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/z8f6423ft020sc

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.

Block Diagram

Figure 1 displays the block diagram of the architecture of the Z8 Encore! XP 64K Series Flash Microcontrollers.

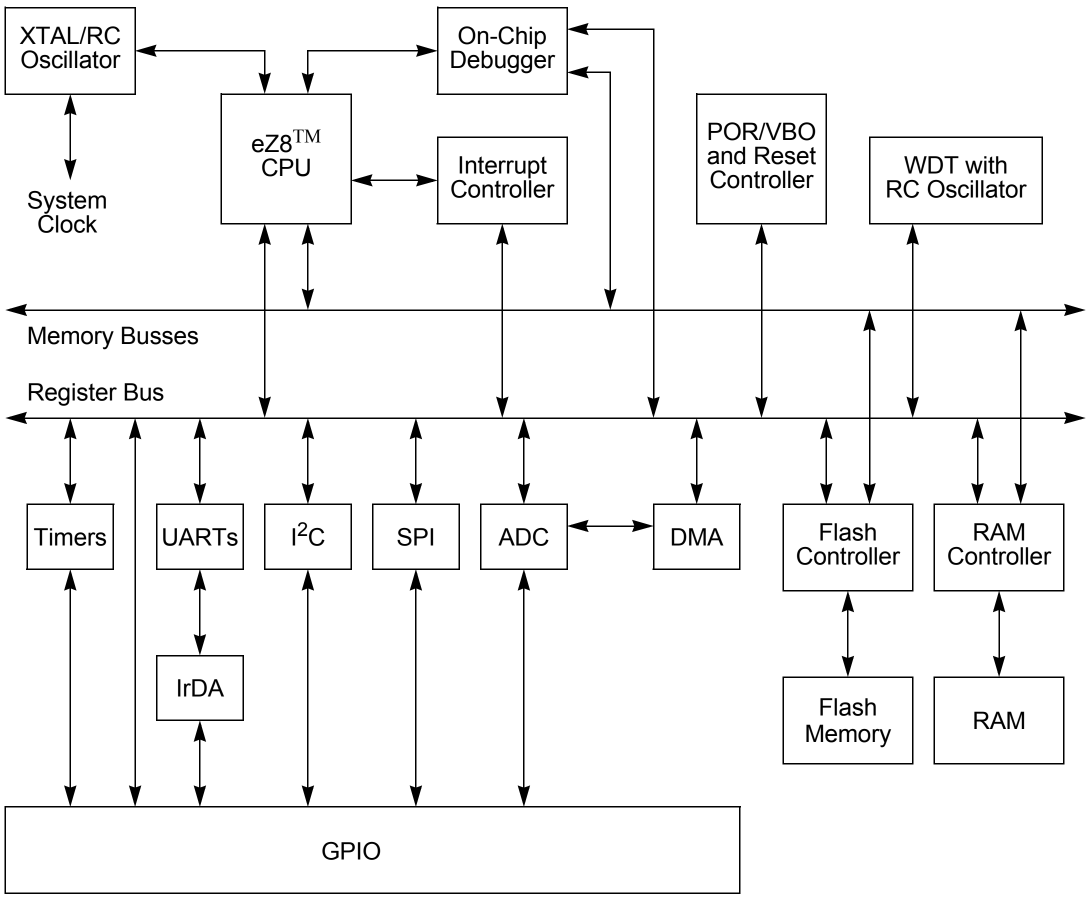

Figure 1. Z8 Encore! XP 64K Series Flash Microcontrollers Block Diagram

CPU and Peripheral Overview

eZ8™ CPU Features

The latest 8-bit eZ8 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 eZ8 CPU features include:

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

For more information on the eZ8 CPU, refer to *eZ8™ CPU Core User Manual (UM0128)* available for download at www.zilog.com.

General-Purpose Input/Output

The 64K Series features seven 8-bit ports (Ports A-G) and one 4-bit port (Port H) for general-purpose input/output (GPIO). Each pin is individually programmable. All ports (except B and H) support 5 V-tolerant inputs.

Flash Controller

The Flash Controller programs and erases the Flash memory.

10-Bit Analog-to-Digital Converter

The Analog-to-Digital Converter converts an analog input signal to a 10-bit binary number. The ADC accepts inputs from up to 12 different analog input sources.

UARTs

Each UART is full-duplex and capable of handling asynchronous data transfers. The UARTs support 8- and 9-bit data modes, selectable parity, and an efficient bus transceiver Driver Enable signal for controlling a multi-transceiver bus, such as RS-485.

IRQ0 Enable Low Bit
IRQ0ENL (FC2H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D0: ADC IRQ Enable Hit Bit
- D1: SPI IRQ Enable Low Bit
- D2: I2C IRQ Enable Low Bit
- D3: UART 0 Transmitter IRQ Enable
- D4: UART 0 Receiver IRQ Enable Low
- D5: Timer 0 IRQ Enable Low Bit
- D6: Timer 1 IRQ Enable Low Bit
- D7: Timer 2 IRQ Enable Low Bit

Interrupt Request 1
IRQ1 (FC3H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D7–D0: Port A or D Pin Interrupt Request
  0 = IRQ from corresponding pin [7:0] is not pending
  1 = IRQ from corresponding pin [7:0] is awaiting service

IRQ1 Enable High Bit
IRQ1ENH (FC4H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D7–D0: Port A or D Pin IRQ Enable High Bit

IRQ1 Enable Low Bit
IRQ1ENL (FC5H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D7–D0: Port A or D Pin IRQ Enable Low Bit

Interrupt Request 2
IRQ2 (FC6H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D3–D0: Port C Pin Interrupt Request
  0 = IRQ from corresponding pin [3:0] is not pending
  1 = IRQ from corresponding pin [3:0] is awaiting service
- D4: DMA Interrupt Request
- D5: UART 1 Transmitter Interrupt
- D6: UART 1 Receiver Interrupt Request
- D7: Timer 3 Interrupt Request

For all of the above peripherals:
0 = Peripheral IRQ is not pending
1 = Peripheral IRQ is awaiting service

IRQ2 Enable High Bit
IRQ2ENH (FC7H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D3–D0: Port C Pin IRQ Enable High Bit
- D4: DMA IRQ Enable High Bit
- D5: UART 1 Transmitter IRQ Enable
- D6: UART 1 Receiver IRQ Enable High
- D7: Timer 3 IRQ Enable High Bit

IRQ2 Enable Low Bit
IRQ2ENL (FC8H - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D3–D0: Port C Pin IRQ Enable Low Bit
- D4: DMA IRQ Enable Low Bit
- D5: UART 1 Transmitter IRQ Enable
- D6: UART 1 Receiver IRQ Enable Low
- D7: Timer 3 IRQ Enable Low Bit

Interrupt Edge Select
IRQES (FCDH - Read/Write)

D7 D6 D5 D4 D3 D2 D1 D0

- D7–D0: Port A or D Interrupt Edge Select
  0 = Falling edge
  1 = Rising edge

Table 9. Reset Sources and Resulting Reset Type

Operating Mode	Reset Source	Reset Type
NORMAL or HALT modes	Power-On Reset/Voltage Brownout	system reset
	Watchdog Timer time-out when configured for Reset	system reset
	$\overline{\text{RESET}}$ pin assertion	system reset
	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 Brownout	system reset
	$\overline{\text{RESET}}$ pin assertion	system reset
	DBG pin driven Low	system reset

Power-On Reset

Each device in the 64K Series contains 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 POR Counter is enabled and counts 66 cycles of the Watchdog Timer oscillator. After the POR counter times out, the XTAL Counter is enabled to count a total of 16 system clock pulses. The devices are held in the Reset state until both the POR Counter and XTAL counter have timed out. After the 64K Series devices exit the Power-On Reset state, the eZ8 CPU fetches the Reset vector. Following Power-On Reset, the POR status bit in the Watchdog Timer Control (WDTCTL) register is set to 1.

Figure 8 displays Power-On Reset operation. For the POR threshold voltage (V_{POR}), see Electrical Characteristics on page 215.

Port A–H Data Direction Sub-Registers

The Port A–H Data Direction sub-register is accessed through the Port A–H Control register by writing 01H to the Port A–H Address register (Table 16).

Table 16. Port A–H Data Direction Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET	1							
R/W	R/W							
ADDR	If 01H in Port A–H Address Register, accessible through Port A–H Control Register							

DD[7:0]—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–H Output Data register is driven onto the port pin.
1 = Input. The port pin is sampled and the value written into the Port A–H Input Data Register. The output driver is tri-stated.

Port A–H Alternate Function Sub-Registers

The Port A–H Alternate Function sub-register (Table 17) is accessed through the Port A–H Control register by writing `02H` to the Port A–H Address register. The Port A–H Alternate Function sub-registers select the alternate functions for the selected pins. To determine the alternate function associated with each port pin, see GPIO Alternate Functions on page 59.

Caution: *Do not enable alternate function for GPIO port pins which do not have an associated alternate function. Failure to follow this guideline may result in unpredictable operation.*

Table 17. Port A–H Alternate Function Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	AF7	AF6	AF5	AF4	AF3	AF2	AF1	AF0
RESET	0							
R/W	R/W							
ADDR	If 02H in Port A–H Address Register, accessible through Port A–H Control Register							

where *x* indicates the specific GPIO Port C pin number (0 through 3).

IRQ0 Enable High and Low Bit Registers

The IRQ0 Enable High and Low Bit registers (see Table 28 and Table 29 on page 75) form a priority encoded enabling for interrupts in the Interrupt Request 0 register. Priority is generated by setting bits in each register. Table 27 describes the priority control for IRQ0.

Table 27. IRQ0 Enable and Priority Encoding

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

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

Table 28. IRQ0 Enable High Bit Register (IRQ0ENH)

BITS	7	6	5	4	3	2	1	0
FIELD	T2ENH	T1ENH	T0ENH	U0RENH	U0TENH	I2CENH	SPIENH	ADCENH
RESET	0							
R/W	R/W							
ADDR	FC1H							

T2ENH—Timer 2 Interrupt Request Enable High Bit
T1ENH—Timer 1 Interrupt Request Enable High Bit
T0ENH—Timer 0 Interrupt Request Enable High Bit
U0RENH—UART 0 Receive Interrupt Request Enable High Bit
U0TENH—UART 0 Transmit Interrupt Request Enable High Bit
I2CENH—I^2C Interrupt Request Enable High Bit
SPIENH—SPI Interrupt Request Enable High Bit
ADCENH—ADC Interrupt Request Enable High Bit

One-Shot time-out, first set the TPOL bit in the Timer Control 1 Register to the start value before beginning ONE-SHOT mode. Then, after starting the timer, set TPOL to the opposite bit value.

Follow the steps below for configuring a timer for ONE-SHOT mode and initiating the count:

1. Write to the Timer Control 1 register to:
   - Disable the timer
   - Configure the timer for ONE-SHOT 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
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. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output alternate function
6. Write to the Timer Control 1 register to enable the timer and initiate counting

In ONE-SHOT mode, the system clock always provides the timer input. The timer period is given by 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) upon timer Reload.

Follow the steps below for configuring a timer for CONTINUOUS mode and initiating the count:

1. Write to the Timer Control 1 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)

Table 42. Timer 0-3 Reload Low Byte Register (TxRL)

BITS	7	6	5	4	3	2	1	0
FIELD	TRL							
RESET	1							
R/W	R/W							
ADDR	F03H, F0BH, F13H, F1BH							

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

Timer 0-3 PWM High and Low Byte Registers

The Timer 0-3 PWM High and Low Byte (TxPWMH and TxPWML) registers (see Table 43 and Table 44 on page 92) are used for Pulse-Width Modulator (PWM) operations. These registers also store the Capture values for the Capture and Capture/COMPARE modes.

Table 43. Timer 0-3 PWM High Byte Register (TxPWMH)

BITS	7	6	5	4	3	2	1	0
FIELD	PWMH							
RESET	0							
R/W	R/W							
ADDR	F04H, F0CH, F14H, F1CH							

Table 44. Timer 0-3 PWM Low Byte Register (TxPWML)

BITS	7	6	5	4	3	2	1	0
FIELD	PWML							
RESET	0							
R/W	R/W							
ADDR	F05H, F0DH, F15H, F1DH							

Write Transaction with a 7-Bit Address

Figure 29 displays the data transfer format for a 7-bit addressed slave. Shaded regions indicate data transferred from the I^{2}C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I^{2}C Controller.

S	Slave Address	W = 0	A	Data	A	Data	A	Data	A/$\overline{\text{A}}$	P/S

Figure 29. 7-Bit Addressed Slave Data Transfer Format

Follow the steps below for a transmit operation to a 7-bit addressed slave:

1. Software asserts the `IEN` bit in the I^{2}C Control register.
2. Software asserts the `TXI` bit of the I^{2}C Control register to enable Transmit interrupts.
3. The I^{2}C interrupt asserts, because the I^{2}C Data register is empty
4. Software responds to the `TDRE` bit by writing a 7-bit slave address plus write bit (=0) to the I^{2}C Data register.
5. Software asserts the START bit of the I^{2}C Control register.
6. The I^{2}C Controller sends the START condition to the I^{2}C slave.
7. The I^{2}C Controller loads the I^{2}C Shift register with the contents of the I^{2}C Data register.
8. After one bit of address has been shifted out by the SDA signal, the Transmit interrupt is asserted (TDRE = 1).
9. Software responds by writing the transmit data into the I^{2}C Data register.
10. The I^{2}C Controller shifts the rest of the address and write bit out by the SDA signal.
11. If the I^{2}C slave sends an acknowledge (by pulling the SDA signal low) during the next high period of SCL the I^{2}C Controller sets the `ACK` bit in the I^{2}C Status register. Continue with step 12.

    If the slave does not acknowledge, the Not Acknowledge interrupt occurs (NCKI bit is set in the Status register, ACK bit is cleared). Software responds to the Not Acknowledge interrupt by setting the STOP and FLUSH bits and clearing the TXI bit. The I^{2}C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore the following steps).
12. The I^{2}C Controller loads the contents of the I^{2}C Shift register with the contents of the I^{2}C Data register.

7. The I^{2}C Controller loads the I^{2}C Shift register with the contents of the I^{2}C Data register.
8. After one bit of address is shifted out by the SDA signal, the Transmit interrupt is asserted.
9. Software responds by writing the second byte of address into the contents of the I^{2}C Data register.
10. The I^{2}C Controller shifts the rest of the first byte of address and write bit out the SDA signal.
11. If the I^{2}C slave sends an acknowledge by pulling the SDA signal low during the next high period of SCL the I^{2}C Controller sets the ACK bit in the I^{2}C Status register. Continue with step 12.

    If the slave does not acknowledge the first address byte, the I^{2}C Controller sets the NCKI bit and clears the ACK bit in the I^{2}C Status register. Software responds to the Not Acknowledge interrupt by setting the STOP and FLUSH bits and clearing the TXI bit. The I^{2}C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore following steps).
12. The I^{2}C Controller loads the I^{2}C Shift register with the contents of the I^{2}C Data register (2nd byte of address).
13. The I^{2}C Controller shifts the second address byte out the SDA signal. After the first bit has been sent, the Transmit interrupt is asserted.
14. Software responds by setting the STOP bit in the I^{2}C Control register. The TXI bit can be cleared at the same time.
15. Software polls the STOP bit of the I^{2}C Control register. Hardware deasserts the STOP bit when the transaction is completed (STOP condition has been sent).
16. Software checks the ACK bit of the I^{2}C Status register. If the slave acknowledged, the ACK bit is = 1. If the slave does not acknowledge, the ACK bit is = 0. The NCKI interrupt do not occur because the STOP bit was set.

Write Transaction with a 10-Bit Address

Figure 31 displays the data transfer format for a 10-bit addressed slave. Shaded regions indicate data transferred from the I^{2}C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I^{2}C Controller.

S	Slave Address 1st 7 bits	W = 0	A	Slave Address 2nd Byte	A	Data	A	Data	A/$\overline{A}$	P/S

Figure 31. 10-Bit Addressed Slave Data Transfer Format

IEN—I^{2}C Enable
1 = The I^{2}C transmitter and receiver are enabled.
0 = The I^{2}C transmitter and receiver are disabled.

START—Send Start Condition
This bit sends the Start condition. Once asserted, it is cleared by the I^{2}C Controller after it sends the START condition or if the `IEN` bit is deasserted. If this bit is 1, it cannot be cleared to 0 by writing to the register. After this bit is set, the Start condition is sent if there is data in the I^{2}C Data or I^{2}C Shift register. If there is no data in one of these registers, the I^{2}C Controller waits until the Data register is written. If this bit is set while the I^{2}C Controller is shifting out data, it generates a START condition after the byte shifts and the acknowledge phase completes. If the `STOP` bit is also set, it also waits until the STOP condition is sent before the sending the START condition.

STOP—Send Stop Condition
This bit causes the I^{2}C Controller to issue a Stop condition after the byte in the I^{2}C Shift register has completed transmission or after a byte has been received in a receive operation. Once set, this bit is reset by the I^{2}C Controller after a Stop condition has been sent or by deasserting the `IEN` bit. If this bit is 1, it cannot be cleared to 0 by writing to the register.

BIRQ—Baud Rate Generator Interrupt Request
This bit allows the I^{2}C Controller to be used as an additional timer when the I^{2}C Controller is disabled. This bit is ignored when the I^{2}C Controller is enabled.
1 = An interrupt occurs every time the baud rate generator counts down to one.
0 = No baud rate generator interrupt occurs.

TXI—Enable TDRE interrupts
This bit enables the transmit interrupt when the I^{2}C Data register is empty (TDRE = 1).
1 = Transmit interrupt (and DMA transmit request) is enabled.
0 = Transmit interrupt (and DMA transmit request) is disabled.

NAK—Send NAK
This bit sends a Not Acknowledge condition after the next byte of data has been read from the I^{2}C slave. Once asserted, it is deasserted after a Not Acknowledge is sent or the `IEN` bit is deasserted. If this bit is 1, it cannot be cleared to 0 by writing to the register.

FLUSH—Flush Data
Setting this bit to 1 clears the I^{2}C Data register and sets the TDRE bit to 1. This bit allows flushing of the I^{2}C Data register when a Not Acknowledge interrupt is received after the data has been sent to the I^{2}C Data register. Reading this bit always returns 0.

FILTEN—I^{2}C Signal Filter Enable
This bit enables low-pass digital filters on the SDA and SCL input signals. These filters reject any input pulse with periods less than a full system clock cycle. The filters introduce a 3-system clock cycle latency on the inputs.
1 = low-pass filters are enabled.
0 = low-pass filters are disabled.

Table 88. ADC Data Low Bits Register (ADCD_L)

BITS	7	6	5	4	3	2	1	0
FIELD	ADCD_L		Reserved					
RESET	X							
R/W	R							
ADDR	F73H							

ADCD_L—ADC Data Low Bits
These are the least significant two bits of the 10-bit ADC output. These bits are undefined after a Reset.

Reserved
These bits are reserved and are always undefined.

Reserved
These bits are reserved and must be 0.

FSTAT—Flash Controller Status
00_0000 = Flash Controller locked
00_0001 = First unlock command received
00_0010 = Second unlock command received
00_0011 = Flash Controller unlocked
00_0100 = Flash Sector Protect register selected
00_1xxx = Program operation in progress
01_0xxx = Page erase operation in progress
10_0xxx = Mass erase operation in progress

Page Select Register

The Page Select (FPS) register (Table 94) selects one of the 128 available Flash memory pages to be erased or programmed. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory locations with the 7 most significant bits of the address given by the PAGE field are erased to FFH.

The Page Select register shares its Register File address with the Flash Sector Protect Register. The Page Select register cannot be accessed when the Flash Sector Protect register is enabled.

Table 94. Page Select Register (FPS)

BITS	7	6	5	4	3	2	1	0
FIELD	INFO_EN	PAGE						
RESET	0							
R/W	R/W							
ADDR	FF9H							

INFO_EN—Information Area Enable
0 = Information Area is not selected.
1 = Information Area is selected. The Information area is mapped into the Flash Memory address space at addresses FE00H through FFFFH.

PAGE—Page Select
This 7-bit field selects the Flash memory page for Programming and Page Erase operations. Flash Memory Address[15:9] = PAGE[6:0].

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.

Reserved
These Option Bits are reserved for future use and must always be 1.This setting is the default for unprogrammed (erased) Flash.

FWP—Flash Write Protect (Flash version only)

FWP	Description
0	Programming, Page Erase, and Mass Erase through User Code is disabled. Mass Erase is available through the On-Chip Debugger.
1	Programming, and Page Erase are enabled for all of Flash Program Memory.

Flash Memory Address 0001H

Table 99. Options Bits at Flash Memory Address 0001H

BITS	7	6	5	4	3	2	1	0
FIELD	Reserved							
RESET	U							
R/W	R/W							
ADDR	Program Memory 0001H							
Note: U = Unchanged by Reset. R = Read-Only. R/W = Read/Write.								

Reserved
These Option Bits are reserved for future use and must always be 1. This setting is the default for unprogrammed (erased) Flash.

I²C Timing

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

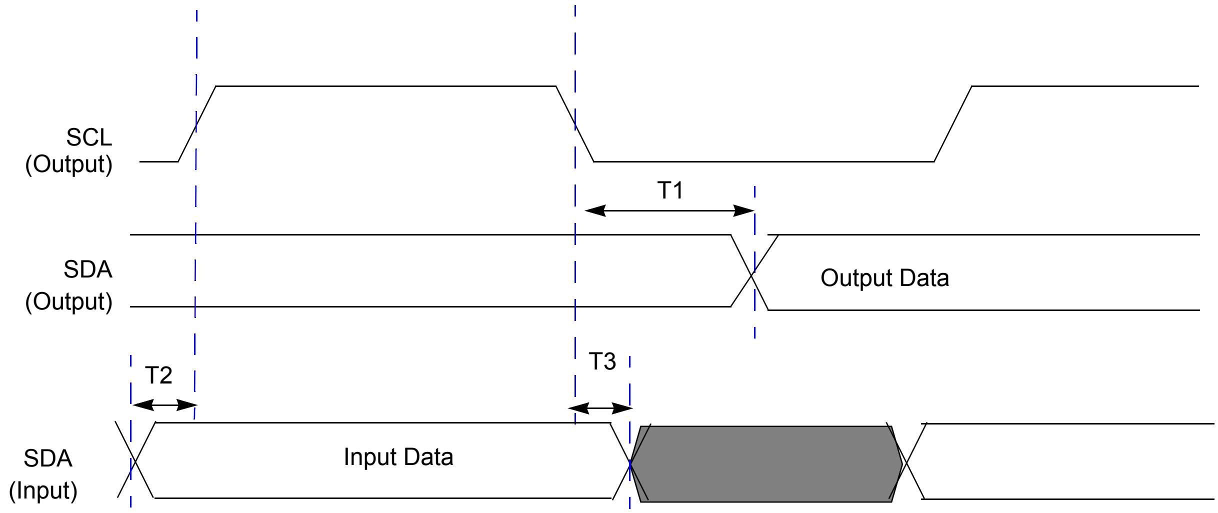

Figure 55. I²C Timing

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 133. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Flags Z	Flags S	Flags V	Flags D	Flags 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	-	*	*	0	-	-	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	-	*	*	0	-	-	2	2
BRK	Debugger Break			00	-	-	-	-	-	-	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	-	*	*	0	-	-	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	-	-	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	-	-	-	-	-	-	3	3
			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
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