] PAGE	**Page Select** This 7-bit field identifies the Flash memory page for Page Erase and page unlocking. Program Memory Address[15:9] = PAGE[6:0]. For the Z8F08xx devices, the upper 3 bits must be zero. For the Z8F04xx devices, the upper 4 bits must be zero. For Z8F02xx devices, the upper 5 bits must always be 0. For the Z8F01xx 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 83. 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
[7:0] SPROT*n*	**Sector Protection** 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 Z8F082A Series product, please refer to Table 78 on page 146 and to Figure 21, which follows the table. • For Z8F08xA and Z8F04xA devices, all bits are used. • For Z8F02xA devices, the upper 4 bits are unused. • For Z8F01xA devices, the upper 6 bits are unused.

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 20 kHz or above 20 MHz.

Table 84. 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] FFREQH	**Flash Frequency High Byte** High byte of the 16-bit Flash Frequency value.

Table 85. 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] FFREQL	**Flash Frequency Low Byte** Low byte of the 16-bit Flash Frequency value.

Flash Option Bits

Programmable Flash option bits allow user configuration of certain aspects of Z8 Encore! XP F082A Series operation. The feature configuration data is stored in Flash program memory and loaded into holding registers during Reset. The features available for control through the Flash option bits include:

- Watchdog Timer time-out response selection–interrupt or system reset
- Watchdog Timer always on (enabled at Reset)
- The ability to prevent unwanted read access to user code in Program Memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory
- Voltage Brown-Out configuration-always enabled or disabled during Stop Mode to reduce Stop Mode power consumption
- Oscillator mode selection-for high, medium and low power crystal oscillators, or external RC oscillator
- Factory trimming information for the internal precision oscillator and low voltage detection
- Factory calibration values for ADC, temperature sensor and Watchdog Timer compensation
- Factory serialization and randomized lot identifier (optional)

Operation

This section describes the type and configuration of the programmable Flash option bits.

Option Bit Configuration By Reset

Each time the Flash option bits are programmed or erased, the device must be Reset for the change to take effect. During any reset operation (System Reset, Power-On Reset, or Stop Mode Recovery), the Flash option bits are automatically read from Flash program memory and written to the Option Configuration registers. The Option Configuration registers control the operation of the devices within the Z8 Encore! XP F082A Series. Option bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access.

Option Bit Types

This section describes the five types of Flash option bits.

User Option Bits

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

Trim Option Bits

The trim option bits are contained in the information page of the Flash memory. 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 See the Flash Information Area section on page 17.

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 and are unaffected by mass erasure of the device's Flash memory. See the Reading the Flash Information Page section below and the Serialization Data section on page 173 for more details.

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 and is unaffected by mass erasure of the device's Flash memory. See Reading the Flash Information Page, below, and the Randomized Lot Identifier section on page 174 for more details.

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 86).

Table 86. 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 87).

Table 87. 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.

Flash Program Memory Address 0000H

Table 88. Flash Option Bits at Program Memory Address 0000H

Bit	7	6	5	4	3	2	1	0
Field	WDT_RES	WDT_AO	OSC_SEL[1:0]		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.

Bit	Description (Continued)
[5:4] OSC_SEL[1:0]	**Oscillator Mode Selection** 00 = On-chip oscillator configured for use with external RC networks (<4MHz). 01 = Minimum power for use with very low frequency crystals (32kHz to 1.0MHz). 10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5MHz to 5.0MHz). 11 = Maximum power for use with high frequency crystals (5.0MHz to 20.0MHz). This setting is the default for unprogrammed (erased) Flash.
[3] VBO_AO	**Voltage Brown-Out Protection Always On** 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 Definitions section on page 33). 1 = Voltage Brown-Out Protection is always enabled including during Stop Mode. This setting is the default for unprogrammed (erased) Flash.
[2] FRP	**Flash Read Protect** 0 = User program code is inaccessible. Limited control features are available through the On-Chip Debugger. 1 = User program code is accessible. All On-Chip Debugger commands are enabled. This setting is the default for unprogrammed (erased) Flash.
[1]	**Reserved** This bit is reserved and must be programmed to 1.
[0] FWP	**Flash Write Protect** This Option Bit provides Flash Program Memory protection: 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. 1 = Programming, Page Erase and Mass Erase are enabled for all of Flash program memory.

Flash Program Memory Address 0001H

Table 89. 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.
[4] XTLDIS	**State of the Crystal Oscillator at Reset** This bit only enables the crystal oscillator. Its selection as a system clock must be performed manually. 0 = Crystal oscillator is enabled during reset, resulting in longer reset timing. 1 = Crystal oscillator is disabled during reset, resulting in shorter reset timing. **Caution:** Programming the XTLDIS bit to zero on 8-pin versions of this device prevents any further communication via the debug pin due to the fact that the X_{IN} and DBG functions are shared on pin 2 of this package. Do not program this bit to zero on 8-pin devices unless further debugging or Flash programming is not required.
[3:0]	**Reserved** These bits are reserved and must be programmed to 1111.

Trim Bit Address Space

All available Trim bit addresses and their functions are listed in Table 90 through Table 95.

Trim Bit Address 0000H

Table 90. 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.

Trim Bit Address 0001H

Table 91. 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.

Trim Bit Address 0002H

Table 92. 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] IPO_TRIM	**Internal Precision Oscillator Trim Byte** Contains trimming bits for the Internal Precision Oscillator.

Trim Bit Address 0003H

Note: The LVD is available on 8-pin devices only.

Table 93. Trim Option Bits at Address 0003H (TLVD)

Bit	7	6	5	4	3	2	1	0
Field	Reserved			LVD_TRIM				
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 0023H							

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

Bit	Description
[7:5]	**Reserved** These bits are reserved and must be programmed to 111.
[4:0] LVD_TRIM	**Low Voltage Detect Trimm** This trimming affects the low voltage detection threshold. Each LSB represents a 50mV change in the threshold level. Alternatively, the low voltage threshold may be computed from the options bit value by the following equation: $LVD_LVL = 3.6\ V - LVD_TRIM \times 0.05\ V$ These values are tabulated in Table 94.

Table 94. LVD Trim Values

LVD_TRIM	LVD Threshold (V) Typical	Description
00000	3.60	Maximum LVD threshold
00001	3.55	
00010	3.50	
00011	3.45	
00100	3.40	
00101	3.35	
00110	3.30	
00111	3.25	
01000	3.20	
01001	3.15	
01010	3.10	Default on Reset
01011	3.05	
01100	3.00	
01101	2.95	
01110	2.90	
01111	2.85	
10000	2.80	
10001	2.75	
10010	2.70	
10011 to 11111	2.70 to 1.65	Minimum LVD threshold

Trim Bit Address 0004H

Table 95. Trim Option Bits at 0004H

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 0024H							

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.

Zilog Calibration Data

This section briefly describes the features of the following Flash option bit calibration registers.

ADC Calibration Data: see page 169

Temperature Sensor Calibration Data: see page 171

Watchdog Timer Calibration Data: see page 172

Serialization Data: see page 173

Randomized Lot Identifier: see page 174

ADC Calibration Data

Table 96. 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 described in the Software Compensation Procedure Using Factory Calibration Data section on page 129. The location of each calibration byte is provided in Table 97.

Table 97. ADC Calibration Data Location

Info Page Address	Memory Address	Compensation Usage	ADC Mode	Reference Type
60	FE60	Offset	Single-Ended Unbuffered	Internal 2.0 V
08	FE08	Gain High Byte	Single-Ended Unbuffered	Internal 2.0 V
09	FE09	Gain Low Byte	Single-Ended Unbuffered	Internal 2.0 V
63	FE63	Offset	Single-Ended Unbuffered	Internal 1.0 V
0A	FE0A	Gain High Byte	Single-Ended Unbuffered	Internal 1.0 V
0B	FE0B	Gain Low Byte	Single-Ended Unbuffered	Internal 1.0 V
66	FE66	Offset	Single-Ended Unbuffered	External 2.0 V
0C	FE0C	Gain High Byte	Single-Ended Unbuffered	External 2.0 V
0D	FE0D	Gain Low Byte	Single-Ended Unbuffered	External 2.0 V
69	FE69	Offset	Single-Ended 1x Buffered	Internal 2.0 V
0E	FE0E	Gain High Byte	Single-Ended 1x Buffered	Internal 2.0 V
0F	FE0F	Gain Low Byte	Single-Ended 1x Buffered	Internal 2.0 V
6C	FE6C	Offset	Single-Ended 1x Buffered	External 2.0 V
10	FE10	Gain High Byte	Single-Ended 1x Buffered	External 2.0 V
11	FE11	Gain Low Byte	Single-Ended 1x Buffered	External 2.0 V
6F	FE6F	Offset	Differential Unbuffered	Internal 2.0 V

Table 97. ADC Calibration Data Location (Continued)

Info Page Address	Memory Address	Compensation Usage	ADC Mode	Reference Type
12	FE12	Positive Gain High Byte	Differential Unbuffered	Internal 2.0 V
13	FE13	Positive Gain Low Byte	Differential Unbuffered	Internal 2.0 V
30	FE30	Negative Gain High Byte	Differential Unbuffered	Internal 2.0 V
31	FE31	Negative Gain Low Byte	Differential Unbuffered	Internal 2.0 V
72	FE72	Offset	Differential Unbuffered	Internal 1.0 V
14	FE14	Positive Gain High Byte	Differential Unbuffered	Internal 1.0 V
15	FE15	Positive Gain Low Byte	Differential Unbuffered	Internal 1.0 V
32	FE32	Negative Gain High Byte	Differential Unbuffered	Internal 1.0 V
33	FE33	Negative Gain Low Byte	Differential Unbuffered	Internal 1.0 V
75	FE75	Offset	Differential Unbuffered	External 2.0 V
16	FE16	Positive Gain High Byte	Differential Unbuffered	External 2.0 V
17	FE17	Positive Gain Low Byte	Differential Unbuffered	External 2.0 V
34	FE34	Negative Gain High Byte	Differential Unbuffered	External 2.0 V
35	FE35	Negative Gain Low Byte	Differential Unbuffered	External 2.0 V
78	FE78	Offset	Differential 1x Buffered	Internal 2.0 V
18	FE18	Positive Gain High Byte	Differential 1x Buffered	Internal 2.0 V
19	FE19	Positive Gain Low Byte	Differential 1x Buffered	Internal 2.0 V
36	FE36	Negative Gain High Byte	Differential 1x Buffered	Internal 2.0 V
37	FE37	Negative Gain Low Byte	Differential 1x Buffered	Internal 2.0 V
7B	FE7B	Offset	Differential 1x Buffered	External 2.0 V
1A	FE1A	Positive Gain High Byte	Differential 1x Buffered	External 2.0 V
1B	FE1B	Positive Gain Low Byte	Differential 1x Buffered	External 2.0 V
38	FE38	Negative Gain High Byte	Differential 1x Buffered	External 2.0 V
39	FE39	Negative Gain Low Byte	Differential 1x Buffered	External 2.0 V

Temperature Sensor Calibration Data

Table 98. Temperature Sensor Calibration High Byte at 003A (TSCALH)

Bit	7	6	5	4	3	2	1	0
Field	TSCALH							
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 003A							

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

Bit	Description
[7:0] TSCALH	**Temperature Sensor Calibration High Byte** The TSCALH and TSCALL bytes combine to form the 12-bit temperature sensor offset calibration value. For more details, see Temperature Sensor Operation on page 139.

Table 99. Temperature Sensor Calibration Low Byte at 003B (TSCALL)

Bit	7	6	5	4	3	2	1	0
Field	TSCALL							
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 003B							

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

Bit	Description
[7:0] TSCALL	**Temperature Sensor Calibration Low Byte** The TSCALH and TSCALL bytes combine to form the 12-bit temperature sensor offset calibration value. For usage details, see the Temperature Sensor Operation section on page 144.

Watchdog Timer Calibration Data

Table 100. Watchdog Calibration High Byte at 007EH (WDTCALH)

Bit	7	6	5	4	3	2	1	0
Field	WDTCALH							
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 007EH							

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

Bit	Description
[7:0] WDTCALH	**Watchdog Timer Calibration High Byte** The WDTCALH and WDTCALL bytes, when loaded into the Watchdog Timer reload registers result in a one second time-out at room temperature and 3.3V supply voltage. To use the Watchdog Timer calibration, user code must load WDTU with 0x00, WDTH with WDTCALH and WDTL with WDTCALL.

Table 101. Watchdog Calibration Low Byte at 007FH (WDTCALL)

Bit	7	6	5	4	3	2	1	0
Field	WDTCALL							
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 007FH							

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

Bit	Description
[7:0] WDTCALL	**Watchdog Timer Calibration Low Byte** The WDTCALH and WDTCALL bytes, when loaded into the Watchdog Timer reload registers result in a one second time-out at room temperature and 3.3V supply voltage. To use the Watchdog Timer calibration, user code must load WDTU with 0x00, WDTH with WDTCALH and WDTL with WDTCALL.

Serialization Data

Table 102. 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 103.

Table 103. 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 104. 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 105.

Table 105. 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.

Table 105. Randomized Lot ID Locations (Continued)

Info Page Address	Memory Address	Usage
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).

Nonvolatile Data Storage

The Z8 Encore! XP F082A Series devices contain a nonvolatile data storage (NVDS) element of up to 128 bytes. This memory can perform over 100,000 write cycles.

Operation

The NVDS is implemented by special purpose Zilog software stored in areas of program memory, which are not user-accessible. These special-purpose routines use the Flash memory to store the data. The routines incorporate a dynamic addressing scheme to maximize the write/erase endurance of the Flash.

Note: Different members of the Z8 Encore! XP F082A Series feature multiple NVDS array sizes; see the Part Selection Guide section on page 2 for details. Devices containing 8KB of Flash memory do not include the NVDS feature.

NVDS Code Interface

Two routines are required to access the NVDS: a write routine and a read routine. Both of these routines are accessed with a CALL instruction to a predefined address outside of the user-accessible program memory. Both the NVDS address and data are single-byte values. Because these routines disturb the working register set, user code must ensure that any required working register values are preserved by pushing them onto the stack or by changing the working register pointer just prior to NVDS execution.

During both read and write accesses to the NVDS, interrupt service is NOT disabled. Any interrupts that occur during the NVDS execution must take care not to disturb the working register and existing stack contents or else the array may become corrupted. Disabling interrupts before executing NVDS operations is recommended.

Use of the NVDS requires 15 bytes of available stack space. Also, the contents of the working register set are overwritten.

For correct NVDS operation, the Flash Frequency registers must be programmed based on the system clock frequency (see the Flash Operation Timing Using the Flash Frequency Registers section on page 149).

Byte Write

To write a byte to the NVDS array, the user code must first push the address, then the data byte onto the stack. The user code issues a CALL instruction to the address of the byte-write routine (0x10B3). At the return from the sub-routine, the write status byte resides in working register R0. The bit fields of this status byte are defined in Table 106. The contents of the status byte are undefined for write operations to illegal addresses. Also, user code must pop the address and data bytes off the stack.

The write routine uses 13 bytes of stack space in addition to the two bytes of address and data pushed by the user. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS writes exhibit a nonuniform execution time. In general, a write takes 251µs (assuming a 20MHz system clock). Every 400 to 500 writes, however, a maintenance operation is necessary. In this rare occurrence, the write takes up to 61ms to complete. Slower system clock speeds result in proportionally higher execution times.

NVDS byte writes to invalid addresses (those exceeding the NVDS array size) have no effect. Illegal write operations have a 2µs execution time.

Table 106. Write Status Byte

Bit	7	6	5	4	3	2	1	0
Field	Reserved				RCPY	PF	AWE	DWE
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] RCPY	**Recopy Subroutine Executed** A recopy subroutine was executed. These operations take significantly longer than a normal write operation.
[2] PF	**Power Failure Indicator** A power failure or system reset occurred during the most recent attempted write to the NVDS array.
[1] AWE	**Address Write Error** An address byte failure occurred during the most recent attempted write to the NVDS array.
[0] DWE	**Data Write Error** A data byte failure occurred during the most recent attempted write to the NVDS array.

Byte Read

To read a byte from the NVDS array, user code must first push the address onto the stack. User code issues a `CALL` instruction to the address of the byte-read routine (`0x1000`). At the return from the sub-routine, the read byte resides in working register R0 and the read status byte resides in working register R1. The contents of the status byte are undefined for read operations to illegal addresses. Also, the user code must pop the address byte off the stack.

The read routine uses 9 bytes of stack space in addition to the one byte of address pushed by the user. Sufficient memory must be available for this stack usage.

Because of the Flash memory architecture, NVDS reads exhibit a nonuniform execution time. A read operation takes between 44 µs and 489 µs (assuming a 20 MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from invalid addresses (those exceeding the NVDS array size) return 0xff. Illegal read operations have a 2 µs execution time.

The status byte returned by the NVDS read routine is zero for successful read, as determined by a CRC check. If the status byte is nonzero, there was a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have a CRC error.

Power Failure Protection

The NVDS routines employ error checking mechanisms to ensure a power failure endangers only the most recently written byte. Bytes previously written to the array are not perturbed.

A system reset (such as a pin reset or Watchdog Timer reset) that occurs during a write operation also perturbs the byte currently being written. All other bytes in the array are unperturbed.

Optimizing NVDS Memory Usage for Execution Speed

NVDS read time can vary drastically. This discrepancy is a trade-off for minimizing the frequency of writes that require post-write page erases, as indicated in Table 107. The NVDS read time of address N is a function of the number of writes to addresses other than N since the most recent write to address N, plus the number of writes since the most recent page erase. Neglecting effects caused by page erases and results caused by the initial condition in which the NVDS is blank, a rule of thumb is that every write since the most recent page erase causes read times of unwritten addresses to increase by 1 µs up to a maximum of (511-NVDS_SIZE) µs.

Table 107. NVDS Read Time

Operation	Minimum Latency	Maximum Latency
Read (16 byte array)	875	9961
Read (64 byte array)	876	8952
Read (128 byte array)	883	7609
Write (16 byte array)	4973	5009
Write (64 byte array)	4971	5013
Write (128 byte array)	4984	5023
Illegal Read	43	43
Illegal Write	31	31

If NVDS read performance is critical to your software architecture, you can optimize your code for speed. Try the first suggestion below before attempting the second.

1. Periodically refresh all addresses that are used. The optimal use of NVDS in terms of speed is to rotate the writes evenly among all addresses planned to use, bringing all reads closer to the minimum read time. Because the minimum read time is much less than the write time, however, actual speed benefits are not always realized.
2. Use as few unique addresses as possible to optimize the impact of refreshing, plus minimize the requirement for it.

On-Chip Debugger

The Z8 Encore! XP F082A Series devices contain an integrated On-Chip Debugger (OCD) that provides advanced debugging features including:

- 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 pins available to the user (8-pin product only)

Architecture

The on-chip debugger consists of four primary functional blocks: transmitter, receiver, auto-baud detector/generator and debug controller. Figure 23 displays the architecture of the on-chip debugger.

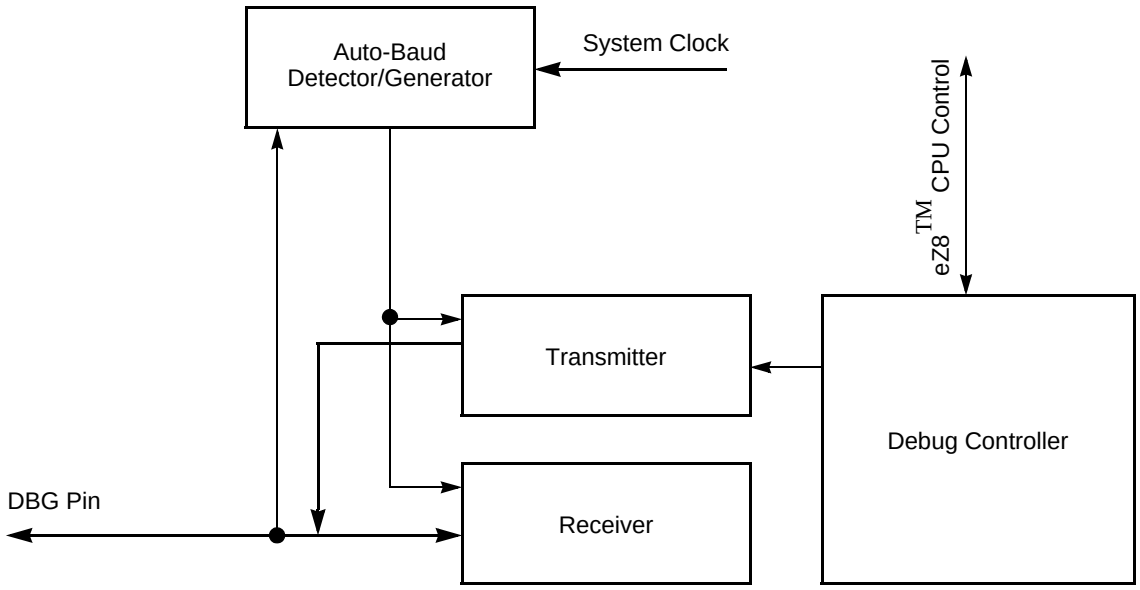


Figure 23. On-Chip Debugger Block Diagram

Operation

This section describes the interface and modes of operation of the On-Chip Debugger.

OCD Interface

The on-chip debugger 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 Z8 Encore! XP F082A 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 24 and Figure 25. The recommended method is the buffered implementation displayed in Figure 25. The DBG pin has a 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 226). For OCD operation at higher data rates or in noisy systems, an external pull-up resistor is recommended.

Caution: For operation of the on-chip debugger, 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.

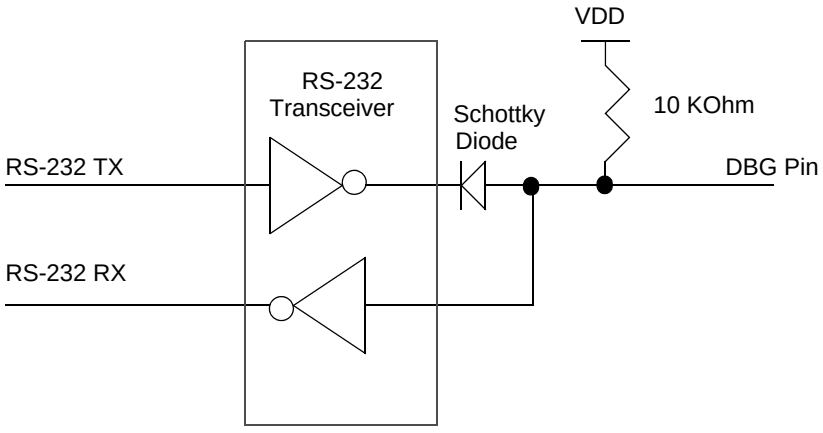


Figure 24. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #1 of 2

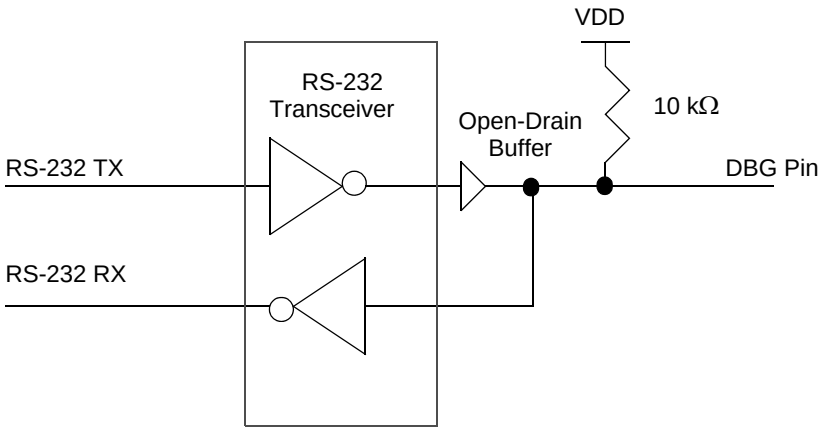


Figure 25. Interfacing the On-Chip Debugger's DBG Pin with an RS-232 Interface; #2 of 2

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 operating characteristics of the devices entering Debug Mode are:

- The device enters Debug Mode after the eZ8 CPU executes a BRK (Breakpoint) instruction
- If the DBG pin is held Low during the final clock cycle of system reset, the part enters Debug Mode immediately (20-/28-pin products only)

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 Auto-Baud Detector/Generator section on page 183).

- 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 may be used to autobaud and cause the device to enter Debug Mode. See the OCD Unlock Sequence (8-Pin Devices Only) section on page 185.

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 transmitted and received by the OCD consists of 1 Start bit, 8 data bits (least-significant bit first) and 1 Stop bit as displayed in Figure 26.

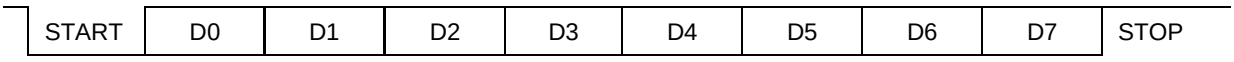


Figure 26. OCD Data Format

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 to avoid this issue.

OCD Auto-Baud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the On-Chip Debugger 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 8. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by 4, but this theoretical maximum is possible only for low noise designs with clean signals. Table 108 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 108. OCD Baud-Rate Limits

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (Kbps)	Recommended Standard PC Baud Rate (bps)	Minimum Baud Rate (Kbps)
20.0	2500.0	1,843,200	39
1.0	125.0	115,200	1.95
0.032768 (32kHz)	4.096	2,400	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 On-Chip Debugger can detect 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 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 Z8 Encore! XP F082A 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 can send 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 5ms 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-/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 186).

Caution: Between Steps 3 and 4, there is an interval during which the 8-pin device is neither in RESET nor Debug Mode. If a device has been erased or has not yet been programmed, all program memory bytes contain FFH. The CPU interprets this value as an illegal instruction; therefore some irregular behavior can occur before entering Debug Mode, and the register values after entering Debug Mode will differ from their specified reset values. However, none of these irregularities prevent the programming of Flash memory. Before beginning system debug, Zilog recommends that some legal code be programmed into the 8-pin device and that a RESET occurs.

Breakpoints

Execution Breakpoints are generated using the BRK instruction (opcode 00H). When the eZ8 CPU decodes a BRK instruction, it signals the On-Chip Debugger. 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 On-Chip Debugger contains a 16-bit Runtime Counter. It counts system clock cycles between Breakpoints. The counter starts counting when the On-Chip Debugger 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 on-chip debugger 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 the Z8 Encore! XP F082A Series device. When this option is enabled, several of the OCD commands are disabled. See Table 109.

Table 110 on page 191 is a summary of the on-chip debugger commands. Each OCD command is described in further detail in the bulleted list following this table. Table 110 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 109. Debug Command Enable/Disable

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	–

Table 109. Debug Command Enable/Disable (Continued)

Debug Command	Command Byte	Enabled when Not in Debug Mode?	Disabled by Flash Read Protect Option Bit
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 list of OCD commands that follows, data and commands sent from the host to the On-Chip Debugger are identified by DBG ← Command/Data. Data sent from the On-Chip Debugger back to the host is identified by DBG → Data.

Read OCD Revision (00H). The Read OCD Revision command determines the version of the On-Chip Debugger. 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 LDCI 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 LDCI 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 CRC command computes and returns the Cyclic Redundancy Check (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 command 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 Instruction 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 Instruction 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 On-Chip Debugger. This register is used to enter or exit Debug Mode and to enable the BRK instruction. It can also reset the Z8 Encore! XP F082A 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 110. 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	**Debug Mode** 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. 0 = The Z8 Encore! XP F082A Series device is operating in Normal Mode. 1 = The Z8 Encore! XP F082A Series device is in Debug Mode.
[6] BRKEN	**Breakpoint Enable** 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. 0 = Breakpoints are disabled. 1 = Breakpoints are enabled.

Bit	Description (Continued)
[5] DBGACK	**Debug Acknowledge** 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. 0 = Debug Acknowledge is disabled. 1 = Debug Acknowledge is enabled.
[4:1]	**Reserved** These bits are reserved and must be programmed to 0000.
[0] RST	**Reset** 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 On-Chip Debugger is not reset. This bit is automatically cleared to 0 at the end of reset. 0 = No effect. 1 = Reset the Flash Read Protect Option Bit device.

OCD Status Register

The OCD Status Register reports status information about the current state of the debugger and the system.

Table 111. 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, that allows disabling of many OCD commands. 1 = FRP bit has no effect.
[4:0]	**Reserved** These bits are reserved and must be programmed to 00000.

Oscillator Control

The Z8 Encore! XP F082A Series devices uses five possible clocking schemes, each user-selectable:

- Internal precision trimmed RC oscillator (IPO)
- On-chip oscillator using off-chip crystal or resonator
- On-chip oscillator using external RC network
- External clock drive
- On-chip low power Watchdog Timer oscillator
- Clock failure detection circuitry

In addition, Z8 Encore! XP F082A Series devices contain clock failure detection and recovery circuitry, allowing continued operation despite a failure of the system clock oscillator.

Operation

This chapter discusses the logic used to select the system clock and handle primary oscillator failures.

System Clock Selection

The oscillator control block selects from the available clocks. Table 112 details each clock source and its usage.

Table 112. Oscillator Configuration and Selection

Clock Source	Characteristics	Required Setup
Internal Precision RC Oscillator	• 32.8kHz or 5.53MHz • High accuracy • No external components required	• Unlock and write Oscillator Control Register (OSCCTL) to enable and select oscillator at either 5.53MHz or 32.8kHz
External Crystal/ Resonator	• 32kHz to 20MHz • Very high accuracy (dependent on crystal or resonator used) • Requires external components	• Configure Flash option bits for correct external oscillator mode • Unlock and write OSCCTL to enable crystal oscillator, wait for it to stabilize and select as system clock (if the XTLDIS option bit has been deasserted, no waiting is required)
External RC Oscillator	• 32kHz to 4MHz • Accuracy dependent on external components	• Configure Flash option bits for correct external oscillator mode • Unlock and write OSCCTL to enable crystal oscillator and select as system clock
External Clock Drive	• 0 to 20MHz • Accuracy dependent on external clock source	• 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	• 10kHz nominal • Low accuracy; no external components required • Very low power consumption	• 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 nonfunctioning oscillator. To prevent this condition, the oscillator control block employs a register unlocking/locking scheme.

OSC Control Register Unlocking/Locking

To write 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 system clock oscillator failure detection circuitry and the Watchdog Timer oscillator failure circuitry must be disabled. If SOFEN 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 OSCCTL Register.

The internal precision oscillator is enabled by default. If the user code changes to a different oscillator, it may be 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.

System Clock Oscillator Failure

The Z8F04xA family devices can generate nonmaskable 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 selected as the system clock oscillator. It is also unavailable if the Watchdog Timer oscillator is disabled, though it is not necessary to enable the Watchdog Timer reset function (see the Watchdog Timer chapter on page 93).

The primary oscillator failure detection circuitry asserts if the system clock frequency drops below 1 kHz ±50%. If an external signal is selected as the system oscillator, it is possible that a very slow but nonfailing clock can generate a failure condition. Under these conditions, do not enable the clock failure circuitry (SOFEN must be deasserted in the OSCCTL Register).

Watchdog Timer Failure

In the event of a Watchdog Timer oscillator failure, a similar nonmaskable 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 system clock 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 looking 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 and all functioning clock sources. In this case, the Z8 Encore! XP F082A Series device ceases functioning and can only be recovered by Power-On-Reset.

Oscillator Control Register Definitions

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. Unlock the Oscillator Control Register by writing the two-step sequence E7H followed by 18H. The register is locked at successful completion of a register write to the OSCCTL.

Table 113. Oscillator Control Register (OSCCTL)

Bit	7	6	5	4	3	2	1	0
Field	INTEN	XTLEN	WDTEN	SOFEN	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] XTLEN	**Crystal Oscillator Enable; this setting overrides the GPIO register control for PA0 and PA1** 1 = Crystal oscillator is enabled. 0 = Crystal oscillator is disabled.
[5] WDTEN	**Watchdog Timer Oscillator Enable** 1 = Watchdog Timer oscillator is enabled. 0 = Watchdog Timer oscillator is disabled.
[4] SOFEN	**System Clock Oscillator Failure Detection Enable** 1 = Failure detection and recovery of system clock oscillator is enabled. 0 = Failure detection and recovery of system clock oscillator is disabled.

Bit	Description (Continued)
[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 5.53 MHz. 001 = Internal precision oscillator functions as system clock at 32 kHz. 010 = Crystal oscillator or external RC oscillator functions as system clock. 011 = Watchdog Timer oscillator functions as system. 100 = External clock signal on PB3 functions as system clock. 101 = Reserved. 110 = Reserved. 111 = Reserved.

Crystal Oscillator

The products in the Z8 Encore! XP F082A Series contain an on-chip crystal oscillator for use with external crystals with 32kHz to 20MHz frequencies. In addition, the oscillator supports external RC networks with oscillation frequencies up to 4MHz or ceramic resonators with frequencies up to 8MHz. The on-chip crystal oscillator can be used to generate the primary system clock for the internal eZ8 CPU and the majority of the on-chip peripherals. Alternatively, the X_{IN} input pin can also accept a CMOS-level clock input signal (32kHz–20MHz). If an external clock generator is used, the X_{OUT} pin must be left unconnected. The Z8 Encore! XP F082A Series products do not contain an internal clock divider. The frequency of the signal on the X_{IN} input pin determines the frequency of the system clock.

Note: Although the X_{IN} pin can be used as an input for an external clock generator, the CLKIN pin is better suited for such use (see the System Clock Selection section on page 193).

Operating Modes

The Z8 Encore! XP F082A Series products support four oscillator modes:

- Minimum power for use with very low frequency crystals (32kHz–1MHz)
- Medium power for use with medium frequency crystals or ceramic resonators (0.5MHz to 8MHz)
- Maximum power for use with high frequency crystals (8MHz to 20MHz)
- On-chip oscillator configured for use with external RC networks (<4MHz)

The oscillator mode is selected via user-programmable Flash option bits. See the Flash Option Bits chapter on page 159 for information.

Crystal Oscillator Operation

The XTLDIS Flash option bit controls whether the crystal oscillator is enabled during reset. The crystal may later be disabled after reset if a new oscillator has been selected as the system clock. If the crystal is manually enabled after reset through the OSCCTL Register, the user code must wait at least 1000 crystal oscillator cycles for the crystal to stabilize. After this, the crystal oscillator may be selected as the system clock.

Note: The stabilization time varies depending on the crystal, resonator or feedback network used. See Table 115 for transconductance values to compute oscillator stabilization times.

Figure 27 displays a recommended configuration for connection with an external fundamental-mode, parallel-resonant crystal operating at 20 MHz. Recommended 20 MHz crystal specifications are provided in Table 114. Printed circuit board layouts must add no more than 4 pF of stray capacitance to either the X_{IN} or X_{OUT} pins. If oscillation does not occur, reduce the values of capacitors C1 and C2 to decrease loading.

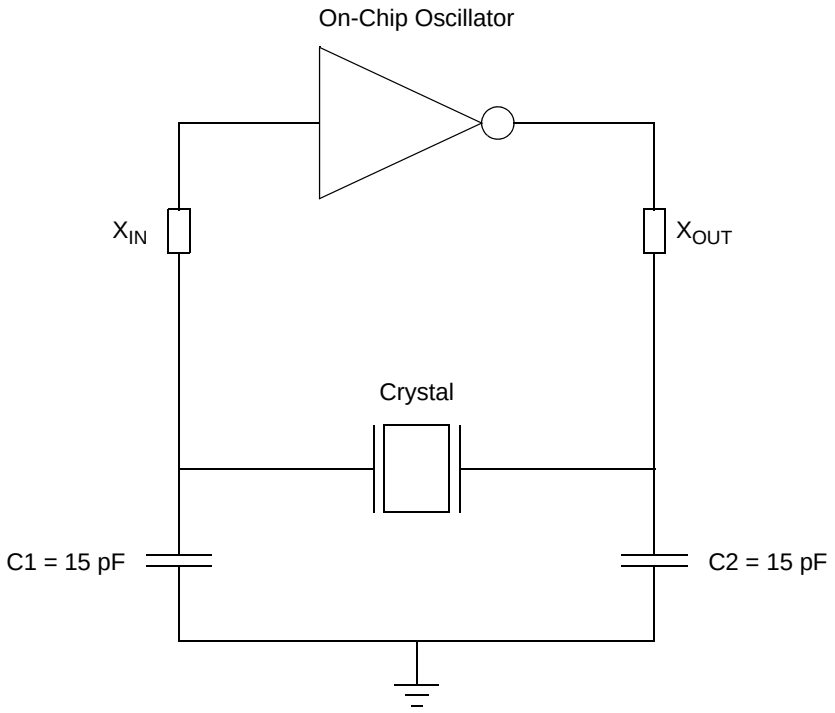


Figure 27. Recommended 20 MHz Crystal Oscillator Configuration

Table 114. Recommended Crystal Oscillator Specifications

Parameter	Value	Units	Comments
Frequency	20	MHz	
Resonance	Parallel		
Mode	Fundamental		
Series Resistance (R_S)	60	W	Maximum
Load Capacitance (C_L)	30	pF	Maximum
Shunt Capacitance (C_0)	7	pF	Maximum
Drive Level	1	mW	Maximum

Table 115. Transconductance Values for Low, Medium and High Gain Operating Modes

Mode	Crystal Frequency Range	Function	Transconductance (mA/V) (Use this range for calculations)		
Low Gain*	32kHz–1MHz	Low Power/Frequency Applications	0.02	0.04	0.09
Medium Gain*	0.5MHz–10MHz	Medium Power/Frequency Applications	0.84	1.7	3.1
High Gain*	8MHz–20MHz	High Power/Frequency Applications	1.1	2.3	4.2

Note: *Printed circuit board layouts must not add more than 4pF of stray capacitance to either the X_{IN} or X_{OUT} pins. if no oscillation occurs, reduce the values of the capacitors C1 and C2 to decrease the loading.

Oscillator Operation with an External RC Network

Figure 28 displays a recommended configuration for connection with an external resistor-capacitor (RC) network.

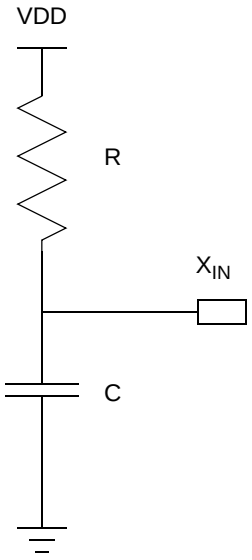


Figure 28. Connecting the On-Chip Oscillator to an External RC Network

An external resistance value of 45kΩ is recommended for oscillator operation with an external RC network. The minimum resistance value to ensure operation is 40 kΩ. The typical oscillator frequency can be estimated from the values of the resistor (*R* in kΩ) and capacitor (*C* in pF) elements using the following equation:

$$\text{Oscillator Frequency (kHz)} = \frac{1\times10^{6}}{(0.4 \times R \times C) + (4 \times C)}$$

Figure 29 displays the typical (3.3V and 25°C) oscillator frequency as a function of the capacitor (C, in pF) employed in the RC network assuming a 45KΩ external resistor. For very small values of C, the parasitic capacitance of the oscillator X_{IN} pin and the printed circuit board must be included in the estimation of the oscillator frequency.

It is possible to operate the RC oscillator using only the parasitic capacitance of the package and printed circuit board. To minimize sensitivity to external parasitics, external capacitance values in excess of 20pF are recommended.

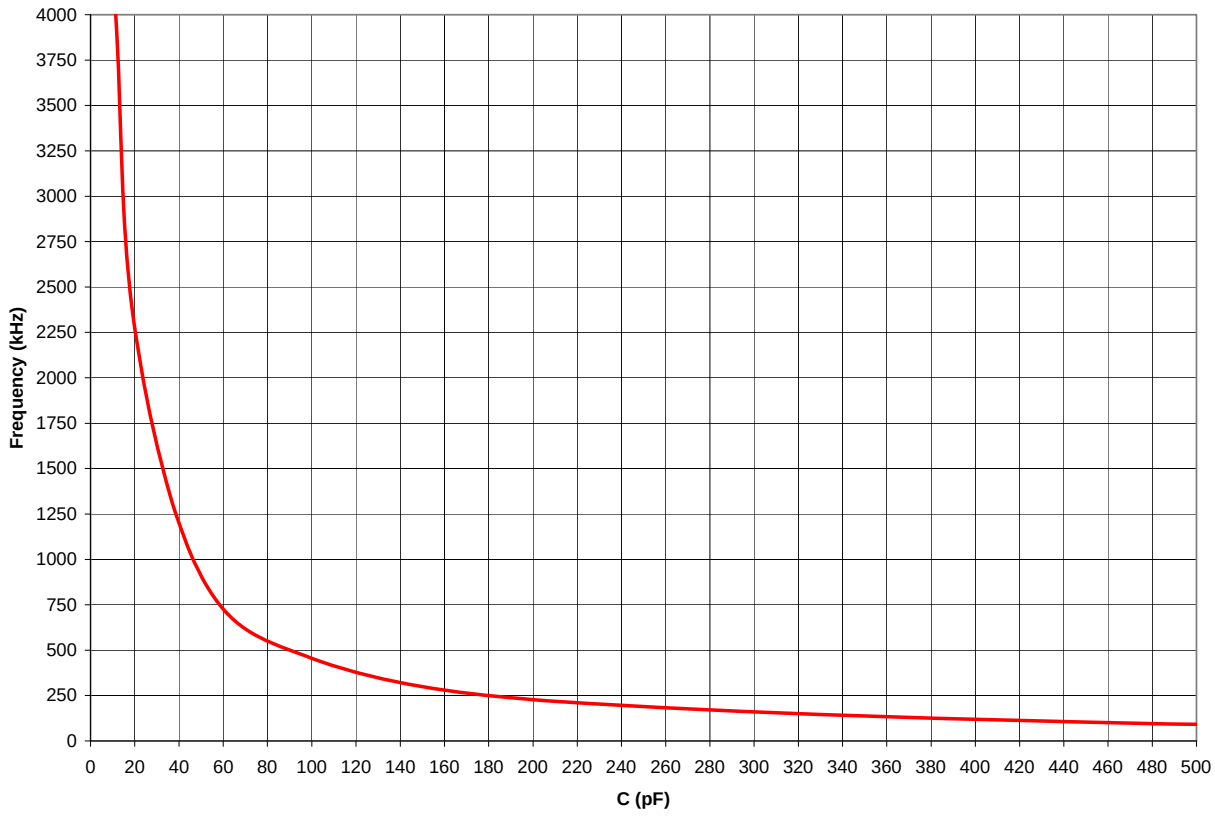


Figure 29. Typical RC Oscillator Frequency as a Function of the External Capacitance with a 45kΩ Resistor

Caution: When using the external RC oscillator mode, the oscillator can stop oscillating if the power supply drops below 2.7V, but before the power supply drops to the Voltage Brown-Out threshold. The oscillator resumes oscillation when the supply voltage exceeds 2.7V.

Internal Precision Oscillator

The internal precision oscillator (IPO) is designed for use without external components. You can either manually trim the oscillator for a nonstandard frequency or use the automatic factory-trimmed version to achieve a 5.53MHz frequency. IPO features 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)
- Trimmed through Flash option bits with user override
- Elimination of crystals or ceramic resonators in applications where very 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 +30ºC and a supply voltage of 3.3V, so accuracy of this operating point is optimal.

If not used, the IPO can be disabled by the Oscillator Control Register (see the Oscillator Control Register Definitions section on page 196).

By default, the oscillator frequency is set by the factory trim value stored in the write-protected Flash information page. However, the user code can override these trim values as described in the Trim Bit Address Space section on page 165.

Select one of two frequencies for the oscillator (5.53MHz and 32.8kHz) using the OSCSEL bits in the the Oscillator Control chapter on page 193.

eZ8 CPU Instruction Set

This chapter describes the following features of the eZ8 CPU instruction set:

Assembly Language Programming Introduction: see page 204

Assembly Language Syntax: see page 205

eZ8 CPU Instruction Notation: see page 206

eZ8 CPU Instruction Classes: see page 207

eZ8 CPU Instruction Summary: see page 212

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 can be 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. This binary format must be followed if manual program coding is preferred or if you 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:

Table 116. 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:

Table 117. Assembly Language Syntax Example 2

Assembly Language Code	ADD	43H,	R8	(ADD dst, src)
Object Code	04	E8	43	(OPC src, dst)

Register file size varies depending on the device type. See the device-specific Z8 Encore! XP Product Specification to determine the exact register file range available.

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 described in Table 118.

Table 118. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B).
cc	Condition code	—	Refer to the Condition Codes section in the eZ8 CPU Core User Manual (UM0128).
DA	Direct address	Addrs	Represents a number in the range 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.
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.

Table 118. Notational Shorthand (Continued)

Notation	Description	Operand	Range
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 119 lists additional symbols that are used throughout the Instruction Summary and Instruction Set Description sections.

Table 119. 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 can be divided functionally into the following groups:

- Arithmetic
- Bit Manipulation

- Block Transfer
- CPU Control
- Load
- Logical
- Program Control
- Rotate and Shift

Tables 120 through 127 list 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 120. 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
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 121. 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 122. 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 123. 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

Table 123. CPU Control Instructions (Continued)

Mnemonic	Operands	Instruction
RCF	—	Reset Carry Flag
SCF	—	Set Carry Flag
SRP	src	Set Register Pointer
STOP	—	Stop Mode
WDT	—	Watchdog Timer Refresh

Table 124. 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 125. 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 126. 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 127. Rotate and Shift Instructions

Mnemonic	Operands	Instruction
BSWAP	dst	Bit Swap
RL	dst	Rotate Left
RLC	dst	Rotate Left through Carry
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 128 summarizes the eZ8 CPU instructions. 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 128. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	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
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	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

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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	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	–	–	–	–	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	–	–	–	–	–	–	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
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
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	Ir	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	Ir	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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
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

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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
JR dst	PC ← PC + X	DA		8B	–	–	–	–	–	–	2	2
JR cc, dst	if cc is true PC ← PC + X	DA		0B-FB	–	–	–	–	–	–	2	2
LD dst, rc	dst ← src	r	IM	0C-FC	–	–	–	–	–	–	2	2
		r	X(r)	C7							3	3
		X(r)	r	D7							3	4
		r	Ir	E3							2	3
		R	R	E4							3	2
		R	IR	E5							3	4
		R	IM	E6							3	2
		IR	IM	E7							3	3
		Ir	r	F3							2	3
		IR	R	F5							3	3
LDC dst, src	dst ← src	r	Irr	C2	–	–	–	–	–	–	2	5
		Ir	Irr	C5							2	9
		Irr	r	D2							2	5
LDCI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	C3	–	–	–	–	–	–	2	9
		Irr	Ir	D3							2	9
LDE dst, src	dst ← src	r	Irr	82	–	–	–	–	–	–	2	5
		Irr	r	92							2	5
LDEI dst, src	dst ← src r ← r + 1 rr ← rr + 1	Ir	Irr	83	–	–	–	–	–	–	2	9
		Irr	Ir	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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
LDX dst, src	dst ← src	r	ER	84	–	–	–	–	–	–	3	2
		Ir	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	Ir	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	Ir	43							2	4
		R	R	44							3	3
		R	IR	45							3	4
		R	IM	46							3	3
		IR	IM	47							3	4

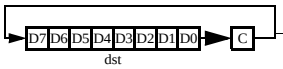
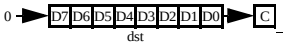
Note: Flags Notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
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
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

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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
RRC dst		R		C0	*	*	*	*	–	–	2	2
		IR		C1							2	3
SBC dst, src	dst ← dst – src - C	r	r	32	*	*	*	*	1	*	2	3
		r	Ir	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 ← dst – src - C	ER	ER	38	*	*	*	*	1	*	4	3
		ER	IM	39							4	3
SCF	C ← 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 ← src		IM	01	–	–	–	–	–	–	2	2
STOP	Stop Mode			6F	–	–	–	–	–	–	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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
SUB dst, src	dst ← dst – src	r	r	22	*	*	*	*	1	*	2	3
		r	Ir	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 ← dst – src	ER	ER	28	*	*	*	*	1	*	4	3
		ER	IM	29							4	3
SWAP dst	dst[7:4] ↔ dst[3:0]	R		F0	X	*	*	X	–	–	2	2
		IR		F1							2	3
TCM dst, src	(NOT dst) AND src	r	r	62	–	*	*	0	–	–	2	3
		r	Ir	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	Ir	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							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 128. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
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
XOR dst, src	dst ← dst XOR src	r	r	B2	–	*	*	0	–	–	2	3
		r	Ir	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 30. Figures 31 and 32 display the eZ8 CPU instructions. Table 129 lists Opcode Map abbreviations.

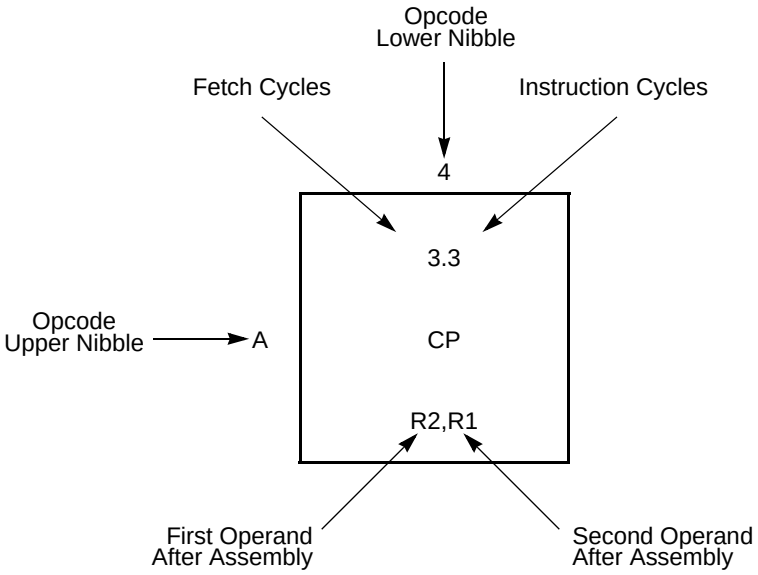


Figure 30. Opcode Map Cell Description

Table 129. 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.

Upper Nibble (Hex) \ Lower Nibble (Hex)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1.1 **BRK**	2.2 **SRP** IM	2.3 **ADD** r1,r2	2.4 **ADD** r1,Ir2	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,Ir2	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,Ir2	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,Ir2	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,Ir2	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,Ir2	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,Ir2	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,Ir2	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,Irr2	2.9 **LDEI** Ir1,Irr2	3.2 **LDX** r1,ER2	3.3 **LDX** Ir1,ER2	3.4 **LDX** IRR2,R1	3.5 **LDX** IRR2,IR1	3.4 **LDX** r1,rr2,X	3.4 **LDX** rr1,r2,X						1.2 **DI**
9	2.2 **RL** R1	2.3 **RL** IR1	2.5 **LDE** r2,Irr1	2.9 **LDEI** Ir2,Irr1	3.2 **LDX** r2,ER1	3.3 **LDX** Ir2,ER1	3.4 **LDX** R2,IRR1	3.5 **LDX** IR2,IRR1	3.3 **LEA** r1,r2,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,Ir2	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,Ir2	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,Irr2	2.9 **LDCI** Ir1,Irr2	2.3 **JP** IRR1	2.9 **LDC** Ir1,Irr2		3.4 **LD** r1,r2,X	3.2 **PUSHX** ER2							1.2 **RCF**
D	2.2 **SRA** R1	2.3 **SRA** IR1	2.5 **LDC** r2,Irr1	2.9 **LDCI** Ir2,Irr1	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,Ir2	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** Ir1,r2	2.8 **MULT** RR1	3.3 **LD** R2,IR1	3.3 **BTJ** p,b,r1,X	3.4 **BTJ** p,b,Ir1,X								

Figure 31. First Opcode Map

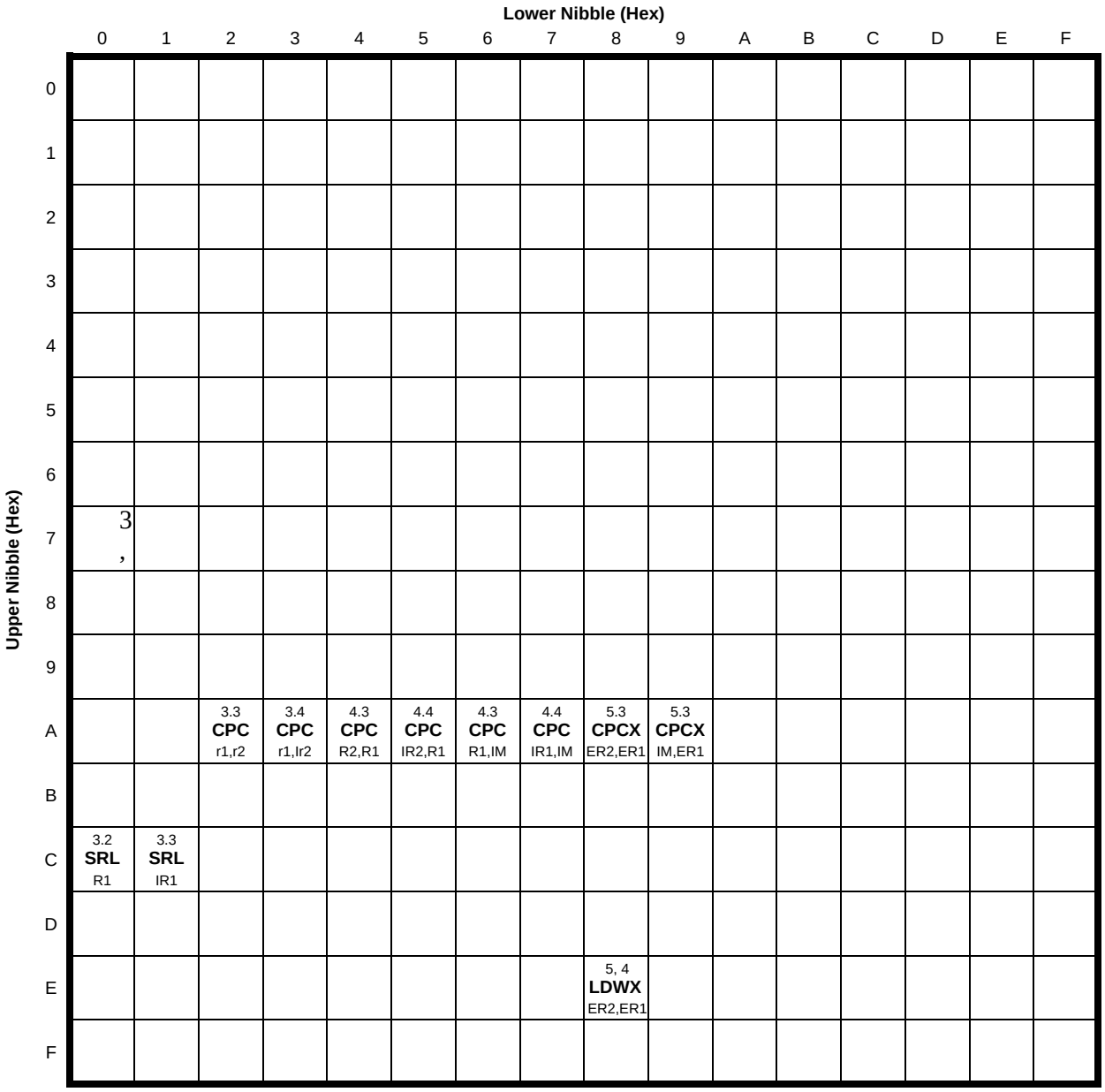


Figure 32. Second Opcode Map after 1FH

Electrical Characteristics

The data in this chapter represents all known data prior to qualification and characterization of the F082A 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 130 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 130. 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	

Table 130. Absolute Maximum Ratings (Continued)

Parameter	Minimum	Maximum	Units	Notes
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 131 lists the DC characteristics of the Z8 Encore! XP F082A Series products. All voltages are referenced to V_{SS}, the primary system ground.

Table 131. DC Characteristics

		T_A = –40°C to +105°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	–	$0.3*V_{DD}$	V	
V_{IH1}	High Level Input Voltage	$0.7*V_{DD}$	–	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	$0.7*V_{DD}$	–	$V_{DD}+0.3$	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 mA; V_{DD} = 3.0 V High Output Drive disabled.

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 131. DC Characteristics (Continued)

		T_A = –40°C to +105°C (unless otherwise specified)				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
V_{OH1}	High Level Output Voltage	2.4	–	–	V	I_{OH} = -2 mA; V_{DD} = 3.0 V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage	–	–	0.6	V	I_{OL} = 20 mA; V_{DD} = 3.3V High Output Drive enabled.
V_{OH2}	High Level Output Voltage	2.4	–	–	V	I_{OH} = -20 mA; V_{DD} = 3.3V High Output Drive enabled.
I_{IH}	Input Leakage Current	–	±0.002	±5	µA	V_{IN} = V_{DD} V_{DD} = 3.3V;
I_{IL}	Input Leakage Current	–	±0.007	±5	µA	V_{IN} = V_{SS} V_{DD} = 3.3V;
I_{TL}	Tristate Leakage Current	–	–	±5	µA	
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 V–3.6 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 132. Power Consumption

		V_{DD} = 2.7 V to 3.6 V				
Symbol	**Parameter**	**Typical[1]**	**Maximum Std Temp[2]**	**Maximum Ext Temp[3]**	**Units**	**Conditions**
I_{DD} Stop	Supply Current in Stop Mode	0.1			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			µA	5.5MHz.
		2.1	2.85	2.85	mA	20MHz.
I_{DD}	Supply Current in ACTIVE Mode (with all peripherals disabled)	2.8			mA	32kHz.
		4.5	5.2	5.2	mA	5.5MHz.
		5.5	6.5	6.5	mA	10MHz.
		7.9	11.5	11.5	mA	20MHz.
I_{DD} WDT	Watchdog Timer Supply Current	0.9	1.0	1.1	µA	
I_{DD} XTAL	Crystal Oscillator Supply Current	40			µA	32kHz.
		230			µA	4MHz.
		760			µA	20MHz.
I_{DD} IPO	Internal Precision Oscillator Supply Current	350	500	550	µA	
I_{DD} VBO	Voltage Brown-Out and Low-Voltage Detect Supply Current	50			µA	For 20-/28-pin devices (VBO only); See Note 4.
						For 8-pin devices; See Note 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.

Notes:
1. Typical conditions are defined as V_{DD} = 3.3V and +30°C.
2. Standard temperature is defined as T_A = 0°C to +70°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°C to +105°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.

Table 132. Power Consumption (Continued)

		V_{DD} = 2.7 V to 3.6 V				
Symbol	**Parameter**	**Typical[1]**	**Maximum Std Temp[2]**	**Maximum Ext Temp[3]**	**Units**	**Conditions**
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} LPO	Low-Power Operational Amplifier Supply Current	3	5	5	µA	Driving a high-impedance load.
I_{DD} TS	Temperature Sensor Supply Current	60			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} = 3.3V and +30°C.
2. Standard temperature is defined as T_A = 0°C to +70°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°C to +105°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.

Figure 33 displays the typical current consumption while operating with all peripherals disabled, at 30 ºC, versus the system clock frequency.

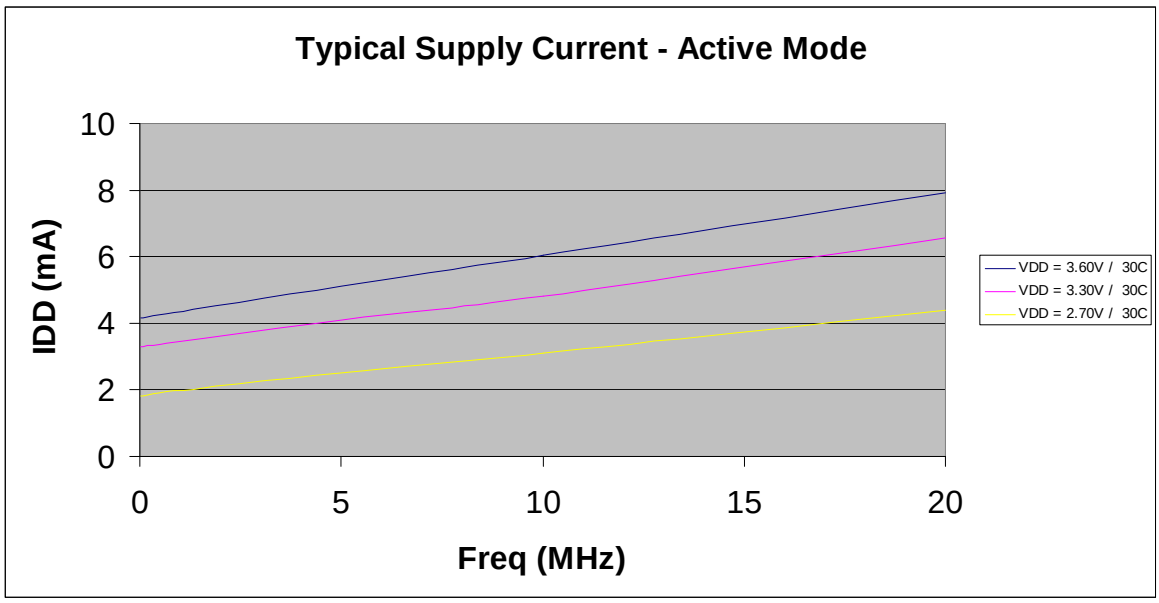


Figure 33. Typical Active Mode I_{DD} Versus System Clock Frequency

AC Characteristics

The section provides information about the AC characteristics and timing. All AC timing information assumes a standard load of 50pF on all outputs.

Table 133. 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	MHz	Program or erasure of the Flash memory
F_{XTAL}	Crystal Oscillator Frequency	–	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver
T_{XIN}	System Clock Period	50	–	ns	$T_{CLK} = 1/F_{sysclk}$
T_{XINH}	System Clock High Time	20	30	ns	T_{CLK} = 50 ns
T_{XINL}	System Clock Low Time	20	30	ns	T_{CLK} = 50 ns
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50 ns
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50 ns

Table 134. 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 135 tabulates the electrical characteristics of the POR and VBO blocks.

Table 135. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = –40°C to +105°C				
Symbol	**Parameter**	**Minimum**	**Typical[1]**	**Maximum**	**Units**	**Conditions**
V_{POR}	Power-On Reset Voltage Threshold	2.20	2.45	2.70	V	$V_{DD} = V_{POR}$
V_{VBO}	Voltage Brown-Out Reset Voltage Threshold	2.15	2.40	2.65	V	$V_{DD} = V_{VBO}$
	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	$V_{DD} > V_{POR}$; 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_{POR}	Power-On Reset Digital Delay		1		ms	5000 Internal Precision Oscillator cycles
T_{SMR}	Stop Mode Recovery with crystal oscillator disabled		16		µs	66 Internal Precision Oscillator cycles
T_{SMR}	Stop Mode Recovery with crystal oscillator enabled		1		ms	5000 Internal Precision Oscillator cycles
T_{VBO}	Voltage Brown-Out Pulse Rejection Period	–	10	–	µs	Period of time in which $V_{DD} < V_{VBO}$ without generating a Reset.

Note: Data in the typical column is from characterization at 3.3V and 30°C. These values are provided for design guidance only and are not tested in production.

Table 135. Power-On Reset and Voltage Brown-Out Electrical Characteristics and Timing

		T_A = –40°C to +105°C				
Symbol	**Parameter**	**Minimum**	**Typical[1]**	**Maximum**	**Units**	**Conditions**
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 3.3V and 30°C. These values are provided for design guidance only and are not tested in production.

Table 136. Flash Memory Electrical Characteristics and Timing

	V_{DD} = 2.7 V to 3.6 V T_A = –40°C to +105°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°C
Endurance	10,000	–	–	cycles	Program/erase cycles

Table 137. Watchdog Timer Electrical Characteristics and Timing

		V_{DD} = 2.7 V to 3.6 V T_A = –40°C to +105°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_{WDTCAL}	WDT Calibrated Time-out	0.98	1	1.02	s	V_{DD} = 3.3V; T_A = 30°C
		0.70	1	1.30	s	V_{DD} = 2.7 V to 3.6 V T_A = 0°C to 70°C
		0.50	1	1.50	s	V_{DD} = 2.7 V to 3.6 V T_A = –40°C to +105°C

Table 138. Non-Volatile Data Storage

	V_{DD} = 2.7 V to 3.6 V T_A = –40°C to +105°C				
Parameter	**Minimum**	**Typical**	**Maximum**	**Units**	**Notes**
NVDS Byte Read Time	34	–	519	µs	With system clock at 20 MHz
NVDS Byte Program Time	0.171	–	39.7	ms	With system clock at 20 MHz
Data Retention	100	–	–	years	25°C
Endurance	160,000	–	–	cycles	Cumulative write cycles for entire memory

Table 139. Analog-to-Digital Converter Electrical Characteristics and Timing

		V_{DD} = 3.0 V to 3.6 V T_A = 0°C to +70°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} = 2.0 V; $R_S \leftarrow$ 3.0 kΩ
	Integral Nonlinearity (INL)	–3.0	–	3.0	LSB[3]	External V_{REF} = 2.0 V; $R_S \leftarrow$ 3.0 kΩ
	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°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
			10258			Temperature sensor measurement

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} = 3.3V and T_A = +30°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 139. Analog-to-Digital Converter Electrical Characteristics and Timing (Continued)

		V_{DD} = 3.0 V to 3.6 V T_A = 0°C to +70°C (unless otherwise stated)				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
	Continuous Conversion Time	–	256	–	System clock cycles	All measurements but temperature sensor
			512			Temperature sensor measurement
	Signal Input Bandwidth	–	10		kHz	As defined by -3 dB point
R_S	Analog Source Impedance[4]	–	–	10	kΩ	In unbuffered mode
				500	kΩ	In buffered modes
Zin	Input Impedance	–	150		kΩ	In unbuffered mode at 20MHz[5]
		10	–		MΩ	In buffered modes
Vin	Input Voltage Range	0		V_{DD}	V	Unbuffered Mode
		0.3		V_{DD}–1.1	V	Buffered Modes These values define the range over which the ADC performs within spec; exceeding these values does not cause damage or instability; see DC Characteristics for absolute pin voltage limits.

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} = 3.3V and T_A = +30°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 140. Low Power Operational Amplifier Electrical Characteristics

		V_{DD} = 2.7 V to 3.6 V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
Av	Open loop voltage gain		80		dB	
GBW	Gain/Bandwidth product		500		kHz	
PM	Phase Margin		50		deg	Assuming 13 pF load capacitance.
V_{osLPO}	Input Offset Voltage		±1	±4	mV	
V_{osLPO}	Input Offset Voltage (Temperature Drift)		1	10	µV/C	
V_{IN}	Input Voltage Range	0.3		V_{DD}–1	V	
V_{OUT}	Output Voltage Range	0.3		V_{DD}–1	V	I_{OUT} = 45 µA.

Table 141. Comparator Electrical Characteristics

		V_{DD} = 2.7 V to 3.6 V T_A = –40°C to +105°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
V_{OS}	Input DC Offset		5		mV	
V_{CREF}	Programmable Internal Reference Voltage		±5		%	20- and 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	

Table 142. Temperature Sensor Electrical Characteristics

		V_{DD} = 2.7 V to 3.6 V				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
T_{AERR}	Temperature Error		±0.5	±2	°C	Over the range +20°C to +30°C (as measured by ADC).[1]
			±1	±5	°C	Over the range +0°C to +70°C (as measured by ADC).
			±2	±7	°C	Over the range +0°C to +105°C (as measured by ADC).
			±7		°C	Over the range –40°C to +105°C (as measured by ADC).
t_{WAKE}	Wakeup Time		80	100	µs	Time required for Temperature Sensor to stabilize after enabling.

Note: Devices are factory calibrated at for maximal accuracy between +20°C and +30°C, so the sensor is maximally accurate in that range. User recalibration for a different temperature range is possible and increases accuracy near the new calibration point.

General Purpose I/O Port Input Data Sample Timing

Figure 34 displays timing of 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.

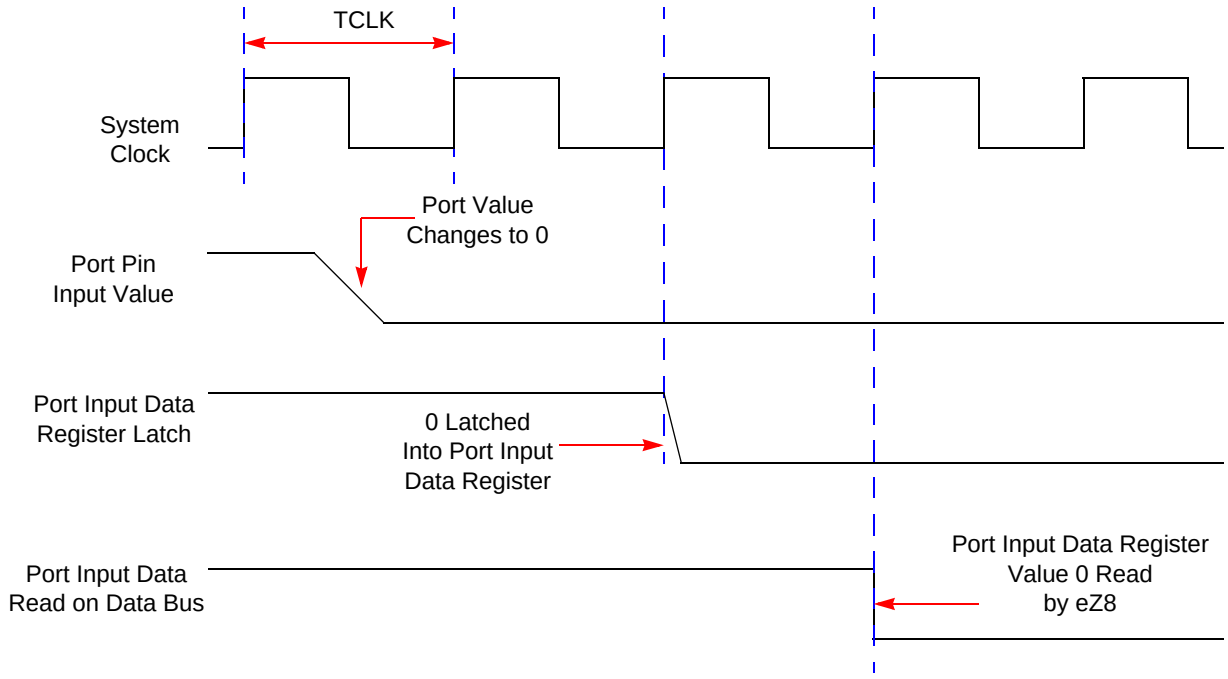


Figure 34. Port Input Sample Timing

Table 143. GPIO Port Input Timing

		Delay (ns)	
Parameter	**Abbreviation**	**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 35 and Table 144 provide timing information for GPIO port pins.

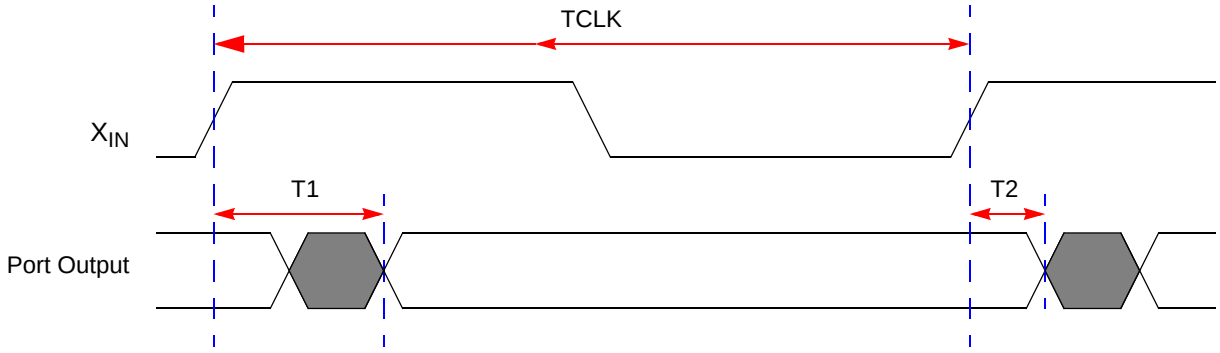


Figure 35. GPIO Port Output Timing

Table 144. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**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 36 and Table 145 provide timing information for the DBG pin. The DBG pin timing specifications assume a 4 ns maximum rise and fall time.

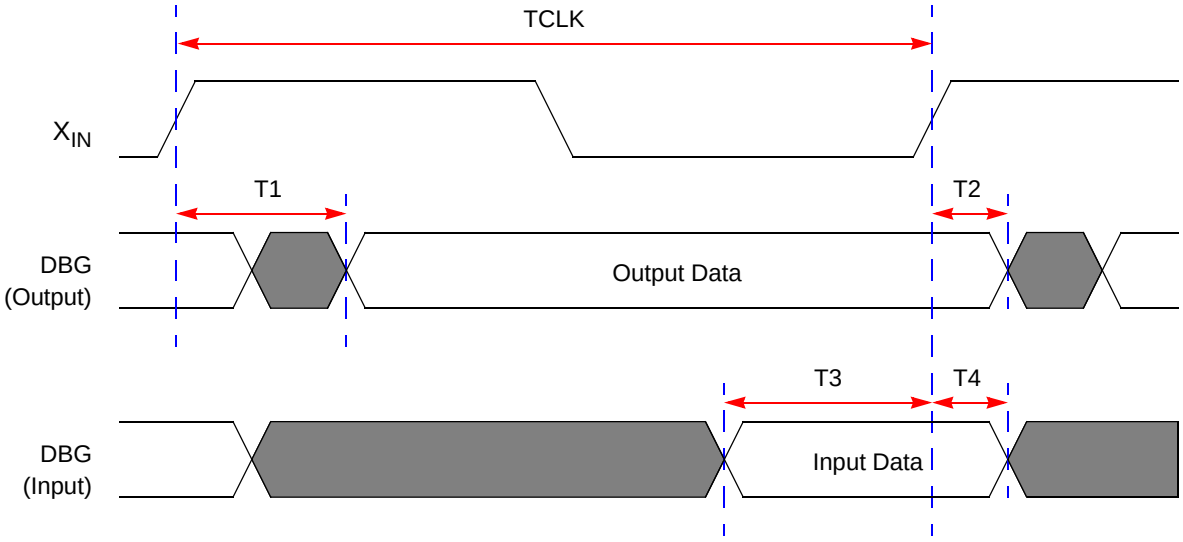


Figure 36. On-Chip Debugger Timing

Table 145. On-Chip Debugger Timing

		Delay (ns)	
Parameter	**Abbreviation**	**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 37 and Table 146 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.

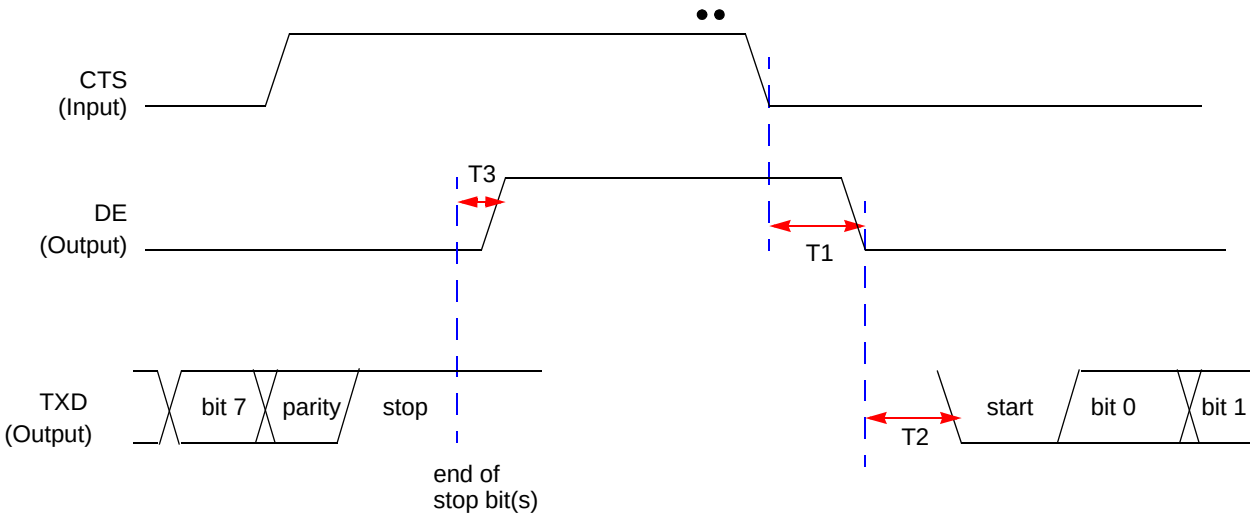


Figure 37. UART Timing With CTS

Table 146. UART Timing With CTS

		Delay (ns)	
Parameter	**Abbreviation**	**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 38 and Table 147 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.

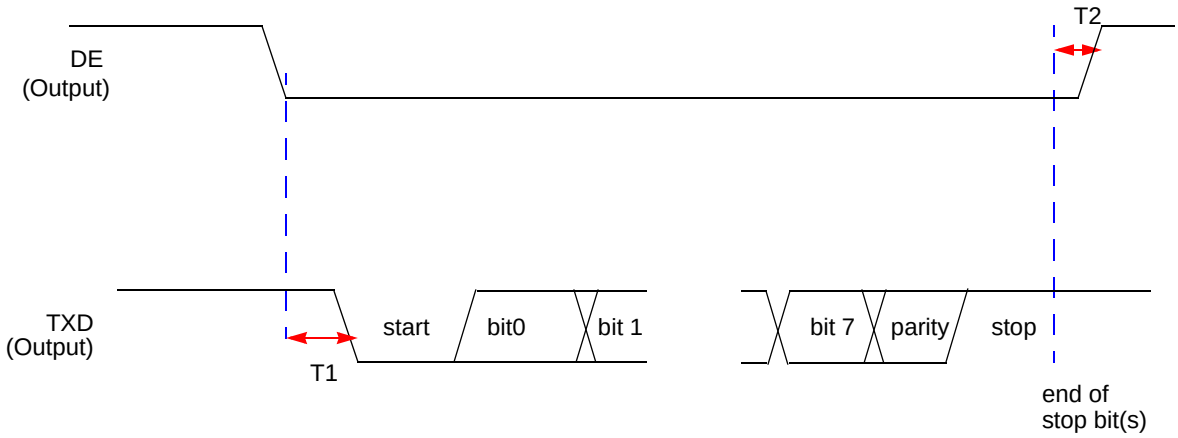


Figure 38. UART Timing Without CTS

Table 147. UART Timing Without CTS

		Delay (ns)	
Parameter	**Abbreviation**	**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 Product Line of MCUs includes the Z8F011A, Z8F012A, Z8F021A, Z8F022A, Z8F041A, Z8F042A, Z8F081A and Z8F082A devices, which are available in the following packages:

- 8-pin Plastic Dual-Inline Package (PDIP)
- 8-Pin Quad Flat No-Lead Package (QFN)/MLF-S[1]
- 8-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Small Outline Integrated Circuit Package (SOIC)
- 20-pin Small Shrink Outline Package (SSOP)
- 20-pin Plastic Dual-Inline Package (PDIP)
- 28-pin Small Outline Integrated Circuit Package (SOIC)
- 28-pin Small Shrink Outline Package (SSOP)
- 28-pin Plastic Dual-Inline Package (PDIP)

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 F082A Series products from Zilog using the part numbers shown in Table 148. 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 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 8KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0°C to 70°C											
Z8F082APB020SG	8KB	1KB	0	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F082AQB020SG	8KB	1KB	0	6	14	2	4	1	1	1	QFN 8-pin package
Z8F082ASB020SG	8KB	1KB	0	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F082ASH020SG	8KB	1KB	0	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F082AHH020SG	8KB	1KB	0	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F082APH020SG	8KB	1KB	0	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F082ASJ020SG	8KB	1KB	0	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F082AHJ020SG	8KB	1KB	0	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F082APJ020SG	8KB	1KB	0	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F082APB020EG	8KB	1KB	0	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F082AQB020EG	8KB	1KB	0	6	14	2	4	1	1	1	QFN 8-pin package
Z8F082ASB020EG	8KB	1KB	0	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F082ASH020EG	8KB	1KB	0	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F082AHH020EG	8KB	1KB	0	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F082APH020EG	8KB	1KB	0	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F082ASJ020EG	8KB	1KB	0	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F082AHJ020EG	8KB	1KB	0	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F082APJ020EG	8KB	1KB	0	23	20	2	8	1	1	1	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 8KB Flash											
Standard Temperature: 0°C to 70°C											
Z8F081APB020SG	8KB	1KB	0	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F081AQB020SG	8KB	1KB	0	6	13	2	0	1	1	0	QFN 8-pin package
Z8F081ASB020SG	8KB	1KB	0	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F081ASH020SG	8KB	1KB	0	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F081AHH020SG	8KB	1KB	0	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F081APH020SG	8KB	1KB	0	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F081ASJ020SG	8KB	1KB	0	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F081AHJ020SG	8KB	1KB	0	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F081APJ020SG	8KB	1KB	0	25	19	2	0	1	1	0	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F081APB020EG	8KB	1KB	0	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F081AQB020EG	8KB	1KB	0	6	13	2	0	1	1	0	QFN 8-pin package
Z8F081ASB020EG	8KB	1KB	0	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F081ASH020EG	8KB	1KB	0	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F081AHH020EG	8KB	1KB	0	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F081APH020EG	8KB	1KB	0	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F081ASJ020EG	8KB	1KB	0	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F081AHJ020EG	8KB	1KB	0	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F081APJ020EG	8KB	1KB	0	25	19	2	0	1	1	0	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 4 KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0°C to 70°C											
Z8F042APB020SG	4 KB	1KB	128 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F042AQB020SG	4 KB	1KB	128 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F042ASB020SG	4 KB	1KB	128 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F042ASH020SG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F042AHH020SG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F042APH020SG	4 KB	1KB	128 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F042ASJ020SG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F042AHJ020SG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F042APJ020SG	4 KB	1KB	128 B	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F042APB020EG	4 KB	1KB	128 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F042AQB020EG	4 KB	1KB	128 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F042ASB020EG	4 KB	1KB	128 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F042ASH020EG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F042AHH020EG	4 KB	1KB	128 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F042APH020EG	4 KB	1KB	128 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F042ASJ020EG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F042AHJ020EG	4 KB	1KB	128 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F042APJ020EG	4 KB	1KB	128 B	23	20	2	8	1	1	1	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 4 KB Flash											
Standard Temperature: 0°C to 70°C											
Z8F041APB020SG	4 KB	1KB	128 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F041AQB020SG	4 KB	1KB	128 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F041ASB020SG	4 KB	1KB	128 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F041ASH020SG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F041AHH020SG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F041APH020SG	4 KB	1KB	128 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F041ASJ020SG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F041AHJ020SG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F041APJ020SG	4 KB	1KB	128 B	25	19	2	0	1	1	0	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F041APB020EG	4 KB	1KB	128 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F041AQB020EG	4 KB	1KB	128 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F041ASB020EG	4 KB	1KB	128 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F041ASH020EG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F041AHH020EG	4 KB	1KB	128 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F041APH020EG	4 KB	1KB	128 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F041ASJ020EG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F041AHJ020EG	4 KB	1KB	128 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F041APJ020EG	4 KB	1KB	128 B	25	19	2	0	1	1	0	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 2 KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0°C to 70°C											
Z8F022APB020SG	2 KB	512 B	64 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F022AQB020SG	2 KB	512 B	64 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F022ASB020SG	2 KB	512 B	64 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F022ASH020SG	2 KB	512 B	64 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F022AHH020SG	2 KB	512 B	64 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F022APH020SG	2 KB	512 B	64 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F022ASJ020SG	2 KB	512 B	64 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F022AHJ020SG	2 KB	512 B	64 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F022APJ020SG	2 KB	512 B	64 B	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F022APB020EG	2 KB	512 B	64 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F022AQB020EG	2 KB	512 B	64 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F022ASB020EG	2 KB	512 B	64 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F022ASH020EG	2 KB	512 B	64 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F022AHH020EG	2 KB	512 B	64 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F022APH020EG	2 KB	512 B	64 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F022ASJ020EG	2 KB	512 B	64 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F022AHJ020EG	2 KB	512 B	64 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F022APJ020EG	2 KB	512 B	64 B	23	20	2	8	1	1	1	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 2 KB Flash											
Standard Temperature: 0°C to 70°C											
Z8F021APB020SG	2 KB	512 B	64 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F021AQB020SG	2 KB	512 B	64 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F021ASB020SG	2 KB	512 B	64 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F021ASH020SG	2 KB	512 B	64 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F021AHH020SG	2 KB	512 B	64 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F021APH020SG	2 KB	512 B	64 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F021ASJ020SG	2 KB	512 B	64 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F021AHJ020SG	2 KB	512 B	64 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F021APJ020SG	2 KB	512 B	64 B	25	19	2	0	1	1	0	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F021APB020EG	2 KB	512 B	64 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F021AQB020EG	2 KB	512 B	64 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F021ASB020EG	2 KB	512 B	64 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F021ASH020EG	2 KB	512 B	64 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F021AHH020EG	2 KB	512 B	64 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F021APH020EG	2 KB	512 B	64 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F021ASJ020EG	2 KB	512 B	64 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F021AHJ020EG	2 KB	512 B	64 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F021APJ020EG	2 KB	512 B	64 B	25	19	2	0	1	1	0	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 1KB Flash, 10-Bit Analog-to-Digital Converter											
Standard Temperature: 0°C to 70°C											
Z8F012APB020SG	1KB	256 B	16 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F012AQB020SG	1KB	256 B	16 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F012ASB020SG	1KB	256 B	16 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F012ASH020SG	1KB	256 B	16 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F012AHH020SG	1KB	256 B	16 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F012APH020SG	1KB	256 B	16 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F012ASJ020SG	1KB	256 B	16 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F012AHJ020SG	1KB	256 B	16 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F012APJ020SG	1KB	256 B	16 B	23	20	2	8	1	1	1	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F012APB020EG	1KB	256 B	16 B	6	14	2	4	1	1	1	PDIP 8-pin package
Z8F012AQB020EG	1KB	256 B	16 B	6	14	2	4	1	1	1	QFN 8-pin package
Z8F012ASB020EG	1KB	256 B	16 B	6	14	2	4	1	1	1	SOIC 8-pin package
Z8F012ASH020EG	1KB	256 B	16 B	17	20	2	7	1	1	1	SOIC 20-pin package
Z8F012AHH020EG	1KB	256 B	16 B	17	20	2	7	1	1	1	SSOP 20-pin package
Z8F012APH020EG	1KB	256 B	16 B	17	20	2	7	1	1	1	PDIP 20-pin package
Z8F012ASJ020EG	1KB	256 B	16 B	23	20	2	8	1	1	1	SOIC 28-pin package
Z8F012AHJ020EG	1KB	256 B	16 B	23	20	2	8	1	1	1	SSOP 28-pin package
Z8F012APJ020EG	1KB	256 B	16 B	23	20	2	8	1	1	1	PDIP 28-pin package

Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series with 1KB Flash											
Standard Temperature: 0°C to 70°C											
Z8F011APB020SG	1KB	256 B	16 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F011AQB020SG	1KB	256 B	16 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F011ASB020SG	1KB	256 B	16 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F011ASH020SG	1KB	256 B	16 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F011AHH020SG	1KB	256 B	16 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F011APH020SG	1KB	256 B	16 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F011ASJ020SG	1KB	256 B	16 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F011AHJ020SG	1KB	256 B	16 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F011APJ020SG	1KB	256 B	16 B	25	19	2	0	1	1	0	PDIP 28-pin package
Extended Temperature: –40°C to 105°C											
Z8F011APB020EG	1KB	256 B	16 B	6	13	2	0	1	1	0	PDIP 8-pin package
Z8F011AQB020EG	1KB	256 B	16 B	6	13	2	0	1	1	0	QFN 8-pin package
Z8F011ASB020EG	1KB	256 B	16 B	6	13	2	0	1	1	0	SOIC 8-pin package
Z8F011ASH020EG	1KB	256 B	16 B	17	19	2	0	1	1	0	SOIC 20-pin package
Z8F011AHH020EG	1KB	256 B	16 B	17	19	2	0	1	1	0	SSOP 20-pin package
Z8F011APH020EG	1KB	256 B	16 B	17	19	2	0	1	1	0	PDIP 20-pin package
Z8F011ASJ020EG	1KB	256 B	16 B	25	19	2	0	1	1	0	SOIC 28-pin package
Z8F011AHJ020EG	1KB	256 B	16 B	25	19	2	0	1	1	0	SSOP 28-pin package
Z8F011APJ020EG	1KB	256 B	16 B	25	19	2	0	1	1	0	PDIP 28-pin package

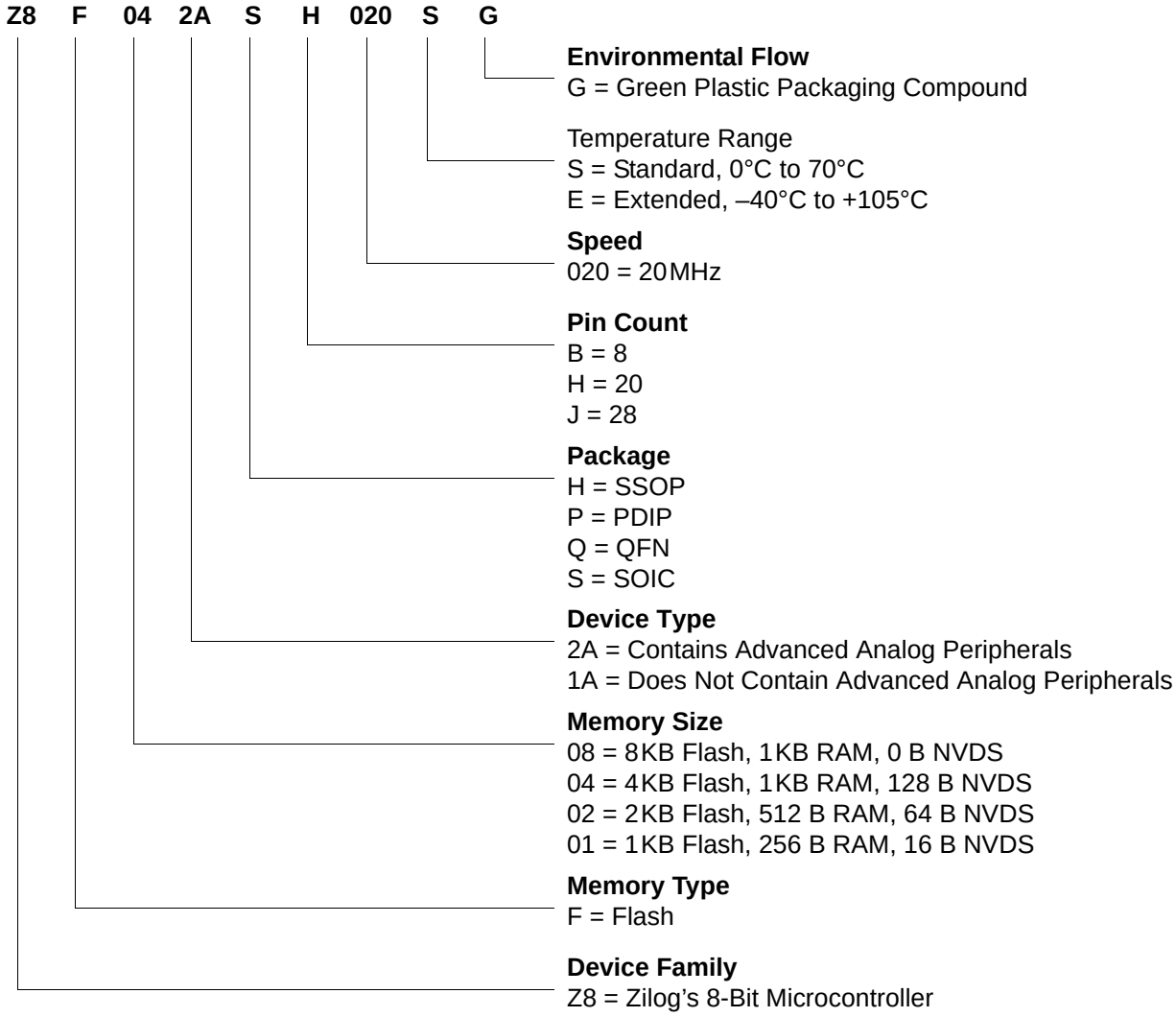
Table 148. Z8 Encore! XP F082A Series Ordering Matrix

Part Number	Flash	RAM	NVDS	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	UART with IrDA	Comparator	Temperature Sensor	Description
Z8 Encore! XP F082A Series Development Kit											
Z8F08A28100KITG											Z8 Encore! XP F082A Series 28-Pin Development Kit
Z8F04A28100KITG											Z8 Encore! XP F042A Series 28-Pin Development Kit
Z8F04A08100KITG											Z8 Encore! XP F042A Series 8-Pin Development Kit
ZUSBSC00100ZACG											USB Smart Cable Accessory Kit
ZUSBOPTSC01ZACG											USB Opto-Isolated 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 Z8F042ASH020SG is an 8-bit Flash MCU with 4KB of Program Memory, equipped with advanced analog peripherals in a 20-pin SOIC package, operating within a 0ºC to +70ºC temperature range and built using lead-free solder.



Index

Numerics

10-bit ADC 6

A

absolute maximum ratings 226
AC characteristics 232
ADC 208
  architecture 124
  block diagram 125
  continuous conversion 127
  control register 134, 135
  control register definitions 133
  data high byte register 136
  data low bits register 137
  electrical characteristics and timing 236
  operation 125
  single-shot conversion 126
ADCCTL register 134, 135
ADCDH register 136
ADCDL register 137
ADCX 208
ADD 208
add - extended addressing 208
add with carry 208
add with carry - extended addressing 208
additional symbols 207
address space 15
ADDX 208
analog signals 11
analog-to-digital converter (ADC) 124
AND 210
ANDX 210
arithmetic instructions 208
assembly language programming 204
assembly language syntax 205

B

B 207
b 206
baud rate generator, UART 110
BCLR 209
binary number suffix 207
BIT 209
bit 206
  clear 209
  manipulation instructions 209
  set 209
  set or clear 209
  swap 209
  test and jump 211
  test and jump if non-zero 211
  test and jump if zero 211
bit jump and test if non-zero 211
bit swap 211
block diagram 3
block transfer instructions 209
BRK 211
BSET 209
BSWAP 209, 211
BTJ 211
BTJNZ 211
BTJZ 211

C

CALL procedure 211
Capture Mode 87, 88
Capture/Compare Mode 88
cc 206
CCF 209
characteristics, electrical 226
clear 210
CLR 210
COM 210
compare 87
compare - extended addressing 208
Compare Mode 87
compare with carry 208

compare with carry - extended addressing 208
complement 210
complement carry flag 209
condition code 206
continuous conversion (ADC) 127
Continuous Mode 87
control register definition, UART 110
Control Registers 15, 18
Counter modes 87
CP 208
CPC 208
CPCX 208
CPU and peripheral overview 4
CPU control instructions 209
CPX 208
Customer Feedback Form 265

D

DA 206, 208
data memory 17
DC characteristics 227
debugger, on-chip 180
DEC 208
decimal adjust 208
decrement 208
decrement and jump non-zero 211
decrement word 208
DECW 208
destination operand 207
device, port availability 36
DI 209
direct address 206
disable interrupts 209
DJNZ 211
dst 207

E

EI 209
electrical characteristics 226
    ADC 236
    flash memory and timing 234
    GPIO input data sample timing 240
    Watchdog Timer 235, 238
enable interrupt 209
ER 206
extended addressing register 206
external pin reset 26
eZ8 CPU features 4
eZ8 CPU instruction classes 207
eZ8 CPU instruction notation 206
eZ8 CPU instruction set 204
eZ8 CPU instruction summary 212

F

FCTL register 155, 161, 162
features, Z8 Encore! 1
first opcode map 224
FLAGS 207
flags register 207
flash
    controller 6
    option bit address space 162
    option bit configuration - reset 159
    program memory address 0000H 162
    program memory address 0001H 164
flash memory 146
    arrangement 147
    byte programming 151
    code protection 149
    configurations 146
    control register definitions 153, 161
    controller bypass 152
    electrical characteristics and timing 234
    flash control register 155, 161, 162
    flash option bits 150
    flash status register 155
    flow chart 148
    frequency high and low byte registers 157
    mass erase 152
    operation 147
    operation timing 149
    page erase 152
    page select register 156, 157
FPS register 156, 157
FSTAT register 155

G

Gated Mode 88
general-purpose I/O 36
GPIO 6, 36
  alternate functions 37
  architecture 37
  control register definitions 44
  input data sample timing 240
  interrupts 44
  port A-C pull-up enable sub-registers 50, 51
  port A-H address registers 45
  port A-H alternate function sub-registers 47
  port A-H control registers 46
  port A-H data direction sub-registers 46
  port A-H high drive enable sub-registers 48
  port A-H input data registers 52
  port A-H output control sub-registers 47
  port A-H output data registers 52, 53
  port A-H stop mode recovery sub-registers 49
  port availability by device 36
  port input timing 240
  port output timing 241

H

H 207
HALT 209
Halt Mode 33, 209
hexadecimal number prefix/suffix 207

I

I2C 6
IM 206
immediate data 206
immediate operand prefix 207
INC 208
increment 208
increment word 208
INCW 208
indexed 207
indirect address prefix 207
indirect register 206
indirect register pair 206
indirect working register 206
indirect working register pair 206
infrared encoder/decoder (IrDA) 120
Instruction Set 204
instruction set, eZ8 CPU 204
instructions
  ADC 208
  ADCX 208
  ADD 208
  ADDX 208
  AND 210
  ANDX 210
  arithmetic 208
  BCLR 209
  BIT 209
  bit manipulation 209
  block transfer 209
  BRK 211
  BSET 209
  BSWAP 209, 211
  BTJ 211
  BTJNZ 211
  BTJZ 211
  CALL 211
  CCF 209
  CLR 210
  COM 210
  CP 208
  CPC 208
  CPCX 208
  CPU control 209
  CPX 208
  DA 208
  DEC 208
  DECW 208
  DI 209
  DJNZ 211
  EI 209
  HALT 209
  INC 208
  INCW 208
  IRET 211
  JP 211

LD 210
LDC 210
LDCI 209, 210
LDE 210
LDEI 209
LDX 210
LEA 210
logical 210
MULT 208
NOP 209
OR 210
ORX 210
POP 210
POPX 210
program control 211
PUSH 210
PUSHX 210
RCF 209, 210
RET 211
RL 211
RLC 211
rotate and shift 211
RR 211
RRC 211
SBC 208
SCF 209, 210
SRA 211
SRL 211
SRP 210
STOP 210
SUB 208
SUBX 208
SWAP 211
TCM 209
TCMX 209
TM 209
TMX 209
TRAP 211
Watchdog Timer refresh 210
XOR 210
XORX 210
instructions, eZ8 classes of 207
interrupt control register 69
interrupt controller 55
architecture 55
interrupt assertion types 58
interrupt vectors and priority 58
operation 57
register definitions 60
software interrupt assertion 59
interrupt edge select register 67
interrupt request 0 register 60
interrupt request 1 register 61
interrupt request 2 register 62
interrupt return 211
interrupt vector listing 55
interrupts
UART 108
IR 206
Ir 206
IrDA
architecture 120
block diagram 120
control register definitions 123
operation 120
receiving data 122
transmitting data 121
IRET 211
IRQ0 enable high and low bit registers 62
IRQ1 enable high and low bit registers 64
IRQ2 enable high and low bit registers 65
IRR 206
Irr 206

J

JP 211
jump, conditional, relative, and relative conditional 211

L

LD 210
LDC 210
LDCI 209, 210
LDE 210
LDEI 209, 210
LDX 210

LEA 210
load 210
load constant 209
load constant to/from program memory 210
load constant with auto-increment addresses 210
load effective address 210
load external data 210
load external data to/from data memory and auto-increment addresses 209
load external to/from data memory and auto-increment addresses 210
load using extended addressing 210
logical AND 210
logical AND/extended addressing 210
logical exclusive OR 210
logical exclusive OR/extended addressing 210
logical instructions 210
logical OR 210
logical OR/extended addressing 210
low power modes 32

M

master interrupt enable 57
memory
  data 17
  program 15
mode
  Capture 87, 88
  Capture/Compare 88
  Continuous 87
  Counter 87
  Gated 88
  One-Shot 87
  PWM 87, 88
modes 87
MULT 208
multiply 208
multiprocessor mode, UART 105

N

NOP (no operation) 209
notation
  b 206
  cc 206
  DA 206
  ER 206
  IM 206
  IR 206
  Ir 206
  IRR 206
  Irr 206
  p 206
  R 206
  r 206
  RA 206
  RR 206
  rr 206
  vector 207
  X 207
notational shorthand 206

O

OCD
  architecture 180
  auto-baud detector/generator 183
  baud rate limits 184
  block diagram 180
  breakpoints 185
  commands 186
  control register 191
  data format 183
  DBG pin to RS-232 Interface 181
  debug mode 182
  debugger break 211
  interface 181
  serial errors 184
  status register 192
  timing 242
OCD commands
  execute instruction (12H) 190
  read data memory (0DH) 190
  read OCD control register (05H) 188
  read OCD revision (00H) 187
  read OCD status register (02H) 187
  read program counter (07H) 188

read program memory (0BH) 189
read program memory CRC (0EH) 190
read register (09H) 189
read runtime counter (03H) 187
step instruction (10H) 190
stuff instruction (11H) 190
write data memory (0CH) 189
write OCD control register (04H) 188
write program counter (06H) 188
write program memory (0AH) 189
write register (08H) 188
on-chip debugger (OCD) 180
on-chip debugger signals 11
on-chip oscillator 198
One-Shot Mode 87
opcode map
abbreviations 223
cell description 222
first 224
second after 1FH 225
Operational Description 22, 32, 36, 55, 70, 93, 99, 120, 124, 139, 140, 144, 146, 159, 176, 180, 193, 198, 203
OR 210
ordering information 246
ORX 210
oscillator signals 11

P

p 206
Packaging 245
part selection guide 2
PC 207
peripheral AC and DC electrical characteristics 233
pin characteristics 12
Pin Descriptions 8
polarity 206
POP 210
pop using extended addressing 210
POPX 210
port availability, device 36
port input timing (GPIO) 240
port output timing, GPIO 241
power supply signals 12
Power-on and Voltage Brownout electrical characteristics and timing 233
Power-On Reset (POR) 24
program control instructions 211
program counter 207
program memory 15
PUSH 210
push using extended addressing 210
PUSHX 210
PWM mode 87, 88
PxADDR register 45
PxCTL register 46

R

R 206
r 206
RA
register address 206
RCF 209, 210
receive
IrDA data 122
receiving UART data-interrupt-driven method 104
receiving UART data-polled method 103
register 206
ADC control (ADCCTL) 134, 135
ADC data high byte (ADCDH) 136
ADC data low bits (ADCDL) 137
flash control (FCTL) 155, 161, 162
flash high and low byte (FFREQH and FREEQL) 157
flash page select (FPS) 156, 157
flash status (FSTAT) 155
GPIO port A-H address (PxADDR) 45
GPIO port A-H alternate function sub-registers 47
GPIO port A-H control address (PxCTL) 46
GPIO port A-H data direction sub-registers 46
OCD control 191
OCD status 192
UARTx baud rate high byte (UxBRH) 117
UARTx baud rate low byte (UxBRL) 117
UARTx Control 0 (UxCTL0) 111, 117

UARTx control 1 (UxCTL1) 112
UARTx receive data (UxRXD) 116
UARTx status 0 (UxSTAT0) 114
UARTx status 1 (UxSTAT1) 115
UARTx transmit data (UxTXD) 116
Watchdog Timer control (WDTCTL) 30, 96, 141, 196
Watchdog Timer reload high byte (WDTH) 97
Watchdog Timer reload low byte (WDTL) 98
Watchdog Timer reload upper byte (WDTU) 97
register file 15
register pair 206
register pointer 207
reset
and stop mode characteristics 23
and Stop Mode Recovery 22
carry flag 209
sources 24
RET 211
return 211
RL 211
RLC 211
rotate and shift instructions 211
rotate left 211
rotate left through carry 211
rotate right 211
rotate right through carry 211
RP 207
RR 206, 211
rr 206
RRC 211

S

SBC 208
SCF 209, 210
second opcode map after 1FH 225
set carry flag 209, 210
set register pointer 210
shift right arithmetic 211
shift right logical 211
signal descriptions 10
single-shot conversion (ADC) 126
software trap 211
source operand 207
SP 207
SRA 211
src 207
SRL 211
SRP 210
stack pointer 207
Stop 210
Stop Mode 32
stop mode 210
Stop Mode Recovery
sources 27
using a GPIO port pin transition 28
using Watchdog Timer time-out 28
stop mode recovery
sources 29
using a GPIO port pin transition 29
SUB 208
subtract 208
subtract - extended addressing 208
subtract with carry 208
subtract with carry - extended addressing 208
SUBX 208
SWAP 211
swap nibbles 211
symbols, additional 207

T

TCM 209
TCMX 209
test complement under mask 209
test complement under mask - extended addressing 209
test under mask 209
test under mask - extended addressing 209
timer signals 10
timers 70
architecture 70
block diagram 71
Capture Mode 79, 80, 87, 88
Capture/Compare Mode 83, 88
Compare Mode 81, 87

Continuous Mode 72, 87
Counter Mode 73, 74
Counter modes 87
Gated Mode 82, 88
One-Shot Mode 71, 87
operating mode 71
PWM mode 76, 77, 87, 88
reading the timer count values 84
reload high and low byte registers 91
timer control register definitions 85
timer output signal operation 84
timers 0-3
control registers 85, 86
high and low byte registers 89, 92
TM 209
TMX 209
transmit
IrDA data 121
transmitting UART data-polled method 101
transmitting UART dat-interrupt-driven method 102
TRAP 211

U

UART 6
architecture 99
baud rate generator 110
baud rates table 118
control register definitions 110
controller signals 10
interrupts 108
multiprocessor mode 105
receiving data using interrupt-driven method 104
receiving data using the polled method 103
transmitting data using the interrupt-driven method 102
transmitting data using the polled method 101
x baud rate high and low registers 117
x control 0 and control 1 registers 110
x status 0 and status 1 registers 114, 115
UxBRH register 117
UxBRL register 117
UxCTL0 register 111, 117
UxCTL1 register 112
UxRXD register 116
UxSTAT0 register 114
UxSTAT1 register 115
UxTXD register 116

V

vector 207
Voltage Brownout reset (VBR) 25

W

Watchdog Timer
approximate time-out delay 93
approximate time-out delays 140
CNTL 25
control register 96
electrical characteristics and timing 235, 238
interrupt in normal operation 94
interrupt in Stop Mode 94
operation 140
refresh 94, 210
reload unlock sequence 95
reload upper, high and low registers 97
reset 26
reset in normal operation 95
reset in Stop Mode 95
time-out response 94
WDTCTL register 30, 96, 141, 196
WDTH register 97
WDTL register 98
working register 206
working register pair 206
WTDU register 97

X

X 207
XOR 210
XORX 210

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.

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View Z8F022ASB020SG on WIN SOURCE
- Zilog Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management