



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	eZ8
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	31
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z8f4801vn020ec

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



List of Tables

Table 1.	Z8F640x Family Part Selection Guide2
Table 2.	Z8F640x Family Package Options
Table 3.	Signal Descriptions
Table 4.	Pin Characteristics of the Z8F640x family 15
Table 5.	Z8F640x Family Program Memory Maps 18
Table 6.	Z8F640x Family Data Memory Maps 19
Table 7.	Register File Address Map 20
Table 8.	Reset and STOP Mode Recovery Characteristics
	and Latency
Table 9.	Reset Sources and Resulting Reset Type
Table 10.	STOP Mode Recovery Sources and Resulting Action $\dots 29$
Table 11.	Port Availability by Device and Package Type
Table 12.	Port Alternate Function Mapping
Table 13.	Port A-H GPIO Address Registers (PxADDR) 37
Table 14.	GPIO Port Registers and Sub-Registers
Table 15.	Port A-H Control Registers (PxCTL)
Table 16.	Port A-H Data Direction Sub-Registers 39
Table 17.	Port A-H Alternate Function Sub-Registers
Table 18.	Port A-H Output Control Sub-Registers
Table 19.	Port A-H High Drive Enable Sub-Registers
Table 20.	Port A-H Input Data Registers (PxIN)
Table 21.	Port A-H STOP Mode Recovery Source Enable
	Sub-Registers
Table 22.	Port A-H Output Data Register (PxOUT)
Table 23.	Interrupt Vectors in Order of Priority
Table 24.	Interrupt Request 0 Register (IRQ0)
Table 25.	Interrupt Request 1 Register (IRQ1)
Table 26.	Interrupt Request 2 Register (IRQ2)
Table 27.	IRQ0 Enable and Priority Encoding
Table 28.	IRQ0 Enable High Bit Register (IRQ0ENH) 51
Table 29.	IRQ0 Enable Low Bit Register (IRQ0ENL) $\dots 52$
Table 30.	IRQ1 Enable and Priority Encoding $\hdots52$
Table 31.	IRQ1 Enable Low Bit Register (IRQ1ENL) 53





Figure 59. Z8Fxx02 in 64-Pin Low-Profile Quad Flat Package (LQFP)





Figure 61. Z8Fxx03 in 80-Pin Quad Flat Package (QFP)



Signal Mnemonic	I/O	Description
UART Controllers		
TXD0 / TXD1	0	Transmit Data. These signals are the transmit outputs from the UARTs. The TXD signals are multiplexed with general-purpose I/O pins.
RXD0 / RXD1	Ι	Receive Data. These signals are the receiver inputs for the UARTs and IrDAs. The RXD signals are multiplexed with general-purpose I/O pins.
CTS0 / CTS1	Ι	Clear To Send. These signals are control inputs for the UARTs. The $\overline{\text{CTS}}$ signals are multiplexed with general-purpose I/O pins.
Timers (Timer 3 is u	navailab	le in the 40-and 44-pin packages)
TOOUT / TIOUT/ T2OUT / T3OUT	0	Timer Output 0-3. These signals are output pins from the timers. The Timer Output signals are multiplexed with general-purpose I/O pins. T2OUT is not supported in the 40-pin package. T3OUT is not supported in the 40- and 44-pin packages.
T0IN / T1IN/ T2IN / T3IN	Ι	Timer Input 0-3. These signals are used as the capture, gating and counter inputs. The Timer Input signals are multiplexed with general-purpose I/O pins. T3IN is not supported in the 40- and 44-pin packages.
Analog		
ANA[11:0]	Ι	Analog Input. These signals are inputs to the analog-to-digital converter (ADC). The ADC analog inputs are multiplexed with general-purpose I/O pins.
VREF	Ι	Analog-to-digital converter reference voltage input. The VREF pin should be left unconnected (or capacitively coupled to analog ground) if the internal voltage reference is selected as the ADC reference voltage.
Oscillators		
XIN	Ι	External Crystal Input. This is the input pin to the crystal oscillator. A crystal can be connected between it and the XOUT pin to form the oscillator.
XOUT	0	External Crystal Output. This pin is the output of the crystal oscillator. A crystal can be connected between it and the XIN pin to form the oscillator. When the system clock is referred to in this manual, it refers to the frequency of the signal at this pin.
RCOUT	0	RC Oscillator Output. This signal is the output of the RC oscillator. It is multiplexed with a general-purpose I/O pin.
On-Chip Debugger		
DBG	I/O	Debug. This pin is the control and data input and output to and from the On-Chip Debugger. For operation of the On-chip debugger, all power pins (V_{DD} and AV_{DD} must be supplied with power, and all ground pins (V_{SS} and AV_{SS} must be grounded. This pin is open-drain and must have an external pull-up resistor to ensure proper operation.

Table 2. Signal Descriptions (Continued)



Address (Hex)	Register Description	Mnemonic	Reset (Hex)	Page #
F62	SPI Status	SPISTAT	01	108
F63	SPI Mode	SPIMODE	00	109
F64-F65	Reserved	—	XX	
F66	SPI Baud Rate High Byte	SPIBRH	FF	110
F67	SPI Baud Rate Low Byte	SPIBRL	FF	110
F68-F69	Reserved	—	XX	
Analog-to-Digit	al Converter (ADC)			
F70	ADC Control	ADCCTL	20	135
F71	Reserved	_	XX	
F72	ADC Data High Byte	ADCD_H	XX	137
F73	ADC Data Low Bits	ADCD_L	XX	137
F74-FAF	Reserved	_	XX	
DMA 0				
FB0	DMA0 Control	DMA0CTL	00	124
FB1	DMA0 I/O Address	DMA0IO	XX	125
FB2	DMA0 End/Start Address High Nibble	DMA0H	XX	126
FB3	DMA0 Start Address Low Byte	DMA0START	XX	127
FB4	DMA0 End Address Low Byte	DMA0END	XX	128
DMA 1	-			
FB8	DMA1 Control	DMA1CTL	00	124
FB9	DMA1 I/O Address	DMA1IO	XX	125
FBA	DMA1 End/Start Address High Nibble	DMA1H	XX	126
FBB	DMA1 Start Address Low Byte	DMA1START	XX	127
FBC	DMA1 End Address Low Byte	DMA1END	XX	128
DMA ADC	· · · · · · · · · · · · · · · · · · ·			
FBD	DMA_ADC Address	DMAA_ADDR	XX	128
FBE	DMA_ADC Control	DMAACTL	00	130
FBF	DMA_ADC Status	DMAASTAT	00	131
Interrupt Contr	oller			
FC0	Interrupt Request 0	IRQ0	00	48
FC1	IRQ0 Enable High Bit	IRQOENH	00	51
FC2	IRQ0 Enable Low Bit	IRQ0ENL	00	51
FC3	Interrupt Request 1	IRQ1	00	49
FC4	IRQ1 Enable High Bit	IRQ1ENH	00	52
FC5	IRQ1 Enable Low Bit	IRQ1ENL	00	52
FC6	Interrupt Request 2	IRQ2	00	50
FC7	IRQ2 Enable High Bit	IRQ2ENH	00	53
FC8	IRQ2 Enable Low Bit	IRQ2ENL	00	53
FC9-FCC	Reserved	_	XX	
FCD	Interrupt Edge Select	IRQES	00	54
XX=Undefined				

Table 6. Register File Address Map (Continued)



Stop Mode Recovery Using a GPIO Port Pin Transition

Each of the GPIO Port pins may be configured as a Stop Mode Recovery input source. On any GPIO pin enabled as a Stop Mode Recover source, a change in the input pin value (from High to Low or from Low to High) initiates Stop Mode Recovery. In the Watch-Dog Timer Control register, the STOP bit is set to 1.

Caution:

In Stop mode, the GPIO Port Input Data registers (PxIN) are disabled. The Port Input Data registers record the Port transition only if the signal stays on the Port pin through the end of the STOP Mode Recovery delay. Thus, short pulses on the Port pin can initiate STOP Mode Recovery without being written to the Port Input Data register or without initiating an interrupt (if enabled for that pin).



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.

BITS 7 6 5 4 3 2 1 0 TH FIELD 0 0 0 0 0 0 0 0 RESET R/W R/W R/W R/W R/W R/W R/W R/W R/W F00H, F08H, F10H, F18H ADDR

Table 38. Timer 0-3 High Byte Register (TxH)

Table 39>. Timer 0-3 Low Byte Register (TxL)

BITS	7	6	5	4	3	2	1	0
FIELD	TL							
RESET	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADDR				F01H, F09H,	F11H, F19H			

TH and 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 0-3 Reload High and Low Byte (TxRH and TxRL) registers (Tables 40 and 41) 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.

In single-byte DMA transactions to the Timer Reload High Byte register, the temporary holding register is bypassed and the value is written directly to the register. If the DMA is



- 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 data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Transmit Shift register and transmit the data.
- 7. To transmit additional bits, return to Step 5.

Transmitting Data using the Interrupt-Driven Method

The UART Transmitter interrupt indicates the availability of the Transmit Data register to accept new data for transmission. Follow these 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 desired 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 desired priority.
- 5. Write to the UART Control 1 register to enable Multiprocessor (9-bit) mode functions, if desired.
- 6. Write to the UART Control 0 register to:
 - Set the transmit enable bit (TEN) to enable the UART for data transmission
 - Enable parity, if desired, and select either even or odd parity.
 - Set or clear the CTSE bit to enable or disable control from the receiver via the $\overline{\text{CTS}}$ pin.
- 7. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data transmission. When the UART Transmit interrupt is detected, the associated interrupt service routine (ISR) should perform the following:

- 8. Write the data byte to the UART Transmit Data register. The transmitter will automatically transfer the data to the Transmit Shift register and transmit the data.
- 9. Clear the UART Transmit interrupt bit in the applicable Interrupt Request register.
- 10. Execute the IRET instruction to return from the interrupt-service routine and wait for the Transmit Data register to again become empty.





Figure 75. SPI Configured as a Master in a Single Master, Multiple Slave System



Figure 76. SPI Configured as a Slave

Operation

The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (serial clock, transmit, receive and Slave select). The SPI block consists of trans-



1 = DMAx is enabled and initiates a data transfer upon receipt of a request from the trigger source.

DLE—DMAx Loop Enable

0 = DMAx reloads the original Start Address and is then disabled after the End Address data is transferred.

1 = DMAx, after the End Address data is transferred, reloads the original Start Address and continues operating.

DDIR—DMAx Data Transfer Direction

0 =Register File \rightarrow on-chip peripheral control register.

1 = on-chip peripheral control register \rightarrow Register File.

IRQEN—DMAx Interrupt Enable

0 = DMAx does not generate any interrupts.

1 = DMAx generates an interrupt when the End Address data is transferred.

WSEL-Word Select

0 = DMAx transfers a single byte per request.

1 = DMAx transfers a two-byte word per request. The address for the on-chip peripheral control register must be an even address.

RSS-Request Trigger Source Select

The Request Trigger Source Select field determines the peripheral that can initiate a DMA request 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: I²C Receiver Interrupt. DMA1 Control register: I²C Transmitter Interrupt register empty.

111 = Reserved.

DMAx I/O Address Register

The DMAx I/O Address register contains the low byte of the on-chip peripheral address for data transfer. The full 12-bit Register File address is given by {FH, DMAx_IO[7:0]}.



RPEN—Read Protect Option Bit Enabled 0 = The Read Protect Option Bit is disabled (1). 0 = The Read Protect Option Bit is enabled (0), disabling many OCD commands. Reserved

These bits are always 0.

OCD Watchpoint Control Register

The OCD Watchpoint Control register is used to configure the debug Watchpoint.

Table 96. OCD Watchpoint Control/Address (WPTCTL)

BITS	7	6	5	4	3	2	1	0
FIELD	WPW	WPR	WPDM	Reserved	WPTADDR[11:8]			
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

WPW-Watchpoint Break on Write

This bit cannot be set if the Read Protect Option Bit is enabled.

0 = Watchpoint Break on Register File write is disabled.

1 = Watchpoint Break on Register File write is enabled.

WPR-Watchpoint Break on Read

This bit cannot be set if the Read Protect Option Bit is enabled.

0 = Watchpoint Break on Register File read is disabled.

1 = Watchpoint Break on Register File write is enabled.

WPDM-Watchpoint Data Match

If this bit is set, then the Watchpoint only generates a Debug Break if the data being read or written matches the specified Watchpoint data. Either the WPR and/or WPW bits must also be set for this bit to affect operation. This bit cannot be set if the Read Protect Option Bit is enabled.

0 = Watchpoint Break on read and/or write does not require a data match.

1 = Watchpoint Break on read and/or write requires a data match.

Reserved

This bit is reserved and must be 0.

RADDR[11:8]—Register address

These bits specify the upper 4 bits of the Register File address to match when generating a Watchpoint Debug Break. The full 12-bit Register File address is given by {WPTCTL3:0], WPTADDR[7:0]}.



Table 101. DC Characteristics

		$T_A = -40^0 C$ to $105^0 C$				
Symbol	Parameter	Minimum	Typical	Maximum	Units	Conditions
I _{PU}	Weak Pull-up Current	30	100	350	μA	$V_{DD} = 3.0 - 3.6V$
I _{CCS}	Supply Current in Stop Mode		600		μA	$V_{DD} = 3.3 V$

¹ This condition excludes all pins that have on-chip pull-ups, when driven Low.

² These values are provided for design guidance only and are not tested in production.

Figure 91 illustrates the typical current consumption while operating at 25°C, 3.3V, versus the system clock frequency.



Figure 91. Nominal ICC Versus System Clock Frequency



On-Chip Debugger Timing

Figure 95 and Table 109 provide timing information for DBG pins. The timing specifications presume a rise and fall time on DBG of less than 4μ s.



Figure 95. On-Chip Debugger Timing

		Dela	Delay (ns)		
Parameter	Abbreviation	Minimum	Maximum		
DBG					
T ₁	XIN Rise to DBG Valid Delay	-	15		
T ₂	XIN Rise to DBG Output Hold Time	2	_		
T ₃	DBG to XIN Rise Input Setup Time	10	_		
T ₄	DBG to XIN Rise Input Hold Time	5	_		
	DBG frequency		System Clock / 4		

Table	109.	On-Chin	Debugger	Timing
Table	10/1	On-Cmp	Debugger	1 mmg

ZILOG

eZ8 CPU Instruction Set

Assembly Language Programming Introduction

The eZ8 CPU assembly language provides a means for writing an application program without having to be concerned with actual memory addresses or machine instruction formats. A program written in assembly language is called a source program. Assembly language allows the use of symbolic addresses to identify memory locations. It also allows mnemonic codes (opcodes and operands) to represent the instructions themselves. The opcodes identify the instruction while the operands represent memory locations, registers, or immediate data values.

Each assembly language program consists of a series of symbolic commands called statements. Each statement can contain labels, operations, operands and comments.

Labels can be assigned to a particular instruction step in a source program. The label identifies that step in the program as an entry point for use by other instructions.

The assembly language also includes assembler directives that supplement the machine instruction. The assembler directives, or pseudo-ops, are not translated into a machine instruction. Rather, the pseudo-ops are interpreted as directives that control or assist the assembly process.

The source program is processed (assembled) by the assembler to obtain a machine language program called the object code. The object code is executed by the eZ8 CPU. An example segment of an assembly language program is detailed in the following example.

Assembly Language Source Program Example

JP START	; Everything after the semicolon is a comment.
START:	; A label called "START". The first instruction (JP START) in this ; example causes program execution to jump to the point within the ; program where the START label occurs.
LD R4, R7	; A Load (LD) instruction with two operands. The first operand, ; Working Register R4, is the destination. The second operand, ; Working Register R7, is the source. The contents of R7 is ; written into R4.
LD 234H, #%01	; Another Load (LD) instruction with two operands. ; The first operand, Extended Mode Register Address 234H, ; identifies the destination. The second operand, Immediate Data



Condition Codes

The C, Z, S and V flags control the operation of the conditional jump (JP cc and JR cc) instructions. Sixteen frequently useful functions of the flag settings are encoded in a 4-bit field called the condition code (cc), which forms Bits 7:4 of the conditional jump instructions. The condition codes are summarized in Table 117. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the flag test operation is used to decide if the conditional jump is executed.

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0000	0	F	Always False	-
0001	1	LT	Less Than	(S XOR V) = 1
0010	2	LE	Less Than or Equal	(Z OR (S XOR V)) = 1
0011	3	ULE	Unsigned Less Than or Equal	(C OR Z) = 1
0100	4	OV	Overflow	V = 1
0101	5	Ml	Minus	S = 1
0110	6	Z	Zero	Z = 1
0110	6	EQ	Equal	Z = 1
0111	7	С	Carry	C = 1
0111	7	ULT	Unsigned Less Than	C = 1
1000	8	T (or blank)	Always True	-
1001	9	GE	Greater Than or Equal	(S XOR V) = 0
1010	А	GT	Greater Than	(Z OR (S XOR V)) = 0
1011	В	UGT	Unsigned Greater Than	(C = 0 AND Z = 0) = 1
1100	С	NOV	No Overflow	V = 0
1101	D	PL	Plus	S = 0
1110	Е	NZ	Non-Zero	Z = 0
1110	Е	NE	Not Equal	Z = 0
1111	F	NC	No Carry	C = 0
1111	F	UGE	Unsigned Greater Than or Equal	C = 0

Table 117. Condition Codes



Abbreviation	Description	Abbreviation	Description
b	Bit position	IRR	Indirect Register Pair
сс	Condition code	р	Polarity (0 or 1)
X	8-bit signed index or displacement	r	4-bit Working Register
DA	Destination address	R	8-bit register
ER	Extended Addressing register	r1, R1, Ir1, Irr1, IR1, rr1, RR1, IRR1, ER1	Destination address
IM	Immediate data value	r2, R2, Ir2, Irr2, IR2, rr2, RR2, IRR2, ER2	Source address
Ir	Indirect Working Register	RA	Relative
IR	Indirect register	rr	Working Register Pair
Irr	Indirect Working Register Pair	RR	Register Pair

Table 127. Opcode Map Abbreviations



Problem Description or Suggestion

Provide a complete description of the problem or your suggestion. If you are reporting a specific problem, include all steps leading up to the occurrence of the problem. Attach additional pages as necessary.



controller 111 controller signals 13 interrupts 112 operation 111 SDA and SCL signals 111 stop and start conditions 112 I2CBRH register 121 I2CBRL register 121 I2CCTL register 119 I2CDATA register 118 I2CSTAT register 118 IM 184 immediate data 184 immediate operand prefix 185 **INC 187** increment 187 increment word 187 **INCW 187** indexed 184 indirect address prefix 185 indirect register 184 indirect register pair 184 indirect working register 184 indirect working register pair 184 infrared encoder/decoder (IrDA) 95 instruction set, ez8 CPU 182 instructions ADC 187 **ADCX 187** ADD 187 **ADDX 187** AND 190 **ANDX 190** arithmetic 187 **BCLR 188 BIT 188** bit manipulation 188 block transfer 188 **BRK 190 BSET 188** BSWAP 188, 191 **BTJ 190** BTJNZ 190 **BTJZ 190**

CALL 190 CCF 188, 189 **CLR 189** COM 190 CP 187 CPC 187 **CPCX 187** CPU control 189 CPX 187 DA 187 **DEC 187 DECW 187** DI 189 **DJNZ 190** EI 189 **HALT 189 INC 187 INCW 187 IRET 190** JP 190 LD 189 LDC 189 LDCI 188, 189 LDE 189 LDEI 188 LDX 189 LEA 189 load 189 logical 190 **MULT 187** NOP 189 OR 190 ORX 190 POP 189 **POPX 189** program control 190 **PUSH 189** PUSHX 189 RCF 188, 189 **RET 190** RL 191 RLC 191 rotate and shift 191 RR 191



logical exclusive OR/extended addressing 190 logical instructions 190 logical OR 190 logical OR/extended addressing 190 low power modes 31 LQFP 44 lead 207 64 lead 208

Μ

master interrupt enable 46 master-in, slave-out and-in 101 memory data 19 program 18 **MISO 101** mode capture 71 capture/compare 71 continuous 70 counter 70 gated 71 one-shot 70 **PWM 70** modes 71 **MOSI 101 MULT 187** multiply 187 multiprocessor mode, UART 84

N

NOP (no operation) 189 not acknowledge interrupt 112 notation b 184 cc 184 DA 184 ER 184 IM 184 IR 184 IR 184 IRR 184 Irr 184 p 184 R 184 r 184 RA 184 RR 184 rr 184 vector 184 X 184 notational shorthand 184

0

OCD architecture 151 auto-baud detector/generator 154 baud rate limits 154 block diagram 151 breakpoints 155 commands 156 control register 161 data format 154 DBG pin to RS-232 Interface 152 debug mode 153 debugger break 190 interface 152 serial errors 155 status register 162 timing 178 watchpoint address register 164 watchpoint control register 163 watchpoint data register 164 watchpoints 155 OCD commands execute instruction (12H) 160 read data memory (0DH) 160 read OCD control register (05H) 158 read OCD revision (00H) 157 read OCD status register (02H) 157 read program counter (07H) 158 read program memory (0BH) 159 read program memory CRC (0EH) 160 read register (09H) 158 read runtime counter (03H) 157



gated mode 64, 71 one-shot mode 58, 70 operating mode 58 PWM mode 61, 70 reading the timer count values 66 reload high and low byte registers 67 timer control register definitions 66 timer output signal operation 66 timers 0-3 control registers 70 high and low byte registers 66, 69 TM. TMX 188 tools, hardware and software 214 transmit IrDA data 96 transmit interrupt 112 transmitting UART data-polled method 80 transmitting UART data-interrupt-driven method 81 **TRAP 190**

U

UART 4 architecture 78 asynchronous data format without/with parity 80 baud rate generator 85 baud rates table 93 control register definitions 86 controller signals 14 data format 79 interrupts 85 multiprocessor mode 84 receiving data using DMA controller 83 receiving data using interrupt-driven method 82 receiving data using the polled method 82 transmitting data using the interrupt-driven method 81 transmitting data using the polled method 80 x baud rate high and low registers 91 x control 0 and control 1 registers 89 x status 0 and status 1 registers 87 **UxBRH** register 91

UxBRL register 92 UxCTL0 register 89 UxCTL1 register 90 UxRXD register 87 UxSTAT0 register 87 UxSTAT1 register 89 UxTXD register 86

V

vector 184 voltage brown-out reset (VBR) 27

W

watch-dog timer approximate time-out delays 72, 73 CNTL 28 control register 75 electrical characteristics and timing 174 interrupt in normal operation 73 interrupt in stop mode 73 operation 72 refresh 73, 189 reload unlock sequence 74 reload upper, high and low registers 76 reset 28 reset in normal operation 74 reset in stop mode 74 time-out response 73 WDTCTL register 75 WDTH register 76 WDTL register 77 working register 184 working register pair 184 WTDU register 76

X

X 184 XOR 190 XORX 190