Zilog - Z8F6423FT020SC00TR Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

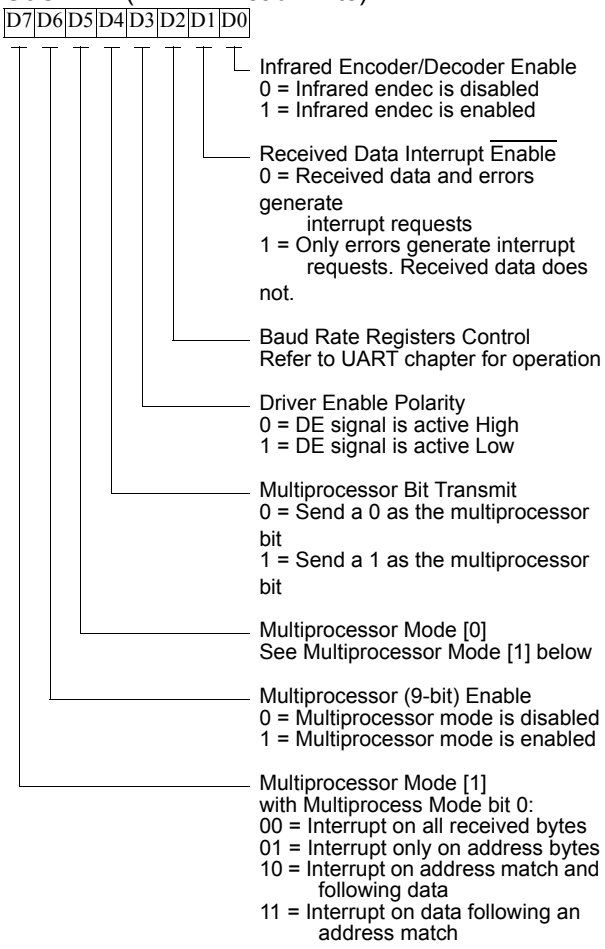
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	60
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	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6423ft020sc00tr

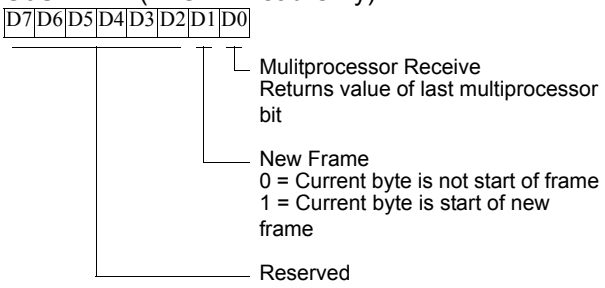
Table 7. Z8 Encore! XP 64K Series Flash Microcontrollers Register File Address Map (Continued)

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	

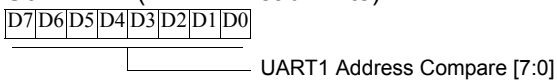
UART1 Control 1
U0CTL1 (F4BH - Read/Write)



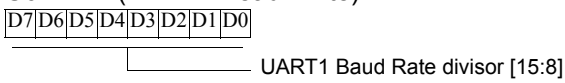
UART1 Status 1
U0STAT1 (F4CH - Read Only)



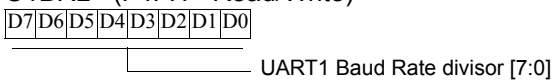
UART1 Address Compare
U0ADDR (F4DH - Read/Write)



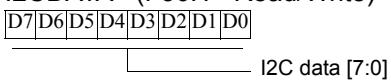
UART1 Baud Rate Generator High Byte
U0BRH (F4EH - Read/Write)



UART1 Baud Rate Generator Low Byte
U1BRL (F4FH - Read/Write)



I^2C Data
I2CDATA (F50H - Read/Write)



DMA1 Control
DMA1CTL (FB8H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Request Trigger Source Select
000 = Timer 0
001 = Timer 1
010 = Timer 2
011 = Timer 3
100 = UART0 Transmit Data register is empty
101 = UART1 Transmit Data register is empty
110 = I2C Transmit Data register is empty
111 = Reserved

Word Select
0 = DMA transfers 1 byte per request
1 = DMA transfers 2 bytes per request

DMA1 Interrupt Enable
0 = DMA1 does not generate interrupts
1 = DMA1 generates an interrupt when End Address data is transferred

DMA1 Data Transfer Direction
0 = Register File to peripheral registers
1 = Peripheral registers to Register File

DMA1 Loop Enable
0 = DMA disables after End Address
1 = DMA reloads Start Address after End Address and continues to run

DMA1 Enable
0 = DMA1 is disabled
1 = DMA1 is enabled

DMA1 I/O Address
DMA1IO (FB9H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

DMA1 Peripheral Register Address
Low byte of on-chip peripheral control registers on Register File page FH

DMA1 Address High Nibble
DMA1H (FBAH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

DMA1 Start Address [11:8]

DMA1 End Address [11:8]

DMA1 Start/Current Address Low Byte
DMA1START (FBBH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

DMA1 Start Address [7:0]

DMA1 End Address Low Byte
DMA1END (FBCH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

DMA1 End Address [7:0]

DMA_ADC Address
DMAA_ADDR (FBDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Reserved

DMA_ADC Address

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 WDT Control register is set.

Stop Mode Recovery

STOP mode is entered by the eZ8 executing a STOP instruction. For detailed STOP mode information, see Low-Power Modes on page 47. During Stop Mode Recovery, the devices are held in reset for 66 cycles of the Watchdog Timer oscillator followed by 16 cycles of the system clock. Stop Mode Recovery only affects the contents of the Watchdog Timer Control register. Stop Mode Recovery does not affect any other values in the Register File, including the Stack Pointer, Register Pointer, Flags, peripheral control registers, and general-purpose RAM.

The eZ8™ CPU fetches the Reset vector at Program Memory addresses 0002H and 0003H and loads that value into the Program Counter. Program execution begins at the Reset vector address. Following Stop Mode Recovery, the STOP bit in the Watchdog Timer Control Register is set to 1. Table 10 lists the Stop Mode Recovery sources and resulting actions.

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

Stop Mode Recovery Using Watchdog Timer Time-Out

If the Watchdog Timer times out during STOP mode, the device undergoes a Stop Mode Recovery sequence. In the Watchdog Timer Control register, the WDT and STOP bits are set to 1. If the Watchdog Timer is configured to generate an interrupt upon time-out and the 64K Series devices are configured to respond to interrupts, the eZ8 CPU services the Watchdog Timer interrupt request following the normal Stop Mode Recovery sequence.

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							

T1I—Timer 1 Interrupt Request
0 = No interrupt request is pending for Timer 1.
1 = An interrupt request from Timer 1 is awaiting service.

T0I—Timer 0 Interrupt Request
0 = No interrupt request is pending for Timer 0.
1 = An interrupt request from Timer 0 is awaiting service.

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.

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.

I^2CI— I^2C Interrupt Request
0 = No interrupt request is pending for the I^2C.
1 = An interrupt request from the I^2C is awaiting service.

SPII—SPI Interrupt Request
0 = No interrupt request is pending for the SPI.
1 = An interrupt request from the SPI is awaiting service.

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 (Table 25) 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 25. Interrupt Request 1 Register (IRQ1)

BITS	7	6	5	4	3	2	1	0
FIELD	PAD7I	PAD6I	PAD5I	PAD4I	PAD3I	PAD2I	PAD1I	PAD0I
RESET	0							
R/W	R/W							
ADDR	FC3H							

Table 47. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10 kHz typical WDT oscillator frequency)	
		Typical	Description
000004	4	400 µs	Minimum time-out delay
FFFFFF	16,777,215	1677.5 s	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 then 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 64K Series devices are operating in DEBUG Mode (through the On-Chip Debugger), the Watchdog Timer is continuously refreshed to prevent spurious 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 Reset. The WDT_RES Option Bit determines the time-out response of the Watchdog Timer. For information on programming of the WDT_RES Option Bit, see Option Bits on page 195.

WDT Interrupt in Normal Operation

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

WDT Interrupt in STOP Mode

If configured to generate an interrupt when a time-out occurs and the 64K 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 Watchdog Timer Control register are set to 1 following WDT time-out in STOP mode. For more information on Stop Mode Recovery, see Reset and Stop Mode Recovery on page 47.

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 Reset state. The WDT status bit in the Watchdog Timer Control register is set to 1. For more information on Reset, see Reset and Stop Mode Recovery on page 47.

WDT Reset in STOP Mode

If enabled in STOP mode and configured to generate a Reset when a time-out occurs and the device is in STOP mode, the Watchdog Timer initiates a Stop Mode Recovery. Both the WDT status bit and the STOP bit in the Watchdog Timer Control register are set to 1 following WDT time-out in STOP mode. Default operation is for the WDT and its RC oscillator to be enabled during STOP mode.

WDT RC Disable in STOP Mode

To minimize power consumption in STOP Mode, the WDT and its RC oscillator can be disabled in STOP mode. The following sequence configures the WDT to be disabled when the 64K Series devices enter STOP Mode following execution of a STOP instruction:

1. Write 55H to the Watchdog Timer Control register (WDTCTL).
2. Write AAH to the Watchdog Timer Control register (WDTCTL).
3. Write 81H to the Watchdog Timer Control register (WDTCTL) to configure the WDT and its oscillator to be disabled during STOP Mode. Alternatively, write 00H to the Watchdog Timer Control register (WDTCTL) as the third step in this sequence to reconfigure the WDT and its oscillator to be enabled during STOP mode.

This sequence only affects WDT operation 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. Follow the steps below 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).
4. Write the Watchdog Timer Reload High Byte register (WDTH).

Table 55. UART Status 1 Register (UxSTAT1)

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved						NEWFRM	MPRX
RESET	0							
R/W	R				R/W		R	
ADDR	F44H and F4CH							

Reserved—Must be 0.

NEWFRM—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.

MPRX—Multiprocessor Receive
Returns the value of the last multiprocessor bit received. Reading from the UART Receive Data register resets this bit to 0.

UART Control 0 and Control 1 Registers

The UART Control 0 and Control 1 registers (see Table 56 and Table 57 on page 118) configure the properties of the UART's transmit and receive operations. The UART Control registers must not been written while the UART is enabled.

Table 56. UART Control 0 Register (UxCTL0)

BITS	7	6	5	4	3	2	1	0
FIELD	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
RESET	0							
R/W	R/W							
ADDR	F42H and F4AH							

TEN—Transmit Enable
This bit enables or disables the transmitter. The enable is also controlled by the $\overline{\text{CTS}}$ signal and the CTSE bit. If the $\overline{\text{CTS}}$ signal is low and the CTSE bit is 1, the transmitter is enabled.
0 = Transmitter disabled.
1 = Transmitter enabled.

Table 59. UART Baud Rate High Byte Register (U*x*BRH)

BITS	7	6	5	4	3	2	1	0
FIELD	BRH							
RESET	1							
R/W	R/W							
ADDR	F46H and F4EH							

Table 60. UART Baud Rate Low Byte Register (U*x*BRL)

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
RESET	1							
R/W	R/W							
ADDR	F47H and F4FH							

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

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

The baud rate error relative to the desired 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 61 provides information on data rate errors for popular baud rates and commonly used crystal oscillator frequencies.

Table 70. I²C Data Register (I2CDATA)

BITS	7	6	5	4	3	2	1	0
FIELD	DATA							
RESET	0							
R/W	R/W							
ADDR	F50H							

I²C Status Register

The Read-only I²C Status register (Table 71) indicates the status of the I²C Controller.

Table 71. I²C Status Register (I2CSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	TDRE	RDRF	ACK	10B	RD	TAS	DSS	NCKI
RESET	1	0						
R/W	R							
ADDR	F51H							

TDRE—Transmit Data Register Empty
When the I²C Controller is enabled, this bit is 1 when the I²C Data register is empty. When this bit is set, an interrupt is generated if the TXI bit is set, except when the I²C Controller is shifting in data during the reception of a byte or when shifting an address and the RD bit is set. This bit is cleared by writing to the I2CDATA register.

RDRF—Receive Data Register Full
This bit is set = 1 when the I²C Controller is enabled and the I²C Controller has received a byte of data. When asserted, this bit causes the I²C Controller to generate an interrupt. This bit is cleared by reading the I²C Data register (unless the read is performed using execution of the On-Chip Debugger's Read Register command).

ACK—Acknowledge
This bit indicates the status of the Acknowledge for the last byte transmitted or received. When set, this bit indicates that an Acknowledge occurred for the last byte transmitted or received. This bit is cleared when IEN = 0 or when a Not Acknowledge occurred for the last byte transmitted or received. It is not reset at the beginning of each transaction and is not reset when this register is read.

Table 74. I²C Baud Rate Low Byte Register (I2CBRL)

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
RESET	FFH							
R/W	R/W							
ADDR	F54H							

BRL = I²C Baud Rate Low Byte
Least significant byte, BRG[7:0], of the I²C Baud Rate Generator's reload value.

Note: *If the DIAG bit in the I²C Diagnostic Control Register is set to 1, a read of the I2CBRL register returns the current value of the I²C Baud Rate Counter[7:0].*

I²C Diagnostic State Register

The I²C Diagnostic State register (Table 75) provides observability of internal state. This is a read only register used for I²C diagnostics and manufacturing test.

Table 75. I²C Diagnostic State Register (I2CDST)

BITS	7	6	5	4	3	2	1	0
FIELD	SCLIN	SDAIN	STPCNT	TXRXSTATE				
RESET	X		0					
R/W	R							
ADDR	F55H							

SCLIN—Value of Serial Clock input signal

SDAIN—Value of the Serial Data input signal

STPCNT—Value of the internal Stop Count control signal

TXRXSTATE—Value of the internal I²C state machine

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.

Direct Memory Access Controller

Overview

The 64K Series Direct Memory Access (DMA) Controller provides three independent Direct Memory Access channels. Two of the channels (DMA0 and DMA1) transfer data between the on-chip peripherals and the Register File. The third channel (DMA_ADC) controls the ADC operation and transfers SINGLE-SHOT mode ADC output data to the Register File.

Operation

DMA0 and DMA1 Operation

DMA0 and DMA1, referred to collectively as DMA*x*, transfer data either from the on-chip peripheral control registers to the Register File, or from the Register File to the on-chip peripheral control registers. The sequence of operations in a DMA*x* data transfer is:

1. DMA*x* trigger source requests a DMA data transfer.
2. DMA*x* requests control of the system bus (address and data) from the eZ8 CPU.
3. After the eZ8 CPU acknowledges the bus request, DMA*x* transfers either a single byte or a two-byte word (depending upon configuration) and then returns system bus control back to the eZ8 CPU.
4. If Current Address equals End Address:
   – DMA*x* reloads the original Start Address
   – If configured to generate an interrupt, DMA*x* sends an interrupt request to the Interrupt Controller
   – If configured for single-pass operation, DMA*x* resets the DEN bit in the DMA*x* Control register to 0 and the DMA is disabled.

   If Current Address does not equal End Address, the Current Address increments by 1 (single-byte transfer) or 2 (two-byte word transfer).

Analog-to-Digital Converter

Overview

The Analog-to-Digital Converter (ADC) converts an analog input signal to a 10-bit binary number. The features of the sigma-delta ADC include:

- 12 analog input sources are multiplexed with general-purpose I/O ports
- Interrupt upon conversion complete
- Internal voltage reference generator
- Direct Memory Access (DMA) controller can automatically initiate data conversion and transfer of the data from 1 to 12 of the analog inputs

Architecture

Figure 34 displays the three major functional blocks (converter, analog multiplexer, and voltage reference generator) of the ADC. The ADC converts an analog input signal to its digital representation. The 12-input analog multiplexer selects one of the 12 analog input sources. The ADC requires an input reference voltage for the conversion. The voltage reference for the conversion may be input through the external VREF pin or generated internally by the voltage reference generator.

Figure 48 displays the maximum current consumption in STOP mode with the VBO disabled and Watchdog Timer enabled versus the power supply voltage. All GPIO pins are configured as outputs and driven High. Disabling the Watchdog Timer and its internal RC oscillator in STOP mode will provide some additional reduction in STOP mode current consumption. This small current reduction would be indistinquishable on the scale of Figure 48.

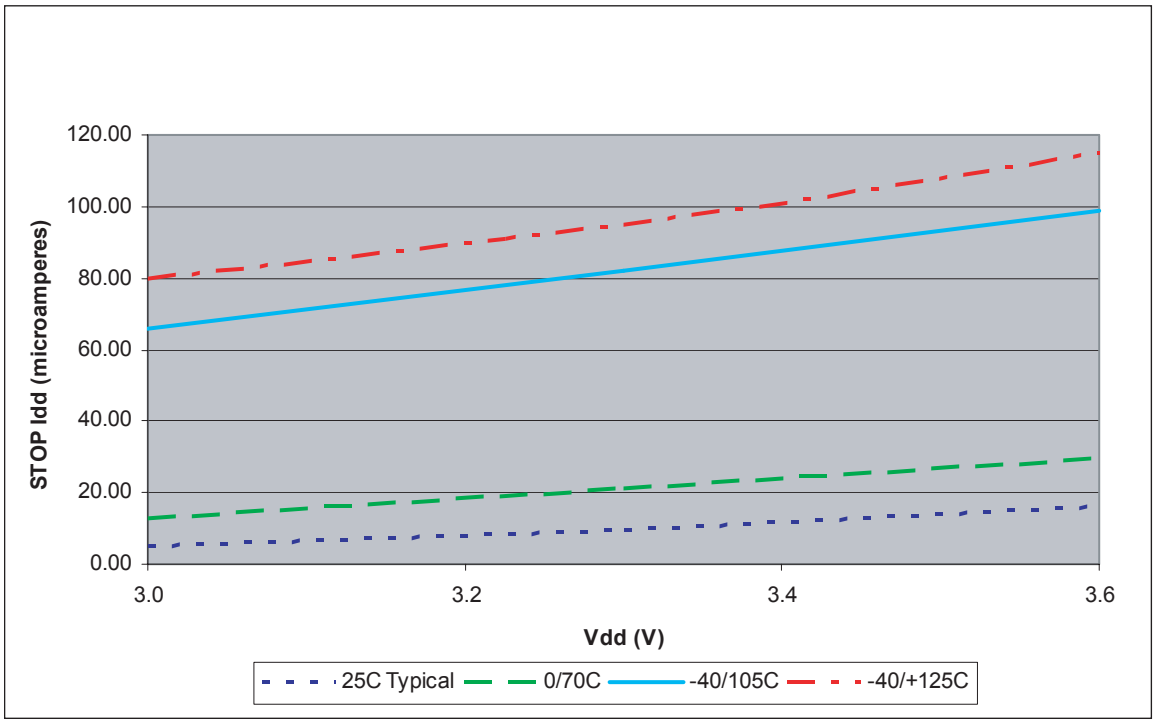


Figure 48. Maximum STOP Mode Idd with VBO Disabled versus Power Supply Voltage

AC Characteristics

The section provides information on the AC characteristics and timing. All AC timing information assumes a standard load of 50 pF on all outputs. Table 113 lists the 64K Series AC characteristics and timing.

Table 113. AC Characteristics

Symbol	Parameter	V_{DD} = 3.0–3.6V T_A = –40 °C to 125 °C		Units	Conditions
		Minimum	Maximum		
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	0.032768	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
T_{XIN}	Crystal Oscillator Clock Period	50	–	ns	$T_{CLK} = 1/F_{sysclk}$
T_{XINH}	System Clock High Time	20		ns	
T_{XINL}	System Clock Low Time	20		ns	
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50 ns. Slower rise times can be tolerated with longer clock periods.
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50 ns. Slower fall times can be tolerated with longer clock periods.

I²C Timing

Figure 55 and Table 119 provide timing information for I²C pins.

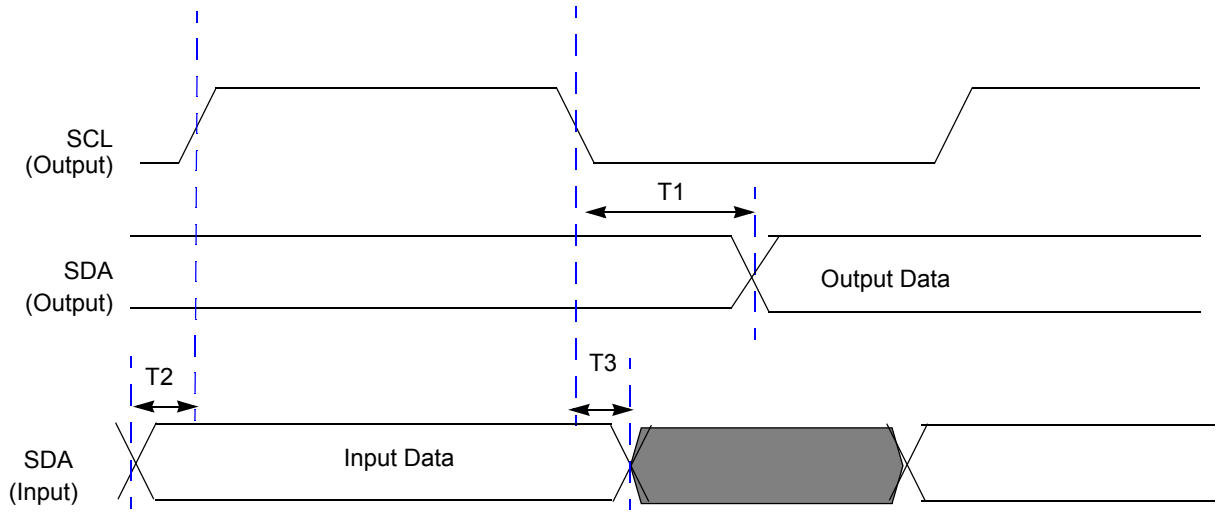


Figure 55. I²C Timing

Table 119. I²C Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
I²C			
T_1	SCL Fall to SDA output delay		SCL period/4
T_2	SDA Input to SCL rising edge Setup Time	0	
T_3	SDA Input to SCL falling edge Hold Time	0	

Table 123. 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. For example,

dst ← dst + src

indicates the source data is added to the destination data and the result is stored in the destination location.

Condition Codes

The C, Z, S and V Flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the Flag settings are encoded in a 4-bit field called the condition code (cc), which forms Bits 7:4 of the conditional jump instructions. The condition codes are summarized in Table 124. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the Flag test operation decides if the conditional jump is executed.

Table 124. Condition Codes

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	–
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1