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	48KB (48K 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	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-BQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4823ft020ec00tr

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.

Table 3. Signal Descriptions (Continued)

Signal Mnemonic	I/O	Description
XIN	I	External Crystal Input. This is the input pin to the crystal oscillator. A crystal can be connected between it and the **XOUT** pin to form the oscillator. This signal is usable with external RC networks and an external clock driver.
XOUT	O	External Crystal Output. This pin is the output of the crystal oscillator. A crystal can be connected between it and the XIN pin to form the oscillator. When the system clock is referred to in this manual, it refers to the frequency of the signal at this pin. This pin must be left unconnected when not using a crystal.
RCOUT	O	RC Oscillator Output. This signal is the output of the RC oscillator. It is multiplexed with a general-purpose I/O pin. This signal must be left unconnected when not using a crystal.
On-Chip Debugger		
DBG	I/O	Debug. This pin is the control and data input and output to and from the On-Chip Debugger. This pin is open-drain.
	Caution:	*For operation of the On-Chip Debugger, all power pins (V_{DD} and AV_{DD}) must be supplied with power and all ground pins (V_{SS} and AV_{SS}) must be properly grounded.* *The DBG pin is open-drain and must have an external pull-up resistor to ensure proper operation.*
Reset		
$\overline{\text{RESET}}$	I	RESET. Generates a Reset when asserted (driven Low).
Power Supply		
VDD	I	Power Supply.
AVDD	I	Analog Power Supply.
VSS	I	Ground.
AVSS	I	Analog Ground.

Pin Characteristics

Table 4 on page 17 provides detailed information on the characteristics for each pin available on the 64K Series products and the data is sorted alphabetically by the pin symbol mnemonic.

Program Memory

The eZ8™ CPU supports 64 KB of Program Memory address space. The Z8 Encore! XP 64K Series Flash Microcontrollers contains 16 KB to 64 KB of on-chip Flash in the Program Memory address space, depending upon the device. Reading from Program Memory addresses outside the available Flash memory addresses returns FFH. Writing to these unimplemented Program Memory addresses produces no effect. Table 5 describes the Program Memory maps for the 64K Series products.

Table 5. Z8 Encore! XP 64K Series Flash Microcontrollers Program Memory Maps

Program Memory Address (Hex)	Function
Z8F162x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-3FFF	Program Memory
Z8F242x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-5FFF	Program Memory
Z8F322x Products	
0000-0001	Option Bits
0002-0003	Reset Vector
0004-0005	WDT Interrupt Vector
0006-0007	Illegal Instruction Trap
0008-0037	Interrupt Vectors*
0038-7FFF	Program Memory
Z8F482x Products	

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

D7	D6	D5	D4	D3	D2	D1	D0

- ADC IRQ Enable Hit Bit
- SPI IRQ Enable Low Bit
- I2C IRQ Enable Low Bit
- UART 0 Transmitter IRQ Enable
- UART 0 Receiver IRQ Enable Low
- Timer 0 IRQ Enable Low Bit
- Timer 1 IRQ Enable Low Bit
- Timer 2 IRQ Enable Low Bit

Interrupt Request 1
IRQ1 (FC3H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	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

- 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

- Port A or D Pin IRQ Enable Low Bit

Interrupt Request 2
IRQ2 (FC6H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	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
- DMA Interrupt Request
- UART 1 Transmitter Interrupt
- UART 1 Receiver Interrupt Request
- 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

- Port C Pin IRQ Enable High Bit
- DMA IRQ Enable High Bit
- UART 1 Transmitter IRQ Enable
- UART 1 Receiver IRQ Enable High
- Timer 3 IRQ Enable High Bit

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

D7	D6	D5	D4	D3	D2	D1	D0

- Port C Pin IRQ Enable Low Bit
- DMA IRQ Enable Low Bit
- UART 1 Transmitter IRQ Enable
- UART 1 Receiver IRQ Enable Low
- Timer 3 IRQ Enable Low Bit

Interrupt Edge Select
IRQES (FCDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

- Port A or D Interrupt Edge Select
  0 = Falling edge
  1 = Rising edge

Port F Control
PFCTL (FE5H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port F Control[7:0]
Provides Access to Port Sub-Registers

Port F Input Data
PFIN (FE6H - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

Port F Input Data [7:0]

Port F Output Data
PFOUT (FE7H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port F Output Data [7:0]

Port G Address
PGADDR (FE8H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port G Control
PGCTL (FE9H - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Control[7:0]
Provides Access to Port Sub-Registers

Port G Input Data
PGIN (FEAH - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Input Data [7:0]

Port G Output Data
PGOUT (FEBH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port G Output Data [7:0]

Port H Address
PHADDR (FECH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Address[7:0]
Selects Port Sub-Registers:
00H = No function
01H = Data direction
02H = Alternate function
03H = Output control (open-drain)
04H = High drive enable
05H = Stop Mode Recovery enable
06H-FFH = No function

Port H Control
PHCTL (FEDH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Control [3:0]
Provides Access to Port Sub-Registers

Reserved

Port H Input Data
PHIN (FEEH - Read Only)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Input Data [3:0]

Reserved

Port H Output Data
PHOUT (FEFH - Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0

Port H Output Data [3:0]

Reserved

Table 8. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Control Registers**	**eZ8™ CPU**	**Reset Latency (Delay)**
System reset	Reset (as applicable)	Reset	66 WDT Oscillator cycles + 16 System Clock cycles
Stop Mode Recovery	Unaffected, except WDT_CTL register	Reset	66 WDT Oscillator cycles + 16 System Clock cycles

System Reset

During a system reset, the 64K Series devices are held in Reset for 66 cycles of the Watchdog Timer oscillator followed by 16 cycles of the system clock. At the beginning of Reset, all GPIO pins are configured as inputs.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and Watchdog Timer oscillator continue to run. The system clock begins operating following the Watchdog Timer oscillator cycle count. The eZ8 CPU and on-chip peripherals remain idle through the 16 cycles of the system clock.

Upon Reset, control registers within the Register File that have a defined Reset value are loaded with their reset values. Other control registers (including the Stack Pointer, Register Pointer, and Flags) and general-purpose RAM are undefined following Reset. The eZ8 CPU fetches the Reset vector at Program Memory addresses `0002H` and `0003H` and loads that value into the Program Counter. Program execution begins at the Reset vector address.

Reset Sources

Table 9 lists the reset sources as a function of the operating mode. The text following provides more detailed information on the individual Reset sources. A Power-On Reset/Voltage Brownout event always takes priority over all other possible reset sources to ensure a full system reset occurs.

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.

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							

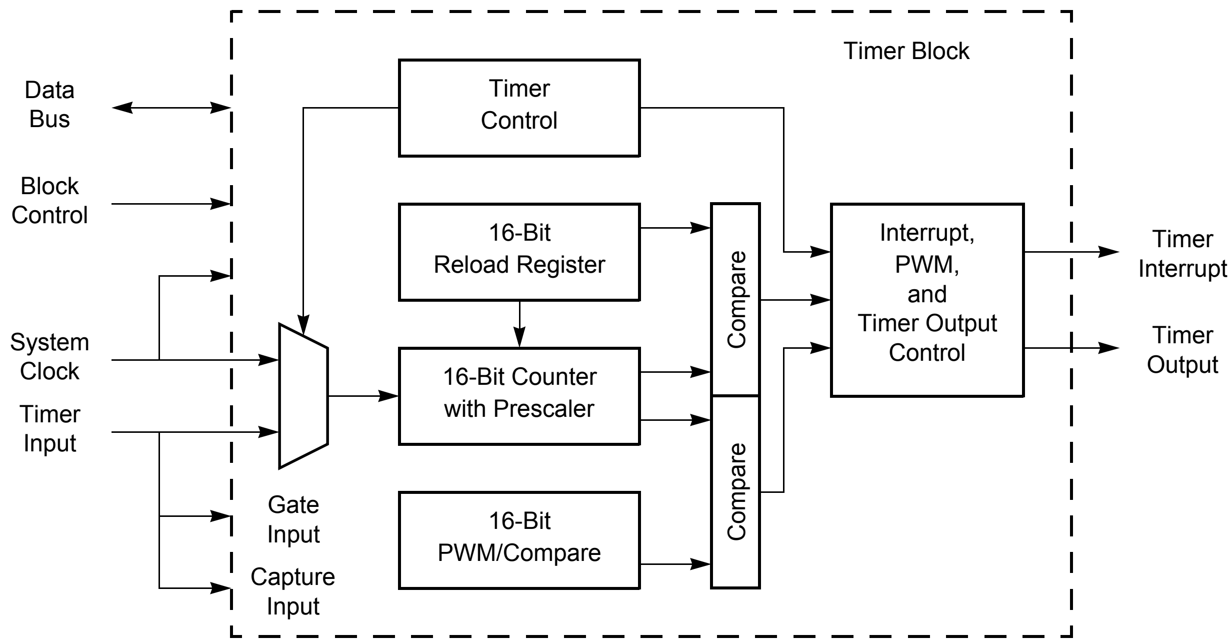

Figure 12. Timer Block Diagram

Operation

The timers are 16-bit up-counters. Minimum time-out delay is set by loading the value `0001H` into the Timer Reload High and Low Byte registers and setting the prescale value to 1. Maximum time-out delay is set by loading the value `0000H` into the Timer Reload High and Low Byte registers and setting the prescale value to 128. If the Timer reaches `FFFFH`, the timer rolls over to `0000H` and continues counting.

Timer Operating Modes

The timers can be configured to operate in the following modes:

ONE-SHOT Mode

In ONE-SHOT mode, the timer counts up to the 16-bit Reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. Upon reaching the Reload value, the timer generates an interrupt and the count value in the Timer High and Low Byte registers is reset to `0001H`. Then, the timer is automatically disabled and stops counting.

Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state for one system clock cycle (from Low to High or from High to Low) upon timer Reload. If it is desired to have the Timer Output make a permanent state change upon

2. Write to the Timer High and Low Byte registers to set the starting count value (usually `0001H`), affecting only the first pass in CONTINUOUS mode. After the first timer Reload in CONTINUOUS mode, counting always begins at the reset value of `0001H`.
3. Write to the Timer Reload High and Low Byte registers to set the Reload value.
4. 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 CONTINUOUS mode, the system clock always provides the timer input. The timer period is given by the following equation:

$$\text{CONTINUOUS Mode Time-Out Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

If an initial starting value other than `0001H` is loaded into the Timer High and Low Byte registers, the ONE-SHOT mode equation must be used to determine the first time-out period.

COUNTER Mode

In COUNTER mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO Port pin Timer Input alternate function. The `TPOL` bit in the Timer Control 1 Register selects whether the count occurs on the rising edge or the falling edge of the Timer Input signal. In COUNTER mode, the prescaler is disabled.

Caution: *The input frequency of the Timer Input signal must not exceed one-fourth the system clock frequency.*

Upon reaching the Reload value stored in the Timer Reload High and Low Byte registers, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes. Also, if the Timer Output alternate function is enabled, the Timer Output pin changes state (from Low to High or from High to Low) at timer Reload.

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

1. Write to the Timer Control 1 register to:
   - Disable the timer
   - Configure the timer for COUNTER mode

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.

Table 61. UART Baud Rates (Continued)

1.20	868	1.20	0.01	1.20	576	1.20	0.00
0.60	1736	0.60	0.01	0.60	1152	0.60	0.00
0.30	3472	0.30	0.01	0.30	2304	0.30	0.00
10.0 MHz System Clock				**5.5296 MHz System Clock**			
Desired Rate	**BRG Divisor**	**Actual Rate**	**Error**	**Desired Rate**	**BRG Divisor**	**Actual Rate**	**Error**
(kHz)	**(Decimal)**	**(kHz)**	**(%)**	**(kHz)**	**(Decimal)**	**(kHz)**	**(%)**
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	1	625.0	0.00	625.0	N/A	N/A	N/A
250.0	3	208.33	-16.67	250.0	1	345.6	38.24
115.2	5	125.0	8.51	115.2	3	115.2	0.00
57.6	11	56.8	-1.36	57.6	6	57.6	0.00
38.4	16	39.1	1.73	38.4	9	38.4	0.00
19.2	33	18.9	0.16	19.2	18	19.2	0.00
9.60	65	9.62	0.16	9.60	36	9.60	0.00
4.80	130	4.81	0.16	4.80	72	4.80	0.00
2.40	260	2.40	-0.03	2.40	144	2.40	0.00
1.20	521	1.20	-0.03	1.20	288	1.20	0.00
0.60	1042	0.60	-0.03	0.60	576	0.60	0.00
0.30	2083	0.30	0.2	0.30	1152	0.30	0.00
3.579545 MHz System Clock				**1.8432 MHz System Clock**			
Desired Rate	**BRG Divisor**	**Actual Rate**	**Error**	**Desired Rate**	**BRG Divisor**	**Actual Rate**	**Error**
(kHz)	**(Decimal)**	**(kHz)**	**(%)**	**(kHz)**	**(Decimal)**	**(kHz)**	**(%)**
1250.0	N/A	N/A	N/A	1250.0	N/A	N/A	N/A
625.0	N/A	N/A	N/A	625.0	N/A	N/A	N/A
250.0	1	223.72	-10.51	250.0	N/A	N/A	N/A
115.2	2	111.9	-2.90	115.2	1	115.2	0.00
57.6	4	55.9	-2.90	57.6	2	57.6	0.00
38.4	6	37.3	-2.90	38.4	3	38.4	0.00
19.2	12	18.6	-2.90	19.2	6	19.2	0.00

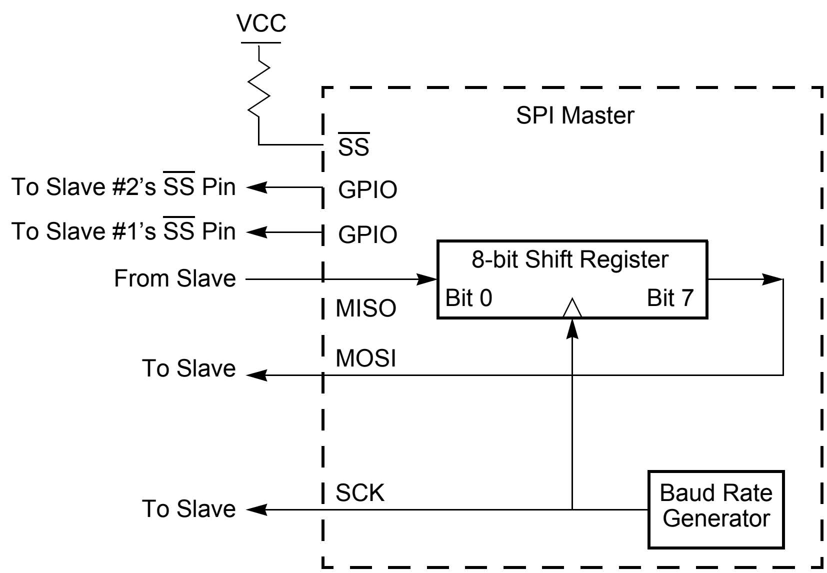

Figure 23. SPI Configured as a Master in a Single Master, Multiple Slave System

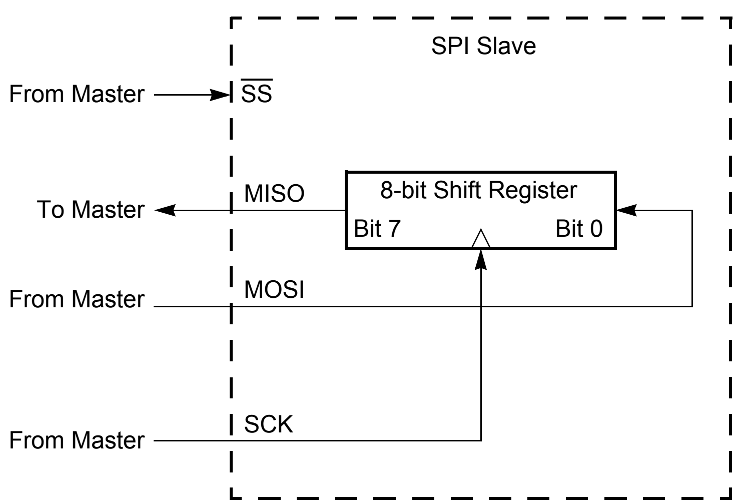

Figure 24. SPI Configured as a Slave

Operation

The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (serial clock, transmit, receive and Slave select). The SPI block consists of a transmit/receive shift register, a Baud Rate (clock) Generator and a control unit.

SPI Baud Rate High and Low Byte Registers

The SPI Baud Rate High and Low Byte registers (Table 68 and Table 69) combine to form a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator.

When configured as a general purpose timer, the SPI BRG interrupt interval is calculated using the following equation:

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

Table 68. SPI Baud Rate High Byte Register (SPIBRH)

BITS	7	6	5	4	3	2	1	0
FIELD	BRH							
RESET	1							
R/W	R/W							
ADDR	F66H							

BRH = SPI Baud Rate High Byte
Most significant byte, BRG[15:8], of the SPI Baud Rate Generator's reload value.

Table 69. SPI Baud Rate Low Byte Register (SPIBRL)

BITS	7	6	5	4	3	2	1	0
FIELD	BRL							
RESET	1							
R/W	R/W							
ADDR	F67H							

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

- Master receives from a 7-bit slave
- Master receives from a 10-bit slave

SDA and SCL Signals

I^2C sends all addresses, data and acknowledge signals over the SDA line, most-significant bit first. SCL is the common clock for the I^2C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I^2C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a high level. When the slave releases the clock, the I^2C Controller continues the transaction. All data is transferred in bytes and there is no limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the high period of SCL.

I^2C Interrupts

The I^2C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The Transmit interrupt is enabled by the IEN and TXI bits of the Control register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I^2C Controller and neither the `START` or `STOP` bit is set. The Not Acknowledge event sets the NCKI bit of the I^2C Status register and can only be cleared by setting the `START` or `STOP` bit in the I^2C Control register. When this interrupt occurs, the I^2C Controller waits until either the STOP or START bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I^2C Controller (master reading data from slave). This procedure sets the RDRF bit of the I^2C Status register. The RDRF bit is cleared by reading the I^2C Data register. The RDRF bit is set during the acknowledge phase. The I^2C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

ADC Control Register Definitions

ADC Control Register

The ADC Control register selects the analog input channel and initiates the analog-to-digital conversion.

Table 86. ADC Control Register (ADCCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	CEN	Reserved	VREF	CONT	ANAIN[3:0]			
RESET	0		1	0				
R/W	R/W							
ADDR	F70H							

CEN—Conversion Enable
0 = Conversion is complete. Writing a 0 produces no effect. The ADC automatically clears this bit to 0 when a conversion has been completed.
1 = Begin conversion. Writing a 1 to this bit starts a conversion. If a conversion is already in progress, the conversion restarts. This bit remains 1 until the conversion is complete.

Reserved—Must be 0.

$\overline{\text{VREF}}$
0 = Internal voltage reference generator enabled. The VREF pin should be left unconnected (or capacitively coupled to analog ground) if the internal voltage reference is selected as the ADC reference voltage.
1 = Internal voltage reference generator disabled. An external voltage reference must be provided through the VREF pin.

CONT
0 = Single-shot conversion. ADC data is output once at completion of the 5129 system clock cycles.
1 = Continuous conversion. ADC data updated every 256 system clock cycles.

ANAIN—Analog Input Select
These bits select the analog input for conversion. Not all Port pins in this list are available in all packages for the Z8F642x family Z8R642x family of products. For information on the Port pins available with each package style, see Signal and Pin Descriptions on page 7. Do not enable unavailable analog inputs.
0000 = ANA0
0001 = ANA1
0010 = ANA2
0011 = ANA3

Flash Memory Address 0000H

Table 98. Flash Option Bits At Flash Memory Address 0000H

BITS	7	6	5	4	3	2	1	0
FIELD	WDT_RES	WDT_AO	OSC_SEL[1:0]		VBO_AO	RP	Reserved	FWP
RESET	U							
R/W	R/W							
ADDR	Program Memory 0000H							
Note: U = Unchanged by Reset. R/W = Read/Write.								

WDT_RES—Watchdog Timer Reset
0 = Watchdog Timer time-out generates an interrupt request. Interrupts must be globally enabled for the eZ8 CPU to acknowledge the interrupt request.
1 = Watchdog Timer time-out causes a Short Reset. This setting is the default for unprogrammed (erased) Flash.

WDT_AO—Watchdog Timer Always On
0 = Watchdog Timer is automatically enabled upon application of system power. Watchdog Timer can not be disabled except during STOP Mode (if configured to power down during STOP Mode).
1 = Watchdog Timer is enabled upon execution of the WDT instruction. Once enabled, the Watchdog Timer can only be disabled by a Reset or Stop Mode Recovery. This setting is the default for unprogrammed (erased) Flash.

OSC_SEL[1:0]—Oscillator Mode Selection
00 = On-chip oscillator configured for use with external RC networks (<4 MHz).
01 = Minimum power for use with very low frequency crystals (32 kHz to 1.0 MHz).
10 = Medium power for use with medium frequency crystals or ceramic resonators (0.5 MHz to 10.0 MHz).
11 = Maximum power for use with high frequency crystals (8.0 MHz to 20.0 MHz). This setting is the default for unprogrammed (erased) Flash.

VBO_AO—Voltage Brownout Protection Always On
0 = Voltage Brownout Protection is disabled in STOP mode to reduce total power consumption.
1 = Voltage Brownout Protection is always enabled including during STOP mode. This setting is the default for unprogrammed (erased) Flash.

RP—Read Protect
0 = User program code is inaccessible. Limited control features are available through

- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a Reset.
- Driving the DBG pin Low while the device is in STOP mode initiates a system reset.

OCD Data Format

The OCD interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 Start bit, 8 data bits (least-significant bit first), and 1 Stop bit (see Figure 39).

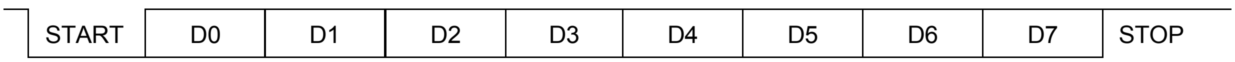

Figure 39. OCD Data Format

OCD Auto-Baud Detector/Generator

To run over a range of baud rates (bits per second) with various system clock frequencies, the On-Chip Debugger has an Auto-Baud Detector/Generator. After a reset, the OCD is idle until it receives data. The OCD requires that the first character sent from the host is the character `80H`. The character `80H` has eight continuous bits Low (one Start bit plus 7 data bits). The Auto-Baud Detector measures this period and sets the OCD Baud Rate Generator accordingly.

The Auto-Baud Detector/Generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. For optimal operation, the maximum recommended baud rate is the system clock frequency divided by 8. The theoretical maximum baud rate is the system clock frequency divided by 4. This theoretical maximum is possible for low noise designs with clean signals. Table 100 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 100. OCD Baud-Rate Limits

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (kbits/s)	Minimum Baud Rate (kbits/s)
20.0	2500	39.1
1.0	125.0	1.96
0.032768 (32 kHz)	4.096	0.064

If the OCD receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. The Auto-Baud Detector/Generator can then be reconfigured by sending `80H`.