



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6422ar020sc2104

Manual Objectives

This Product Specification provides detailed operating information for the Flash devices within Zilog's Z8 Encore! XP[®] 64K Series Flash Microcontrollers Microcontroller (MCU) products. Within this document, the Z8F642x, Z8F482x, Z8F322x, Z8F242x, and Z8F162x devices are referred to collectively as the Z8 Encore! XP[®] 64K Series Flash Microcontrollers unless specifically stated otherwise.

About This Manual

Zilog[®] recommends that you read and understand everything in this manual before setting up and using the product. However, we recognize that there are different styles of learning. Therefore, we have designed this Product Specification to be used either as a *how to* procedural manual or a reference guide to important data.

Intended Audience

This document is written for Zilog customers who are experienced at working with microcontrollers, integrated circuits, or printed circuit assemblies.

Manual Conventions

The following assumptions and conventions are adopted to provide clarity and ease of use:

Courier Typeface

Commands, code lines and fragments, bits, equations, hexadecimal addresses, and various executable items are distinguished from general text by the use of the `Courier` typeface. Where the use of the font is not indicated, as in the Index, the name of the entity is presented in upper case.

- Example: `FLAGS[1]` is `smrf`.

Hexadecimal Values

Hexadecimal values are designated by uppercase *H* suffix and appear in the `Courier` typeface.

- Example: R1 is set to `F8H`.

Brackets

The square brackets, `[]`, indicate a register or bus.

- Example: For the register `R1[7:0]`, R1 is an 8-bit register, `R1[7]` is the most significant bit, and `R1[0]` is the least significant bit.

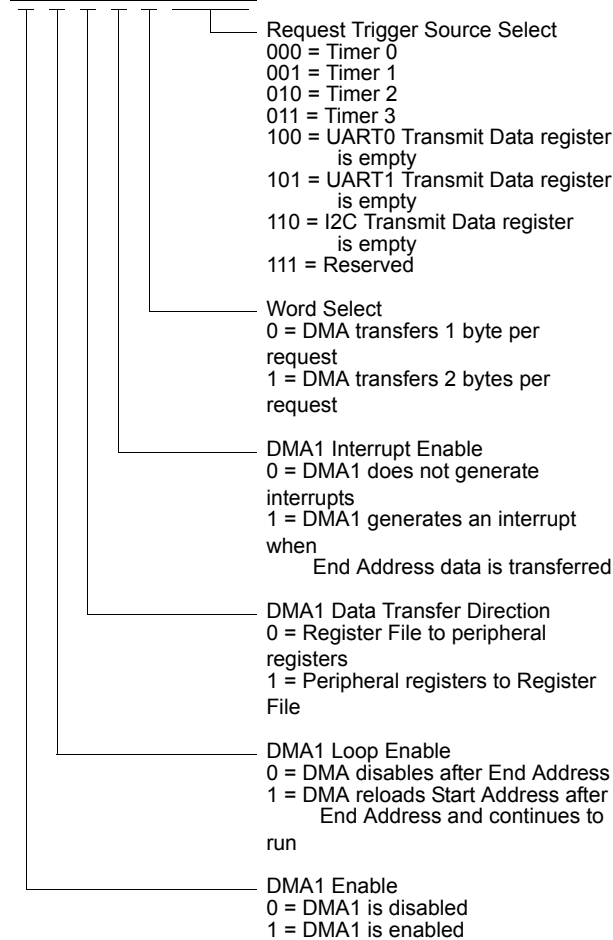
Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
F61	SPI Control	SPICTL	00	137
F62	SPI Status	SPISTAT	01	139
F63	SPI Mode	SPIMODE	00	140
F64	SPI Diagnostic State	SPIDST	00	141
F65	Reserved	—	XX	
F66	SPI Baud Rate High Byte	SPIBRH	FF	142
F67	SPI Baud Rate Low Byte	SPIBRL	FF	142
F68-F6F	Reserved	—	XX	
Analog-to-Digital Converter				
F70	ADC Control	ADCCTL	20	179
F71	Reserved	—	XX	
F72	ADC Data High Byte	ADCD_H	XX	180
F73	ADC Data Low Bits	ADCD_L	XX	180
F74-FAF	Reserved	—	XX	
DMA 0				
FB0	DMA0 Control	DMA0CTL	00	167
FB1	DMA0 I/O Address	DMA0IO	XX	169
FB2	DMA0 End/Start Address High Nibble	DMA0H	XX	169
FB3	DMA0 Start Address Low Byte	DMA0START	XX	170
FB4	DMA0 End Address Low Byte	DMA0END	XX	170
DMA 1				
FB8	DMA1 Control	DMA1CTL	00	167
FB9	DMA1 I/O Address	DMA1IO	XX	169
FBA	DMA1 End/Start Address High Nibble	DMA1H	XX	169
FBB	DMA1 Start Address Low Byte	DMA1START	XX	170
FBC	DMA1 End Address Low Byte	DMA1END	XX	170
DMA ADC				
FBD	DMA_ADC Address	DMAA_ADDR	XX	171
FBE	DMA_ADC Control	DMAACTL	00	172
FBF	DMA_ADC Status	DMAASTAT	00	173
Interrupt Controller				
FC0	Interrupt Request 0	IRQ0	00	71
FC1	IRQ0 Enable High Bit	IRQ0ENH	00	74
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	74
FC3	Interrupt Request 1	IRQ1	00	72
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	75
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	75
FC6	Interrupt Request 2	IRQ2	00	73
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	76
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	76
FC9-FCC	Reserved	—	XX	

DMA1 Control

DMA1CTL (FB8H - Read/Write)

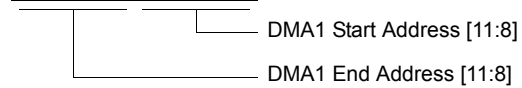
D7 D6 D5 D4 D3 D2 D1 D0



DMA1 Address High Nibble

DMA1H (FBAH - Read/Write)

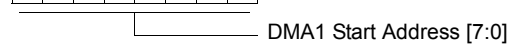
D7 D6 D5 D4 D3 D2 D1 D0



DMA1 Start/Current Address Low Byte

DMA1START (FBBH - Read/Write)

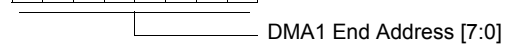
D7 D6 D5 D4 D3 D2 D1 D0



DMA1 End Address Low Byte

DMA1END (FBCH - Read/Write)

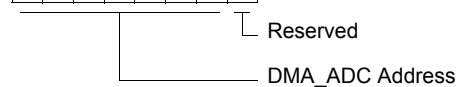
D7 D6 D5 D4 D3 D2 D1 D0



DMA_ADC Address

DMAA_ADDR (FBDH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0



DMA1 I/O Address

DMA1IO (FB9H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

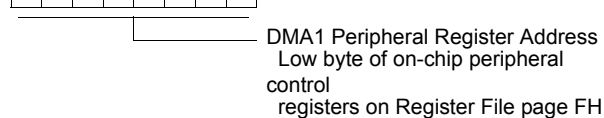


Table 11. Port Availability by Device and Package Type (Continued)

Device	Packages	Port A	Port B	Port C	Port D	Port E	Port F	Port G	Port H
Z8X4823	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]
Z8X6421	40-pin	[7:0]	[7:0]	[6:0]	[6:3, 1:0]	-	-	-	-
Z8X6421	44-pin	[7:0]	[7:0]	[7:0]	[6:0]	-	-	-	-
Z8X6422	64- and 68-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7]	[3]	[3:0]
Z8X6423	80-pin	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[7:0]	[3:0]

Architecture

Figure 10 displays a simplified block diagram of a GPIO port pin. In Figure 10, the ability to accommodate alternate functions and variable port current drive strength are not illustrated.

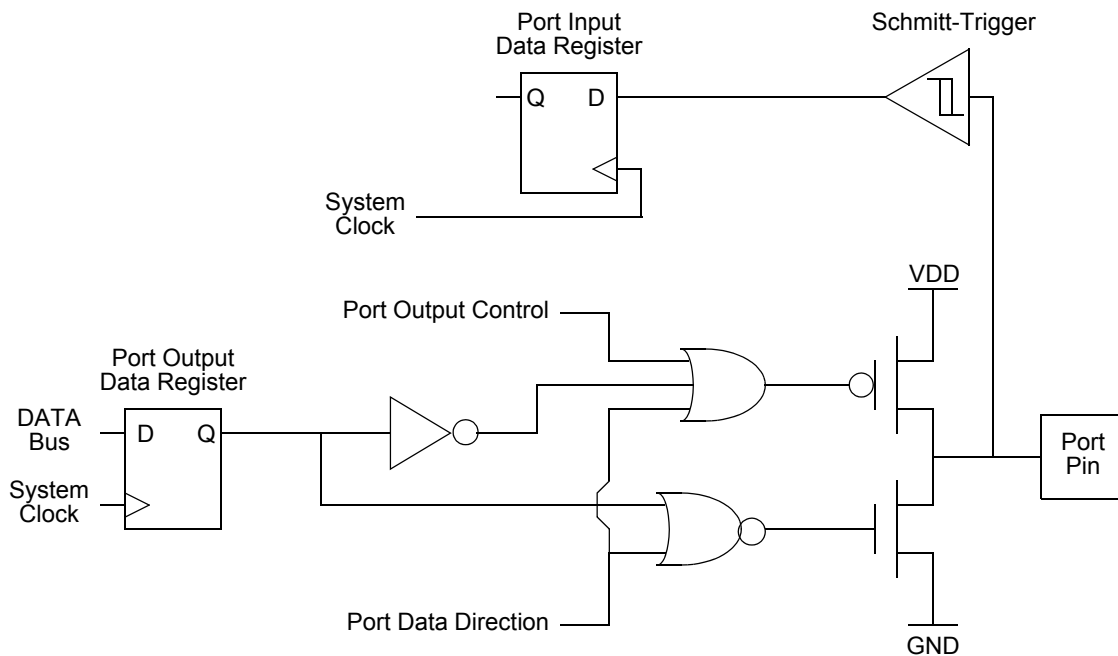


Figure 10. GPIO Port Pin Block Diagram

GPIO Control Register Definitions

Four registers for each Port provide access to GPIO control, input data, and output data. [Table 13](#) lists these Port registers. Use the Port A–H Address and Control registers together to provide access to sub-registers for Port configuration and control.

Table 13. GPIO Port Registers and Sub-Registers

Port Register Mnemonic	Port Register Name
PxADDR	Port A–H Address Register (Selects sub-registers)
PxCTL	Port A–H Control Register (Provides access to sub-registers)
PxIN	Port A–H Input Data Register
PxOUT	Port A–H Output Data Register
Port Sub-Register Mnemonic	Port Register Name
PxDD	Data Direction
PxAF	Alternate Function
PxOC	Output Control (Open-Drain)
PxDD	High Drive Enable
PxSMRE	Stop Mode Recovery Source Enable

Port A–H Address Registers

The Port A–H Address registers select the GPIO Port functionality accessible through the Port A–H Control registers. The Port A–H Address and Control registers combine to provide access to all GPIO Port control ([Table 14](#)).

Table 14. Port A–H GPIO Address Registers (PxADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	PADDR[7:0]							
RESET	00H							
R/W	R/W							
ADDR	FD0H, FD4H, FD8H, FDCH, FE0H, FE4H, FE8H, FECH							

- Executing a Trap instruction.
- Illegal Instruction 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 were enabled with identical interrupt priority (all as Level 2 interrupts, for example), then interrupt priority would be assigned from highest to lowest as specified in [Table 23](#) on page 68. Level 3 interrupts always have higher priority than Level 2 interrupts which, in turn, always have 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 23](#) on page 68. Reset, Watchdog Timer interrupt (if enabled), and Illegal Instruction Trap always have highest 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: *The following style of coding to clear bits in the Interrupt Request registers is NOT recommended. All incoming interrupts that are received between execution of the first LDX command and the last LDX command are lost.*

Poor coding style that can result in lost interrupt requests:

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

To avoid missing interrupts, the following style of coding to clear bits in the Interrupt Request 0 register is recommended:

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 desired 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.

Timers

Overview

The 64K Series products contain up to four 16-bit reloadable timers that can be used for timing, event counting, or generation of pulse width modulated signals. The timers' features include:

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

In addition to the timers described in this chapter, the Baud Rate Generators for any unused UART, SPI, or I²C peripherals may also be used to provide basic timing functionality. For information on using the Baud Rate Generators as timers, see the respective serial communication peripheral. Timer 3 is unavailable in the 44-pin package devices.

Architecture

Figure 12 displays the architecture of the timers.

- Set the prescale value
- 2. Write to the Timer High and Low Byte registers to set the starting count value. This 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. If desired, 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. Write to the Timer Control 1 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 desired transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control 1 Register. The timer input is the system clock.

Every subsequent desired 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.

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.

Follow the steps below for configuring a timer for CAPTURE/COMPARE mode and initiating the count:

1. Write to the Timer Control 1 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. If desired, 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.

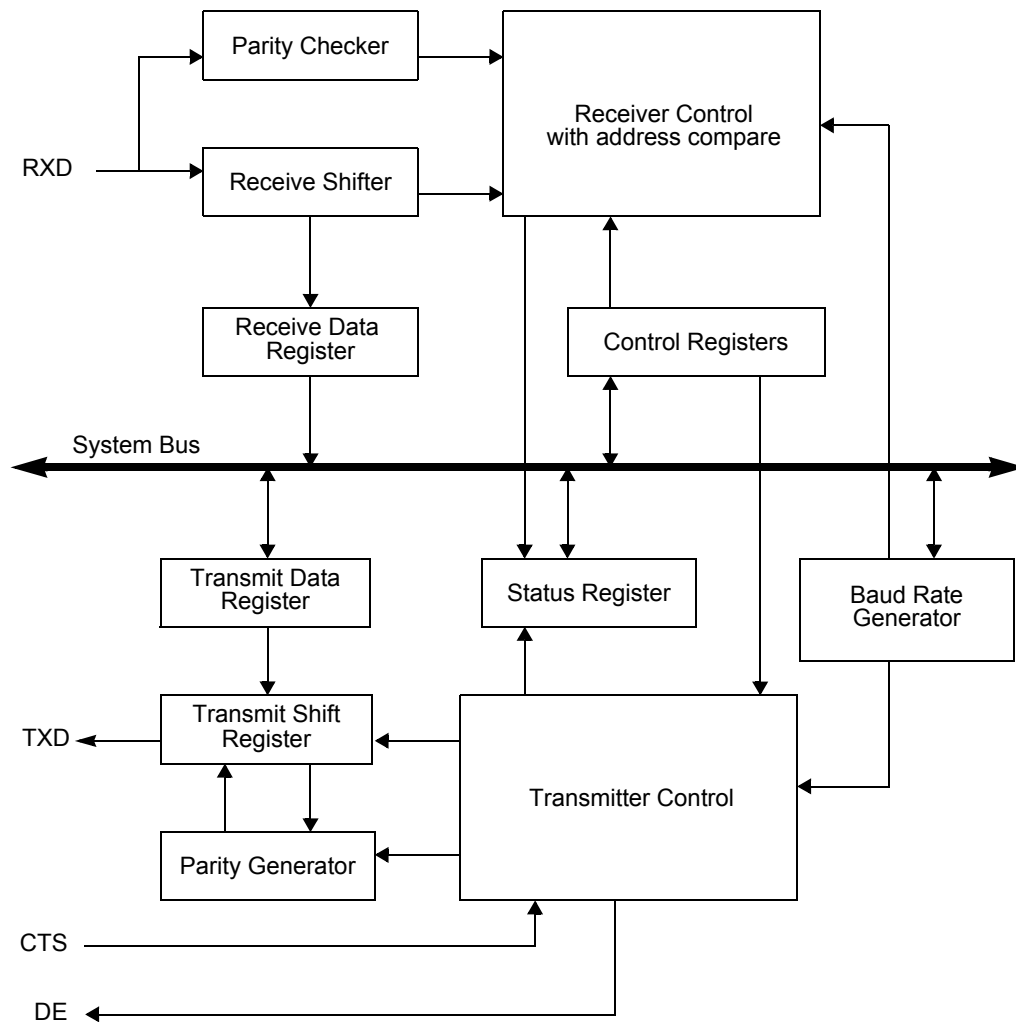


Figure 13. UART Block Diagram

Operation

Data Format

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 optionally added to the data stream. Each character begins with an active Low Start bit and ends with either 1 or 2 active High Stop bits.

[Figure 14](#) and [Figure 15](#) on page 105 displays the asynchronous data format employed by the UART without parity and with parity, respectively.

Receiving Data using the Interrupt-Driven Method

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

1. Write to the UART Baud Rate High and Low Byte registers to set the desired 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 desired 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 desired.
 - Set the MULTIPROCESSOR Mode Select (`MPEN`) to Enable MULTIPROCESSOR mode.
 - Set the MULTIPROCESSOR Mode Bits, `MPMD[1:0]`, to select the desired 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 desired 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 performs the following:

1. Check the UART Status 0 register to determine the source of the interrupt - error, break, or received data.
2. If the interrupt was caused by data available, read the 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]`.

(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 can function as a basic 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 register to 0.
2. Load the desired 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 on the infrared operation, see [Infrared Encoder/Decoder](#) on page 125.

UART Transmit Data Register

Data bytes written to the UART Transmit Data register ([Table 52](#)) 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 52. UART Transmit Data Register (UxTXD)

BITS	7	6	5	4	3	2	1	0
FIELD	TXD							
RESET	X							
R/W	W							
ADDR	F40H and F48H							

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. This action allows the Endec to tolerate jitter and baud rate errors in the incoming data stream. 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 [UART Control Register Definitions](#) on page 114.



Caution: *To prevent spurious signals during IrDA data transmission, set the `IREN` bit in the `UARTx Control 1` register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.*

During an SPI transfer, data is sent and received simultaneously by both the Master and the Slave SPI devices. Separate signals are required for data and the serial clock. When an SPI transfer occurs, a multi-bit (typically 8-bit) character is shifted out one data pin and a multi-bit character is simultaneously shifted in on a second data pin. An 8-bit shift register in the Master and another 8-bit shift register in the Slave are connected as a circular buffer. The SPI shift register is single-buffered in the transmit and receive directions. New data to be transmitted cannot be written into the shift register until the previous transmission is complete and receive data (if valid) has been read.

SPI Signals

The four basic SPI signals are:

- [Master-In/Slave-Out](#)
- [Master-Out/Slave-In](#)
- [Serial Clock](#)
- [Slave Select](#)

Each signal is described in both Master and Slave modes.

Master-In/Slave-Out

The Master-In/Slave-Out (MISO) pin is configured as an input in a Master device and as an output in a Slave device. It is one of the two lines that transfer serial data, with the most significant bit sent first. The MISO pin of a Slave device is placed in a high-impedance state if the Slave is not selected. When the SPI is not enabled, this signal is in a high-impedance state.

Master-Out/Slave-In

The Master-Out/Slave-In (MOSI) pin is configured as an output in a Master device and as an input in a Slave device. It is one of the two lines that transfer serial data, with the most significant bit sent first. When the SPI is not enabled, this signal is in a high-impedance state.

Serial Clock

The Serial Clock (SCK) synchronizes data movement both in and out of the device through its MOSI and MISO pins. In MASTER mode, the SPI's Baud Rate Generator creates the serial clock. The Master drives the serial clock out its own SCK pin to the Slave's SCK pin. When the SPI is configured as a Slave, the SCK pin is an input and the clock signal from the Master synchronizes the data transfer between the Master and Slave devices. Slave devices ignore the SCK signal, unless the \overline{SS} pin is asserted. When configured as a slave, the SPI block requires a minimum SCK period of greater than or equal to 8 times the system (XIN) clock period.

TXRXSTATE	State Description
1_1101	10-bit addressing: Bit 3 of 2nd address byte 7-bit addressing: Bit 3 of address byte
1_1110	10-bit addressing: Bit 2 of 2nd address byte 7-bit addressing: Bit 2 of address byte
1_1111	10-bit addressing: Bit 1 of 2nd address byte 7-bit addressing: Bit 1 of address byte

I²C Diagnostic Control Register

The I²C Diagnostic register (Table 76) provides control over diagnostic modes. This register is a read/write register used for I²C diagnostics.

Table 76. I²C Diagnostic Control Register (I2CDIAG)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved							DIAG
RESET	0							
R/W	R							R/W
ADDR	F56H							

DIAG = Diagnostic Control Bit - Selects read back value of the Baud Rate Reload registers.

0 = NORMAL mode. Reading the Baud Rate High and Low Byte registers returns the baud rate reload value.

1 = DIAGNOSTIC mode. Reading the Baud Rate High and Low Byte registers returns the baud rate counter value.

Flash Control Register Definitions

Flash Control Register

The Flash Control register ([Table 92](#)) unlocks the Flash Controller for programming and erase operations, or to select the Flash Sector Protect register.

The Write-only Flash Control Register shares its Register File address with the Read-only Flash Status Register.

Table 92. Flash Control Register (FCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	FCMD							
RESET	0							
R/W	W							
ADDR	FF8H							

FCMD—Flash Command

73H = First unlock command.

8CH = Second unlock command.

95H = Page erase command.

63H = Mass erase command

5EH = Flash Sector Protect register select.

* All other commands, or any command out of sequence, lock the Flash Controller.

Flash Status Register

The Flash Status register ([Table 93](#)) 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 93. Flash Status Register (FSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved		FSTAT					
RESET	0							
R/W	R							
ADDR	FF8H							

Option Bits

Overview

Option Bits allow user configuration of certain aspects of the 64K Series operation. The feature configuration data is stored in the Flash Memory and read during Reset. The features available for control via the Option Bits are:

- Watchdog Timer time-out response selection—interrupt or Reset.
- Watchdog Timer enabled at Reset.
- The ability to prevent unwanted read access to user code in Flash Memory.
- The ability to prevent accidental programming and erasure of the user code in Flash Memory.
- Voltage Brownout 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.

Operation

Option Bit Configuration By Reset

Each time the Option Bits are programmed or erased, the device must be Reset for the change to take place. During any reset operation (System Reset, Reset, or Stop Mode Recovery), the Option Bits are automatically read from the Flash Memory and written to Option Configuration registers. The Option Configuration registers control operation of the devices within the 64K 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 Address Space

The first two bytes of Flash Memory at addresses 0000H (see [Table 98](#) on page 196) and 0001H (see [Table 99](#) on page 197) are reserved for the user Option Bits. The byte at Flash Memory address 0000H configures user options. The byte at Flash Memory address 0001H is reserved for future use and must remain unprogrammed.

Figure 47 displays the maximum current consumption in STOP mode with the VBO and Watchdog Timer enabled versus the power supply voltage. All GPIO pins are configured as outputs and driven High.

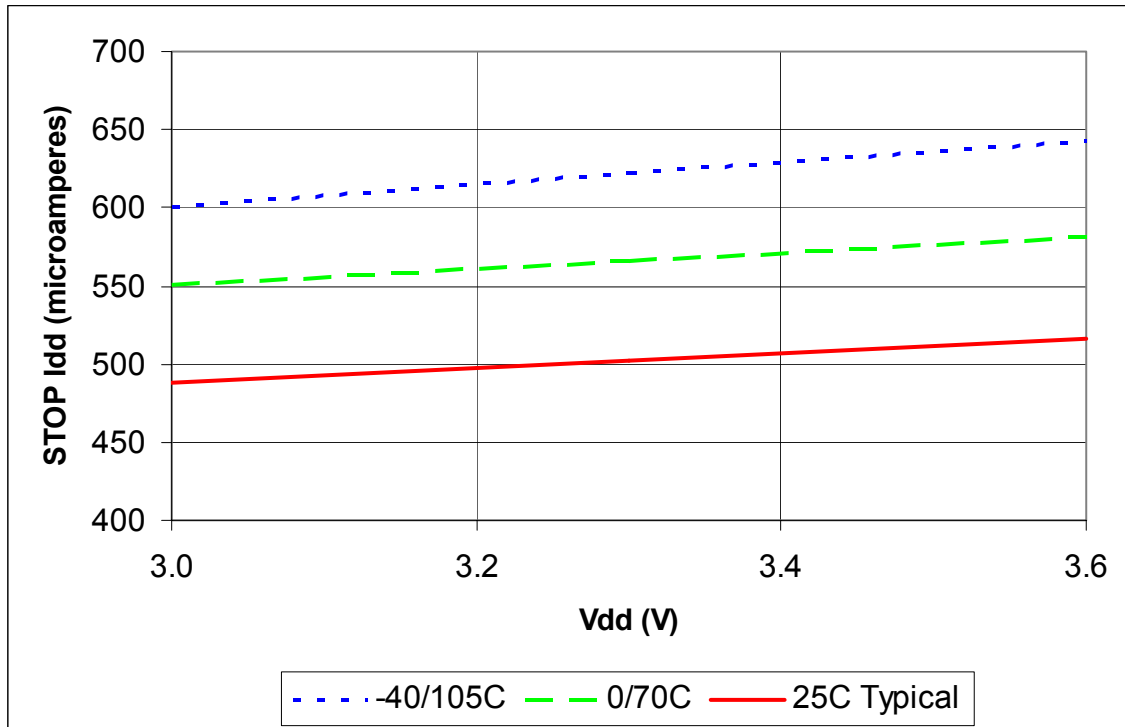


Figure 47. Maximum STOP Mode Idd with VBO enabled versus Power Supply Voltage

Table 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
COM dst	dst ← ~dst	R		60	-	*	*	0	-	-	2	2
		IR		61							2	3
CP dst, src	dst - src	r	r	A2	*	*	*	*	-	-	2	3
		r	lr	A3							2	4
		R	R	A4							3	3
		R	IR	A5							3	4
		R	IM	A6							3	3
		IR	IM	A7							3	4
CPC dst, src	dst - src - C	r	r	1F A2	*	*	*	*	-	-	3	3
		r	lr	1F A3							3	4
		R	R	1F A4							4	3
		R	IR	1F A5							4	4
		R	IM	1F A6							4	3
		IR	IM	1F A7							4	4
CPCX dst, src	dst - src - C	ER	ER	1F A8	*	*	*	*	-	-	5	3
		ER	IM	1F A9							5	3
CPX dst, src	dst - src	ER	ER	A8	*	*	*	*	-	-	4	3
		ER	IM	A9							4	3
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

Table 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Opcode(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
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	lr	63							2	4
		R	R	64							3	3
		R	IR	65							3	4
		R	IM	66							3	3
		IR	IM	67							3	4
TCMX dst, src	(NOT dst) AND src	ER	ER	68	-	*	*	0	-	-	4	3
		ER	IM	69							4	3
TM dst, src	dst AND src	r	r	72	-	*	*	0	-	-	2	3
		r	lr	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	-	*	*	0	-	-	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	-	-	-	-	-	-	2	6
WDT				5F	-	-	-	-	-	-	1	2

Ordering Information

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	I ² C	SPI	UARTs with IrDA	Description
Z8F642x with 64 KB Flash, 10-Bit Analog-to-Digital Converter										
Standard Temperature: 0 °C to 70 °C										
Z8F6421PM020SC	64 KB	4 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F6421AN020SC	64 KB	4 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F6421VN020SC	64 KB	4 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F6422AR020SC	64 KB	4 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F6422VS020SC	64 KB	4 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F6423FT020SC	64 KB	4 KB	60	24	4	12	1	1	2	QFP 80-pin package
Extended Temperature: –40 °C to +105 °C										
Z8F6421PM020EC	64 KB	4 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F6421AN020EC	64 KB	4 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F6421VN020EC	64 KB	4 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F6422AR020EC	64 KB	4 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F6422VS020EC	64 KB	4 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F6423FT020EC	64 KB	4 KB	60	24	4	12	1	1	2	QFP 80-pin package
Automotive/Industrial Temperature: –40 °C to +125 °C										
Z8F6421PM020AC	64 KB	4 KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F6421AN020AC	64 KB	4 KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F6421VN020AC	64 KB	4 KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F6422AR020AC	64 KB	4 KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F6422VS020AC	64 KB	4 KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F6423FT020AC	64 KB	4 KB	60	24	4	12	1	1	2	QFP 80-pin package