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, DMA, POR, PWM, WDT
Number of I/O	46
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f6422ar020eg

Introduction

Zilog's Z8 Encore! XP F64xx Series MCU family of products are a line of Zilog microcontroller products based upon the 8-bit eZ8 CPU. The Z8 Encore! XP F64xx Series adds Flash memory to Zilog's extensive line of 8-bit microcontrollers. The Flash in-circuit programming capability allows for faster development time and program changes in the field. The new eZ8 CPU is upward-compatible with existing Z8 instructions. The rich-peripheral set of the Z8 Encore! XP F64xx Series makes it suitable for a variety of applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Features

The features of Z8 Encore! XP F64xx Series include:

- 20MHz eZ8 CPU
- Up to 64KB Flash with in-circuit programming capability
- Up to 4KB register RAM
- 12-channel, 10-bit Analog-to-Digital Converter (ADC)
- Two full-duplex 9-bit UARTs with bus transceiver Driver Enable control
- Inter-integrated circuit (I^2C)
- Serial Peripheral Interface (SPI)
- Two Infrared Data Association (IrDA)-compliant infrared encoder/decoders
- Up to four 16-bit timers with capture, compare and PWM capability
- Watchdog Timer (WDT) with internal RC oscillator
- Three-channel DMA
- Up to 60 input/output (I/O) pins
- 24 interrupts with configurable priority
- On-Chip Debugger
- Voltage Brown-Out (VBO) Protection
- Power-On Reset (POR)
- Operating voltage of 3.0V to 3.6V with 5V-tolerant inputs
- 0°C to +70°C, –40°C to +105°C, and –40°C to +125°C operating temperature ranges

DMA Controller

The Z8 Encore! XP F64xx Series feature three channels of DMA. Two of the channels are for register RAM to and from I/O operations. The third channel automatically controls the transfer of data from the ADC to the memory.

Interrupt Edge Select Register

The Interrupt Edge Select (IRQES) Register, shown in Table 36, determines whether an interrupt is generated for the rising edge or falling edge on the selected GPIO port input pin. The Interrupt Port Select Register selects between Port A and Port D for the individual interrupts.

Table 36. 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*** The minimum pulse width should be greater than 1 system clock to guarantee capture of the edge triggered interrupt. Shorter pulses may be captured but not guaranteed. 0 = An interrupt request is generated on the falling edge of the PA*x*/PD*x* input. 1 = An interrupt request is generated on the rising edge of the PA*x*/PD*x* input.

Note: x indicates specific GPIO port pins in the range [7:0].

Interrupt Port Select Register

The Port Select (IRQPS) Register, shown in Table 37, determines the port pin that generates the PAx/PDx interrupts. This register allows either Port A or Port D pins to be used as interrupts. The Interrupt Edge Select Register controls the active interrupt edge.

Table 37. Interrupt Port Select Register (IRQPS)

Bit	7	6	5	4	3	2	1	0
Field	PAD7S	PAD6S	PAD5S	PAD4S	PAD3S	PAD2S	PAD1S	PAD0S
RESET	0							
R/W	R/W							
Address	FCEH							

Bit	Description
[7:0] PADxS	**PA*x*/PD*x* Selection** 0 = PA*x* is used for the interrupt for PA*x*/PD*x* interrupt request. 1 = PD*x* is used for the interrupt for PA*x*/PD*x* interrupt request.

Note: x indicates specific GPIO port pins in the range [7:0].

$$\text{COMPARE Mode Time (s)} = \frac{(\text{Compare Value} - \text{Start Value}) \times \text{Prescale}}{\text{System Clock Frequency (Hz)}}$$

GATED Mode

In GATED Mode, the timer counts only when the timer input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control 1 Register. When the timer input signal is asserted, counting begins. A timer interrupt is generated when the timer input signal is deasserted or a timer reload occurs. To determine if a timer input signal deassertion generated the interrupt, read the associated GPIO input value and compare to the value stored in the TPOL bit.

The timer counts up to the 16-bit reload value stored in the Timer Reload High and Low Byte registers. The timer input is the system clock. When 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 (assuming the timer input signal is still asserted). Also, if the timer output alternate function is enabled, the timer output pin changes state (from Low to High or from High to Low) at timer reset.

Observe the following procedure for configuring a timer for GATED Mode and initiating the count:

1. Write to the Timer Control 1 Register to:
   - Disable the timer
   - Configure the timer for GATED Mode
   - Set the prescale value
2. Write to the Timer High and Low Byte registers to set the starting count value. This only affects the first pass in GATED Mode. After the first timer reset in GATED Mode, counting always begins at the reset value of 0001H.
3. Write to the Timer Reload High and Low Byte registers to set the reload value.
4. If appropriate, enable the timer interrupt and set the timer interrupt priority by writing to the relevant interrupt registers.
5. Configure the associated GPIO port pin for the timer input alternate function.
6. Write to the Timer Control 1 Register to enable the timer.
7. Assert the timer input signal to initiate the counting.

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 input is the system clock.

ister. This operation allows accurate reads of the full 16-bit timer count value while enabled. When the timers are not enabled, a read from the Timer Low Byte Register returns the actual value in the counter.

Timer Output Signal Operation

A timer output is a GPIO port pin alternate function. Generally, the timer output is toggled every time the counter is reloaded.

Timer Control Register Definitions

This section defines the features of the following Timer Control registers.

Timer 0–3 High and Low Byte Registers: see page 72

Timer Reload High and Low Byte Registers: see page 74

Timer 0–3 PWM High and Low Byte Registers: see page 75

Timer 0–3 Control 0 Registers: see page 76

Timer 0–3 Control 1 Registers: see page 77

Timers 0–2 are available in all packages. Timer 3 is only available in 64-, 68- and 80-pin packages.

Timer 0–3 High and Low Byte Registers

The Timer 0–3 High and Low Byte (TxH and TxL) registers, shown in Tables 39 and 40, contain the current 16-bit timer count value. When the timer is enabled, a read from TxH causes the value in TxL to be stored in a temporary holding register. A read from TMRL always returns this temporary register when the timers are enabled. When the timer is disabled, reads from the TMRL read the register directly.

Writing to the Timer High and Low Byte registers while the timer is enabled is not recommended. There are no temporary holding registers available for write operations, so simultaneous 16-bit writes are not possible. If either the Timer High or Low Byte registers are written during counting, the 8-bit written value is placed in the counter (High or Low Byte) at the next clock edge. The counter continues counting from the new value.

Timer 3 is unavailable in 44-pin packages.

since the previous pulse was detected). This gives the endec a sampling window of minus four baud rate clocks to plus eight baud rate clocks around the expected time of an incoming pulse. If an incoming pulse is detected inside this window this process is repeated. If the incoming data is a logical 1 (no pulse), the endec returns to the initial state and waits for the next falling edge. As each falling edge is detected, the endec clock counter is reset, resynchronizing the endec to the incoming signal. This action allows the endec to tolerate jitter and baud rate errors in the incoming data stream. Resynchronizing the endec does not alter the operation of the UART, which ultimately receives the data. The UART is only synchronized to the incoming data stream when a start bit is received.

Infrared Encoder/Decoder Control Register Definitions

All infrared endec configuration and status information is set by the UART control registers as defined in the UART Control Register Definitions section on page 98.

Caution: To prevent spurious signals during IrDA data transmission, set the IREN bit in the UARTx Control 1 Register to 1 to enable the Infrared Encoder/Decoder before enabling the GPIO Port alternate function for the corresponding pin.

Serial Peripheral Interface

The Serial Peripheral Interface is a synchronous interface allowing several SPI-type devices to be interconnected. SPI-compatible devices include EEPROMs, Analog-to-Digital Converters, and ISDN devices. Features of the SPI include:

- Full-duplex, synchronous, character-oriented communication
- Four-wire interface
- Data transfers rates up to a maximum of one-half the system clock frequency
- Error detection
- Dedicated Baud Rate Generator

Architecture

The SPI may be configured as either a Master (in single or multimaster systems) or a Slave as displayed in Figures 22 through 24.

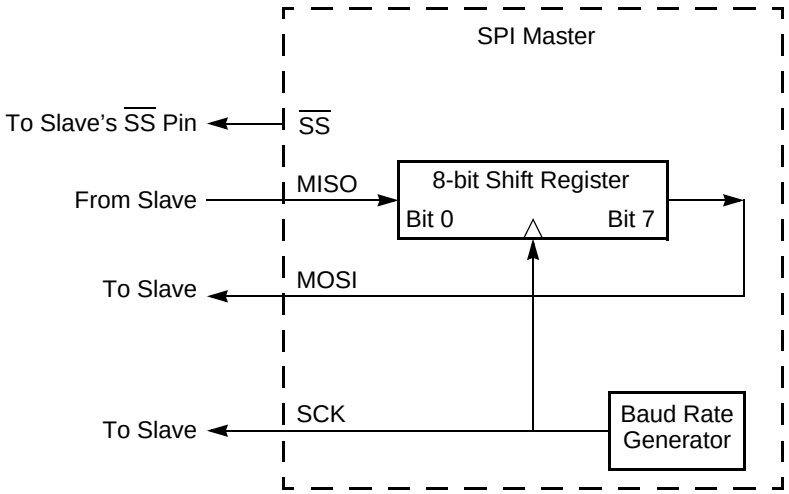

Figure 22. SPI Configured as a Master in a Single-Master, Single-Slave System

Transfer Format PHASE Equals One

Figure 26 displays the timing diagram for an SPI transfer in which PHASE is 1. Two waveforms are depicted for SCK, one for CLKPOL reset to 0 and another for CLKPOL set to 1.

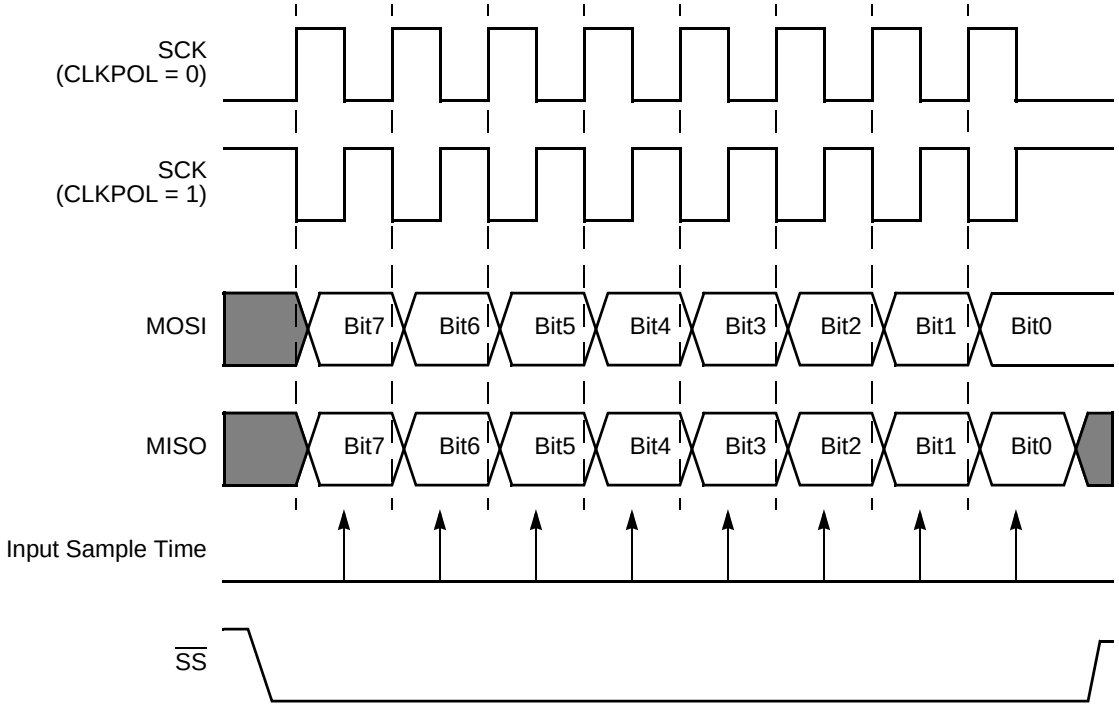

Figure 26. SPI Timing When PHASE is 1

Multimaster Operation

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

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

SDA and SCL Signals

I²C sends all addresses, data and acknowledge signals over the SDA line, most significant bit first. SCL is the common clock for the I²C Controller. When the SDA and SCL pin alternate functions are selected for their respective GPIO ports, the pins are automatically configured for open-drain operation.

The master (I²C) is responsible for driving the SCL clock signal, although the clock signal can become skewed by a slow slave device. During the low period of the clock, the slave pulls the SCL signal Low to suspend the transaction. The master releases the clock at the end of the low period and notices that the clock remains low instead of returning to a High level. When the slave releases the clock, the I²C Controller continues the transaction. All data is transferred in bytes and there is no limit to the amount of data transferred in one operation. When transmitting data or acknowledging read data from the slave, the SDA signal changes in the middle of the low period of SCL and is sampled in the middle of the High period of SCL.

I²C Interrupts

The I²C Controller contains four sources of interrupts—Transmit, Receive, Not Acknowledge and baud rate generator. These four interrupt sources are combined into a single interrupt request signal to the Interrupt Controller. The transmit interrupt is enabled by the IEN and TXI bits of the Control Register. The Receive and Not Acknowledge interrupts are enabled by the IEN bit of the Control Register. The baud rate generator interrupt is enabled by the BIRQ and IEN bits of the Control Register.

Not Acknowledge interrupts occur when a Not Acknowledge condition is received from the slave or sent by the I²C Controller and neither the start or stop bit is set. The Not Acknowledge event sets the NCKI bit of the I²C Status Register and can only be cleared by setting the start or stop bit in the I²C Control Register. When this interrupt occurs, the I²C Controller waits until either the stop or start bit is set before performing any action. In an interrupt service routine, the NCKI bit should always be checked prior to servicing transmit or receive interrupt conditions because it indicates the transaction is being terminated.

Receive interrupts occur when a byte of data has been received by the I²C Controller (master reading data from slave). This procedure sets the RDRF bit of the I²C Status Register. The RDRF bit is cleared by reading the I²C Data Register. The RDRF bit is set during the acknowledge phase. The I²C Controller pauses after the acknowledge phase until the receive interrupt is cleared before performing any other action.

Transmit interrupts occur when the TDRE bit of the I²C Status Register sets and the TXI bit in the I²C Control Register is set. transmit interrupts occur under the following conditions when the transmit data register is empty:

- The I²C Controller is enabled

If the slave were to Not Acknowledge at this point (this should not happen because the slave did acknowledge the first two address bytes), software would respond by setting the stop and flush bits and clearing the TXI bit. The I2C Controller sends the stop condition on the bus and clears the stop and NCKI bits. The transaction is complete (ignore the following steps).

19. The I2C Controller shifts in a byte of data from the I2C slave on the SDA signal. The I2C Controller sends a Not Acknowledge to the I2C slave if the NAK bit is set (last byte), else it sends an Acknowledge.
20. The I2C Controller asserts the receive interrupt (RDRF bit set in the Status Register).
21. Software responds by reading the I2C Data Register which clears the RDRF bit. If there is only one more byte to receive, set the NAK bit of the I2C Control Register.
22. If there are one or more bytes to transfer, return to Step 19.
23. After the last byte is shifted in, a Not Acknowledge interrupt is generated by the I2C Controller.
24. Software responds by setting the stop bit of the I2C Control Register.
25. A stop condition is sent to the I2C slave and the stop and NCKI bits are cleared.

I2C Control Register Definitions

This section defines the features of the following I2C Control registers.

I2C Data Register: see page 141

I2C Status Register: see page 142

I2C Control Register: see page 144

I2C Baud Rate High and Low Byte Registers: see page 145

I2C Diagnostic State Register: see page 147

I2C Diagnostic Control Register: see page 149

I2C Data Register

The I2C Data Register, shown in Table 71, holds the data that is to be loaded into the I2C Shift Register during a write to a slave. This register also holds data that is loaded from the I2C Shift Register during a read from a slave. The I2C Shift Register is not accessible in the Register File address space, but is used only to buffer incoming and outgoing data.

Table 74. I²C Baud Rate High Byte Register (I2CBRH)

Bit	7	6	5	4	3	2	1	0
Field	BRH							
RESET	FFH							
R/W	R/W							
Address	F53H							

Bit	Description
[7:0] BRH	**I²C Baud Rate High Byte** Most significant byte, BRG[15:8], of the I²C Baud Rate Generator's reload value. **Note:** If the DIAG bit in the I²C Diagnostic Control Register is set to 1, a read of the I2CBRH Register returns the current value of the I²C Baud Rate Counter[15:8].

Table 75. I²C Baud Rate Low Byte Register (I2CBRL)

Bit	7	6	5	4	3	2	1	0
Field	BRL							
RESET	FFH							
R/W	R/W							
Address	F54H							

Bit	Description
[7:0] BRL	**I²C Baud Rate Low Byte** Least significant byte, BRG[7:0], of the I²C Baud Rate Generator's reload value. **Note:** If the DIAG bit in the I²C Diagnostic Control Register is set to 1, a read of the I2CBRL Register returns the current value of the I²C Baud Rate Counter[7:0].

Bit	Description (Continued)
[3] WSEL	**Word Select** 0 = DMA*x* transfers a single byte per request. 1 = DMA*x* transfers a two-byte word per request. The address for the on-chip peripheral control register must be an even address.
[2:0] RSS	**Request Trigger Source Select** The Request Trigger Source Select field determines the peripheral that can initiate a DMA transfer. The corresponding interrupts do not need to be enabled within the Interrupt Controller to initiate a DMA transfer. However, if the Request Trigger Source can enable or disable the interrupt request sent to the Interrupt Controller, the interrupt request must be enabled within the Request Trigger Source block. 000 = Timer 0. 001 = Timer 1. 010 = Timer 2. 011 = Timer 3. 100 = DMA0 Control Register: UART0 Received Data Register contains valid data. DMA1 Control Register: UART0 Transmit Data Register empty. 101 = DMA0 Control Register: UART1 Received Data Register contains valid data. DMA1 Control Register: UART1 Transmit Data Register empty. 110 = DMA0 Control Register: I^2C Receiver Interrupt. DMA1 Control Register: I^2C Transmitter Interrupt Register empty. 111 = Reserved.

DMA*x* I/O Address Register

The DMA*x* I/O Address Register, shown in Table 79, contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is provided by {FH, DMA*x*_IO[7:0]}. When the DMA is configured for two-byte word transfers, the DMA*x* I/O Address Register must contain an even-numbered address.

Table 79. DMA*x* I/O Address Register (DMA*x*IO)

Bit	7	6	5	4	3	2	1	0
Field	DMA_IO							
RESET	X							
R/W	R/W							
Address	FB1H, FB9H							

Bit	Description
[7:0] DMA_IO	**DMA On-Chip Peripheral Control Register Address** This byte sets the low byte of the on-chip peripheral control register address on Register File Page FH (addresses F00H to FFFH).

AC Characteristics

This section provides AC characteristics and timing data which assumes a standard load of 50 pF on all outputs. Table 114 lists the Z8 Encore! XP F64xx Series AC characteristics and timing.

Table 114. AC Characteristics

		V_{DD} = 3.0 V–3.6 V T_A = –40°C to 125°C			
Symbol	**Parameter**	**Minimum**	**Maximum**	**Units**	**Conditions**
F_{SYSCLK}	System Clock Frequency	–	20.0	MHz	Read-only from Flash memory.
		0.032768	20.0	MHz	Program or erasure of Flash memory.
F_{XTAL}	Crystal Oscillator Frequency	0.032768	20.0	MHz	System clock frequencies below the crystal oscillator minimum require an external clock driver.
T_{XIN}	Crystal Oscillator Clock Period	50	–	ns	$T_{CLK} = 1/F_{SYSCLK}$
T_{XINH}	System Clock High Time	20		ns	
T_{XINL}	System Clock Low Time	20		ns	
T_{XINR}	System Clock Rise Time	–	3	ns	T_{CLK} = 50 ns. Slower rise times can be tolerated with longer clock periods.
T_{XINF}	System Clock Fall Time	–	3	ns	T_{CLK} = 50 ns. Slower fall times can be tolerated with longer clock periods.

Table 136. eZ8 CPU Instruction Summary (Continued)

Assembly Mnemonic	Symbolic Operation	Address Mode dst	Address Mode src	Opcode(s) (Hex)	Flags C	Z	S	V	D	H	Fetch Cycles	Instr. Cycles
TM dst, src	dst AND src	r	r	72	–	*	*	0	–	–	2	3
		r	Ir	73							2	4
		R	R	74							3	3
		R	IR	75							3	4
		R	IM	76							3	3
		IR	IM	77							3	4
TMX dst, src	dst AND src	ER	ER	78	–	*	*	0	–	–	4	3
		ER	IM	79							4	3
TRAP Vector	SP ← SP – 2 @SP ← PC SP ← SP – 1 @SP ← FLAGS PC ← @Vector		Vector	F2	–	–	–	–	–	–	2	6
WDT				5F	–	–	–	–	–	–	1	2
XOR dst, src	dst ← dst XOR src	r	r	B2	–	*	*	0	–	–	2	3
		r	Ir	B3							2	4
		R	R	B4							3	3
		R	IR	B5							3	4
		R	IM	B6							3	3
		IR	IM	B7							3	4
XORX dst, src	dst ← dst XOR src	ER	ER	B8	–	*	*	0	–	–	4	3
		ER	IM	B9							4	3

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

Hex Address: F47

Table 178. UART Baud Rate Low Byte Register (U*x*BRL)

Bit7	7	6	5	4	3	2	1	0
Field	BRL							
RESET	1							
R/W	R/W							
Address	F47H and F4FH							

Hex Address: F48

Table 179. UART Transmit Data Register (U*x*TXD)

Bit	7	6	5	4	3	2	1	0
Field	TXD							
RESET	X							
R/W	W							
Address	F40H and F48H							

Table 180. UART Receive Data Register (U*x*RXD)

Bit	7	6	5	4	3	2	1	0
Field	RXD							
RESET	X							
R/W	R							
Address	F40H and F48H							

Hex Address: F49

Table 181. UART Status 0 Register (U*x*STAT0)

Bit	7	6	5	4	3	2	1	0
Field	RDA	PE	OE	FE	BRKD	TDRE	TXE	CTS
RESET	0						1	X
R/W	R							
Address	F41H and F49H							

Hex Address: FBA

Table 211. DMA*x* Address High Nibble Register (DMA*x*H)

Bit	7	6	5	4	3	2	1	0
Field	DMA_END_H				DMA_START_H			
RESET	X							
R/W	R/W							
Address	FB2H, FBAH							

Hex Address: FBB

Table 212. DMA*x* Start/Current Address Low Byte Register (DMA*x*START)

Bit	7	6	5	4	3	2	1	0
Field	DMA_START							
RESET	X							
R/W	R/W							
Address	FB3H, FBBH							

Hex Address: FBC

Table 213. DMA*x* End Address Low Byte Register (DMA*x*END)

Bit	7	6	5	4	3	2	1	0
Field	DMA_END							
RESET	X							
R/W	R/W							
Address	FB4H, FBCH							

Hex Address: FBD

Table 214. DMA_ADC Address Register (DMAA_ADDR)

Bit	7	6	5	4	3	2	1	0
Field	DMAA_ADDR							Reserved
RESET	X							
R/W	R/W							
Address	FBDH							

Table 271. Z8 Encore! XP F64xx Series Ordering Matrix

Part Number	Flash	RAM	I/O Lines	Interrupts	16-Bit Timers w/PWM	10-Bit A/D Channels	I^2C	SPI	UARTs with IrDA	Description
Z8F482x with 48KB Flash, 10-Bit Analog-to-Digital Converter										
Standard Temperature: 0°C to 70°C										
Z8F4821PM020SG	48KB	4KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F4821AN020SG	48KB	4KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F4821VN020SG	48KB	4KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F4822AR020SG	48KB	4KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F4822VS020SG	48KB	4KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F4823FT020SG	48KB	4KB	60	24	4	12	1	1	2	QFP 80-pin package
Extended Temperature: –40°C to +105°C										
Z8F4821PM020EG	48KB	4KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F4821AN020EG	48KB	4KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F4821VN020EG	48KB	4KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F4822AR020EG	48KB	4KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F4822VS020EG	48KB	4KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F4823FT020EG	48KB	4KB	60	24	4	12	1	1	2	QFP 80-pin package
Automotive/Industrial Temperature: –40°C to +125°C										
Z8F4821PM020AG	48KB	4KB	29	23	3	8	1	1	2	PDIP 40-pin package
Z8F4821AN020AG	48KB	4KB	31	23	3	8	1	1	2	LQFP 44-pin package
Z8F4821VN020AG	48KB	4KB	31	23	3	8	1	1	2	PLCC 44-pin package
Z8F4822AR020AG	48KB	4KB	46	24	4	12	1	1	2	LQFP 64-pin package
Z8F4822VS020AG	48KB	4KB	46	24	4	12	1	1	2	PLCC 68-pin package
Z8F4823FT020AG	48KB	4KB	60	24	4	12	1	1	2	QFP 80-pin package

Index

Numerics

A

B

C

S