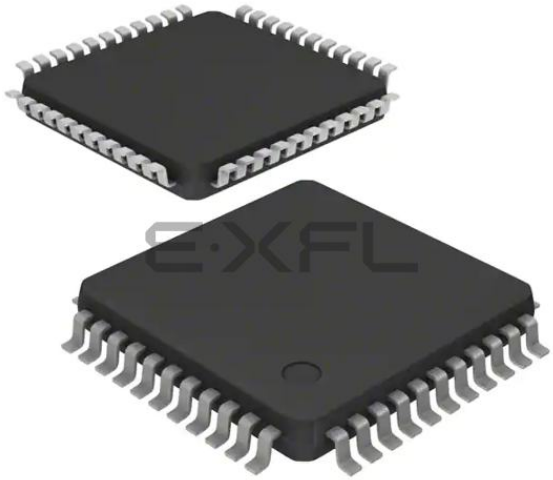

Welcome to E-XFL.COM

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	31
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1621an020ec00tr

Warning: DO NOT USE IN LIFE SUPPORT

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

 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concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded.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!, Z8 Encore! XP, Z8 Encore! MC, Crimzon, eZ80, and ZNEO are trademarks or registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.

ISO 9001:2000
FS 507510

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit www.zilog.com/quality.

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

Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page No
FCD	Interrupt Edge Select	IRQES	00	78
FCE	Interrupt Port Select	IRQPS	00	78
FCF	Interrupt Control	IRQCTL	00	79
GPIO Port A				
FD0	Port A Address	PAADDR	00	61
FD1	Port A Control	PACTL	00	62
FD2	Port A Input Data	PAIN	XX	66
FD3	Port A Output Data	PAOUT	00	66
GPIO Port B				
FD4	Port B Address	PBADDR	00	61
FD5	Port B Control	PBCTL	00	62
FD6	Port B Input Data	PBIN	XX	66
FD7	Port B Output Data	PBOUT	00	66
GPIO Port C				
FD8	Port C Address	PCADDR	00	61
FD9	Port C Control	PCCTL	00	62
FDA	Port C Input Data	PCIN	XX	66
FDB	Port C Output Data	PCOUT	00	66
GPIO Port D				
FDC	Port D Address	PDADDR	00	61
FDD	Port D Control	PDCTL	00	62
FDE	Port D Input Data	PDIN	XX	66
FDF	Port D Output Data	PDOUT	00	66
GPIO Port E				
FE0	Port E Address	PEADDR	00	61
FE1	Port E Control	PECTL	00	62
FE2	Port E Input Data	PEIN	XX	66
FE3	Port E Output Data	PEOUT	00	66
GPIO Port F				
FE4	Port F Address	PFADDR	00	61
FE5	Port F Control	PFCTL	00	62
FE6	Port F Input Data	PFIN	XX	66
FE7	Port F Output Data	PFOUT	00	66
GPIO Port G				
FE8	Port G Address	PGADDR	00	61
FE9	Port G Control	PGCTL	00	62
FEA	Port G Input Data	PGIN	XX	66
FEB	Port G Output Data	PGOUT	00	66
GPIO Port H				
FEC	Port H Address	PHADDR	00	61
FED	Port H Control	PHCTL	00	62
FEE	Port H Input Data	PHIN	XX	66

UART0 Control 1
U0CTL1 (F43H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- Infrared Encoder/Decoder Enable
  0 = Infrared endec is disabled
  1 = Infrared endec is enabled
- Received Data Interrupt Enable
  0 = Received data and errors generate interrupt requests
  1 = Only errors generate interrupt requests. Received data does not.
- Baud Rate Registers Control
  Refer to UART chapter for operation
- Driver Enable Polarity
  0 = DE signal is active High
  1 = DE signal is active Low
- Multiprocessor Bit Transmit
  0 = Send a 0 as the multiprocessor bit
  1 = Send a 1 as the multiprocessor bit
- Multiprocessor Mode [0]
  See Multiprocessor Mode [1] below
- Multiprocessor (9-bit) Enable
  0 = Multiprocessor mode is disabled
  1 = Multiprocessor mode is enabled
- Multiprocessor Mode [1]
  with Multiprocess Mode bit 0:
  00 = Interrupt on all received bytes
  01 = Interrupt only on address bytes
  10 = Interrupt on address match and following data
  11 = Interrupt on data following an address match

UART0 Status 1
U0STAT1 (F44H - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

- Mulitprocessor Receive
  Returns value of last multiprocessor bit
- New Frame
  0 = Current byte is not start of frame
  1 = Current byte is start of new frame
- Reserved

UART0 Address Compare
U0ADDR (F45H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- UART0 Address Compare [7:0]

UART0 Baud Rate Generator High Byte
U0BRH (F46H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- UART0 Baud Rate divisor [15:8]

UART0 Baud Rate Generator Low Byte
U0BRL (F47H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- UART0 Baud Rate divisor [7:0]

UART1 Transmit Data
U1TXD (F48H - Write Only)

D7	D6	D5	D4	D3	D2	D1	D0

- UART1 transmitter data byte[7:0]

UART1 Receive Data
U1RXD (F48H - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

- UART receiver data byte [7:0]

Flags

FLAGS (FFC - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D0 — F1 - User Flag 1
- D1 — F2 - User Flag 2
- D2 — H - Half Carry
- D3 — D - Decimal Adjust
- D4 — V - Overflow Flag
- D5 — S - Sign Flag
- D6 — Z - Zero Flag
- D7 — C - Carry Flag

Register Pointer

RP (FFDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D3:D0 — Working Register Page Address
- D7:D6 — Working Register Group Address

Stack Pointer High Byte

SPH (FFEH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D7:D0 — Stack Pointer [15:8]

Stack Pointer Low Byte

SPL (FFFH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- D7:D0 — Stack Pointer [7:0]

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.

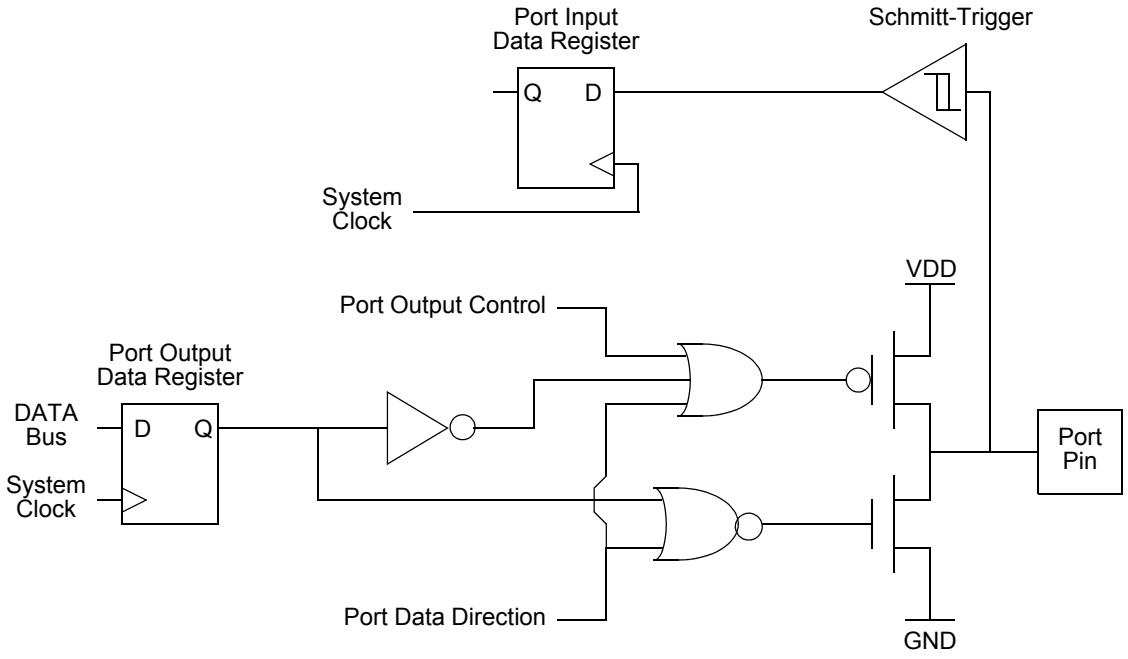

Figure 10. GPIO Port Pin Block Diagram

Table 19. Port A–H High Drive Enable Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	PHDE7	PHDE6	PHDE5	PHDE4	PHDE3	PHDE2	PHDE1	PHDE0
RESET	0							
R/W	R/W							
ADDR	If 04H in Port A-H Address Register, accessible through Port A-H Control Register							

PHDE[7:0]—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.

Port A–H Stop Mode Recovery Source Enable Sub-Registers

The Port A–H Stop Mode Recovery Source Enable sub-register (Table 20) is accessed through the Port A–H Control register by writing `05H` to the Port A–H Address register. Setting the bits in the Port A–H Stop Mode Recovery Source Enable sub-registers 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 20. Port A–H Stop Mode Recovery Source Enable Sub-Registers

BITS	7	6	5	4	3	2	1	0
FIELD	PSMRE7	PSMRE6	PSMRE5	PSMRE4	PSMRE3	PSMRE2	PSMRE1	PSMRE0
RESET	0							
R/W	R/W							
ADDR	If 05H in Port A–H Address Register, accessible through Port A–H Control Register							

PSMRE[7:0]—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.

Architecture

Figure 11 displays a block diagram of the interrupt controller.

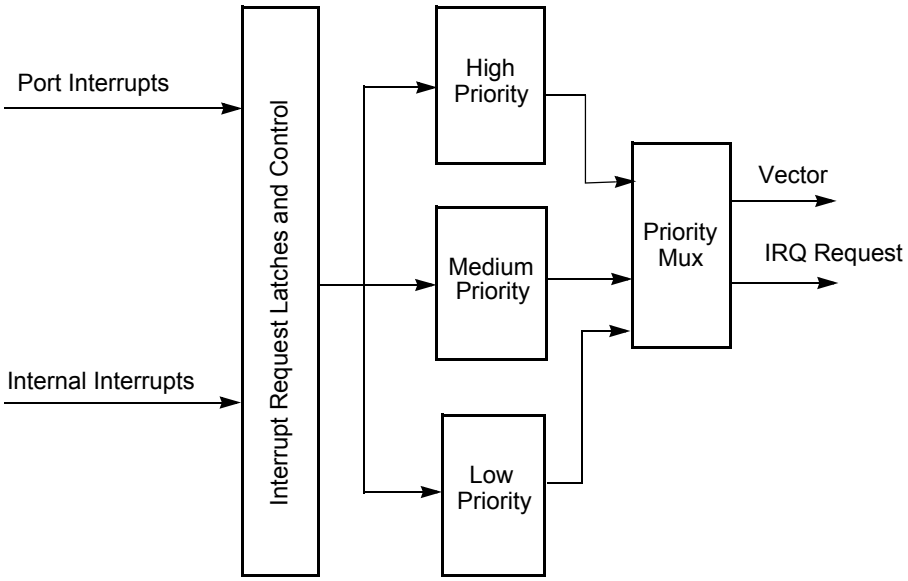

Figure 11. Interrupt Controller Block Diagram

Operation

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:

- Executing an Enable Interrupt (EI) instruction.
- Executing an Return from Interrupt (IRET) instruction.
- Writing a 1 to the IRQE bit in the Interrupt Control register.

Interrupts are globally disabled by any of the following actions:

- Execution of a Disable Interrupt (DI) instruction.
- eZ8 CPU acknowledgement of an interrupt service request from the interrupt controller.
- Writing a 0 to the IRQE bit in the Interrupt Control register.
- Reset.

C1ENL—Port C1 Interrupt Request Enable Low Bit
C0ENL—Port C0 Interrupt Request Enable Low Bit

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) register (Table 36) determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port input pin. The Interrupt Port Select register selects between Port A and Port D for the individual interrupts.

Table 36. Interrupt Edge Select Register (IRQES)

BITS	7	6	5	4	3	2	1	0
FIELD	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0							
R/W	R/W							
ADDR	FCDH							

IES*x*—Interrupt Edge Select *x*
The minimum pulse width should be greater than 1 system clock to guarantee capture of the edge triggered interrupt. Shorter pulses may be captured but not guaranteed.
0 = An interrupt request is generated on the falling edge of the PA*x*/PD*x* input.
1 = An interrupt request is generated on the rising edge of the PA*x*/PD*x* input.
where *x* indicates the specific GPIO Port pin number (0 through 7).

Interrupt Port Select Register

The Port Select (IRQPS) register (Table 37) determines the port pin that generates the PAx/PDx interrupts. This register allows either Port A or Port D pins to be used as interrupts. The Interrupt Edge Select register controls the active interrupt edge.

Table 37. Interrupt Port Select Register (IRQPS)

BITS	7	6	5	4	3	2	1	0
FIELD	PAD7S	PAD6S	PAD5S	PAD4S	PAD3S	PAD2S	PAD1S	PAD0S
RESET	0							
R/W	R/W							
ADDR	FCEH							

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.

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

1. Write to the Timer Control 1 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. 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, then the interrupt was generated by a Reload.
5. If desired, 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 Input alternate function.
7. Write to the Timer Control 1 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.

1 = Infrared Encoder/Decoder is enabled. The UART transmits and receives data through the Infrared Encoder/Decoder.

UART Address Compare Register

The UART Address Compare register (Table 58) stores the multi-node network address of the UART. When the MPMD[1] bit of UART Control Register 0 is set, all incoming address bytes are compared to the value stored in the Address Compare register. Receive interrupts and RDA assertions only occur in the event of a match.

Table 58. UART Address Compare Register (U*x*ADDR)

BITS	7	6	5	4	3	2	1	0
FIELD	COMP_ADDR							
RESET	0							
R/W	R/W							
ADDR	F45H and F4DH							

COMP_ADDR—Compare Address
This 8-bit value is compared to the incoming address bytes.

UART Baud Rate High and Low Byte Registers

The UART Baud Rate High and Low Byte registers (see Table 59 and Table 60 on page 121) combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. 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 UART BRG interrupt interval is calculated using the following equation:

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

Transfer Format PHASE Equals Zero

Figure 25 displays the timing diagram for an SPI transfer in which PHASE is cleared to 0. The two SCK waveforms show polarity with CLKPOL reset to 0 and with CLKPOL set to one. The diagram may be interpreted as either a Master or Slave timing diagram because the SCK Master-In/Slave-Out (MISO) and Master-Out/Slave-In (MOSI) pins are directly connected between the Master and the Slave.

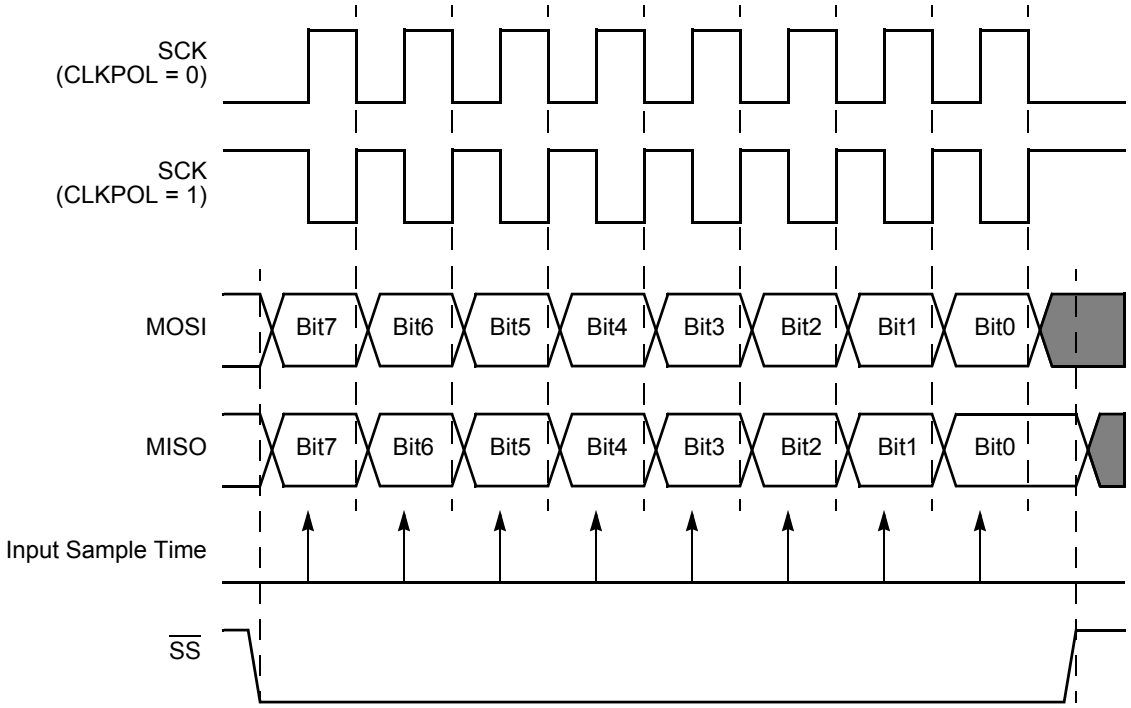

Figure 25. SPI Timing When PHASE is 0

Transfer Format PHASE Equals One

Figure 26 on page 134 displays the timing diagram for an SPI transfer in which PHASE is one. Two waveforms are depicted for SCK, one for CLKPOL reset to 0 and another for CLKPOL set to 1.

Architecture

Figure 27 displays the architecture of the I²C Controller.

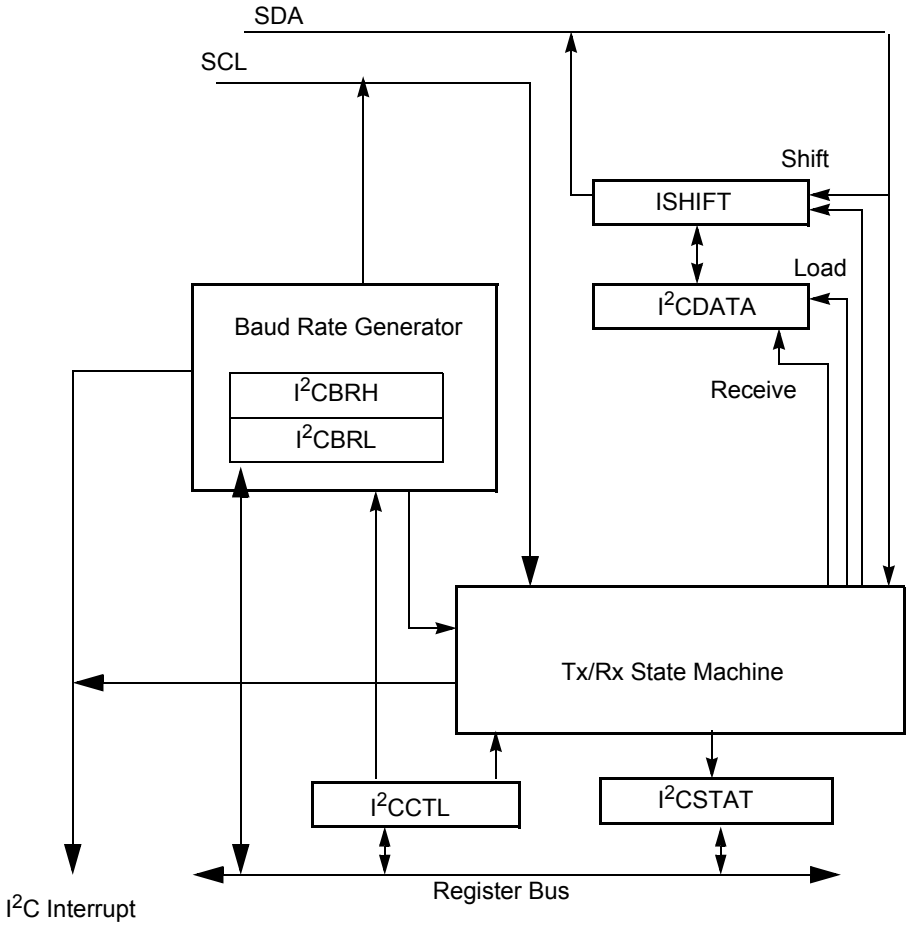

Figure 27. I²C Controller Block Diagram

Operation

The I²C Controller operates in MASTER mode to transmit and receive data. Only a single master is supported. Arbitration between two masters must be accomplished in software. I²C supports the following operations:

- Master transmits to a 7-bit slave
- Master transmits to a 10-bit slave

7. The I2C Controller loads the I2C Shift register with the contents of the I2C 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 I2C Data register.
10. The I2C Controller shifts the rest of the first byte of address and write bit out the SDA signal.
11. If the I2C slave sends an acknowledge by pulling the SDA signal low during the next high period of SCL the I2C Controller sets the ACK bit in the I2C Status register. Continue with step 12.

    If the slave does not acknowledge the first address byte, the I2C Controller sets the NCKI bit and clears the ACK bit in the I2C Status register. Software responds to the Not Acknowledge interrupt by setting the STOP and FLUSH bits and clearing the TXI bit. The I2C Controller sends the STOP condition on the bus and clears the STOP and NCKI bits. The transaction is complete (ignore following steps).
12. The I2C Controller loads the I2C Shift register with the contents of the I2C Data register (2nd byte of address).
13. The I2C 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 I2C Control register. The TXI bit can be cleared at the same time.
15. Software polls the STOP bit of the I2C 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 I2C 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 I2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I2C Controller.

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

Figure 31. 10-Bit Addressed Slave Data Transfer Format

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
0_0000	Idle State
0_0001	START State
0_0010	Send/Receive data bit 7
0_0011	Send/Receive data bit 6
0_0100	Send/Receive data bit 5
0_0101	Send/Receive data bit 4
0_0110	Send/Receive data bit 3
0_0111	Send/Receive data bit 2
0_1000	Send/Receive data bit 1
0_1001	Send/Receive data bit 0
0_1010	Data Acknowledge State
0_1011	Second half of data Acknowledge State used only for not acknowledge
0_1100	First part of STOP state
0_1101	Second part of STOP state
0_1110	10-bit addressing: Acknowledge State for 2nd address byte 7-bit addressing: Address Acknowledge State
0_1111	10-bit address: Bit 0 (Least significant bit) of 2nd address byte 7-bit address: Bit 0 (Least significant bit) (R/W) of address byte
1_0000	10-bit addressing: Bit 7 (Most significant bit) of 1st address byte
1_0001	10-bit addressing: Bit 6 of 1st address byte
1_0010	10-bit addressing: Bit 5 of 1st address byte
1_0011	10-bit addressing: Bit 4 of 1st address byte
1_0100	10-bit addressing: Bit 3 of 1st address byte
1_0101	10-bit addressing: Bit 2 of 1st address byte
1_0110	10-bit addressing: Bit 1 of 1st address byte
1_0111	10-bit addressing: Bit 0 (R/W) of 1st address byte
1_1000	10-bit addressing: Acknowledge state for 1st address byte
1_1001	10-bit addressing: Bit 7 of 2nd address byte 7-bit addressing: Bit 7 of address byte
1_1010	10-bit addressing: Bit 6 of 2nd address byte 7-bit addressing: Bit 6 of address byte
1_1011	10-bit addressing: Bit 5 of 2nd address byte 7-bit addressing: Bit 5 of address byte
1_1100	10-bit addressing: Bit 4 of 2nd address byte 7-bit addressing: Bit 4 of address byte

1 = DMA*x* is enabled and initiates a data transfer upon receipt of a request from the trigger source.

DLE—DMA*x* Loop Enable
0 = DMA*x* reloads the original Start Address and is then disabled after the End Address data is transferred.
1 = DMA*x*, after the End Address data is transferred, reloads the original Start Address and continues operating.

DDIR—DMA*x* Data Transfer Direction
0 = Register File → on-chip peripheral control register.
1 = on-chip peripheral control register → Register File.

IRQEN—DMA*x* Interrupt Enable
0 = DMA*x* does not generate any interrupts.
1 = DMA*x* generates an interrupt when the End Address data is transferred.

WSEL—Word Select
0 = DMA*x* transfers a single byte per request.
1 = DMA*x* transfers a two-byte word per request. The address for the on-chip peripheral control register must be an even address.

RSS—Request Trigger Source Select
The Request Trigger Source Select field determines the peripheral that can initiate a DMA transfer. The corresponding interrupts do not need to be enabled within the Interrupt Controller to initiate a DMA transfer. However, if the Request Trigger Source can enable or disable the interrupt request sent to the Interrupt Controller, the interrupt request must be enabled within the Request Trigger Source block.
000 = Timer 0.
001 = Timer 1.
010 = Timer 2.
011 = Timer 3.
100 = DMA0 Control register: UART0 Received Data register contains valid data. DMA1 Control register: UART0 Transmit Data register empty.
101 = DMA0 Control register: UART1 Received Data register contains valid data. DMA1 Control register: UART1 Transmit Data register empty.
110 = DMA0 Control register: I^2C Receiver Interrupt. DMA1 Control register: I^2C Transmitter Interrupt register empty.
111 = Reserved.

DMA*x* I/O Address Register

The DMA*x* I/O Address register (Table 78) contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is given by {FH,

While the Flash Controller programs the Flash memory, the eZ8 CPU idles but the system clock and on-chip peripherals continue to operate. Interrupts that occur when a Programming operation is in progress are serviced once the Programming operation is complete. To exit Programming mode and lock the Flash Controller, write 00H to the Flash Control register.

User code cannot program Flash Memory on a page that lies in a protected sector. When user code writes memory locations, only addresses located in the unlocked page are programmed. Memory writes outside of the unlocked page are ignored.

Caution: *Each memory location must not be programmed more than twice before an erase occurs.*

Follow the steps below to program the Flash from user code:

1. Write 00H to the Flash Control register to reset the Flash Controller.
2. Write the page of memory to be programmed to the Page Select register.
3. Write the first unlock command 73H to the Flash Control register.
4. Write the second unlock command 8CH to the Flash Control register.
5. Re-write the page written in step 2 to the Page Select register.
6. Write Flash Memory using LDC or LDCI instructions to program the Flash.
7. Repeat step 6 to program additional memory locations on the same page.
8. Write 00H to the Flash Control register to lock the Flash Controller.

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 Page Select register identifies the page to be erased. 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. Interrupts that occur when the Page Erase operation is in progress are serviced once the Page Erase operation is complete. When the Page Erase operation is complete, the Flash Controller returns to its locked state. Only pages located in unprotected sectors can be erased.

Follow the steps below to perform a Page Erase operation:

1. Write 00H to the Flash Control register to reset the Flash Controller.
2. Write the page to be erased to the Page Select register.
3. Write the first unlock command 73H to the Flash Control register.
4. Write the second unlock command 8CH to the Flash Control register.

eZ8 CPU loops on the BRK instruction.
0 = BRK instruction sets DBGMODE to 1.
1 = eZ8 CPU loops on BRK instruction.

Reserved
These bits are reserved and must be 0.

RST—Reset
Setting this bit to 1 resets the 64K Series devices. The devices go 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 when the reset finishes.
0 = No effect
1 = Reset the 64K Series device

OCD Status Register

The OCD Status register (Table 103) reports status information about the current state of the debugger and the system.

Table 103. OCD Status Register (OCDSTAT)

BITS	7	6	5	4	3	2	1	0
FIELD	IDLE	HALT	RPEN	Reserved				
RESET	0							
R/W	R							

IDLE—CPU idling
This bit is set if the part is in DEBUG mode (DBGMODE is 1), or if a BRK instruction occurred since the last time OCDCTL was written. This can be used to determine if the CPU is running or if it is idling.
0 = The eZ8 CPU is running.
1 = The eZ8 CPU is either stopped or looping on a BRK instruction.

HALT—HALT Mode
0 = The device is not in HALT mode.
1 = The device is in HALT mode.

RPEN—Read Protect Option Bit Enabled
0 = The Read Protect Option Bit is disabled (1).
1 = The Read Protect Option Bit is enabled (0), disabling many OCD commands.

Reserved
These bits are always 0.

Table 112. Analog-to-Digital Converter Electrical Characteristics and Timing

Symbol	Parameter	V_{DD} = 3.0–3.6 V T_A = –40 °C to 125 °C			Units	Conditions
		Minimum	Typical	Maximum		
	Resolution	10	–	–	bits	External V_{REF} = 3.0 V;
	Differential Nonlinearity (DNL)	-1.0		+1.0	lsb	Guaranteed by design
	Integral Nonlinearity (INL)	-3.0	±1.0	3.0	lsb	External V_{REF} = 3.0 V
	DC Offset Error	-35	–	25	mV	
	DC Offset Error	-50	–	25	mV	44-pin LQFP, 44-pin PLCC, and 68-pin PLCC packages.
V_{REF}	Internal Reference Voltage	1.9	2.0	2.4	V	V_{DD} = 3.0 - 3.6 V T_A = -40 °C to 105 °C
VC_{REF}	Voltage Coefficient of Internal Reference Voltage	–	78	–	mV/V	V_{REF} variation as a function of AVDD.
TC_{REF}	Temperature Coefficient of Internal Reference Voltage	–	1	–	mV/°C	
	Single-Shot Conversion Period	–	5129	–	cycles	System clock cycles
	Continuous Conversion Period	–	256	–	cycles	System clock cycles
R_S	Analog Source Impedance	–	–	150	Ω	Recommended
Zin	Input Impedance		150		kΩ	
V_{REF}	External Reference Voltage			AVDD	V	AVDD <= VDD. When using an external reference voltage, decoupling capacitance should be placed from VREF to AVSS.
I_{REF}	Current draw into VREF pin when driving with external source.		25.0	40.0	µA	