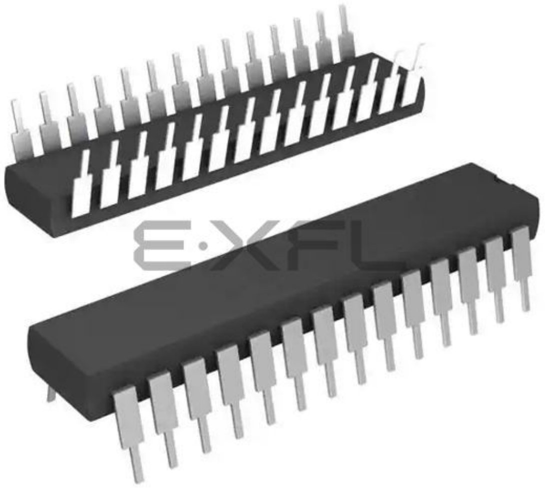
Zilog - Z8F1680PJ020EG Datasheet



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	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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f1680pj020eg

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 9. Reset and Stop Mode Recovery Characteristics and Latency

	Reset Characteristics and Latency		
Reset Type	**Control Registers**	**eZ8 CPU**	**Reset Latency (Delay)**
System Reset (non-POR Reset)	Reset (as applicable)	Reset	68 Internal Precision Oscillator Cycles after IPO starts up
System Reset (POR Reset)	Reset (as applicable)	Reset	68 Internal Precision Oscillator Cycles + 50ms Wait time
System Reset with Crystal Oscillator Enabled	Reset (as applicable)	Reset	568–10068 Internal Precision Oscillator Cycles after IPO starts up; see Table 141 on page 280 for a description of the EXTLTMG user option bit.
Stop Mode Recovery	Unaffected, except RSTSTAT and OSCCTL registers	Reset	4 Internal Precision Oscillator Cycles after IPO starts up

During a System Reset or Stop Mode Recovery, the Internal Precision Oscillator (IPO) requires 4µs to start up. When the reset type is a System Reset, the F1680 Series MCU is held in Reset for 68 IPO cycles. If the crystal oscillator is enabled in Flash option bits, the Reset period is increased to 568–10068 IPO cycles. For more details, see Table 141 on page 280 for a description of the EXTLTMG user option bit. When the reset type is a Stop Mode Recovery, the F1680 Series MCU goes to NORMAL Mode immediately after 4 IPO cycles. The total Stop Mode Recovey delay is less than 6µs. When a Reset occurs due to a VBO condition, this delay is measured from the time the supply voltage first exceeds the VBO level (discussed later in this chapter). When a Reset occurs due to a POR condition, this delay is measured from the time that the supply voltage first exceeds the POR level. If the external pin reset remains asserted at the end of the Reset period, the device remains in reset until the pin is deasserted.

Note: After a Stop Mode Recovery, the external crystal oscillator is unstable. Use software to wait until it is stable before you can use it as main clock.

At the beginning of Reset, all GPIO pins are configured as inputs with pull-up resistor disabled, except PD0 that is shared with the Reset pin. On Reset, the Port D0 pin is configured as a bidirectional open-drain Reset. The pin is internally driven Low during port reset, after which the user code can reconfigure this pin as a general-purpose output.

During Reset, the eZ8 CPU and on-chip peripherals are idle; however, the on-chip crystal oscillator and WDT oscillator continue to function.

Table 88. Mode Status Fields

MULTIPROCESSOR Mode Status Field	**NEWFRM** Status bit denoting the start of a new frame. Reading the LIN-UART Receive Data Register resets this bit to 0. 0 = The current byte is not the first data byte of a new frame. 1 = The current byte is the first data byte of a new frame.
	Multiprocessor Receive (MPRX) Returns the value of the last multiprocessor bit received. Reading from the LIN-UART Receive Data Register resets this bit to 0.
Digital Noise Filter Mode Status Field	**Noise Event (NE); MSEL = 001b** This bit is asserted if digital noise is detected on the receive data line when the data is sampled (center of bit-time). If this bit is set, it does not mean that the receive data is corrupted (though it can be in extreme cases), means that one or more of the noise filter data samples near the center of the bit-time did not match the average data value.
LIN Mode Status Field	**Noise Event (NE); MSEL = 010b** This bit is asserted if some noise level is detected on the receive data line when the data is sampled (center of bit-time). If this bit is set, it does not indicate that the receive data is corrupt (though it can be in extreme cases), means that one or more of the 16x data samples near the center of the bit-time did not match the average data value.
	RxBreakLength LIN mode received break length. This field can be read following a break (LIN Wake-up or Break) so that the software can determine the measured duration of the break. If the break exceeds 15 bit times the value saturates at 1111b.
Hardware Revision Mode Status Field	**Noise Event (NE); MSEL = 111b** This field indicates the hardware revision of the LIN-UART block. 00_xxx LIN UART hardware revision. 01_xxx Reserved. 10_xxx Reserved. 11_xxx Reserved.

A receive interrupt is generated by the RDRNE status bit when the ESPI block is enabled, the DIRQE bit is set and a character transfer completes. At the end of the character transfer, the contents of the Shift Register are transferred into the Data Register, causing the RDRNE bit to assert. The RDRNE bit is cleared when the Data Buffer is read as empty. If information is being transmitted but not received by the software application, the receive interrupt can be eliminated by selecting Transmit Only mode (ESPIEN1,0 = 10) in either MASTER or SLAVE modes. When information is being sent and received under interrupt control, RDRNE and TDRE will both assert simultaneously at the end of a character transfer. Since the new receive data is in the Data Register, the receive interrupt must be serviced before the transmit interrupt.

ESPI error interrupts occur if any of the TUND, COL, ABT and ROVR bits in the ESPI Status Register are set. These bits are cleared by writing a 1. If the ESPI is disabled (ESPIEN1, 0 = 00), an ESPI interrupt can be generated by a Baud Rate Generator time-out. This timer function must be enabled by setting the BRGCTL bit in the ESPICTL register. This timer interrupt does not set any of the bits of the ESPI Status Register.

16.3.7. ESPI Baud Rate Generator

In ESPI MASTER Mode, the Baud Rate Generator creates a lower frequency serial clock (SCK) for data transmission synchronization between the Master and the external Slave. The input to the Baud Rate Generator is the system clock. The ESPI Baud Rate High and Low Byte registers combine to form a 16-bit reload value, BRG[15:0], for the ESPI Baud Rate Generator. The ESPI baud rate is calculated using the following equation:

$$\text{SPI Baud Rate (bits § s)} = \frac{\text{System Clock Frequency (Hz)}}{2 \times \text{BRG[15:0]}}$$

Minimum baud rate is obtained by setting BRG[15:0] to `0000H` for a clock divisor value of (2 x 65536 = 131072).

When the ESPI is disabled, the Baud Rate Generator can function as a basic 16-bit timer with interrupt on time-out. Observe the following steps to configure the Baud Rate Generator as a timer with interrupt on time-out:

1. Disable the ESPI by clearing the ESPIEN1,0 bits in the ESPI Control Register.
2. Load the appropriate 16-bit count value into the ESPI Baud Rate High and Low Byte registers.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the BRGCTL bit in the ESPI Control Register to 1.

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

Bit	Description (Continued)
[6,0] ESPIEN1, ESPIEN0	**ESPI Enable and Direction Control** 00 = The ESPI block is disabled. BRG can be used as a general-purpose timer by setting BRGCTL = 1. 01 = Receive Only Mode. Use this setting in SLAVE Mode if software application is receiving data but not sending. TDRE will not assert. Transmitted data will be all 1s. Not valid in MASTER Mode since Master must source data to drive the transfer. 10 = Transmit Only Mode Use this setting in MASTER or SLAVE Mode when the software application is sending data but not receiving. RDRNE will not assert. 11 = Transmit/Receive Mode Use this setting if the software application is both sending and receiving information. Both TDRE and RDRNE will be active.
[5] BRGCTL	**Baud Rate Generator Control** The function of this bit depends upon ESPIEN1,0. When ESPIEN1,0 = 00, this bit allows enabling the BRG to provide periodic interrupts. **If the ESPI is disabled** 0 = The Baud Rate Generator timer function is disabled. Reading the Baud Rate High and Low registers returns the BRG reload value. 1 = The Baud Rate Generator timer function and time-out interrupt is enabled. Reading the Baud Rate High and Low registers returns the BRG Counter value. **If the ESPI is enabled** 0 = Reading the Baud Rate High and Low registers returns the BRG reload value. If MMEN = 1, the BRG is enabled to generate SCK. If MMEN = 0, the BRG is disabled. 1 = Reading the Baud Rate High and Low registers returns the BRG Counter value. If MMEN = 1, the BRG is enabled to generate SCK. If MMEN = 0 the BRG is enabled to provide a Slave SCK time-out. See the SLAVE Mode Abort error description on page 211. **Caution:** If reading the counter one byte at a time while the BRG is counting keep in mind that the values will not be in sync. Zilog recommends reading the counter using (2-byte) word reads.
[4] PHASE	**Phase Select** Sets the phase relationship of the data to the clock. For more information about operation of the PHASE bit, see the ESPI Clock Phase and Polarity Control section on page 201.
[3] CLKPOL	**Clock Polarity** 0 = SCK idles Low (0). 1 = SCK idles High (1).
[2] WOR	**Wire-OR (Open-Drain) Mode Enabled** 0 = ESPI signal pins not configured for open-drain. 1 = All four ESPI signal pins (SCK, $\overline{\text{SS}}$, MISO and MOSI) configured for open-drain function. This setting is typically used for multi-Master and/or Multi-Slave configurations.
[1] MMEN	**ESPI MASTER Mode Enable** This bit controls the data I/O pin selection and SCK direction. 0 = Data out on MISO, data in on MOSI (used in SPI SLAVE Mode), SCK is an input. 1 = Data out on MOSI, data in on MISO (used in SPI MASTER Mode), SCK is an output.

4. If this operation is a single-byte transfer, the software asserts the NAK bit of the I^{2}C Control Register so that after the first byte of data has been read by the I^{2}C controller, a Not Acknowledge instruction is sent to the I^{2}C slave.
5. The I^{2}C controller sends a start condition.
6. The I^{2}C controller sends the address and Read bit out via the SDA signal.
7. The I^{2}C slave acknowledges the address by pulling the SDA signal Low during the next High period of SCL.

   If the slave does not acknowledge the address byte, the I^{2}C controller sets the NCKI bit in the I^{2}C Status Register, sets the ACKV bit and clears the ACK bit in the I^{2}C State Register. The software responds to the Not Acknowledge interrupt by setting the stop bit and clearing the TXI bit. The I^{2}C controller flushes the Transmit Data Register, sends a stop condition on the bus and clears the stop and NCKI bits. The transaction is complete and the following steps can be ignored.
8. The I^{2}C controller shifts in the first byte of data from the I^{2}C slave on the SDA signal.
9. The I^{2}C controller asserts the receive interrupt.
10. The software responds by reading the I^{2}C Data Register. If the next data byte is to be the final byte, the software must set the NAK bit of the I^{2}C Control Register.
11. The I^{2}C controller sends a Not Acknowledge to the I^{2}C slave if the next byte is the final byte; otherwise, it sends an Acknowledge.
12. If there are more bytes to transfer, the I^{2}C controller returns to Step 7.
13. A NAK interrupt (NCKI bit in I2CISTAT) is generated by the I^{2}C controller.
14. The software responds by setting the stop bit of the I^{2}C Control Register.
15. A stop condition is sent to the I^{2}C slave.

17.2.5.7. Master Read Transaction with a 10-Bit Address

Figure 46 displays the read transaction format for a 10-bit addressed Slave.

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

Figure 46. Data Transfer Format—Master Read Transaction with a 10-Bit Address

The first 7 bits transmitted in the first byte are `11110XX`. The two `XX` bits are the two most-significant bits of the 10-bit address. The lowest bit of the first byte transferred is the write control bit.

Observe the following data transfer procedure for a Read operation to a 10-bit addressed slave:

S	Start
W	Write
A	Acknowledge
$\overline{A}$	Not Acknowledge
P	Stop

17.2.6.5. Slave Receive Transaction with 7-Bit Address

The data transfer format for writing data from a Master to a Slave in 7-bit address mode is displayed in Figure 47. The procedure that follows describes the I^{2}C Master/Slave Controller operating as a slave in 7-bit addressing mode and receiving data from the bus master.

S	Slave Address	W=0	A	Data	A	Data	A	Data	$A/\overline{A}$	P/S

Figure 47. Data Transfer Format—Slave Receive Transaction with 7-Bit Address

1. The software configures the controller for operation as a slave in 7-bit addressing mode, as follows:
   a. Initialize the MODE field in the I^{2}C Mode Register for either SLAVE ONLY mode or MASTER/SLAVE Mode with 7-bit addressing.
   b. Optionally set the GCE bit.
   c. Initialize the SLA[6:0] bits in the I^{2}C Slave Address Register.
   d. Set IEN = 1 in the I^{2}C Control Register. Set NAK = 0 in the I^{2}C Control Register.
2. The bus master initiates a transfer, sending the address byte. In SLAVE Mode, the I^{2}C controller recognizes its own address and detects that R/$\overline{W}$ bit = 0 (written from the master to the slave). The I^{2}C controller acknowledges indicating it is available to accept the transaction. The SAM bit in the I2CISTAT Register is set to 1, causing an interrupt. The RD bit in the I2CISTAT Register is cleared to 0, indicating a Write to the slave. The I^{2}C controller holds the SCL signal Low waiting for the software to load the first data byte.
3. The software responds to the interrupt by reading the I2CISTAT Register (which clears the SAM bit). After seeing the SAM bit to 1, the software checks the RD bit. Because RD = 0, no immediate action is required until the first byte of data is received. If software is only able to accept a single byte, it sets the NAK bit in the I2CCTL Register at this time.
4. The Master detects the Acknowledge and sends the byte of data.

Bit	Description
[3] BRKPC	**Break when PC == OCDCNTR** If this bit is set to 1, then the OCDCNTR Register is used as a hardware breakpoint. When the program counter matches the value in the OCDCNTR Register, DBGMODE is automatically set to 1. If this bit is set, the OCDCNTR Register does not count when the CPU is running. 0 = OCDCNTR is set up as a counter. 1 = OCDCNTR generates a hardware break when PC == OCDCNTR.
[2] BRKZRO	**Break when OCDCNTR == 0000H** If this bit is set, then the OCD automatically sets the DBGMODE bit when the OCDCNTR Register counts down to 0000H. If this bit is set, the OCDCNTR Register is not reset when the part exits DEBUG Mode. 0 = OCD does not generate BRK when OCDCNTR decrements to 0000H. 1 = OCD sets DBGMODE to 1 when OCDCNTR decrements to 0000H.
[1]	Reserved; must be 0.
[0] RST	**Reset** Setting this bit to 1 resets the device. The controller goes through a normal POR sequence with the exception that the On-Chip Debugger is not reset. This bit is automatically cleared to 0 when the reset finishes. 0 = No effect. 1 = Reset the device.

Table 186. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode		Op Code(s) (Hex)	Flags						Fetch Cycles	Instr. Cycles
		dst	src		C	Z	S	V	D	H		
LDX dst, src	dst ← src	r	ER	84	–	–	–	–	–	–	3	2
		Ir	ER	85							3	3
		R	IRR	86							3	4
		IR	IRR	87							3	5
		r	X(rr)	88							3	4
		X(rr)	r	89							3	4
		ER	r	94							3	2
		ER	Ir	95							3	3
		IRR	R	96							3	4
		IRR	IR	97							3	5
		ER	ER	E8							4	2
		ER	IM	E9							4	2
LEA dst, X(src)	dst ← src + X	r	X(r)	98	–	–	–	–	–	–	3	3
		rr	X(rr)	99							3	5
MULT dst	dst[15:0] ← dst[15:8] * dst[7:0]	RR		F4	–	–	–	–	–	–	2	8
NOP	No operation			0F	–	–	–	–	–	–	1	2
OR dst, src	dst ← dst OR src	r	r	42	–	*	*	0	–	–	2	3
		r	Ir	43							2	4
		R	R	44							3	3
		R	IR	45							3	4
		R	IM	46							3	3
		IR	IM	47							3	4
ORX dst, src	dst ← dst OR src	ER	ER	48	–	*	*	0	–	–	4	3
		ER	IM	49							4	3
POP dst	dst ← @SP SP ← SP + 1	R		50	–	–	–	–	–	–	2	2
		IR		51							2	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 200. IPO Electrical Characteristics (Continued)

		V_{DD} = 1.8 to 3.6V T_A = –40°C to +105°C			V_{DD} = 2.7 to 3.6V T_A = 0°C to +70°C				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
F_{IPO}	Output Frequency	10.6168	11.0592	11.5016	10.78272	11.0592	11.33568	MHz	±2.5% 2.7 to 3.6 V, 0–70°C; ±4% 1.8 to 2.7 V, 0–70°C ±4% 1.8 to 3.6 V, –40–105°C
	Divided-by-2 Output Frequency	5.3084	5.5296	5.7508	5.39136	5.5296	5.66784		
	Divided-by-4 Output Frequency	2.6542	2.7648	2.8754	2.69568	2.7648	2.83392		
	Divided-by-8 Output Frequency	1.3271	1.3824	1.4377	1.34784	1.3824	1.41696		
	Divided-by-16 Output Frequency	0.6636	0.6912	0.7188	0.67392	0.6912	0.70848		
	Divided-by-32 Output Frequency	0.3318	0.3456	0.3594	0.33696	0.3456	0.35424		
	Divided-by-128 Output Frequency	0.0829	0.0864	0.0899	0.08424	0.0864	0.08856		
	Divided-by-256 Output Frequency	0.0415	0.0432	0.0449	0.04212	0.0432	0.04428		
	Duty Cycle of Output	45		55	45		55	%	

Table 201. Low Voltage Detect Electrical Characteristics

		T_A = 0°C to +70°C T_A = –40°C to +105°C				
		V_{DD} = 1.8 to 3.6 V				
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
I_{DD}LVD	LVD Active Current	–	–	50	µA	
I_{DDQ}LVD	LVD Quiescent Current	–	5	–	nA	
V_{TH}	Detected Source Voltage	V_{TP} – 10%	V_{TP}[1]	V_{TP} + 10%	V	

29.4.4. UART Timing

Figure 78 and Table 207 provide timing information for the UART pins for situations in which CTS is used for flow control. The CTS to DE assertion delay (T1) assumes that the Transmit Data Register has been loaded with data prior to CTS assertion.

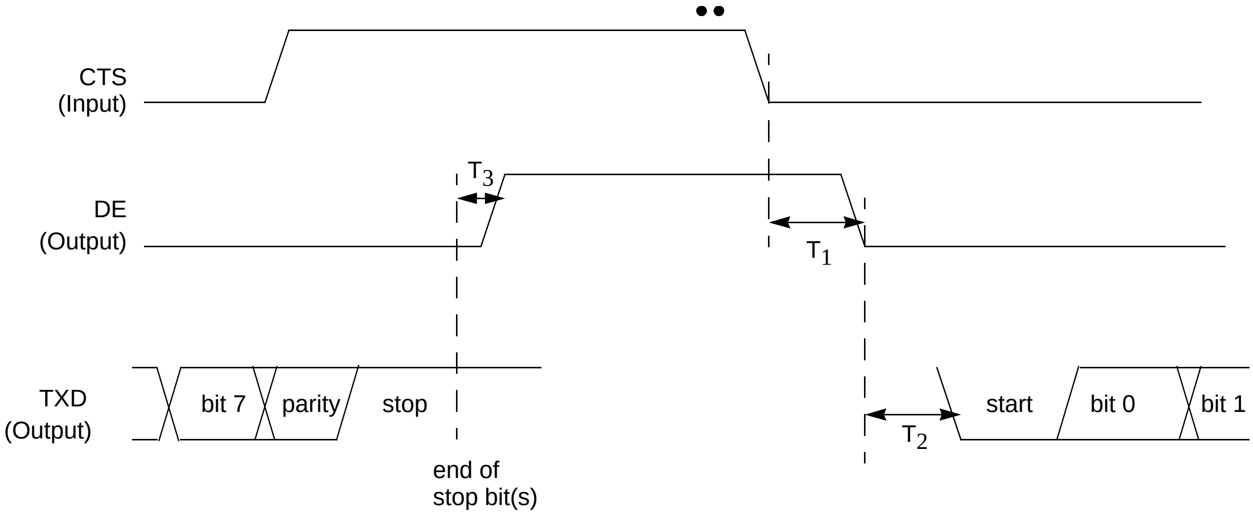


Figure 78. UART Timing With CTS

Table 207. UART Timing with CTS

		Delay (ns)	
Parameter	**Abbreviation**	**Min**	**Max**
UART			
T_1	CTS Fall to DE output delay	2 * X_{IN} period	2 * X_{IN} period + 1 bit time
T_2	DE assertion to TXD falling edge (start bit) delay	± 5	
T_3	End of stop bit(s) to DE deassertion delay	± 5	

Figure 79 and Table 208 provide timing information for the UART pins for situations in which CTS is not used for flow control. DE asserts after the Transmit Data Register has been written. DE remains asserted for multiple characters as long as the Transmit Data Register is written with the next character before the current character has completed.