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	Active
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, IrDA, LINbus, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1680hh020eg

Chapter 3. Address Space

The eZ8 CPU can access the following three distinct address spaces:

- The Register File contains addresses for general-purpose registers, eZ8 CPU, peripherals and GPIO 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 contain data only

These three address spaces are covered briefly in the following sections. For more details about the eZ8 CPU and its address space, refer to the eZ8 CPU Core User Manual (UM0128), available for download at www.zilog.com.

3.1. Register File

The Register File address space in the Z8 Encore!® MCU is 4KB (4096 bytes). The Register File is composed of two sections: control registers and general-purpose registers. When instructions are executed, registers defined as sources are read and registers defined as destinations are written. 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, on-chip peripherals and the input/output ports. These registers are located at addresses F00H to FFFH. Some of the addresses within the 256 B control register sections are reserved (that is, unavailable). Reading from a reserved Register File address returns an undefined value. Zilog does not recommend writing to the reserved Register File addresses because doing so can produce unpredictable results.

The on-chip Register RAM always begins at address 000H in the Register File address space. The F1680 Series MCU contains 1KB or 2KB of on-chip Register RAM. Reading from Register File addresses outside the available RAM addresses (and not within the control register address space) returns an undefined value. Writing to these Register File addresses produces no effect.

In addition, the F1680 Series MCU contains 1KB of on-chip Program RAM. Normally it is used as Program RAM and is present in the Program Memory address space (see the Program Memory section on page 20). However, it can also be used as additional Register RAM present in the Register File address space 800H–BFFH (1KB Program RAM, 2KB Register RAM), or 400H–7FFH (1KB Program RAM, 1KB Register RAM), if you do not

Chapter 5. Reset, Stop Mode Recovery and Low-Voltage Detection

The Reset Controller within the F1680 Series MCU controls Reset and Stop Mode Recovery operations and provides indication of low-voltage supply conditions. During the operation, the following events cause a Reset:

- Power-On Reset (POR)
- Voltage Brown-Out (VBO) protection
- Watchdog Timer (WDT) time-out (when configured by the WDT_RES Flash option bit to initiate a Reset)
- External $\overline{\text{RESET}}$ pin assertion (when the alternate RESET function is enabled by the GPIO register)
- On-Chip Debugger initiated Reset (OCDCTL[0] set to 1)

When the device is in STOP Mode, a Stop Mode Recovery is initiated by each of the following:

- Watchdog Timer time-out
- GPIO Port input pin transition on an enabled Stop Mode Recovery source
- Interrupt from a timer or comparator enabled for STOP Mode operation

The low-voltage detection circuitry on the device features the following:

- The low-voltage detection threshold level is user-defined
- It generates an interrupt when the supply voltage drops below a user-defined level

5.1. Reset Types

The F1680 Series MCU provides various types of Reset operation. Stop Mode Recovery is considered a form of Reset. Table 9 lists the types of Reset and their operating characteristics. The System Reset is longer, if the external crystal oscillator is enabled by the Flash option bits allowing additional time for oscillator start-up.

eZ8 CPU services the Timer interrupt request following the normal Stop Mode Recovery sequence.

5.3.3. Stop Mode Recovery Using Comparator Interrupt

If Comparator enabled for STOP Mode operation interrupts during STOP Mode, the device undergoes a Stop Mode Recovery sequence. In the Reset Status Register, the stop bit is set to 1. If the F1680 Series MCU is configured to respond to interrupts, the eZ8 CPU services the comparator interrupt request following the normal Stop Mode Recovery sequence.

5.3.4. Stop Mode Recovery Using GPIO Port Pin Transition

Each of the GPIO port pins can be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recovery source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. In the Reset Status Register, the stop bit is set to 1.

! **Caution:** In STOP Mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin until the end of the Stop Mode Recovery delay. As a result, short pulses on the Port pin can initiate Stop Mode Recovery without being written to the Port Input Data Register or without initiating an interrupt (if enabled for that pin).

5.3.5. Stop Mode Recovery Using External $\overline{\text{RESET}}$ Pin

When the F1680 Series MCU is in STOP Mode and the external $\overline{\text{RESET}}$ pin is driven Low, a System Reset occurs. Because of a glitch filter operating on the $\overline{\text{RESET}}$ pin, the Low pulse must be greater than the minimum width specified, or it is ignored. For details, see the Electrical Characteristics chapter on page 349.

5.4. Low-Voltage Detection

In addition to the VBO Reset described earlier, it is also possible to generate an interrupt when the supply voltage drops below a user-selected value. For more details about the available Low-Voltage Detection (LVD) threshold levels, see the Trim Option Bits at Address 0000H (TTEMP0) section on page 282.

When the supply voltage drops below the LVD threshold, the LVD bit of the RSTSTAT Register is set to 1. This bit remains 1 until the low-voltage condition elapses. Reading or

transitions from a Low to High. This configuration ensures a time gap between the removal of one PWM output and the assertion of its complement.

Observe the following steps to configure a timer for PWM DUAL OUTPUT Mode and initiate the PWM operation:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for PWM DUAL OUTPUT Mode. Setting the mode also involves writing to TMODE[3] bit in the TxCTL0 Register
   - Set the prescale value
   - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output Alternate Function
2. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`). This value only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of `0001H`.
3. Write to the Timer PWM0 High and Low Byte registers to set the PWM value.
4. Write to the Timer Control 0 Register:
   - To set the PWM deadband delay value
   - To choose the timer clock source
5. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
6. Write to the Timer Reload High and Low Byte registers to set the reload value (PWM period). The reload value must be greater than the PWM value.
7. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
8. Configure the associated GPIO port pin for the Timer Output and Timer Output Complement alternate functions.
9. Write to the Timer Control 1 Register to enable the timer and initiate counting.

The PWM period is calculated using the following equation:

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 PWM time-out period.

If TPOL is set to 0, the ratio of the PWM output High time to the total period is calculated using the following equation:

9.2.3.12. CAPTURE/COMPARE Mode

In CAPTURE/COMPARE Mode, the timer begins counting on the first external Timer Input transition. The appropriate transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control 1 Register. The Timer counts timer clocks up to the 16-bit reload value.

Every subsequent appropriate transition (after the first) of the Timer Input signal captures the current count value. The Capture value is written to the Timer PWM0 High and Low Byte registers. When the Capture event occurs, an interrupt is generated, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in Timer Control 0 Register is set to indicate the timer interrupt is due to an input capture event.

If no Capture event occurs, the timer counts up to the 16-bit Compare value stored in the Timer Reload High and Low Byte registers. Upon reaching the Compare value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to 0001H and counting resumes. The INPCAP bit in Timer Control 0 Register is cleared to indicate the timer interrupt is not due to an input capture event.

Observe the following steps to configure a timer for CAPTURE/COMPARE Mode and initiate the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for CAPTURE/COMPARE 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 Control 2 Register to choose the timer clock source.
4. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
5. Write to the Timer Reload High and Low Byte registers to set the Compare value.
6. If required, enable the timer interrupt and set the timer-interrupt priority by writing to the relevant interrupt registers. By default, the timer interrupt will be generated for both input capture and reload events. If required, configure the timer interrupt to be generated only at the input Capture event or the Reload event by setting TICONFIG field of the Timer Control 0 Register.
7. Configure the associated GPIO port pin for the Timer Input alternate function.
8. Write to the Timer Control 1 Register to enable the timer.

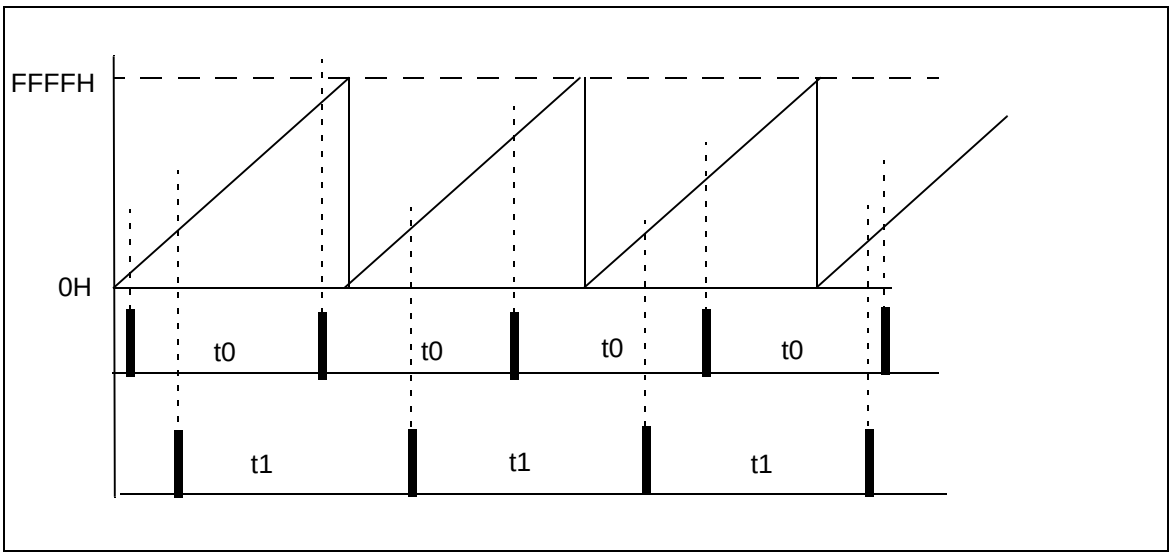

Figure 18. Count Max Mode with Channel Compare

10.7. Multi-Channel Timer Control Register Definitions

This section defines the features of the following Multi-Channel Timer Control registers.

Multi-Channel Timer High and Low Byte Registers: see page 130

Multi-Channel Timer Reload High and Low Byte Registers: see page 130

Multi-Channel Timer Subaddress Register: see page 131

Multi-Channel Timer Subregister x (0, 1, or 2): see page 132

Multi-Channel Timer Control 0, Control 1 Registers: see page 132

Multi-Channel Timer Channel Status 0 and Status 1 Registers: see page 135

Multi-Channel Timer Channel-y Control Registers: see page 137

Multi-Channel Timer Channel-y High and Low Byte Registers: see page 139

10.7.1. Multi-Channel Timer Address Map

Table 69 defines the byte address offsets for the Multi-channel Timer registers. For saving address space, a subaddress is used for the Timer Control 0, Timer Control 1, Channel Status 0, Channel Status 1, Channel-y Control, and Channel-y High and Low byte registers. Only the Timer High and Low Byte registers and the Reload High and Low Byte registers can be directly accessed.

Bit	Description (Continued)
[5] CHIEN	**Channel Interrupt Enable** This bit enables generation of channel interrupt. A channel interrupt is generated whenever there is a capture/compare event on the Timer Channel. 0 = Channel interrupt is disabled. 1 = Channel interrupt is enabled.
[4] CHUE	**Channel Update Enable** This bit determines whether writes to the Channel High and Low Byte registers are buffered when TEN = 1. Writes to these registers are not buffered when TEN = 0 regardless of the value of this bit. 0 = Writes to the Channel High and Low Byte registers are buffered when TEN = 1 and only take affect on the next end of cycle count. 1 = Writes to the Channel High and Low Byte registers are not buffered when TEN = 1.
[3]	Reserved; must be 0.
[2:0] CHOP	**Channel Operation Method** This field determines the operating mode of the channel. For a detailed description of the operating modes, see Count Up/Down Mode on page 123. 000 = One-Shot Compare operation. 001 = Continuous Compare operation. 010 = PWM Output operation. 011 = Capture operation. 100 – 111 = Reserved.

ModeStatus[4:0] field is set. By observing this bit, an indication of the level of noise in the network can be obtained.

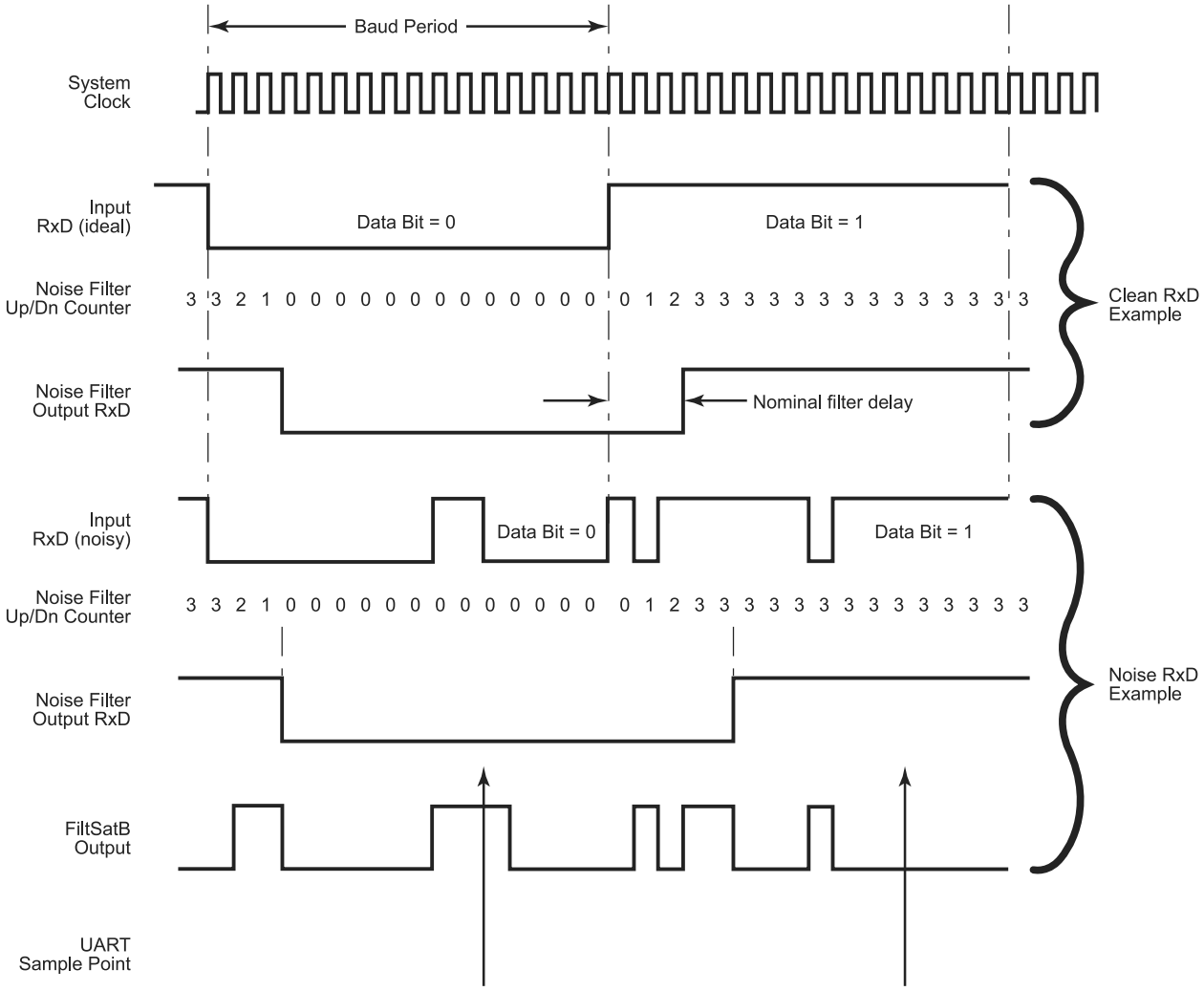

Figure 26. Noise Filter Operation

12.3.5. LIN-UART Control 0 Register

The LIN-UART Control 0 Register, shown in Table 89, configures the basic properties of LIN-UART's transmit and receive operations. A more detailed discussion of each bit follows the table.

Table 89. LIN-UART Control 0 Register (U0CTL0 = F42H)

Bit	7	6	5	4	3	2	1	0
Field	TEN	REN	CTSE	PEN	PSEL	SBRK	STOP	LBEN
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
Address	F42H, F4AH							

Note: R/W = Read/Write.

Bit	Description
[7] TEN	**Transmit Enable** This bit enables or disables the transmitter. The enable is also controlled by the $\overline{CTS}$ signal and the CTSE bit. If the $\overline{CTS}$ signal is Low and the CTSE bit is 1, the transmitter is enabled. 0 = Transmitter disabled. 1 = Transmitter enabled.
[6] REN	**Receive Enable** This bit enables or disables the receiver. 0 = Receiver disabled. 1 = Receiver enabled.
[5] CTSE	**Clear To Send Enable** 0 = The $\overline{CTS}$ signal has no effect on the transmitter. 1 = The LIN-UART recognizes the $\overline{CTS}$ signal as an enable control for the transmitter.
[4] PEN	**Parity Enable** This bit enables or disables parity. Even or odd is determined by the PSEL bit. 0 = Parity is disabled. This bit is overridden by the MPEN bit. 1 = The transmitter sends data with an additional parity bit and the receiver receives an additional parity bit.
[3] PSEL	**Parity Select** 0 = Even parity is sent as an additional parity bit for the transmitter/receiver. 1 = Odd parity is sent as an additional parity bit for the transmitter/receiver.

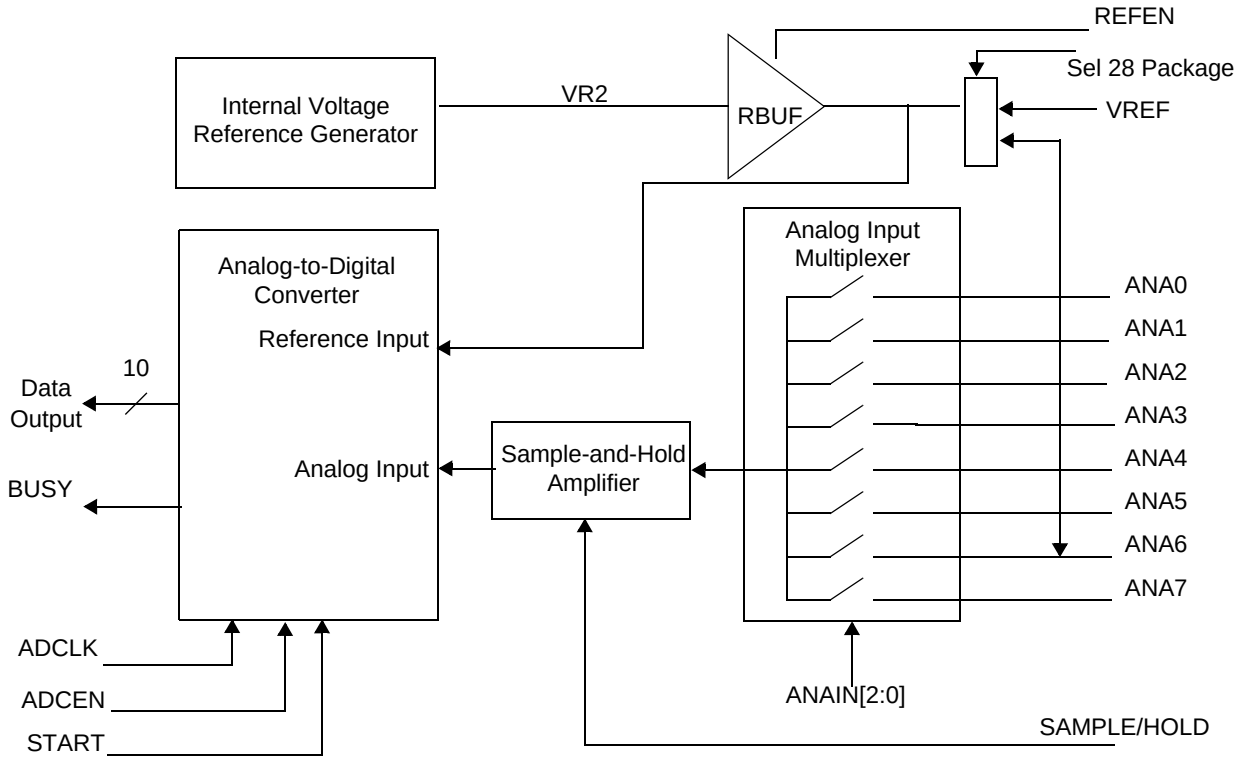

Figure 30. Analog-to-Digital Converter Block Diagram

14.2.1. ADC Timing

Each ADC measurement consists of 3 phases:

1. Input sampling (programmable, minimum of 1.8µs).
2. Sample-and-hold amplifier settling (programmable, minimum of 0.5µs).
3. Conversion is 13 ADCLK cycles.

Figure 31 displays the timing of an ADC conversion.

Bit	Description
[1] TEOF	**Transmit End of Frame** This bit is used in MASTER Mode to indicate that the data in the Transmit Data Register is the last byte of the transfer or frame. When the last byte has been sent $\overline{SS}$ (and SSV) will change state and TEOF will automatically clear. 0 = The data in the Transmit Data Register is not the last character in the message. 1 = The data in the Transmit Data Register is the last character in the message.
[0] SSV	**Slave Select Value** When SSIO = 1, writes to this register will control the value output on the $\overline{SS}$ pin. For more details, see the SSMD field of the ESPI Mode Register on page 217.

16.4.3. ESPI Control Register

The ESPI Control Register, shown in Table 111, configures the ESPI for transmit and receive operations.

Table 111. ESPI Control Register

Bits	7	6	5	4	3	2	1	0
Field	DIRQE	ESPIEN1	BRGCTL	PHASE	CLKPOL	WOR	MMEN	ESPIEN0
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
Address	F62H							

Bit	Description
[7] DIRQE	**Data Interrupt Request Enable** This bit is used to disable or enable data (TDRE and RDRNE) interrupts. Disabling the data interrupts is needed to control data transfer by polling. Error interrupts are not disabled. To block all ESPI interrupt sources, clear the ESPI interrupt enable bit in the Interrupt Controller. 0 = TDRE and RDRNE assertions do not cause an interrupt. Use this setting if controlling data transfer by software polling of TDRE and RDRNE. The TUND, COL, ABT and ROVR bits will cause an interrupt. 1 = TDRE and RDRNE assertions will cause an interrupt. TUND, COL, ABT and ROVR will also cause interrupts. Use this setting when controlling data transfer via interrupt handlers.

Table 138. Flash Frequency High Byte Register (FFREQH)

Bits	7	6	5	4	3	2	1	0
Field	FFREQH							
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
Address	FFAH							

Bit	Description
[7:0] FFREQH	**Flash Frequency High Byte** High byte of the 16-bit Flash Frequency value.

Table 139. Flash Frequency Low Byte Register (FFREQL)

Bits	7	6	5	4	3	2	1	0
Field	FFREQL							
Reset	0							
R/W	R/W							
Address	FFBH							

Bit	Description
[7:0] FFREQL	**Flash Frequency Low Byte** Low byte of the 16-bit Flash Frequency value.

Chapter 21. Flash Option Bits

Programmable Flash option bits allow user configuration of certain aspects of Z8 Encore! XP F1680 Series MCU operation. The feature configuration data is stored in Flash program memory and are read during Reset. The features available for control through the Flash option bits include:

- Watchdog Timer time-out response selection–interrupt or System Reset
- Watchdog Timer enabled at Reset
- The ability to prevent unwanted read access to user code in Program Memory
- The ability to prevent accidental programming and erasure of all or a portion of the user code in Program Memory
- VBO configuration-always enabled or disabled during STOP Mode to reduce STOP Mode power consumption
- LVD voltage threshold selection
- Oscillator mode selection for high, medium and low-power crystal oscillators or an external RC oscillator
- Factory trimming information for the IPO and Temperature Sensor

21.1. Operation

This section describes the types of option bits and their configuration in the Option Configuration registers.

21.1.1. Option Bit Configuration by Reset

Each time the Flash option bits are programmed or erased, the device must be Reset for the change to take effect. During any Reset operation (System Reset or Stop Mode Recovery), the Flash option bits are automatically read from Flash Program Memory and written to the Option Configuration registers. These Option Configuration registers control operation of the devices within the Z8 Encore! XP F1680 Series MCU. Option bit control is established before the device exits Reset and the eZ8 CPU begins code execution. The Option Configuration registers are not part of the Register File and are not accessible for read or write access.

the return from the subroutine, the read byte resides in working register R0 and the read status byte resides in working register R1. The bit fields of this status byte are defined in Table 160. Also, the user code should pop the address byte off the stack.

The read routine uses 16 bytes of stack space in addition to the 1 byte of address pushed by you. Sufficient memory must be available for this stack usage. Because of the Flash memory architecture, NVDS reads exhibit a nonuniform execution time. A read operation takes between 71 µs and 258 µs (assuming a 20 MHz system clock). Slower system clock speeds result in proportionally higher execution times.

NVDS byte reads from invalid addresses (those exceeding the NVDS array size) return 0xff. Illegal read operations have a 6 µs execution time. The status byte returned by the NVDS read routine is zero for successful read. If the status byte is nonzero, there is a corrupted value in the NVDS array at the location being read. In this case, the value returned in R0 is the byte most recently written to the array that does not have an error.

Table 160. Read Status Byte

Bits	7	6	5	4	3	2	1	0
Field	Reserved			DE	Reserved	FE	IGADDR	Reserved
Default Value	0	0	0	0	0	0	0	0

Bit	Description
[7:5]	Reserved; must be 0.
[4] DE	**Data Error** When reading a NVDS address, if an error is found in the latest data corresponding to the NVDS address, this bit is set to 1. NVDS source code steps forward until it finds valid data at this address.
[3]	Reserved; must be 0.
[2] FE	**Flash Error** If a Flash error is detected, this bit is set to 1.
[1] IGADDR	**Illegal Address** When NVDS byte reads occur from invalid addresses (those exceeding the NVDS array size), this bit is set to 1. Note: When the NVDS array size is 256 bytes, there is no address exceeding the size: therefore the IGADDR bit cannot be used.
[0]	Reserved; must be 0.

22.2.3. Power Failure Protection

The NVDS routines employ error checking mechanisms to ensure a power failure endangers only the most recently written byte. Bytes previously written to the array are not perturbed. For this protection to function, the VBO must be enabled (see the Low-Power

Table 162. OCD Baud-Rate Limits

System Clock Frequency	Maximum Asynchronous Baud Rate (bits/s)	Minimum Baud Rate (bits/s)
20.0 MHz	2.5 M	39.1 k
1.0MHz	125 k	1.96K
32kHz	4096	64

If the OCD receives a Serial Break (ten or more continuous bits Low), the Autobaud Detector/Generator resets. The Autobaud Detector/Generator can then be reconfigured by sending 80H. If the Autobaud Detector overflows while measuring the Autobaud character, the Autobaud Detector will remain reset.

23.2.4. High Speed Synchronous

It is possible to operate the serial On-Chip Debugger at high speeds. To operate at high speeds, data must be synchronized with an external clock. High speed synchronous communication will only work when using an external clock source. To operate in high-speed synchronous mode, simply Autobaud to the appropriate speed. The Autobaud generator will automatically run at the appropriate baud rate.

Slow bus rise times due to the pullup resistor become a limiting factor when operating at high speeds. To compensate for slow rise times, the output driver can be configured to drive the line High. If the TXD (Transmit Drive) bit is set, the line will be driven both High and Low during transmission. The line starts being driven at the beginning of the start bit and stops being driven at the middle of the stop bit. If the TXDH (Transmit Drive High) bit is set, the line will be driven High until the input is High or the center of the bit occurs, whichever is first. If both TXD and TXDH are set, the pin will be driven High for one clock period at the beginning of each 0 to 1 transition. An example of a high-speed synchronous interface is displayed in Figure 60.

27.3. eZ8 CPU Instruction Notation

In the eZ8 CPU Instruction Summary section on page 336, the operands, condition codes, status flags and address modes are represented by the notational shorthand provided in Table 176.

Table 176. Notational Shorthand

Notation	Description	Operand	Range
b	Bit	b	b represents a value from 0 to 7 (000B to 111B)
cc	Condition Code	—	See the Condition Codes overview in the eZ8 CPU Core User Manual (UM0128)
DA	Direct Address	Addrs	Addrs. represents a number in the range of 0000H to FFFFH
ER	Extended Addressing Register	Reg	Reg. represents a number in the range of 000H to FFFH
IM	Immediate Data	#Data	Data is a number between 00H to FFH
Ir	Indirect Working Register	@Rn	n = 0 –15
IR	Indirect Register	@Reg	Reg. represents a number in the range of 00H to FFH
Irr	Indirect Working Register Pair	@RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14
IRR	Indirect Register Pair	@Reg	Reg. represents an even number in the range 00H to FEH
p	Polarity	p	Polarity is a single bit binary value of either 0B or 1B.
r	Working Register	Rn	n = 0–15
R	Register	Reg	Reg. represents a number in the range of 00H to FFH
RA	Relative Address	X	X represents an index in the range of +127 to –128, which is an offset relative to the address of the next instruction
rr	Working Register Pair	RRp	p = 0, 2, 4, 6, 8, 10, 12 or 14
RR	Register Pair	Reg	Reg. represents an even number in the range of 00H to FEH
Vector	Vector Address	Vector	Vector represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to –128 range.

Tables 178 through 185 contain the instructions belonging to each group and the number of operands required for each instruction. Some instructions appear in more than one table as these instructions are to be considered as a subset of more than one category. Within these tables, the source operand is identified as src, the destination operand is dst and a condition code is cc.

Table 178. Arithmetic Instructions

Mnemonic	Operands	Instruction
ADC	dst, src	Add with Carry
ADCX	dst, src	Add with Carry using Extended Addressing
ADD	dst, src	Add
ADDX	dst, src	Add using Extended Addressing
CP	dst, src	Compare
CPC	dst, src	Compare with Carry
CPCX	dst, src	Compare with Carry using Extended Addressing
CPX	dst, src	Compare using Extended Addressing
DA	dst	Decimal Adjust
DEC	dst	Decrement
DECW	dst	Decrement Word
INC	dst	Increment
INCW	dst	Increment Word
MULT	dst	Multiply
SBC	dst, src	Subtract with Carry
SBCX	dst, src	Subtract with Carry using Extended Addressing
SUB	dst, src	Subtract
SUBX	dst, src	Subtract using Extended Addressing

27.5. eZ8 CPU Instruction Summary

Table 186 summarizes the eZ8 CPU instructions. The table identifies the addressing modes employed by the instruction, the effect upon the Flags register, the number of CPU clock cycles required for the instruction fetch and the number of CPU clock cycles required for the instruction execution.

Table 186. eZ8 CPU Instruction Summary

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Op Code(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
ADC dst, src	dst ← dst + src + C	r	r	12	*	*	*	*	0	*	2	3
		r	Ir	13							2	4
		R	R	14							3	3
		R	IR	15							3	4
		R	IM	16							3	3
		IR	IM	17							3	4
ADCX dst, src	dst ← dst + src + C	ER	ER	18	*	*	*	*	0	*	4	3
		ER	IM	19							4	3
ADD dst, src	dst ← dst + src	r	r	02	*	*	*	*	0	*	2	3
		r	Ir	03							2	4
		R	R	04							3	3
		R	IR	05							3	4
		R	IM	06							3	3
		IR	IM	07							3	4
ADDX dst, src	dst ← dst + src	ER	ER	08	*	*	*	*	0	*	4	3
		ER	IM	09							4	3

Flags notation:
* = Value is a function of the result of the operation.
– = Unaffected.
X = Undefined.
0 = Reset to 0.
1 = Set to 1.

Table 197. Comparator Electrical Characteristics

		TA = 0°C to +70°C TA = –40°C to +105°C V_{DD} = 1.8V to 3.6V				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
V_{OS}	Input DC Offset	–	5	–	mV	
V_{CREF_P}	Programmable Internal Reference Voltage Range	0	–	1.8	V	
V_{CREF_D}	Default Internal Reference Voltage	0.90	1.0	1.10	V	
$I_{DD}CMP$	Comparator Active Current	–	–	400	µA	
$I_{DDQ}CMP$	Comparator Quiescent Current	–	5	–	nA	
V_{HYS}	Input Hysteresis	–	8	–	mV	
T_{PROP}	Propagation Delay	–	100	–	ns	

Table 198. Temperature Sensor Electrical Characteristics

		TA = 0°C to +70°C TA = –40°C to +105°C							
		V_{DD} = 2.7 to 3.6V			V_{DD} = 1.8 to 2.7V				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
T_{AERR}	Temperature Sensor Output Error	–7	–	+7	–10	–	+10	°C	–40°C to +105°C (as measured by ADC)
		–1.5	–	+1.5	–3	–	+3	°C	+20°C to +30°C (as measured by ADC)
		–10	–	10	–15	–	15	°C	–40°C to +105°C (as measured by comparator)
$I_{DD}TEMP$	Temperature Sensor Active Current	–	–	100	–	–	100	µA	
$I_{DDQ}TEMP$	Temperature Sensor Quiescent Current	–	5	–	–	5	–	nA	
T_{WAKE}	Time for Wake up	–	80	100	–	80	100	µs	