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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LED, LVD, POR, PWM, Temp Sensor, WDT
Number of I/O	23
Program Memory Size	24KB (24K 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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f2480hj020eg

Table 5. Pin Characteristics (20-, 28-, 40- and 44-pin Devices) (Continued)

Symbol Mnemonic	Direction	Reset Direction	Active Low or Active High	Tristate Output	Internal Pull-up or Pull-down	Schmitt Trigger Input	Open Drain Output	5V Tolerance
PE[6:0]	I/O	I	N/A	Yes	Programmable pull-up	Yes	Yes, programmable	Yes, 5V tolerant inputs unless pull-ups are enabled
RESET/ PD0	I/O	I/O (defaults to $\overline{\text{RESET}}$)	Low (in RESET mode)	Yes (PD0 only)	Programmable for PD0; always On for $\overline{\text{RESET}}$	Yes	Programmable for PD0; always On for $\overline{\text{RESET}}$	Yes, 5V tolerant inputs unless pull-ups are enabled
V_{DD}	N/A	N/A	N/A	N/A			N/A	N/A
V_{SS}	N/A	N/A	N/A	N/A			N/A	N/A

Table 7. F1680 Series MCU Flash Memory Information Area Map

Program Memory Address (Hex)	Function
FE00–FE3F	Zilog option bits
FE40–FE53	Part Number: 20-character ASCII alphanumeric code Left-justified and filled with FH
FE54–FE5F	Reserved
FE60–FE7F	Zilog calibration data (only use the first two bytes FE60 and FE61)
FE80–FFFF	Reserved

Table 8. Register File Address Map (Continued)

Address (Hex)	Register Description	Mnemonic	Reset (Hex)[1]	Page #
F17	Timer 2 Control 1	T2CTL1	00	113
F28	Timer 2 PWM1 High Byte	T2PWM1H	00	111
F29	Timer 2 PWM1 Low Byte	T2PWM1L	00	111
F2A	Timer 2 Control 2	T2CTL2	00	117
F2B	Timer 2 Status	T2STA	00	118
F2E	Timer 2 Noise Filter Control	T2NFC	00	119
F2F–F3F	Reserved	—	XX	
LIN UART 0				
F40	LIN UART0 Transmit Data	U0TXD	XX	163
	LIN UART0 Receive Data	U0RXD	XX	164
F41	LIN UART0 Status 0—Standard UART Mode	U0STAT0	0000011Xb	165
	LIN UART0 Status 0—LIN Mode	U0STAT0	00000110b	166
F42	LIN UART0 Control 0	U0CTL0	00	170
F43	LIN UART0 Control 1—Multiprocessor Control	U0CTL1	00	172
	LIN UART0 Control 1—Noise Filter Control	U0CTL1	00	174
	LIN UART0 Control 1—LIN Control	U0CTL1	00	175
F44	LIN UART0 Mode Select and Status	U0MDSTAT	00	168
F45	UART0 Address Compare	U0ADDR	00	177
F46	UART0 Baud Rate High Byte	U0BRH	FF	177
F47	UART0 Baud Rate Low Byte	U0BRL	FF	178
LIN UART 1				
F48	LIN UART1 Transmit Data	U1TXD	XX	163
	LIN UART1 Receive Data	U1RXD	XX	164
F49	LIN UART1 Status 0—Standard UART Mode	U1STAT0	0000011Xb	165
	LIN UART1 Status 0—LIN Mode	U1STAT0	00000110b	166
F4A	LIN UART1 Control 0	U1CTL0	00	170

Notes:
1. XX=Undefined.
2. The Reserved space can be configured as General-Purpose Register File RAM depending on the user option bits (see the User Option Bits chapter on page 277) and the on-chip PRAM size (see the Ordering Information chapter on page 372). If the PRAM is programmed as General-Purpose Register File RAM on Reserved space, the starting address always begins immediately after the end of General-Purpose Register File RAM.

7.4. Direct LED Drive

The Port C pins provide a current synchronized output capable of driving an LED without requiring an external resistor. The output synchronizes current at programmable levels of 3mA, 7mA, 13mA and 20mA. This mode is enabled through the Alternate Function sub-register AFS1 and is programmable through the LED control registers. For proper function, the LED anode must be connected to V_{DD} and the cathode to the GPIO pin.

Using all Port C pins in LED drive mode with maximum current can result in excessive total current. For the maximum total current for the applicable package, see the Electrical Characteristics chapter on page 349.

7.5. Shared Reset Pin

On all the devices, the Port D0 pin shares function with a bidirectional reset pin. Unlike all other I/O pins, this pin does not default to GPIO pin on power-up. This pin acts as a bidirectional input/open-drain output reset with an internal pull-up until user software reconfigures it as GPIO PD0. The Port D0 pin is output-only when in GPIO Mode, and must be configured as an output. PD0 supports the High Drive feature but not the Stop Mode Recovery feature.

7.6. Crystal Oscillator Override

For systems using the crystal oscillator, PA0 and PA1 is used to connect the crystal. When the main crystal oscillator is enabled (see the Oscillator Control1 Register section on page 320), the GPIO settings are overridden and PA0 and PA1 is disabled.

7.7. 32kHz Secondary Oscillator Override

For systems using a 32kHz secondary oscillator, PA2 and PA3 is used to connect a watch crystal. When the 32kHz secondary oscillator is enabled (see the Oscillator Control1 Register section on page 320), the GPIO settings are overridden and PA2 and PA3 is disabled.

7.8. 5V Tolerance

All GPIO pins, including those that share functionality with an ADC, crystal or comparator signals are 5V-tolerant and can handle inputs higher than V_{DD} even with the pull-ups enabled.

Table 18. Port Alternate Function Mapping, 28-Pin Parts[1,2]

Port	Pin	Mnemonic	Alternate Function Description	Alternate Function Set Register AFS1
Port A	PA0	T0IN/$\overline{\text{T0OUT}}$	Timer 0 Input/Timer 0 Output Complement	AFS1[0]: 0
		Reserved		AFS1[0]: 1
	PA1	T0OUT	Timer 0 Output	AFS1[1]: 0
		Reserved		AFS1[1]: 1
	PA2	DE0	UART 0 Driver Enable	AFS1[2]: 0
		Reserved		AFS1[2]: 1
	PA3	$\overline{\text{CTS0}}$	UART 0 Clear to Send	AFS1[3]: 0
		Reserved		AFS1[3]: 1
	PA4	RXD0/IRRX0	UART 0/IrDA 0 Receive Data	AFS1[4]: 0
		Reserved		AFS1[4]: 1
	PA5	TXD0/IRTX0	UART 0/IrDA 0 Transmit Data	AFS1[5]: 0
		Reserved		AFS1[5]: 1
	PA6	T1IN/$\overline{\text{T1OUT}}$	Timer 1 Input/Timer 1 Output Complement	AFS1[6]: 0
		SCL	I^2C Serial Clock	AFS1[6]: 1
	PA7	T1OUT	Timer 1 Output	AFS1[7]: 0
		SDA	I^2C Serial Data	AFS1[7]: 1

Notes:
1. Because there are at most two choices of alternate functions for some pins in Ports A and B, the Alternate Function Set Register (AFS2) is implemented but not used to select the function. The alternate function selection must also be enabled, as described in the Port A–E Alternate Function Subregisters section on page 61.
2. Because there is only one alternate function for each Port D and Port E pin, the Alternate Function Set registers are not implemented for Ports D and E. Enabling the alternate function selections automatically enables the associated alternate function, as described in the Port A–E Alternate Function Subregisters section on page 61.

7.11.14. LED Drive Level Registers

Two LED Drive Level registers consist of the LED Drive Level High Bit Register (LEDLVLH[7:0]) and the LED Drive Level Low Bit Register (LEDLVLL[7:0]), as shown in Tables 34 and 35. Two control bits, LEDLVLH[x] and LEDLVLL[x], are used to select one of four programmable current drive levels for each associated Port C[x] pin. Each Port C pin is individually programmable.

Table 34. LED Drive Level High Bit Register (LEDLVLH)

Bits	7	6	5	4	3	2	1	0
Field	LEDLVLH							
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	F83H							

Table 35. LED Drive Level Low Bit Register (LEDLVLL)

Bits	7	6	5	4	3	2	1	0
Field	LEDLVLL							
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	F84H							

Bit	Description
[7:0] LEDLVLH, LEDLVLL	**LED Drive Level High Bit** **LED Drive Level Low Bit** These bits are used to set the LED drive current. {LEDLVLH[x], LEDLVLL[x]}, in which x=Port C[0] to Port C[7]. Select one of the following four programmable current drive levels for each Port C pin. 00 = 3 mA 01 = 7 mA 10 = 13 mA 11 = 20 mA

8.4.1. Interrupt Request 0 Register

The Interrupt Request 0 (IRQ0) Register, shown in Table 37, 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 37. 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
Address	FC0H							

Bit	Description
[7] T2I	**Timer 2 Interrupt Request** 0 = No interrupt request is pending for Timer 2. 1 = An interrupt request from Timer 2 is awaiting service.
[6] T1I	**Timer 1 Interrupt Request** 0 = No interrupt request is pending for Timer 1. 1 = An interrupt request from Timer 1 is awaiting service.
[5] T0I	**Timer 0 Interrupt Request** 0 = No interrupt request is pending for Timer 0. 1 = An interrupt request from Timer 0 is awaiting service.
[4] 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.
[3] 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.
[2] I2CI	**I^2C Interrupt Request** 0 = No interrupt request is pending for the I^2C. 1 = An interrupt request from I^2C is awaiting service.
[1] SPII	**SPI Interrupt Request** 0 = No interrupt request is pending for the SPI. 1 = An interrupt request from the SPI is awaiting service.
[0] ADCI	**ADC Interrupt Request** 0 = No interrupt request is pending for the ADC. 1 = An interrupt request from the ADC is awaiting service.

Bit	Description
[5] PA5CS	**PA5/Comparator 1 Selection** 0 = PA5 is used for the interrupt for PA5CS interrupt request. 1 = The Comparator 1 is used for the interrupt for PA5CS interrupt request.
[4:1] PAD*x*S	**PAx/PDx Selection** 0 = PAx is used for the interrupt for PAx/PDx interrupt request 1 = PDx is used for the interrupt for PAx/PDx interrupt request; an *x* indicates the specific GPIO port pin number (1–4).
[0]	Reserved; must be 0.

8.4.9. Interrupt Control Register

The Interrupt Control (IRQCTL) Register, shown in Table 51, contains the master enable bit for all interrupts.

Table 51. 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
Address	FCFH							

Bit	Description
[7] IRQE	**Interrupt Request Enable** This bit is set to 1 by executing an Enable Interrupts (EI) 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, a Reset, or by a direct register write of a 0 to this bit. 0 = Interrupts are disabled. 1 = Interrupts are enabled.
[6:0]	Reserved; must be 0.

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

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for CAPTURE RESTART Mode. Setting the mode also involves writing to TMODE[3] bit in the TxCTL0 Register
   - Set the prescale value
   - Set the Capture edge (rising or falling) for the Timer Input
2. Write to the Timer Control 2 Register to choose the timer clock source.
3. Write to the Timer Control 0 Register to set the timer interrupt configuration field TICONFIG.
4. Write to the Timer High and Low Byte registers to set the starting count value (typically `0001H`).
5. Write to the Timer Reload High and Low Byte registers to set the reload value.
6. Clear the Timer PWM High and Low Byte registers to `0000H`. This allows user software to determine if interrupts are 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 is generated by a Reload.
7. 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.
8. Configure the associated GPIO port pin for the Timer Input alternate function.
9. 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:

$$\text{Capture Elapsed Time (s)} = \frac{(\text{Capture Value} - \text{Start Value}) \times \text{Prescale}}{\text{Timer Clock Frequency (Hz)}}$$

11.2.1. Watchdog Timer Reload High and Low Byte Registers

The Watchdog Timer Reload High and Low Byte (WDTH, WDTL) registers, shown in Tables 81 and 82, form the 16-bit reload value that is loaded into the Watchdog Timer when a WDT instruction executes; this 16-bit reload value is {WDTH[7:0], WDTL[7:0]}. Writing to these registers following the unlock sequence sets the appropriate reload value. Reading from these registers returns the current WDT count value.

Table 81. Watchdog Timer Reload High Byte Register (WDTH = FF2H)

Bit	7	6	5	4	3	2	1	0
Field	WDTH							
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	FF2H							

Table 82. Watchdog Timer Reload Low Byte Register (WDTL = FF3H)

Bit	7	6	5	4	3	2	1	0
Field	WDTL							
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	FF3H							

Bit	Description
[7:0] WDTH, WDTL	**Watchdog Timer Reload High and Low Bytes** WDTH: The WDT Reload High Byte is the most significant byte, or bits [15:8] of the 16-bit WDT reload value. WDTL: The WDT Reload Low Byte is the least significant byte, or bits [7:0] of the 16-bit WDT reload value.

Wake-up message if it requires the master to initiate a LIN message frame. Following the Wake-up message, the master wakes up and initiates a new message. A Wake-up message is accomplished by pulling the bus Low for at least 250 µs but less than 5 ms. Transmitting a 00H character is one way to transmit the Wake-up message.

If the CPU is in STOP Mode, the LIN-UART is not active and the Wake-up message must be detected by a GPIO edge detect Stop Mode Recovery. The duration of the Stop Mode Recovery sequence can preclude making an accurate measurement of the Wake-up message duration.

If the CPU is in HALT or OPERATIONAL mode, the LIN-UART (if enabled) times the duration of the Wake-up and provides an interrupt following the end of the break sequence if the duration is ≥ 3 bit times. The total duration of the Wake-up message in bit times can be obtained by reading the RxBreakLength field in the Mode Select and Status register. After a Wake-up message has been detected, the LIN-UART can be placed (by software) either into LIN Master or LIN Slave Wait for Break states as appropriate. If the break duration exceeds 15 bit times, the RxBreakLength field contains the value Fh. If the LIN-UART is disabled, Wake-up message is detected via a port pin interrupt and timed by software. If the device is in STOP Mode, the High to Low transition on the port pin will bring the device out of STOP Mode.

The LIN Sleep state is selected by software setting LinState[1:0] = 00. The decision to move from an active state to sleep state is based on the LIN messages as interpreted by software.

12.1.10.5. LIN Slave Operation

LIN SLAVE Mode is selected by setting LMST = 0, LSLV = 1, ABEN = 1 or 0 and LinState[1:0] = 01b (Wait for Break state). The LIN slave detects the start of a new message by the break which appears to the Slave as a break of at least 11 bit times in duration. The LIN-UART detects the break and generates an interrupt to the CPU. The duration of the break is observable in the RxBreakLength field of the Mode Select and Status register. A break of less than 11 bit times in duration does not generate a break interrupt when the LIN-UART is in Wait for Break state. If the break duration exceeds 15 bit times, the RxBreakLength field contains the value Fh.

Following the break, the LIN-UART hardware automatically transits to the *Autobaud* state, where it autobauds by timing the duration of the first 8 bit times of the Synch character as defined in the LIN standard. The duration of the autobaud period is measured by the BRG Counter which will update every 8th system clock cycle between the start bit and the beginning of bit 7 of the autobaud sequence. At the end of the autobaud period, the duration measured by the BRG counter (auto baud period divided by 8) is automatically transferred to the Baud Reload High and Low registers if the ABEN bit of the LIN control register is set. If the BRG Counter overflows before reaching the start of bit 7 in the autobaud sequence the Autobaud Overrun Error interrupt occurs, the OE bit in the Status 0 Register is set and the Baud Reload registers are not updated. To autobaud within 2% of the master's baud rate, the slave system clock must be a minimum of 100 times the baud

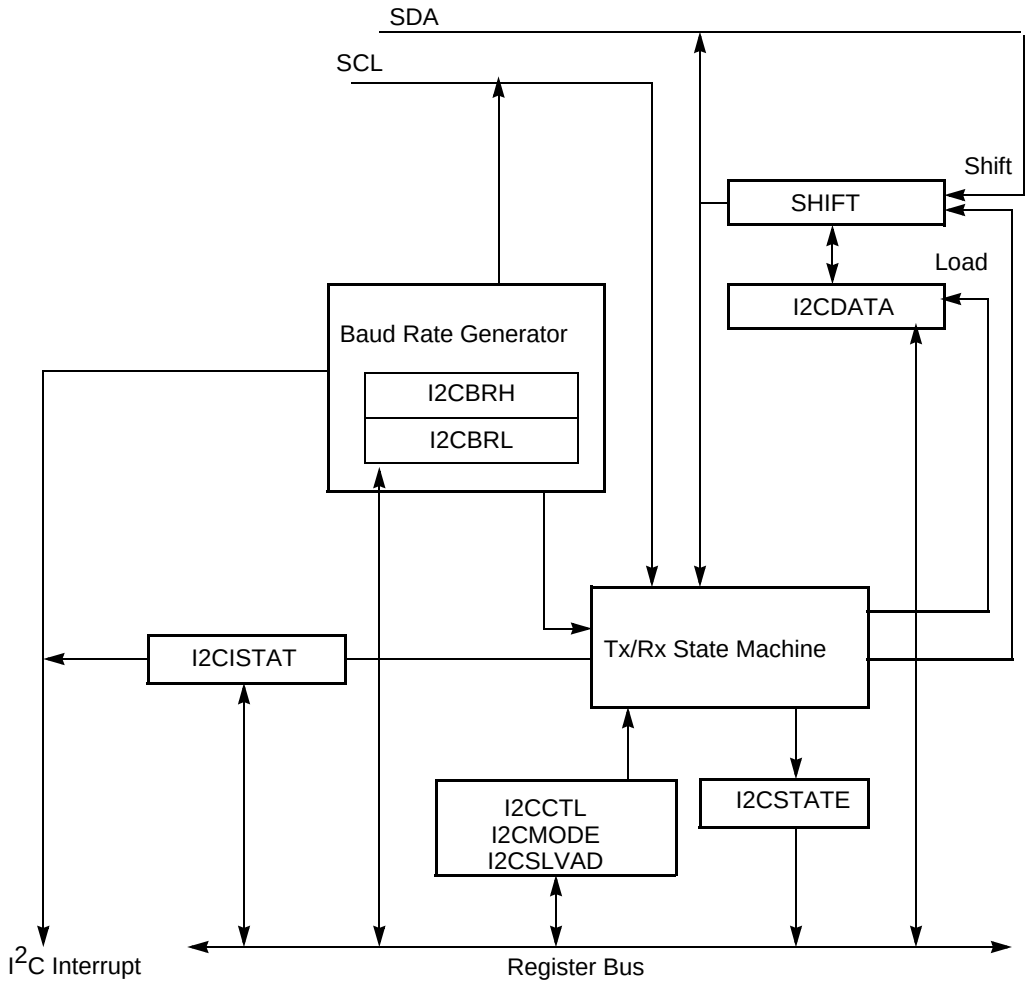


Figure 42. I²C Controller Block Diagram

17.1.1. I²C Master/Slave Controller Registers

Table 118 summarizes the I²C Master/Slave controller's software-accessible registers.

Table 118. I²C Master/Slave Controller Registers

Name	Abbreviation	Description
I²C Data	I2CDATA	Transmit/Receive Data Register.
I²C Interrupt Status	I2CISTAT	Interrupt status register.
I²C Control	I2CCTL	Control Register—basic control functions.

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 186. eZ8 CPU Instruction Summary (Continued)

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
AND dst, src	dst ← dst AND src	r	r	52	–	*	*	0	–	–	2	3
		r	Ir	53							2	4
		R	R	54							3	3
		R	IR	55							3	4
		R	IM	56							3	3
		IR	IM	57							3	4
ANDX dst, src	dst ← dst AND src	ER	ER	58	–	*	*	0	–	–	4	3
		ER	IM	59							4	3
ATM	Block all interrupt and DMA requests during execution of the next 3 instructions			2F	–	–	–	–	–	–	1	2
BCLR bit, dst	dst[bit] ← 0	r		E2	–	*	*	0	–	–	2	2
BIT p, bit, dst	dst[bit] ← p	r		E2	–	*	*	0	–	–	2	2
BRK	Debugger Break			00	–	–	–	–	–	–	1	1
BSET bit, dst	dst[bit] ← 1	r		E2	–	*	*	0	–	–	2	2
BSWAP dst	dst[7:0] ← dst[0:7]	R		D5	X	*	*	0	–	–	2	2
BTJ p, bit, src, dst	if src[bit] = p PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJNZ bit, src, dst	if src[bit] = 1 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
BTJZ bit, src, dst	if src[bit] = 0 PC ← PC + X		r	F6	–	–	–	–	–	–	3	3
			Ir	F7							3	4
CALL dst	SP ← SP -2 @SP ← PC PC ← dst	IRR		D4	–	–	–	–	–	–	2	6
		DA		D6							3	3
CCF	C ← ~C			EF	*	–	–	–	–	–-	1	2

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

The currents in Table 190 represent the power consumption without any peripherals active (unless otherwise noted). For design guidance, total power consumption will be the sum of all active peripheral currents plus the appropriate current characteristics shown below.

Table 190. Supply Current Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C				
Symbol	**Parameter**	**Min**	**Typical[1]**	**Max**	**Units**	**Conditions[2]**
I_{DDA1}	Active Mode Device Current Executing from Flash		8.5		mA	Typical: 20MHz[3, 4, 5, 6], V_{DD} = 3V, Flash, 25°C
I_{DDA2}	Active Mode Device Current Executing from PRAM		6		mA	Typical: 20MHz[3, 4, 5, 6], V_{DD} = 3V, PRAM, 25°C
I_{DDH}	Halt Mode Device Current		TBD		mA	Typical: 20MHz[3, 4, 5], V_{DD} typical, 25°C
I_{DDS1}	Stop Mode Device Current		2.5		µA	Typical: WDT, V_{DD} typical, 25 °C, all peripherals including VBO disabled[3, 4, 6]
I_{DDS2}	Stop Mode Device Current		<1		µA	Typical: V_{DD} typical, 25°C, all peripherals disabled including VBO and WDT[3, 4, 6]

Notes
1. These values are provided for design guidance only and are not tested in production.
2. Typical conditions are defined as 3.3 V at 25°C, unless otherwise noted.
3. All internal pull ups are disabled and all push-pull outputs are unloaded.
4. All open-drain outputs are pulled up to V_{DD}/AV_{DD} and are at a High state.
5. System clock source is an external square wave clock signal driven through the CLK-IN pin.
6. All inputs are at V_{DD}/AV_{DD} or V_{SS}/AV_{SS} as appropriate.

Figure 73 displays the STOP Mode supply current versus ambient temperature and V_{DD} level with all peripherals disabled.

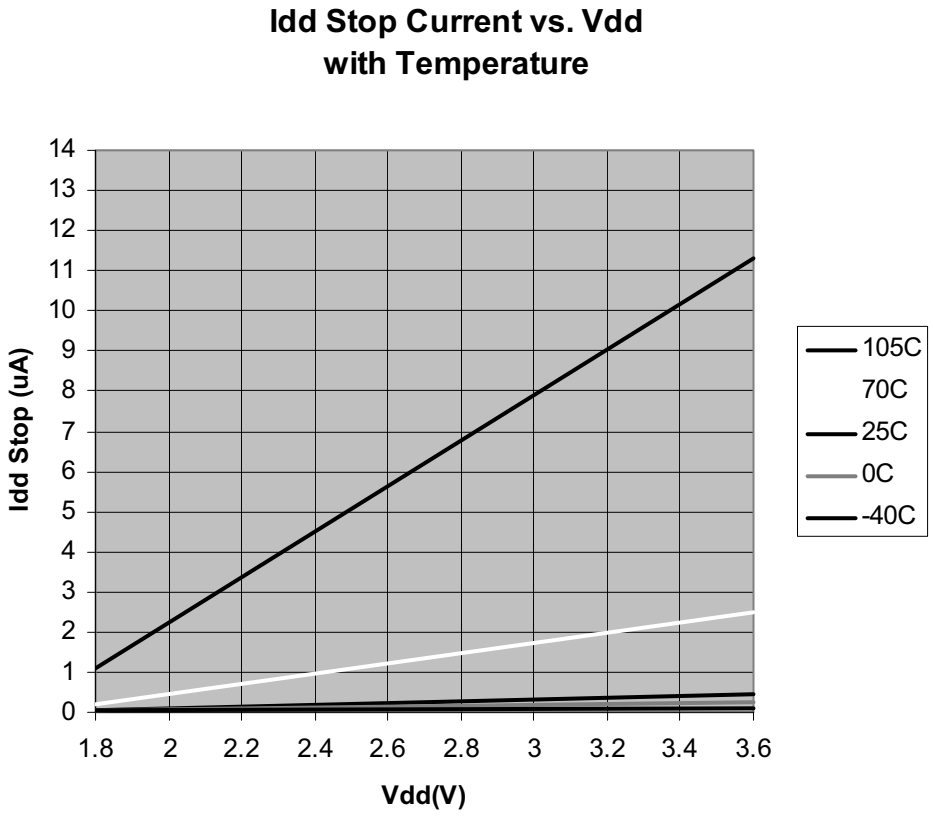


Figure 73. STOP Mode Current Consumption as a Function of V_{DD} with Temperature as a Parameter; all Peripherals Disabled

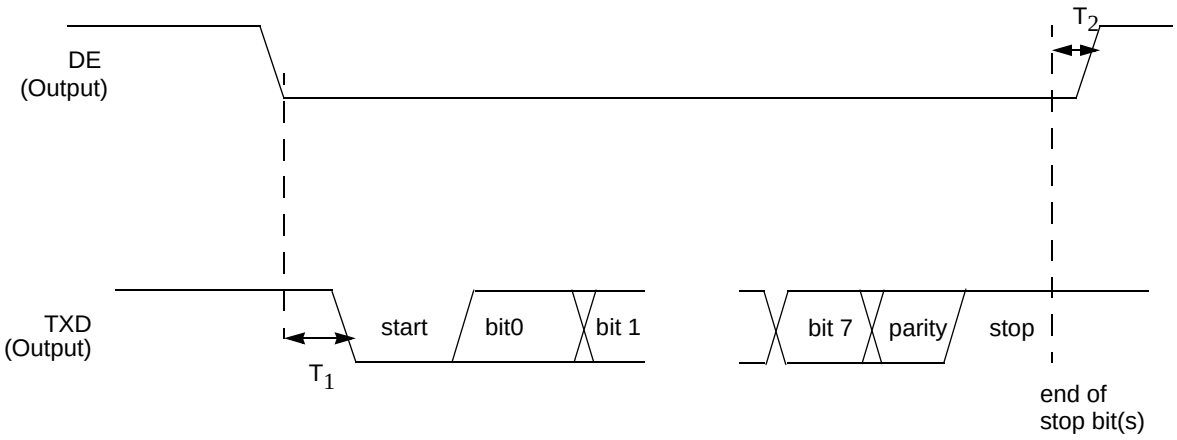


Figure 79. UART Timing Without CTS

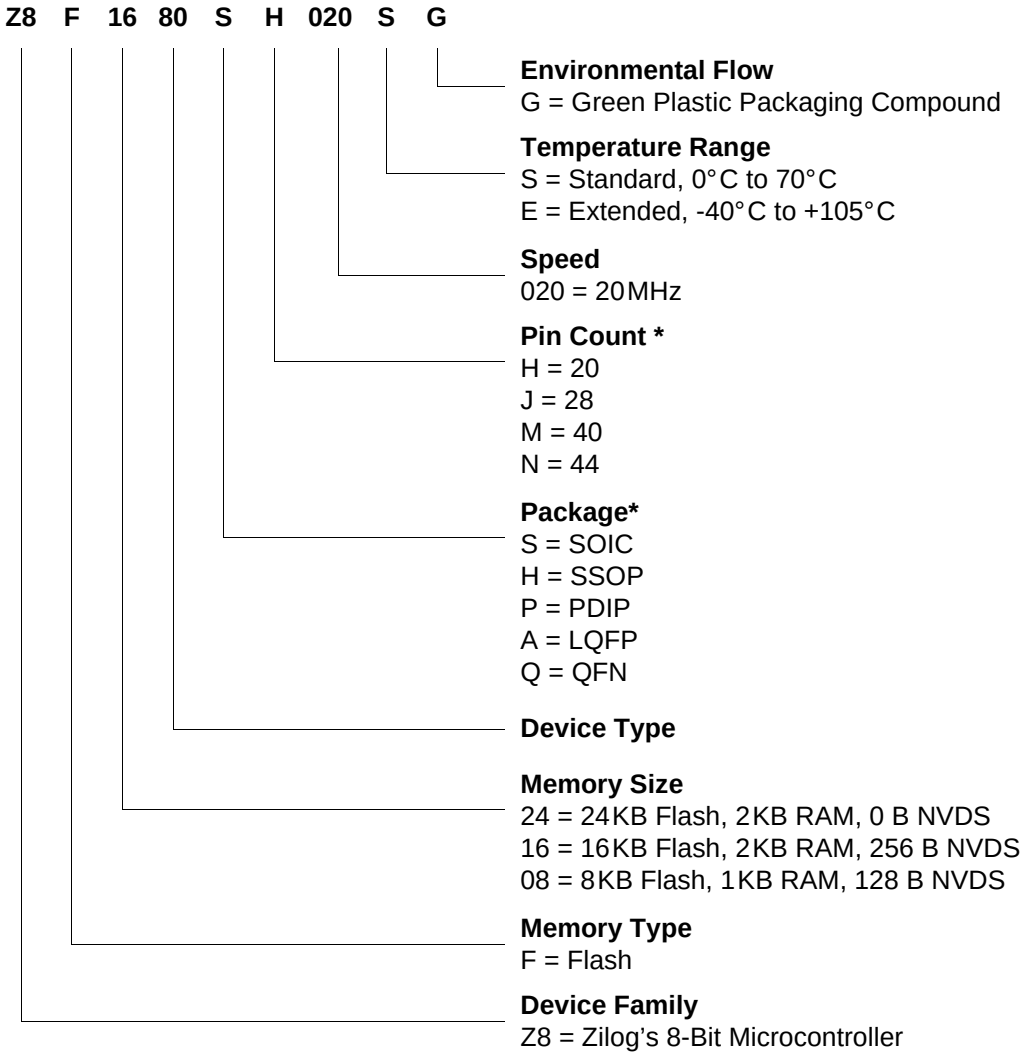
Table 208. UART Timing Without CTS

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
UART			
T_1	DE assertion to TXD falling edge (start bit) delay	1 * X_{IN} period	1 bit time
T_2	End of stop bit(s) to DE deassertion delay (Tx Data Register is empty)	± 5	

31.1. Part Number Suffix Designations

Zilog part numbers consist of a number of components, as indicated in the following example.

Example. Part number Z8F1680SH020SG is an 8-bit, 20 MHz Flash Motor Controller with 16 KB of Program memory in a 20-pin SOIC package, operating within a 0ºC to +70ºC temperature range and built using lead-free solder.



Index