



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	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	46
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	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4802ar020ec



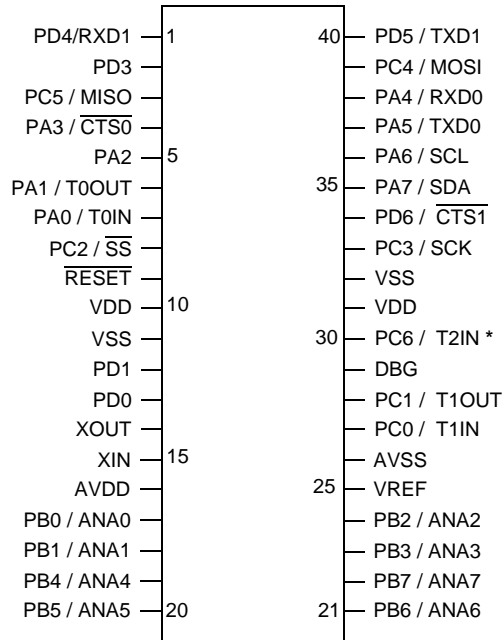
External Pin Reset	29
Stop Mode Recovery	29
Stop Mode Recovery Using Watch-Dog Timer Time-Out	29
Stop Mode Recovery Using a GPIO Port Pin Transition	30
Low-Power Modes	31
Overview	31
Stop Mode	31
Halt Mode	31
General-Purpose I/O	33
Overview	33
GPIO Port Availability By Device	33
Architecture	34
GPIO Alternate Functions	34
GPIO Interrupts	36
GPIO Control Register Definitions	36
Port A-H Address Registers	37
Port A-H Control Registers	38
Port A-H Input Data Registers	42
Port A-H Output Data Register	43
Interrupt Controller	44
Overview	44
Interrupt Vector Listing	44
Architecture	46
Operation	46
Master Interrupt Enable	46
Interrupt Vectors and Priority	47
Interrupt Assertion Types	47
Interrupt Control Register Definitions	48
Interrupt Request 0 Register	48
Interrupt Request 1 Register	49
Interrupt Request 2 Register	50
IRQ0 Enable High and Low Bit Registers	51
IRQ1 Enable High and Low Bit Registers	52
IRQ2 Enable High and Low Bit Registers	53
Interrupt Edge Select Register	54
Interrupt Port Select Register	55
Interrupt Control Register	56
Timers	57
Overview	57
Architecture	57
Operation	58



Table 32.	IRQ2 Enable and Priority Encoding	53
Table 33.	IRQ1 Enable High Bit Register (IRQ1ENH)	53
Table 34.	IRQ2 Enable Low Bit Register (IRQ2ENL)	54
Table 35.	IRQ2 Enable High Bit Register (IRQ2ENH)	54
Table 36.	Interrupt Edge Select Register (IRQES)	55
Table 37.	Interrupt Port Select Register (IRQPS)	55
Table 38.	Interrupt Control Register (IRQCTL)	56
Table 39.	Timer 0-3 High Byte Register (TxH)	67
Table 40.	Timer 0-3 Low Byte Register (TxL)	67
Table 41.	Timer 0-3 Reload High Byte Register (TxRH)	68
Table 42.	Timer 0-3 Reload Low Byte Register (TxRL)	68
Table 43.	Timer 0-3 PWM High Byte Register (TxPWMH)	69
Table 44.	Timer 0-3 PWM Low Byte Register (TxPWML)	69
Table 45.	Timer 0-3 Control Register (TxCTL)	70
Table 46.	Watch-Dog Timer Approximate Time-Out Delays	73
Table 47.	Watch-Dog Timer Control Register (WDTCTL)	75
Table 48.	Watch-Dog Timer Reload Upper Byte Register (WDTU) ..	76
Table 49.	Watch-Dog Timer Reload High Byte Register (WDTH) ..	76
Table 50.	Watch-Dog Timer Reload Low Byte Register (WDTL) ..	77
Table 51.	UARTx Transmit Data Register (UxTXD)	86
Table 52.	UARTx Receive Data Register (UxRXD)	87
Table 53.	UARTx Status 0 Register (UxSTAT0)	87
Table 54.	UARTx Control 0 Register (UxCTL0)	89
Table 55.	UARTx Status 1 Register (UxSTAT1)	89
Table 56.	UARTx Control 1 Register (UxCTL1)	90
Table 57.	UARTx Baud Rate High Byte Register (UxBRH)	91
Table 58.	UARTx Baud Rate Low Byte Register (UxBRL)	92
Table 59.	UART Baud Rates	93
Table 60.	SPI Clock Phase (PHASE) and Clock Polarity (CLKPOL) Operation	102
Table 61.	SPI Data Register (SPIDATA)	106
Table 62.	SPI Control Register (SPICTL)	107
Table 63.	SPI Status Register (SPISTAT)	108
Table 64.	SPI Mode Register (SPIMODE)	109
Table 65.	SPI Baud Rate High Byte Register (SPIBRH)	110
Table 66.	SPI Baud Rate Low Byte Register (SPIBRL)	110

Pin Configurations

Figures 56 through 61 illustrate the pin configurations for all of the packages available in the Z8 Encore!® MCU family. Refer to Table 2 for a description of the signals.



Note: Timer 3 is not supported.

* T2OUT is not supported.

Figure 56. Z8Fxx01 in 40-Pin Dual Inline Package (DIP)



Address Space

Overview

The eZ8 CPU can access three distinct address spaces:

- The Register File contains addresses for the general-purpose registers and the eZ8 CPU, peripheral, and general-purpose I/O port control registers.
- The Program Memory contains addresses for all memory locations having executable code and/or data.
- The Data Memory contains addresses for all memory locations that hold data only.

These three address spaces are covered briefly in the following subsections. For more detailed information regarding the eZ8 CPU and its address space, refer to the *eZ8 CPU User Manual* available for download at www.zilog.com.

Register File

The Register File address space in the Z8 Encore!® is 4KB (4096 bytes). The Register File is composed of two sections—control registers and general-purpose registers. When instructions are executed, registers are read from when defined as sources and written to when defined as destinations. The architecture of the eZ8 CPU allows all general-purpose registers to function as accumulators, address pointers, index registers, stack areas, or scratch pad memory.

The upper 256 bytes of the 4KB Register File address space are reserved for control of the eZ8 CPU, the on-chip peripherals, and the I/O ports. These registers are located at addresses from F00H to FFFH. Some of the addresses within the 256-byte control register section are reserved (unavailable). Reading from an reserved Register File addresses returns an undefined value. Writing to reserved Register File addresses is not recommended and can produce unpredictable results.

The on-chip RAM always begins at address 000H in the Register File address space. The Z8F640x family products contain 2KB to 4KB of on-chip RAM depending upon the device. Reading from Register File addresses outside the available RAM addresses (and not within in the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect. Refer to the **Part Selection Guide** section of the **Introduction** chapter to determine the amount of RAM available for the specific Z8F640x family device.

Reset and Stop Mode Recovery

Overview

The Reset Controller within the Z8F640x family devices controls Reset and STOP Mode Recovery operation. In typical operation, the following events cause a Reset to occur:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO)
- Watch-Dog Timer time-out (when configured via the WDT_RES Option Bit to initiate a reset)
- External $\overline{\text{RESET}}$ pin assertion
- On-Chip Debugger initiated Reset (OCDCTL[1] set to 1)

When the Z8F640x family device is in Stop mode, a Stop Mode Recovery is initiated by either of the following:

- Watch-Dog Timer time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source
- DBG pin driven Low

Reset Types

The Z8F640x family provides several different types of Reset operation. Stop Mode Recovery is considered a form of Reset. The type of Reset is a function of both the current operating mode of the Z8F640x family device and the source of the Reset. Table 7 lists the types of Reset and their operating characteristics. The System Reset is longer than the Short Reset to allow additional time for external oscillator start-up.

Table 7. Reset and Stop Mode Recovery Characteristics and Latency

Reset Type	Reset Characteristics and Latency		
	Control Registers	eZ8 CPU	Reset Latency (Delay)
System Reset	Reset (as applicable)	Reset	514 WDT Oscillator cycles + 16 System Clock cycles
Short Reset	Reset (as applicable)	Reset	66 WDT Oscillator cycles + 16 System Clock cycles
Stop Mode Recovery	Unaffected, except WDT_CTL register	Reset	514 WDT Oscillator cycles + 16 System Clock cycles

Table 12. 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)
PxHDE	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 13).

Table 13. 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							

Interrupt Control Register Definitions

For all interrupts other than the Watch-Dog Timer interrupt, the interrupt control registers enable individual interrupts, set interrupt priorities, and indicate interrupt requests.

Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) register (Table 23) stores the interrupt requests for both vectored and polled interrupts. When a request is presented to the interrupt controller, the corresponding bit in the IRQ0 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 0 register to determine if any interrupt requests are pending

Table 23. Interrupt Request 0 Register (IRQ0)

BITS	7	6	5	4	3	2	1	0
FIELD	T2I	T1I	T0I	U0RXI	U0TXI	I2CI	SPII	ADCI
RESET	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FC0H							

T2I—Timer 2 Interrupt Request

0 = No interrupt request is pending for Timer 2.

1 = An interrupt request from Timer 2 is awaiting service.

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.

Interrupt Control Register

The Interrupt Control (IRQCTL) register (Table 37) contains the master enable bit for all interrupts.

Table 37. Interrupt Control Register (IRQCTL)

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

IRQE—Interrupt Request Enable

This bit is set to 1 by execution of an EI (Enable Interrupts) or IRET (Interrupt Return) instruction, or by a direct register write of a 1 to this bit. It is reset to 0 by executing a DI instruction, eZ8 CPU acknowledgement of an interrupt request, or Reset.

0 = Interrupts are disabled.

1 = Interrupts are enabled.

Reserved

These bits must be 0.

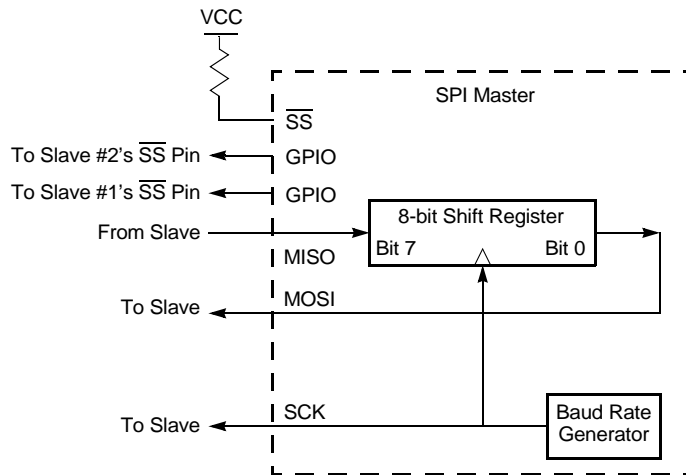


Figure 75. SPI Configured as a Master in a Single Master, Multiple Slave System

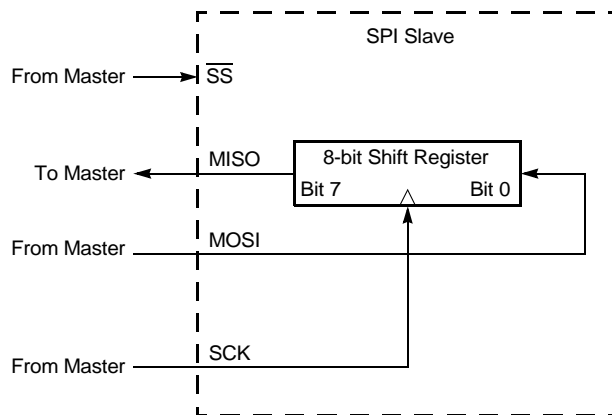


Figure 76. SPI Configured as a Slave

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 trans-

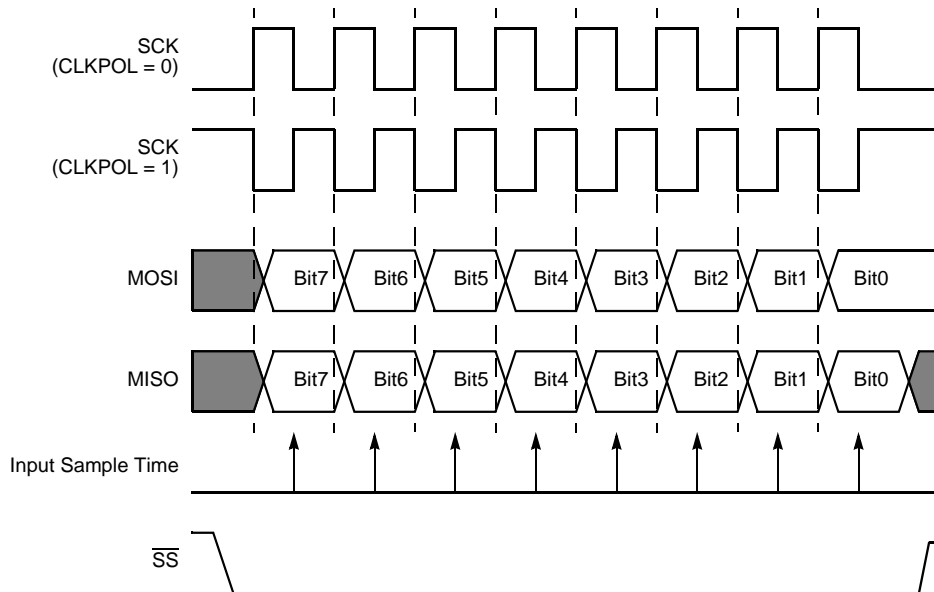


Figure 78. SPI Timing When PHASE is 1

Multi-Master Operation

In a multi-master SPI system, all SCK pins are tied together, all MOSI pins are tied together and all MISO pins are tied together. All SPI pins must then be configured in open-drain mode to prevent bus contention. At any one time, only one SPI device is configured as the Master and all other SPI devices on the bus are configured as Slaves. The Master enables a single Slave by asserting the \overline{SS} pin on that Slave only. Then, the single Master drives data out its SCK and MOSI pins to the SCK and MOSI pins on the Slaves (including those which are not enabled). The enabled Slave drives data out its MISO pin to the MISO Master pin.

For a Master device operating in a multi-master system, if the \overline{SS} pin is configured as an input and is driven Low by another Master, the COL bit is set to 1 in the SPI Status Register. The COL bit indicates the occurrence of a multi-master collision (mode fault error condition).

1. Software writes the I²C Data register with a 7-bit slave address followed by a 1 (read).
2. Software asserts the START bit of the I²C Control register.
3. Software asserts the NAK bit of the I²C Control register so that after the first byte of data has been read by the I²C Controller, a Not Acknowledge is sent to the I²C slave.
4. The I²C Controller sends the START condition.
5. The I²C Controller sends the address and read bit by the SDA signal.
6. The I²C slave sends an Acknowledge by pulling the SDA signal Low during the next high period of SCL.
7. The I²C Controller reads the first byte of data from the I²C slave.
8. The I²C Controller asserts the Receive interrupt.
9. Software responds by reading the I²C Data register.
10. The I²C Controller sends a NAK to the I²C slave.
11. A NAK interrupt is generated by the I²C Controller.
12. Software responds by setting the STOP bit of the I²C Control register.
13. A STOP condition is sent to the I²C slave.

Reading a Transaction with a 10-Bit Address

Figure 82 illustrates the receive format for a 10-bit addressed slave. The 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	S	Slave Address 1st 7 bits	R=1	A	Data	A	Data	\bar{A}	P
---	-----------------------------	-----	---	---------------------------	---	---	-----------------------------	-----	---	------	---	------	-----------	---

Figure 82. Receive Data Format for a 10-Bit Addressed Slave

The first seven bits transmitted in the first byte are 11110XX. The two bits XX are the two most-significant bits of the 10-bit address. The lowest bit of the first byte transferred is the write signal.

The data transfer format for a receive operation on a 10-bit addressed slave is as follows:

1. Software writes an address 11110B followed by the two address bits and a 0 (write).
2. Software asserts the START bit of the I²C Control register.
3. The I²C Controller sends the Start condition.

**START—Send Start Condition**

This bit sends the Start condition. Once asserted, it is cleared by the I²C Controller after it sends the START condition or by deasserting the IEN bit. After this bit is set, the Start condition is sent if there is data in the I²C Data or I²C Shift register. If there is no data in one of these registers, the I²C Controller waits until data is loaded. If this bit is set while the I²C Controller is shifting out data, it generates a START condition after the byte shifts and the acknowledge phase completed. If the STOP bit is also set, it also waits until the STOP condition is sent before the START condition. If this bit is 1, it cannot be cleared to 0 by writing to the register. This bit clears when the I²C is disabled.

STOP—Send Stop Condition

This bit causes the I²C Controller to issue a Stop condition after the byte in the I²C Shift register has completed transmission or after a byte has been received in a receive operation. Once set, this bit is reset by the I²C 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. This bit clears when the I²C is disabled.

BIRQ—Baud Rate Generator Interrupt Request

This bit causes an interrupt to occur every time the baud rate generator counts down to zero. This bit allows the I²C Controller to be used as an additional counter when it is not being used elsewhere. This bit must only be set when the I²C Controller is disabled.

TXI—Enable TDRE interrupts

This bit enables interrupts when the I²C Data register is empty on the I²C Controller.

NAK—Send NAK

This bit sends a Not Acknowledge condition after the next byte of data has been read from the I²C slave. Once asserted, it is deasserted after a Not Acknowledge is sent or the IEN bit is deasserted.

FLUSH—Flush Data

Setting this bit to 1 clears the I²C Data register and sets the TDRE bit to 1. This bit allows flushing of the I²C Data register when an NAK is received after the data has been sent to the I²C Data register. Reading this bit always returns 0.

FILTEN—I²C Signal Filter Enable

Setting this bit to 1 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.

When the DMA is configured for two-byte word transfers, the DMAx I/O Address register must contain an even numbered address.

Table 72. DMAx I/O Address Register (DMAxIO)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_IO							
RESET	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FB1H, FB9H							

DMA_IO—DMA on-chip peripheral control register address

This byte sets the low byte of the on-chip peripheral control register address on Register File Page FH (addresses F00H to FFFH).

DMAx Address High Nibble Register

The DMAx Address High register specifies the upper four bits of address for the Start/Current and End Addresses of DMAx.

Table 73. DMAx Address High Nibble Register (DMAxH)

BITS	7	6	5	4	3	2	1	0
FIELD	DMA_END_H				DMA_START_H			
RESET	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR	FB2H, FHAH							

DMA_END_H—DMAx End Address High Nibble

These bits, used with the DMAx End Address Low register, form a 12-bit End Address. The full 12-bit address is given by {DMA_END_H[3:0], DMA_END[7:0]}.

DMA_START_H—DMAx Start/Current Address High Nibble

These bits, used with the DMAx Start/Current Address Low register, form a 12-bit Start/Current Address. The full 12-bit address is given by {DMA_START_H[3:0], DMA_START[7:0]}.

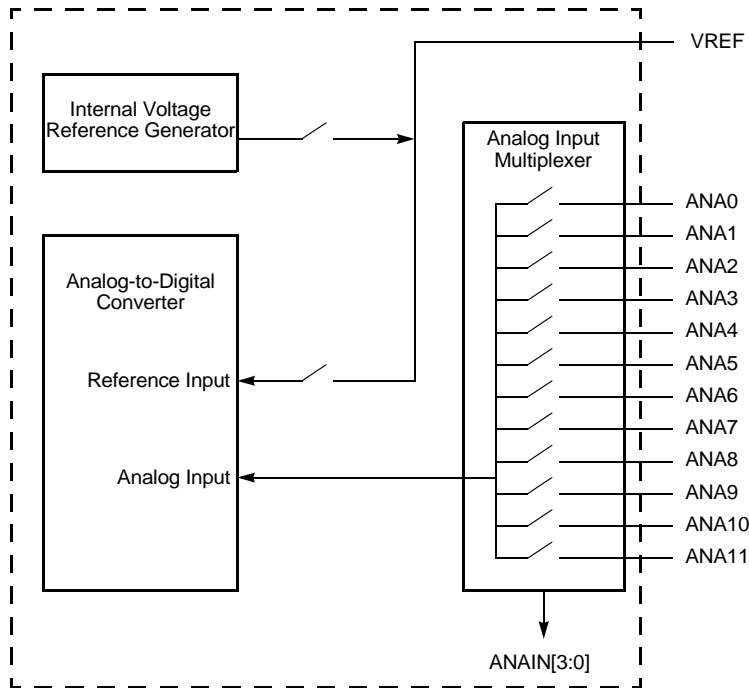


Figure 83. Analog-to-Digital Converter Block Diagram

Operation

Automatic Power-Down

If the ADC is idle (no conversions in progress) for 160 consecutive system clock cycles, portions of the ADC are automatically powered-down. From this power-down state, the ADC requires 40 system clock cycles to power-up. The ADC powers up when a conversion is requested using the ADC Control register.

Single-Shot Conversion

When configured for single-shot conversion, the ADC performs a single analog-to-digital conversion on the selected analog input channel. After completion of the conversion, the ADC shuts down. The steps for setting up the ADC and initiating a single-shot conversion are as follows:

1. Enable the desired analog inputs by configuring the general-purpose I/O pins for alternate function. This configuration disables the digital input and output drivers.
2. Write to the ADC Control register to configure the ADC and begin the conversion. The bit fields in the ADC Control register can be written simultaneously:
 - Write to `ANAIN[3 : 0]` to select one of the 12 analog input sources.
 - Clear `CONT` to 0 to select a single-shot conversion.
 - Write to `VREF` to enable or disable the internal voltage reference generator.
 - Set `CEN` to 1 to start the conversion.
3. `CEN` remains 1 while the conversion is in progress. A single-shot conversion requires 5129 system clock cycles to complete. If a single-shot conversion is requested from an ADC powered-down state, the ADC uses 40 additional clock cycles to power-up before beginning the 5129 cycle conversion.
4. When the conversion is complete, the ADC control logic performs the following operations:
 - 10-bit data result written to `{ADCD_H[7:0], ADCD_L[7:6]}`.
 - `CEN` resets to 0 to indicate the conversion is complete.
 - An interrupt request is sent to the Interrupt Controller.
5. If the ADC remains idle for 160 consecutive system clock cycles, it is automatically powered-down.

Continuous Conversion

When configured for continuous conversion, the ADC continuously performs an analog-to-digital conversion on the selected analog input. Each new data value over-writes the previous value stored in the ADC Data registers. An interrupt is generated only at the end of the first conversion after enabling.



Caution: In Continuous mode, users must be aware that ADC updates are limited by the input signal bandwidth of the ADC and the latency of the ADC and its digital filter. Step changes at the input are not seen at the next output from the ADC. The response of the ADC (in all modes) is limited by the input signal bandwidth and the latency.

The steps for setting up the ADC and initiating continuous conversion are as follows:

1. Enable the desired analog input by configuring the general-purpose I/O pins for alternate function. This disables the digital input and output driver.
2. Write to the ADC Control register to configure the ADC for continuous conversion. The bit fields in the ADC Control register may be written simultaneously:
 - Write to `ANAIN[3 : 0]` to select one of the 12 analog input sources.



- Power-on reset
- Voltage Brownout reset
- Asserting the $\overline{\text{RESET}}$ pin Low to initiate a Reset.
- Driving the DBG pin Low while the Z8F640x family 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 1.5 Stop bits (Figure 89)



Figure 89. OCD Data Format

OCD Auto-Baud Detector/Generator

To run over a range of baud rates (data 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 Z8F640x family device 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 92 lists minimum and recommended maximum baud rates for sample crystal frequencies.

Table 92. 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 (32KHz)	4.096	0.064



zero). If the Z8F640x family device is not in Debug mode or if the Read Protect Option Bit is enabled, this command returns FFH for all the data values.

```
DBG <-- 09H
DBG <-- {4'h0, Register Address[11:8]}
DBG <-- Register Address[7:0]
DBG <-- Size[7:0]
DBG --> 1-256 data bytes
```

- **Write Program Memory (0AH)**—The Write Program Memory command writes data to Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. If the Z8F640x family device is not in Debug mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG <-- 0AH
DBG <-- Program Memory Address[15:8]
DBG <-- Program Memory Address[7:0]
DBG <-- Size[15:8]
DBG <-- Size[7:0]
DBG <-- 1-65536 data bytes
```

- **Read Program Memory (0BH)**—The Read Program Memory command reads data from Program Memory. This command is equivalent to the LDC and LDCI instructions. Data can be read 1-65536 bytes at a time (65536 bytes can be read by setting size to zero). If the Z8F640x family device is not in Debug mode or if the Read Protect Option Bit is enabled, this command returns FFH for the data.

```
DBG <-- 0BH
DBG <-- Program Memory Address[15:8]
DBG <-- Program Memory Address[7:0]
DBG <-- Size[15:8]
DBG <-- Size[7:0]
DBG --> 1-65536 data bytes
```

- **Write Data Memory (0CH)**—The Write Data Memory command writes data to Data Memory. This command is equivalent to the LDE and LDEI instructions. Data can be written 1-65536 bytes at a time (65536 bytes can be written by setting size to zero). If the Z8F640x family device is not in Debug mode or if the Read Protect Option Bit is enabled, the data is discarded.

```
DBG <-- 0CH
DBG <-- Data Memory Address[15:8]
DBG <-- Data Memory Address[7:0]
DBG <-- Size[15:8]
DBG <-- Size[7:0]
DBG <-- 1-65536 data bytes
```

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 117. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the flag test operation is used to decide if the conditional jump is executed.

Table 117. 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
0011	3	ULE	Unsigned Less Than or Equal	(C OR Z) = 1
0100	4	OV	Overflow	V = 1
0101	5	MI	Minus	S = 1
0110	6	Z	Zero	Z = 1
0110	6	EQ	Equal	Z = 1
0111	7	C	Carry	C = 1
0111	7	ULT	Unsigned Less Than	C = 1
1000	8	T (or blank)	Always True	–
1001	9	GE	Greater Than or Equal	(S XOR V) = 0
1010	A	GT	Greater Than	(Z OR (S XOR V)) = 0
1011	B	UGT	Unsigned Greater Than	(C = 0 AND Z = 0) = 1
1100	C	NOV	No Overflow	V = 0
1101	D	PL	Plus	S = 0
1110	E	NZ	Non-Zero	Z = 0
1110	E	NE	Not Equal	Z = 0
1111	F	NC	No Carry	C = 0
1111	F	UGE	Unsigned Greater Than or Equal	C = 0



Ordering Information

Table 128. Ordering Information

Part	Flash KB (Bytes)	RAM KB (Bytes)	Max. Speed (MHz)	Temp (°C)	Voltage (V)	Package	Part Number
Z8 Encore![®] with 16KB Flash, Standard Temperature							
Z8 Encore! [®]	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F1601PM020SC
Z8 Encore! [®]	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F1601AN020SC
Z8 Encore! [®]	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F1601VN020SC
Z8 Encore! [®]	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F1602AR020SC
Z8 Encore! [®]	16 (16,384)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F1602VS020SC
Z8 Encore![®] with 24KB Flash, Standard Temperature							
Z8 Encore! [®]	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F2401PM020SC
Z8 Encore! [®]	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F2401AN020SC
Z8 Encore! [®]	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F2401VN020SC
Z8 Encore! [®]	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F2402AR020SC
Z8 Encore! [®]	24 (24,576)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F2402VS020SC
Z8 Encore![®] with 32KB Flash, Standard Temperature							
Z8 Encore! [®]	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F3201PM020SC
Z8 Encore! [®]	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F3201AN020SC
Z8 Encore! [®]	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F3201VN020SC
Z8 Encore! [®]	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F3202AR020SC
Z8 Encore! [®]	32 (32,768)	2 (2048)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F3202VS020SC
Z8 Encore![®] with 48KB Flash, Standard Temperature							
Z8 Encore! [®]	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	PDIP-40	Z8F4801PM020SC
Z8 Encore! [®]	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	LQFP-44	Z8F4801AN020SC
Z8 Encore! [®]	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	PLCC-44	Z8F4801VN020SC
Z8 Encore! [®]	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	LQFP-64	Z8F4802AR020SC
Z8 Encore! [®]	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	PLCC-68	Z8F4802VS020SC
Z8 Encore! [®]	48 (49,152)	4 (4096)	20	0 to +70	3.0 - 3.6	QFP-80	Z8F4803FT020SC



Table 128. Ordering Information (Continued)

Part	Flash KB (Bytes)	RAM KB (Bytes)	Max. Speed (MHz)	Temp (°C)	Voltage (V)	Package	Part Number
Z8 Encore!® with 48KB Flash, Extended Temperature							
Z8 Encore!®	48 (49,152)	4 (4096)	20	-40 to +105	3.0 - 3.6	PDIP-40	Z8F4801PM020EC
Z8 Encore!®	48 (49,152)	4 (4096)	20	-40 to +105	3.0 - 3.6	LQFP-44	Z8F4801AN020EC
Z8 Encore!®	48 (49,152)	4 (4096)	20	-40 to +105	3.0 - 3.6	PLCC-44	Z8F4801VN020EC
Z8 Encore!®	48 (49,152)	4 (4096)	20	-40 to +105	3.0 - 3.6	LQFP-64	Z8F4802AR020EC
Z8 Encore!®	48 (49,152)	4 (4096)	20	-40 to +105	3.0 - 3.6	PLCC-68	Z8F4802VS020EC
Z8 Encore!®	48 (49,152)	4 (4096)	20	-40 to +105	3.0 - 3.6	QFP-80	Z8F4803FT020EC
Z8 Encore!® with 64KB Flash, Extended Temperature							
Z8 Encore!®	64 (65,536)	4 (4096)	20	-40 to +105	3.0 - 3.6	PDIP-40	Z8F6401PM020EC
Z8 Encore!®	64 (65,536)	4 (4096)	20	-40 to +105	3.0 - 3.6	LQFP-44	Z8F6401AN020EC
Z8 Encore!®	64 (65,536)	4 (4096)	20	-40 to +105	3.0 - 3.6	PLCC-44	Z8F6401VN020EC
Z8 Encore!®	64 (65,536)	4 (4096)	20	-40 to +105	3.0 - 3.6	LQFP-64	Z8F6402AR020EC
Z8 Encore!®	64 (65,536)	4 (4096)	20	-40 to +105	3.0 - 3.6	PLCC-68	Z8F6402VS020EC
Z8 Encore!®	64 (65,536)	4 (4096)	20	-40 to +105	3.0 - 3.6	QFP-80	Z8F6403FT020EC
Z8 Encore!® Development Tools							
Z8 Encore!® Developer Kit						Z8ENCORE000ZCO	

Contact ZILOG's worldwide customer support center for more information on ordering the Z8 Encore!®. The customer support center is open from 7 a.m. to 7 p.m. Pacific Time.

The customer support toll-free number for ZiLOG is 1-877-ZiLOGCS (1-877-945-6427). For Z8 Encore!® the customer support toll-free number is 1-866-498-3636. The FAX number for the customer support center is 1-603-316-0345. Customers can also gain access to customer support using the ZiLOG website. Z8 Encore!® has its own web page at www.zilog.com/z8encore.

For customer service, navigate your browser to:

- <http://register.zilog.com/login.asp?login=servicelogin>

For technical support, navigate your browser to:

- <http://register.zilog.com/login.asp?login=supportlogin>