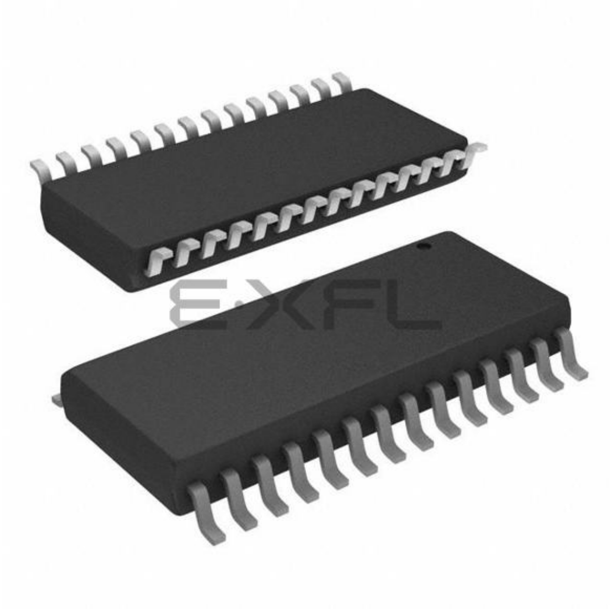
Zilog - Z8F0422SJ020SG Datasheet



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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	19
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f0422sj020sg

Table of Contents

Table 12. Port Alternate Function Mapping (Continued)

Port	Pin	Mnemonic	Alternate Function Description
Port C	PC0	T1IN	Timer 1 Input
	PC1	T1OUT	Timer 1 Output
	PC2	$\overline{\text{SS}}$	SPI Slave Select
	PC3	SCK	SPI Serial Clock
	PC4	MOSI	SPI Master Out Slave In
	PC5	MISO	SPI Master In Slave Out

GPIO Interrupts

Many of GPIO port pins are used as interrupt sources. Some port pins are configured to generate an interrupt request on either the rising edge or falling edge of the pin input signal. Other port pin interrupts generate an interrupt when any edge occurs (both rising and falling). For more details about interrupts using the GPIO pins, see Figure 8.

GPIO Control Register Definitions

Four registers for each port provide access to GPIO control, input data, and output data. Table 13 lists the GPIO port registers and subregisters. Use the Port A–C Address and Control registers together to provide access to subregisters for Port configuration and control.

Table 13. GPIO Port Registers and Subregisters

Port Register Mnemonic	Port Register Name
P*x*ADDR	Port A–C Address Register (selects subregisters)
P*x*CTL	Port A–C Control Register (provides access to subregisters)
P*x*IN	Port A–C Input Data Register
P*x*OUT	Port A–C Output Data Register
Port Subregister Mnemonic	**Port Register Name**
P*x*DD	Data Direction
P*x*AF	Alternate Function
P*x*OC	Output Control (Open-Drain)
P*x*HDE	High Drive Enable
P*x*SMRE	Stop Mode Recovery Source Enable
P*x*PUE	Pull-up Enable

IRQ0 Enable High and Low Bit Registers

Table 28 describes the priority control for IRQ0. The IRQ0 Enable High and Low Bit registers, shown in Tables 29 and 30, form a priority-encoded enabling for interrupts in the Interrupt Request 0 Register. Priority is generated by setting bits in each register.

Table 28. IRQ0 Enable and Priority Encoding

IRQ0ENH[*x*]	IRQ0ENL[*x*]	Priority	Description
0	0	Disabled	Disabled
0	1	Level 1	Low
1	0	Level 2	Nominal
1	1	Level 3	High
Note: *x* indicates register bits in the range [7:0].			

Table 29. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	7	6	5	4	3	2	1	0
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	I2CENH	SPIENH	ADCENH
RESET	0							
R/W	R/W							
Address	FC1H							

Bit	Description
[7]	**Reserved** This bit is reserved and must be programmed to 0.
[6] T1ENH	**Timer 1 Interrupt Request Enable High Bit**
[5] T0ENH	**Timer 0 Interrupt Request Enable High Bit**
[4] U0RENH	**UART 0 Receive Interrupt Request Enable High Bit**
[3] U0TENH	**UART 0 Transmit Interrupt Request Enable High Bit**
[2] I2CENH	**I^{2}C Interrupt Request Enable High Bit**
[1] SPIENH	**SPI Interrupt Request Enable High Bit**
[0] ADCENH	**ADC Interrupt Request Enable High Bit**

Table 36. IRQ2 Enable Low Bit Register (IRQ2ENL)

Bit	7	6	5	4	3	2	1	0
Field	Reserved				C3ENL	C2ENL	C1ENL	C0ENL
RESET	0							
R/W	R/W							
Address	FC8H							

Bit	Description
[7:4]	**Reserved** These bits are reserved and must be programmed to 0000.
[3] C3ENL	**Port C3 Interrupt Request Enable Low Bit**
[2] C2ENL	**Port C2 Interrupt Request Enable Low Bit**
[1] C1ENL	**Port C1 Interrupt Request Enable Low Bit**
[0] C0ENL	**Port C0 Interrupt Request Enable Low Bit**

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 37, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO Port input pin. The minimum pulse width must be greater than 1 system clock to guarantee capture of the edge triggered interrupt. Edge detection for pulses less than 1 system clock are not guaranteed.

Table 37. Interrupt Edge Select Register (IRQES)

Bit	7	6	5	4	3	2	1	0
Field	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
RESET	0							
R/W	R/W							
Address	FCDH							

Bit	Description
[7:0] IESx	**Interrupt Edge Select *x*** 0 = An interrupt request is generated on the falling edge of the PA*x* input. 1 = An interrupt request is generated on the rising edge of the PA*x* input.
Note: *x* indicates register bits in the range [7:0].	

value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the reload value stored in the Timer Reload High and Low Byte registers. Upon reaching the reload value, the timer generates an interrupt, the count value in the Timer High and Low Byte registers is reset to `0001H` and counting resumes.

If the TPOL bit in the Timer Control Register is set to 1, the Timer Output signal begins as a High (1) and then transitions to a Low (0) when the timer value matches the PWM value. The Timer Output signal returns to a High (1) after the timer reaches the reload value and is reset to `0001H`.

If the TPOL bit in the Timer Control Register is set to 0, the Timer Output signal begins as a Low (0) and then transitions to a High (1) when the timer value matches the PWM value. The Timer Output signal returns to a Low (0) after the timer reaches the reload value and is reset to `0001H`.

Observe the following procedure for configuring a timer for PWM Mode and initiating the PWM operation:

1. Write to the Timer Control Register to:
   - Disable the timer
   - Configure the timer for PWM Mode
   - 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 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 PWM High and Low Byte registers to set the PWM value.
4. 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.
5. If required, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
6. Configure the associated GPIO port pin for the Timer Output alternate function.
7. Write to the Timer Control Register to enable the timer and initiate counting.

The PWM period is calculated using the following equation.

$$\text{PWM Period (s)} = \frac{\text{Reload Value} \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

Table 39. Timer 0–1 High Byte Register (TxH)

Bit	7	6	5	4	3	2	1	0
Field	TH							
RESET	0							
R/W	R/W							
Address	F00H, F08H							

Table 40. Timer 0–1 Low Byte Register (TxL)

Bit	7	6	5	4	3	2	1	0
Field	TL							
RESET	0							1
R/W	R/W							
Address	F01H, F09H							

Bit	Description
[7:0] TH, TL	**Timer High and Low Bytes** These 2 bytes, {TMRH[7:0], TMRL[7:0]}, contain the current 16-bit timer count value.

Timer Reload High and Low Byte Registers

The Timer 0–1 Reload High and Low Byte (TxRH and TxRL) registers, shown in Tables 41 and 42, store a 16-bit reload value, {TRH[7:0], TRL[7:0]}. Values written to the Timer Reload High Byte Register are stored in a temporary holding register. When a write to the Timer Reload Low Byte Register occurs, the temporary holding register value is written to the Timer High Byte Register. This operation allows simultaneous updates of the 16-bit Timer reload value.

In COMPARE Mode, the Timer Reload High and Low Byte registers store the 16-bit Compare value.

Table 41. Timer 0–1 Reload High Byte Register (TxRH)

Bit	7	6	5	4	3	2	1	0
Field	TRH							
RESET	1							
R/W	R/W							
Address	F02H, F0AH							

Table 47. Watchdog Timer Approximate Time-Out Delays

WDT Reload Value (Hex)	WDT Reload Value (Decimal)	Approximate Time-Out Delay (with 10 kHz Typical WDT Oscillator Frequency)	
		Typical	Description
000004	4	400 µs	Minimum time-out delay
FFFFFF	16,777,215	1677.5 s	Maximum time-out delay

Watchdog Timer Refresh

When first enabled, the WDT is loaded with the value in the WDT Reload registers. The WDT then counts down to `000000H` unless a WDT instruction is executed by the eZ8 CPU. Execution of the WDT instruction causes the downcounter to be reloaded with the WDT reload value stored in the WDT Reload registers. Counting resumes following the reload operation.

When Z8 Encore! XP® F0822 Series device is operating in DEBUG Mode (using the OCD), the WDT is continuously refreshed to prevent spurious WDT time-outs.

Watchdog Timer Time-Out Response

The WDT times out when the counter reaches `000000H`. A WDT time-out generates either an Interrupt or a Reset. The WDT_RES option bit determines the time-out response of the WDT. For information regarding programming of the WDT_RES option bit, see the Option Bits chapter on page 155.

WDT Interrupt in Normal Operation

If configured to generate an interrupt when a time-out occurs, the WDT issues an interrupt request to the interrupt controller and sets the WDT status bit in the WDT Control Register. If interrupts are enabled, the eZ8 CPU responds to the interrupt request by fetching the WDT interrupt vector and executing the code from the vector address. After time-out and interrupt generation, the WDT counter rolls over to its maximum value of `FFFFFH` and continues counting. The WDT counter is not automatically returned to its reload value.

WDT Reset in STOP Mode

If enabled in STOP Mode and configured to generate a Reset when a time-out occurs and the device is in STOP Mode, the WDT initiates a Stop Mode Recovery. Both the WDT status bit and the stop bit in the WDT Control Register is set to 1 following the WDT time-out in STOP Mode. For more information, see the Reset and Stop Mode Recovery chapter on page 21. Default operation is for the WDT and its RC oscillator to be enabled during STOP Mode.

   - Set or clear the CTSE bit to enable or disable control from the remote receiver using the $\overline{\text{CTS}}$ pin.

5. Check the TDRE bit in the UART Status 0 Register to determine if the Transmit Data Register is empty (indicated by a 1). If empty, continue to Step 6. If the Transmit Data Register is full (indicated by a 0), continue to monitor the TDRE bit until the Transmit Data Register becomes available to receive new data.
6. Write the UART Control 1 Register to select the outgoing address bit:
   - Set the Multiprocessor Bit Transmitter (MPBT) if sending an address byte, clear it if sending a data byte.
7. Write data byte to the UART Transmit Data Register. The transmitter automatically transfers data to the Transmit Shift Register and then transmits the data.
8. If required, and multiprocessor mode is enabled, make any changes to the Multiprocessor Bit Transmitter (MPBT) value.
9. To transmit additional bytes, return to Step 5.

Transmitting Data Using Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data Register to accept new data for transmission. Follow the below steps to configure the UART for interrupt-driven data transmission:

1. Write to the UART Baud Rate High and Low Byte registers to set the required baud rate.
2. Enable the UART pin functions by configuring the associated GPIO port pins for alternate function operation.
3. Execute a DI instruction to disable interrupts.
4. Write to the Interrupt Control registers to enable the UART Transmitter interrupt and set the required priority.
5. If MULTIPROCESSOR Mode is required, write to the UART Control 1 Register to enable MULTIPROCESSOR (9-Bit) Mode functions:
   - Set the Multiprocessor Mode Select (MPEN) to enable MULTIPROCESSOR Mode.
6. Write to the UART Control 0 Register to:
   - Set the transmit enable (TEN) bit to enable the UART for data transmission
   - Enable parity, if required, and if MULTIPROCESSOR Mode is not enabled, and select either even or odd parity.

UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the BRG is the system clock. The UART Baud Rate High and Low Byte registers combine to create a 16-bit baud rate divisor value (BRG[15:0]) that sets the data transmission rate (baud rate) of the UART. The UART data rate is calculated using the following equation:

$$\text{UART Data Rate (bits/s)} = \frac{\text{System Clock Frequency (Hz)}}{16 \text{x} \text{UART Baud Rate Divisor Value}}$$

When the UART is disabled, the BRG functions as a basic 16-bit timer with interrupt upon time-out. Observe the following procedure to configure the BRG as a timer with interrupt upon time-out:

1. Disable the UART by clearing the REN and TEN bits in the UART Control 0 Register to 0.
2. Load the appropriate 16-bit count value into the UART Baud Rate High and Low Byte registers.
3. Enable the BRG timer function and associated interrupt by setting the BKGCTL bit in the UART Control 1 Register to 1.

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

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

UART Control Register Definitions

The UART Control registers support the UART and the associated Infrared Encoder/Decoders. See the Infrared Encoder/Decoder chapter on page 97 for more information about the infrared operation.

UART Transmit Data Register

Data bytes written to the UART Transmit Data Register, shown in Table 53, are shifted out on the TXD*x* pin. The write-only UART Transmit Data Register shares a Register File address with the read-only UART Receive Data Register.

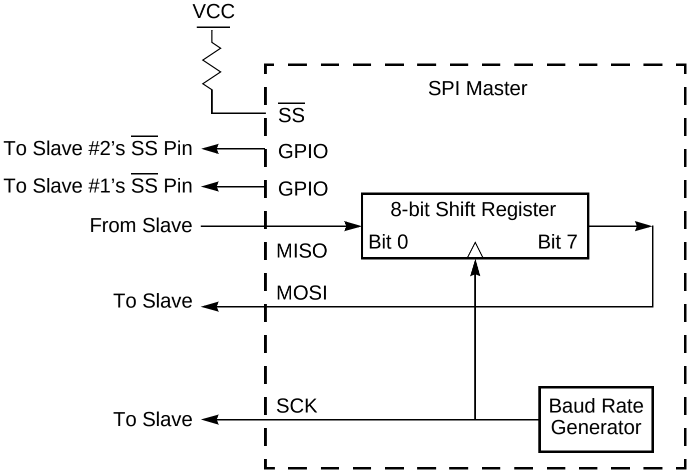


Figure 21. SPI Configured as a Master in a Single Master, Multiple Slave System

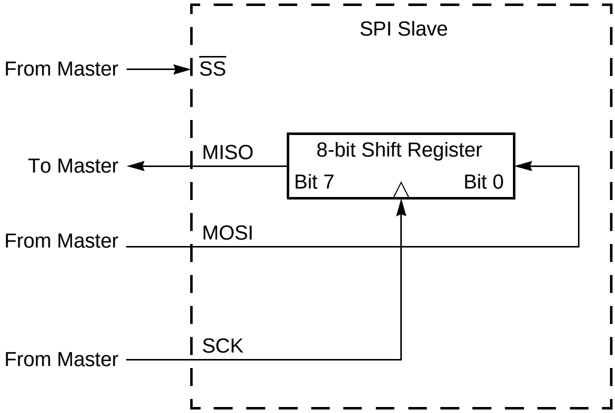


Figure 22. SPI Configured as a Slave

16. If the I^2C Slave sends an acknowledge by pulling the SDA signal Low during the next High period of SCL, the I^2C Controller sets the ACK bit in the I^2C Status Register. Continue to Step 17.

    If the slave does not acknowledge the second address byte or one of the data bytes, the I^2C Controller sets the NCKI bit and clears the ACK bit in the I^2C Status Register. Software responds to the Not Acknowledge interrupt by setting the stop and flush bits and clearing the TXI bit. The I^2C Controller sends the stop condition on the bus and clears the stop and NCKI bits. The transaction is complete; ignore the remainder of this sequence.

17. The I^2C Controller shifts the data out by the SDA signal. After the first bit is sent, the transmit interrupt is asserted.
18. If more bytes remain to be sent, return to Step 14.
19. If the last byte is currently being sent, software sets the stop bit of the I^2C Control Register (or start bit to initiate a new transaction). In the stop case, software simultaneously clears the TXI bit of the I^2C Control Register.
20. The I^2C Controller completes transmission of the last data byte on the SDA signal.
21. The slave can either Acknowledge or Not Acknowledge the last byte. Because either the stop or start bit is already set, the NCKI interrupt does not occur.
22. The I^2C Controller sends the stop (or restart) condition to the I^2C bus and clears the stop (or start) bit.

Read Transaction with a 7-Bit Address

Figure 30 displays the data transfer format for a read operation to a 7-bit addressed slave. The shaded regions indicate data transferred from the I^2C Controller to slaves and unshaded regions indicate data transferred from the slaves to the I^2C Controller.

S	Slave Address	R = 1	A	Data	A	Data	A	P/S

Figure 30. Receive Data Transfer Format for a 7-Bit Addressed Slave

Observe the following procedure for a read operation to a 7-bit addressed slave:

1. Software writes the I^2C Data Register with a 7-bit Slave address plus the read bit (=1).
2. Software asserts the start bit of the I^2C Control Register.
3. If this transfer is a single byte transfer, Software asserts the NAK bit of the I^2C Control Register so that after the first byte of data has been read by the I^2C Controller, a Not Acknowledge is sent to the I^2C Slave.
4. The I^2C Controller sends the start condition.

Bit	Description (Continued)
[5] ACK	**Acknowledge** This bit indicates the status of the Acknowledge for the last byte transmitted or received. When set, this bit indicates that an Acknowledge occurred for the last byte transmitted or received. This bit is cleared when IEN = 0 or when a Not Acknowledge occurred for the last byte transmitted or received. It is not reset at the beginning of each transaction and is not reset when this register is read. **Caution:** When making decisions based on this bit within a transaction, software cannot determine when the bit is updated by hardware. In the case of write transactions, the I^{2}C pauses at the beginning of the Acknowledge cycle if the next transmit data or address byte has not been written (TDRE = 1) and STOP and start = 0. In this case the ACK bit is not updated until the transmit interrupt is serviced and the Acknowledge cycle for the previous byte completes. For examples on usage of the ACK bit, see the Address Only Transaction with a 7-Bit Address section on page 120 and the Address-Only Transaction with a 10-Bit Address section on page 122.
[4] 10B	**10-Bit Address** This bit indicates whether a 10-bit or 7-bit address is being transmitted. After the start bit is set, if the five most-significant bits of the address are 11110B, this bit is set. When set, it is reset after the first byte of the address has been sent.
[3] RD	**Read** This bit indicates the direction of transfer of the data. It is active High during a read. The status of this bit is determined by the least-significant bit of the I^{2}C Shift Register after the start bit is set.
[2] TAS	**Transmit Address State** This bit is active High while the address is being shifted out of the I^{2}C Shift Register.
[1] DSS	**Data Shift State** This bit is active High while data is being shifted to or from the I^{2}C Shift Register.
[0] NCKI	**NACK Interrupt** This bit is set High when a Not Acknowledge condition is received or sent and neither the start nor the stop bit is active. When set, this bit generates an interrupt that can only be cleared by setting the start or stop bit, allowing you to specify whether you want to perform a stop or a repeated start.

Flash Memory

The products in the Z8 Encore! XP® F0822 Series feature either 8KB (8192) or 4KB (4096) bytes of Flash memory with Read/Write/Erase capability. Flash memory is programmed and erased in-circuit by either user code or through the OCD.

The Flash memory array is arranged in 512-byte per page. The 512-byte page is the minimum Flash block size that can be erased. Flash memory is divided into eight sectors which is protected from programming and erase operations on a per sector basis.

Table 81 describes the Flash memory configuration for each device in the Z8F082xfamily. Table 82 lists the sector address ranges. Figure 33 displays the Flash memory arrangement.

Table 81. Flash Memory Configurations

Part Number	Flash Size	Number of Pages	Flash Memory Addresses	Sector Size	Number of Sectors	Pages per Sector
Z8F08xx	8KB (8192)	16	0000H–1FFFH	1KB (1024)	8	2
Z8F04xx	4KB (4096)	8	0000H–0FFFH	0.5KB (512)	8	1

Table 82. Flash Memory Sector Addresses

	Flash Sector Address Ranges	
Sector Number	**Z8F04xx**	**Z8F08xx**
0	0000H–01FFH	0000H–03FFH
1	0200H–03FFH	0400H–07FFH
2	0400H–05FFH	0800H–0BFFH
3	0600H–07FFH	0C00H–0FFFH
4	0800H–09FFH	1000H–13FFH
5	0A00H–0BFFH	1400H–17FFH
6	0C00H–0DFFH	1800H–1BFFH
7	0E00H–0FFFH	1C00H–1FFFH

OCD Status Register

The OCD Status Register, shown in Table 95, reports status information about the current state of the debugger and the system.

Table 95. OCD Status Register (OCDSTAT)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	IDLE	HALT	RPEN	Reserved				
RESET	0							
R/W	R							

Bit	**Description**
[7] IDLE	**CPU Idling** This bit is set if the part is in DEBUG Mode (DBGMODE is 1), or if a BRK instruction occurred since the last time OCDCTL was written. This can be used to determine if the CPU is running or if it is idling. 0 = The eZ8 CPU is running. 1 = The eZ8 CPU is either stopped or looping on a BRK instruction.
[6] HALT	**HALT Mode** 0 = The device is not in HALT Mode. 1 = The device is in HALT Mode.
[5] RPEN	**Read Protect Option Bit Enabled** 0 = The Read Protect option bit is disabled (1). 1 = The Read Protect option bit is enabled (0), disabling many OCD commands.
[4:0]	**Reserved** These bits are reserved and must be programmed to 00000.

Table 97. Absolute Maximum Ratings (Continued)

Parameter	Minimum	Maximum	Units	Notes
20-pin SSOP Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		250	mW	
Maximum current into V_{DD} or out of V_{SS}		69	mA	
20-pin PDIP Package Maximum Ratings at –40°C to 70°C				
Total power dissipation		775	mW	
Maximum current into V_{DD} or out of V_{SS}		215	mA	
20-pin PDIP Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		285	mW	
Maximum current into V_{DD} or out of V_{SS}		79	mA	
28-pin SOIC Package Maximum Ratings at –40°C to 70°C				
Total power dissipation		450	mW	
Maximum current into V_{DD} or out of V_{SS}		125	mA	
28-pin SOIC Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		260	mW	
Maximum current into V_{DD} or out of V_{SS}		73	mA	
28-pin PDIP Package Maximum Ratings at –40°C to 70°C				
Total power dissipation		1100	mW	
Maximum current into V_{DD} or out of V_{SS}		305	mA	
28-pin PDIP Package Maximum Ratings at 70°C to 105°C				
Total power dissipation		400	mW	
Maximum current into V_{DD} or out of V_{SS}		110	mA	

Note: This voltage applies to all pins except the following: V_{DD}, AV_{DD}, V_{REF}, pins that support analog input (Port B), and where otherwise noted.

DC Characteristics

Table 98 lists the DC characteristics of the Z8 Encore! XP® F0822 Series products. All voltages are referenced to V_{SS}, the primary system ground.

Table 98. DC Characteristics

		T_A = –40°C to 105°C				
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Units**	**Conditions**
V_{DD}	Supply Voltage	2.7	–	3.6	V	
V_{IL1}	Low Level Input Voltage	-0.3	–	0.3^*V_{DD}	V	For all input pins except $\overline{\text{RESET}}$, DBG, and X_{IN}.
V_{IL2}	Low Level Input Voltage	–0.3	–	0.2^*V_{DD}	V	For $\overline{\text{RESET}}$, DBG, and X_{IN}.
V_{IH1}	High Level Input Voltage	0.7^*V_{DD}	–	5.5	V	Ports A and C pins when their programmable pull-ups are disabled.
V_{IH2}	High Level Input Voltage	0.7^*V_{DD}	–	V_{DD}+0.3	V	Port B pins. Ports A and C pins when their programmable pull-ups are enabled.
V_{IH3}	High Level Input Voltage	0.8^*V_{DD}	–	V_{DD}+0.3	V	$\overline{\text{RESET}}$, DBG, and X_{IN} pins.
V_{OL1}	Low Level Output Voltage	–	–	0.4	V	I_{OL} = 2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OH1}	High Level Output Voltage	2.4	–	–	V	I_{OH} = –2 mA; V_{DD} = 3.0V High Output Drive disabled.
V_{OL2}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 20 mA; V_{DD} = 3.3V High Output Drive enabled T_A = –40°C to +70°C
V_{OH2}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = –20 mA; V_{DD} = 3.3V High Output Drive enabled; T_A = –40°C to +70°C
V_{OL3}	Low Level Output Voltage High Drive	–	–	0.6	V	I_{OL} = 15mA; V_{DD} = 3.3V High Output Drive enabled; T_A = +70°C to +105°C
V_{OH3}	High Level Output Voltage High Drive	2.4	–	–	V	I_{OH} = 15 mA; V_{DD} = 3.3V High Output Drive enabled; T_A = +70°C to +105°C
V_{RAM}	RAM Data Retention	0.7	–	–	V	
I_{IL}	Input Leakage Current	–5	–	+5	µA	V_{DD} = 3.6V; V_{IN} = V_{DD} or V_{SS}[1]

General Purpose I/O Port Output Timing

Figure 49 and Table 107 provide timing information for GPIO port pins.

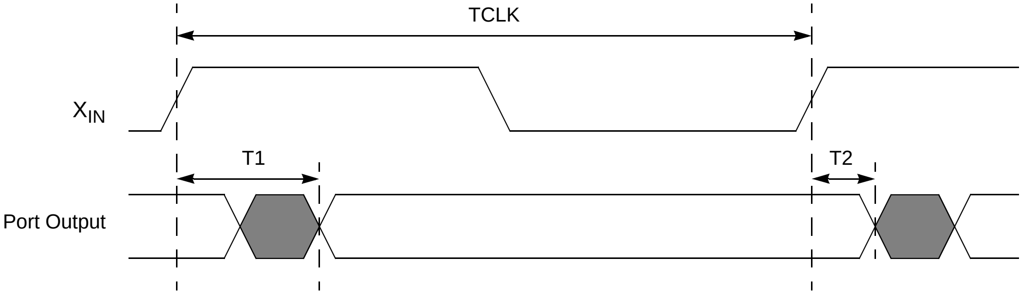


Figure 49. GPIO Port Output Timing

Table 107. GPIO Port Output Timing

		Delay (ns)	
Parameter	**Abbreviation**	**Minimum**	**Maximum**
GPIO Port Pins			
T_1	X_{IN} Rise to Port Output Valid Delay	–	15
T_2	X_{IN} Rise to Port Output Hold Time	2	–

Flags Register

The Flags Register contains the status information regarding the most recent arithmetic, logical, bit manipulation or rotate and shift operation. The Flags Register contains six bits of status information that are set or cleared by CPU operations. Four of the bits (C, V, Z and S) can be tested with conditional jump instructions. Two flags (H and D) cannot be tested and are used for Binary-Coded Decimal (BCD) arithmetic.

The two remaining bits, User Flags (F1 and F2), are available as general-purpose status bits. User Flags are unaffected by arithmetic operations and must be set or cleared by instructions. The User Flags cannot be used with conditional Jumps. They are undefined at initial power-up and are unaffected by Reset. Figure 56 illustrates the flags and their bit positions in the Flags Register.

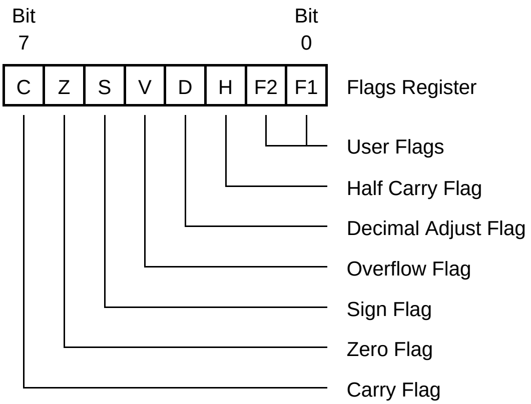


Figure 56. Flags Register

Interrupts, the Software Trap (TRAP) instruction, and Illegal Instruction Traps all write the value of the Flags Register to the stack. Executing an Interrupt Return (IRET) instruction restores the value saved on the stack into the Flags Register.

Hex Address: FC1

Table 173. IRQ0 Enable High Bit Register (IRQ0ENH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1ENH	T0ENH	U0RENH	U0TENH	I2CENH	SPIENH	ADCENH
RESET	0							
R/W	R/W							
Address	FC1H							

Hex Address: FC2

Table 174. IRQ0 Enable Low Bit Register (IRQ0ENL)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	Reserved	T1ENL	T0ENL	U0RENL	U0TENL	I2CENL	SPIENL	ADCENL
RESET	0							
R/W	R/W							
Address	FC2H							

Hex Address: FC3

Table 175. Interrupt Request 1 Register (IRQ1)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
RESET	0							
R/W	R/W							
Address	FC3H							

Hex Address: FC4

Table 176. IRQ1 Enable High Bit Register (IRQ1ENH)

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	PA7ENH	PA6ENH	PA5ENH	PA4ENH	PA3ENH	PA2ENH	PA1ENH	PA0ENH
RESET	0							
R/W	R/W							
Address	FC4H							