



**Crimzon<sup>®</sup> Infrared Microcontrollers**

**ZLF645 Series Flash MCUs  
with Learning Amplification**

**Product Specification**

PS026407-0408



**Warning:** DO NOT USE IN LIFE SUPPORT!

### **LIFE SUPPORT POLICY**

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

### **As used herein**

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

### **Document Disclaimer**

©2008 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8 and Crimzon are registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.



**ISO 9001:2000  
FS 507510**

Zilog products are designed and manufactured under an ISO registered 9001:2000 Quality Management System. For more details, please visit [www.zilog.com/quality](http://www.zilog.com/quality).

# Revision History

Each instance in the revision history table reflects a change to this document from its previous revision. For more details, refer to the corresponding pages or appropriate links given in the table below.

Date	Version	Description	Page Number
April 2008	07	Updated <a href="#">Enabling The Flash Byte Programming Interface</a> section	76
April 2008	06	Deleted “Design Info” subsection and all of its associated text. Updated <a href="#">Flash Memory Overview</a> section; updated <a href="#">Figure 19</a> . Updated <a href="#">Flash Frequency High and Low Byte Registers</a> section. Updated lcc and lcc1 in <a href="#">Table 81</a> . Updated notes in <a href="#">Table 5</a> and <a href="#">Table 6</a> . Removed “Preliminary” and “Precharacterization Product” notice. Added 20-pin QFN package to <a href="#">Pin Description</a> , <a href="#">Table 3</a> , <a href="#">Table 4</a> , <a href="#">Packaging</a> , <a href="#">Table 88</a> , and <a href="#">Part Number Description</a> . Change P31 to P32 at the beginning of <a href="#">Universal Asynchronous Receiver/Transmitter</a> .	All 64, 65 74 160 9, 10 All 5, 6, 7, 171, 180, 182 79
January 2008	05	Updated <a href="#">Flash Operation Timing Using Flash Frequency Registers</a> and <a href="#">Flash Frequency High and Low Byte Registers</a> .	68, 74
January 2008	04	Chapter <a href="#">Reset and Power Management</a> : Updated <a href="#">Table 69</a> .	137
January 2008	03	Replaced Advance with Preliminary and added Precharacterization Product section. Updated <a href="#">Table 81</a> , <a href="#">Table 82</a> , and <a href="#">Table 83</a> .	All
August 2007	02	Updated <a href="#">Memory and Registers</a> section; Added description to <a href="#">Stack</a> . Updated <a href="#">ICP Interface</a> section; <a href="#">Table 30</a> , Field for Bit 0 changed from RST to Reserved. Updated <a href="#">Reset and Power Management</a> section; <a href="#">Watchdog Timer</a> , changed the Field for Bits 6 through Bits 4 from Reserved to Time-Out Select.	42 61 136
		Minor updates throughout the document.	All



# Table of Contents

<b>Architectural Overview</b> .....	<b>1</b>
Features .....	1
Interrupt Sources .....	2
Additional Features .....	2
Functional Block Diagram .....	4
<b>Pin Description</b> .....	<b>5</b>
<b>I/O Port Pin Functions</b> .....	<b>18</b>
RESET (Input, Active Low) .....	18
Port 0 .....	20
Port 1 .....	21
Port 2 .....	22
Port 3 .....	23
Comparator Inputs .....	28
Comparator Outputs .....	28
Port 4 .....	28
Port Configuration Register .....	30
Port 0/1 Mode Register .....	31
Port 0 Register .....	32
Port 1 Register .....	33
Port 2 Mode Register .....	34
Port 2 Register .....	35
Port 3 Mode Register .....	36
Port 3 Register .....	37
Port 4 Mode Register .....	39
Port 4 Register .....	40
<b>Memory and Registers</b> .....	<b>41</b>
Flash Program/Constant Memory .....	41
Register File .....	42
Stack .....	42
Register Pointer Example .....	45
Linear Memory Addressing .....	45
Register Pointer Register .....	48
Stack Pointer Register .....	48



<b>Register File Summary</b> .....	<b>50</b>
<b>ICP Interface</b> .....	<b>53</b>
Enabling ICP Mode .....	53
State of ZLF645 in ICP Mode .....	53
Enabling Flash Accesses Through the ICP .....	54
ICP Interface Logic Architecture .....	54
ICP Interface Operation .....	54
ICP Data Format .....	55
ICP Auto-Baud Detector/Generator .....	55
ICP Serial Errors .....	56
ICP In-Circuit Programming Commands .....	57
In-Circuit Programming Control Register Definitions .....	61
ICP Control Register .....	61
ICP Status Register .....	62
TEST Mode Register .....	63
Exiting ICP Mode .....	63
<b>Flash Controller</b> .....	<b>64</b>
Flash Memory Overview .....	64
Flash Information Block .....	65
Flash Controller Operation .....	66
Flash Operation Timing Using Flash Frequency Registers .....	68
Flash Code Protection Against External Access .....	68
Flash Code Protection Against Accidental Program and Erasure .....	68
Byte Programming .....	69
Page Erase .....	70
Mass Erase .....	70
Flash Control Register Definitions .....	71
Flash Control Register .....	71
Flash Status Register .....	72
Flash Page Select Register .....	72
Flash Sector Protect Register .....	73
Flash Frequency High and Low Byte Registers .....	74
Flash Controller Functions Summary .....	75
<b>Flash Byte Programming Interface</b> .....	<b>76</b>
Enabling The Flash Byte Programming Interface .....	76
Flash Byte Programming Interface Flash Access Restrictions .....	76
<b>Infrared Learning Amplifier</b> .....	<b>78</b>



<b>Universal Asynchronous Receiver/Transmitter</b> .....	<b>79</b>
Architecture .....	80
Operation .....	80
Data Format .....	81
Transmitting Data Using Polled Method .....	81
Transmitting Data Using Interrupt-Driven Method .....	82
Receiving Data Using the Polled Method .....	83
Receiving Data Using the Interrupt-Driven Method .....	83
UART Interrupts .....	84
UART Baud Rate Generator .....	87
UART Receive Data Register/UART Transmit Data Register .....	89
UART Status Register .....	89
UART Control Register .....	90
UART Baud Rate Generator Constant Register .....	91
<b>Timers</b> .....	<b>93</b>
Counter/Timer Functional Blocks .....	94
Input Circuit .....	94
T8 TRANSMIT Mode .....	95
T8 DEMODULATION Mode .....	99
T16 TRANSMIT Mode .....	103
T16 DEMODULATION Mode .....	104
PING-PONG Mode .....	105
Timer Output .....	106
Counter/Timer Registers .....	108
Timer 8 Capture High Register .....	108
Timer 8 Capture Low Register .....	108
Timer 16 Capture High Register .....	109
Timer 16 Capture Low Register .....	110
Counter/Timer 16 High Hold Register .....	110
Counter/Timer 16 Low Hold Register .....	111
Counter/Timer 8 High Hold Register .....	111
Counter/Timer 8 Low Hold Register .....	112
Counter/Timer 8 Control Register .....	113
T8 and T16 Common Functions Register .....	115
Counter/Timer 16 Control Register .....	118
Timer 8/Timer 16 Control Register .....	119
<b>Interrupts</b> .....	<b>121</b>
Interrupt Priority Register .....	124
Interrupt Request Register .....	125



Interrupt Mask Register .....	127
.....	128
<b>Clock .....</b>	<b>129</b>
Crystal 1 Oscillator Pin (XTAL1) .....	130
Crystal 2 Oscillator Pin (XTAL2) .....	130
Internal Clock Signals (SCLK and TCLK) .....	130
<b>Reset and Power Management .....</b>	<b>132</b>
Voltage Brownout Standby .....	134
STOP Mode .....	134
HALT Mode .....	134
Voltage Detection .....	135
Power-On Reset Timer .....	136
Watchdog Timer .....	136
Reset/Stop Mode Recovery Status .....	137
Fast Stop Mode Recovery .....	138
Stop Mode Recovery Interrupt .....	138
Stop Mode Recovery Event Sources .....	138
SMR Register Events .....	139
SMR1 Register Events .....	142
SMR2 Register Events .....	145
SMR3 Register Events .....	147
Stop Mode Recovery Register 4 .....	151
<b>Z8 LXMC CPU Programming Summary .....</b>	<b>152</b>
Addressing Notation .....	152
Flags Register .....	155
Condition Codes .....	156
<b>Electrical Characteristics .....</b>	<b>158</b>
Absolute Maximum Ratings .....	158
Standard Test Conditions .....	159
Capacitance .....	159
DC Characteristics .....	160
AC Characteristics .....	163
<b>Flash Option Bits .....</b>	<b>166</b>
Operation .....	166
Option Bit Shadow Register Loading By Reset .....	166
User Option Bit Locations in Flash Memory .....	167



User Option Bit Shadow Register Access .....	167
User Option Byte 0 and Option Byte 0 Shadow Register Definitions .....	167
User Option Byte 1 and Option Byte 1 Shadow Register Definitions .....	168
<b>Packaging .....</b>	<b>171</b>
<b>Ordering Information .....</b>	<b>180</b>
Part Number Description .....	182
<b>Index .....</b>	<b>183</b>
<b>Customer Support .....</b>	<b>189</b>



# Architectural Overview

Zilog's ZLF645 Series of Flash MCU's are members of the Crimzon<sup>®</sup> family of infrared microcontrollers. This series provides a directly-compatible code upgrade path to other Crimzon MCUs, offers a robust learning function, and features up to 64 KB Flash memory and 1K general-purpose Random Access Memory (RAM). Two timers allow the generation of complex signals while performing other counting operations.

A Universal Asynchronous Receiver/Transmitter (UART) allows the ZLF645 MCU to function as a slave/master database chip. When the UART is not in use, the Baud Rate Generator (BRG) can be used as a third timer. Enhanced Stop Mode Recovery features allow the ZLF645 MCU to recover from STOP mode on any change of logic and on any combination of the 12 SMR inputs. The SMR source can also be used as an interrupt source.

Many high-end remote control units offer a learning function. A learning function allows a replacement remote unit to learn infrared signals from the original remote unit and regenerate the signal. However, the amplifying circuits of many learning remotes are expensive and are not tuned well. The ZLF645 MCU is the first chip to offer a built-in tuned amplification circuit in a wide range of positions and battery voltages. The only external component required is a photodiode.

The ZLF645 MCU greatly reduces the system cost and improves learning function reliability. With all new features, the ZLF645 MCU is excellent for infrared remote control and other MCU applications.

## Features

[Table 1](#) lists the memory, I/O, and power features of the ZLF645 Flash MCU. Additional features are listed below the table.

**Table 1. ZLF645 Flash MCU Features**

Device	Flash (KB)	RAM*	I/O Lines	Voltage Range
ZLF645 Flash MCU	32 or 64	512 B or 1 K	16, 24, or 40	2.0 V–3.6 V

\*General-purpose registers implemented as RAM.

## Interrupt Sources

The ZLF645 MCU supports 23 interrupt sources with 6 interrupt vectors, as given below:

- Three external interrupts.
- Two from T8, T16 time-out and capture.
- Three from UART Tx, UART Rx, and UART BRG.
- One from LVD.
- Fourteen from SMR source P20-P27, P30-P33, P00, and P07:
  - Any change in logic from P20-P27, P30-P33 can generate an interrupt or SMR

## Additional Features

The additional features of ZLF645 MCU include:

- IR learning amplifier.
- Low power consumption—11 mW (typical).
- Three standby modes:
  - STOP—1.7  $\mu$ A (typical)
  - HALT—0.6 mA (typical)
  - Low-voltage reset
- Intelligent counter/timer architecture to automate generation or reception and demodulation of complex waveform, and pulsed signals:
  - One programmable 8-bit counter/timer with two capture registers and two load registers
  - One programmable 16-bit counter/timer with one 16-bit capture register pair and one 16-bit load register pair
  - Programmable input glitch filter for pulse reception
  - The UART baud rate generator can be used as another 8-bit timer, when the UART is not in use
- Six priority interrupts:
  - Three external/UART interrupts
  - Two assigned to counter/timers
  - One low-voltage detection interrupt

- 8-bit UART:
  - R<sub>X</sub> and T<sub>X</sub> interrupts
  - 4800, 9600, 19200, and 38400 baud rates
  - Parity Odd/Even/None
  - Stop bits 1/2
- ICP (In-circuit Flash Programming) interface multiplexed with one of the GPIO's.
- Intelligent Power-On Reset (POR) to provide reduced POR time on detection of stable clock from external crystal oscillator or resonator.
- Low-voltage and high-voltage detection flags.
- Programmable Watchdog Timer (WDT)/POR circuits.
- Two on-board analog comparators with independent reference voltages and programmable interrupt polarity.
- User-selectable options through option bit Flash coding (ON/OFF):
  - Port 0 pins 0–3 pull-up transistors
  - Port 0 pins 4–7 pull-up transistors
  - Port 1 pins 0–3 pull-up transistors
  - Port 1 pins 4–7 pull-up transistors
  - Port 2 pins 0–7 pull-up transistors
  - Port 3 pins 0–3 pull-up transistors
  - Port 4 pins 0–7 pull-up transistors
  - WDT enabled at Power-On Reset
  - Flash lowest half main memory protect
  - Flash entire main memory protect
  - 16-bit addressability for stack pointer
  - No division, divide by 2, divide by 16, or divide by 32 of external clock to system clock

► **Note:** *All signals with an overline, “ $\bar{\phantom{x}}$ ”, are active Low. For example,  $B/\bar{W}$ , in which *WORD* is active Low, and  $\bar{B}/W$ , in which *BYTE* is active Low.*

Power connections use the conventional descriptions listed in [Table 2](#).

**Table 2. Power Connections**

Connection	Device
Power	V <sub>DD</sub>
Ground	V <sub>SS</sub>

## Functional Block Diagram

Figure 1 displays the functional blocks of the ZLF645 Flash MCU.

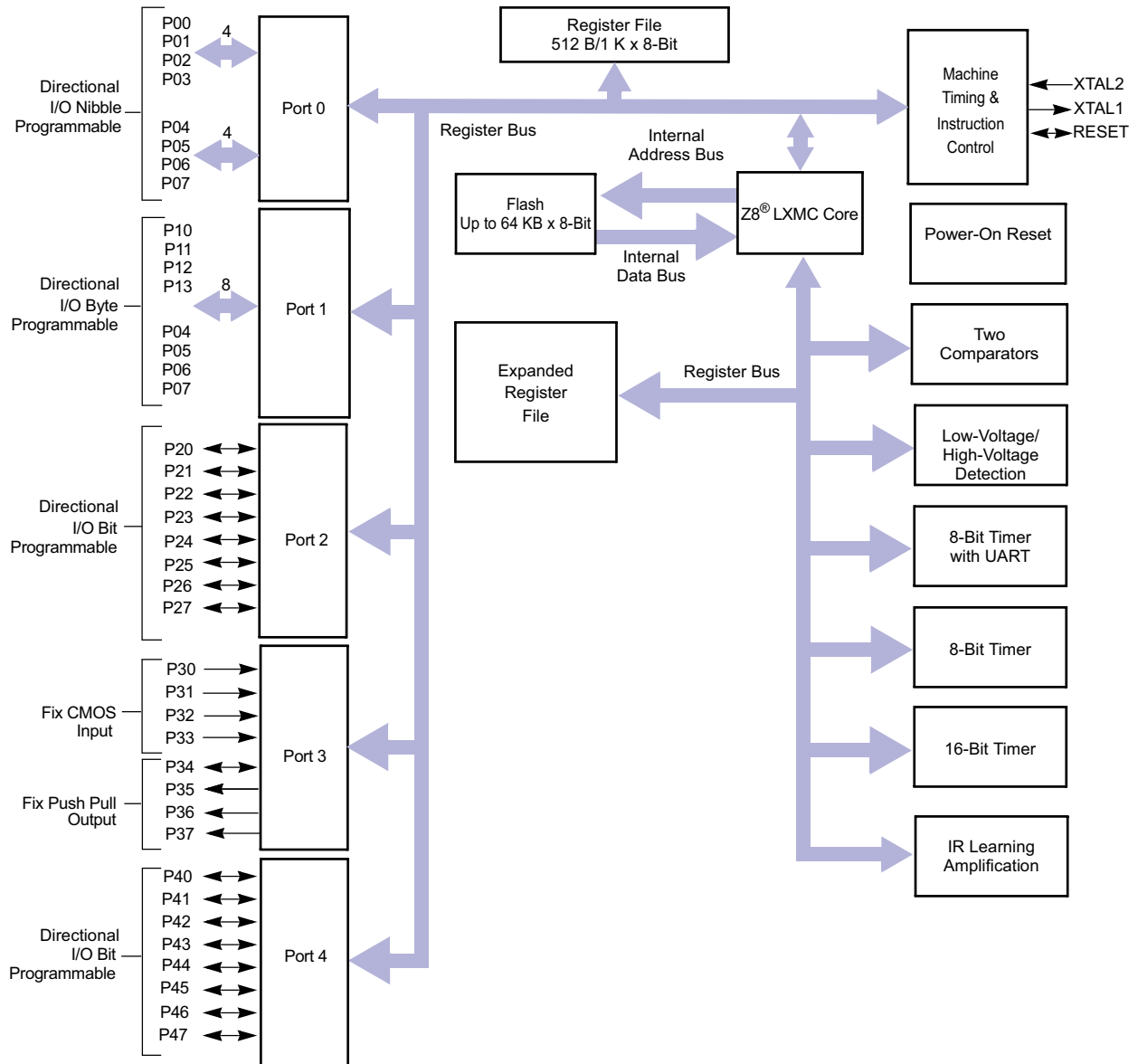


Figure 1. ZLF645 Flash MCU Functional Block Diagram

# Pin Description

Figure 2 displays the pin configuration for ZLF645 MCU 20-pin QFN packages.

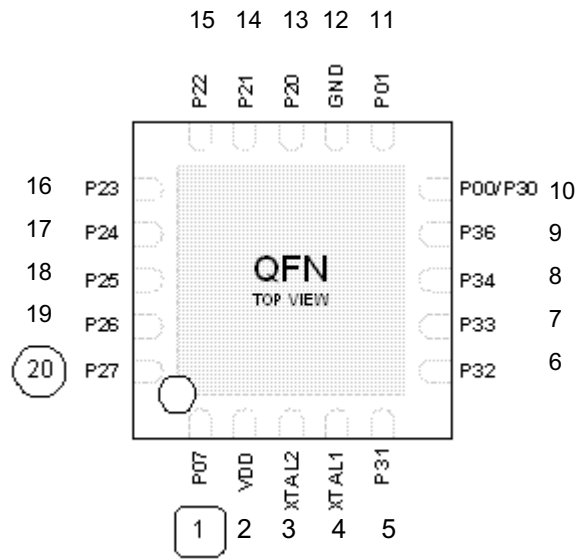


Figure 2. 20-Pin QFN Pin Configuration

Table 3 lists the function and signal directions of each pin within the 20-pin QFN package sequentially by pin number.

**Table 3. 20-Pin QFN Sequential Pin Identification**

Pin No	Symbol	Function	Signal Direction
1	P07	Port 0, bit 7	Input/Output
2	V <sub>DD</sub>	Power Supply	Input
3	XTAL2	Crystal oscillator	Output
4	XTAL1	Crystal oscillator	Input
5	P31	Port 3, bit 1	Input
6	P32	Port 3, bit 2	Input
7	P33	Port 3, bit 3	Input
8	P34	Port 3, bit 4	Input/Output
9	P36	Port 3, bit 6	Output
10	P00	Port 0, bit 0	Input/Output
	P30	Port 3, bit 0	Input
11	P01	Port 0, bit 1	Input/Output
12	GND	Ground	Input
13	P20	Port 2, bit 0	Input/Output
14	P21	Port 2, bit 1	Input/Output
15	P22	Port 2, bit 2	Input/Output
16	P23	Port 2, bit 3	Input/Output
17	P24	Port 2, bit 4	Input/Output
18	P25	Port 2, bit 5	Input/Output
19	P26	Port 2, bit 6	Input/Output
20	P27	Port 2, bit 7	Input/Output

**Note:** When the Port 0 low-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Table 4 lists the function and signal direction of each pin within the 20-pin QFN package by function.

**Table 4. 20-Pin QFN Functional Pin Identification**

Pin No	Symbol	Function	Signal Direction
10	P00	Port 0, bit 0	Input/Output
	P30	Port 3, bit 0	Input
11	P01	Port 0, bit 1	Input/Output
1	P07	Port 0, bit 7	Input/Output
13	P20	Port 2, bit 0	Input/Output
14	P21	Port 2, bit 1	Input/Output
15	P22	Port 2, bit 2	Input/Output
16	P23	Port 2, bit 3	Input/Output
17	P24	Port 2, bit 4	Input/Output
18	P25	Port 2, bit 5	Input/Output
19	P26	Port 2, bit 6	Input/Output
20	P27	Port 2, bit 7	Input/Output
5	P31	Port 3, bit 1	Input
6	P32	Port 3, bit 2	Input
7	P33	Port 3, bit 3	Input
8	P34	Port 3, bit 4	Input/Output
9	P36	Port 3, bit 6	Output
2	V <sub>DD</sub>	Power Supply	Input
12	GND	Ground	Input
4	XTAL1	Crystal oscillator	Input
3	XTAL2	Crystal oscillator	Output

**Note:** When the Port 0 low-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Figure 3 displays the pin configuration for ZLF645 MCU 20-pin PDIP, SOIC, and SSOP packages.

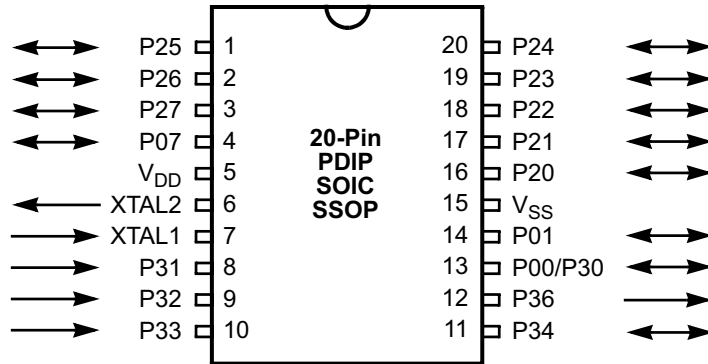


Figure 3. 20-Pin PDIP/SOIC/SSOP Pin Configuration



Table 5 lists the function and signal directions of each pin within the 20-pin PDIP, SOIC, and SSOP packages sequentially by pin number.

**Table 5. 20-Pin PDIP/SOIC/SSOP Sequential Pin Identification**

Pin No	Symbol	Function	Signal Direction
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
4	P07	Port 0, bit 7	Input/Output
5	V <sub>DD</sub>	Power Supply	Input
6	XTAL2	Crystal oscillator	Output
7	XTAL1	Crystal oscillator	Input
8	P31	Port 3, bit 1	Input
9	P32	Port 3, bit 2	Input
10	P33	Port 3, bit 3	Input
11	P34	Port 3, bit 4	Input/Output
12	P36	Port 3, bit 6	Output
13	P00	Port 0, bit 0	Input/Output
	P30	Port 3, bit 0	Input
14	P01	Port 0, bit 1	Input/Output
15	V <sub>SS</sub>	Ground	Input
16	P20	Port 2, bit 0	Input/Output
17	P21	Port 2, bit 1	Input/Output
18	P22	Port 2, bit 2	Input/Output
19	P23	Port 2, bit 3	Input/Output
20	P24	Port 2, bit 4	Input/Output

**Note:** When the Port 0 low-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Table 6 lists the function and signal direction of each pin within the 20-pin PDIP, SOIC, and SSOP packages by function.

**Table 6. 20-Pin PDIP/SOIC/SSOP Functional Pin Identification**

Pin No	Symbol	Function	Signal Direction
13	P00	Port 0, bit 0	Input/Output
	P30	Port 3, bit 0	Input
14	P01	Port 0, bit 1	Input/Output
4	P07	Port 0, bit 7	Input/Output
16	P20	Port 2, bit 0	Input/Output
17	P21	Port 2, bit 1	Input/Output
18	P22	Port 2, bit 2	Input/Output
19	P23	Port 2, bit 3	Input/Output
20	P24	Port 2, bit 4	Input/Output
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
8	P31	Port 3, bit 1	Input
9	P32	Port 3, bit 2	Input
10	P33	Port 3, bit 3	Input
11	P34	Port 3, bit 4	Input/Output
12	P36	Port 3, bit 6	Output
5	V <sub>DD</sub>	Power Supply	Input
15	V <sub>SS</sub>	Ground	Input
7	XTAL1	Crystal oscillator	Input
6	XTAL2	Crystal oscillator	Output

**Note:** When the Port 0 low-nibble pull-up option is enabled and the P30 input is Low, current flows through the pull-up to Ground.

Figure 4 displays the pin configuration of the ZLF645 MCU within the 28-pin PDIP, SOIC, and SSOP packages.

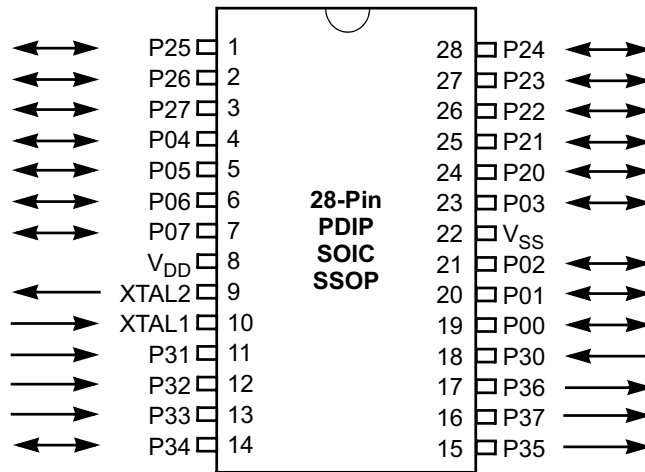


Figure 4. 28-Pin PDIP/SOIC/SSOP Pin Configuration



Table 7 lists the function and signal directions of each pin within the 28-pin PDIP, SOIC, and SSOP packages sequentially by pin number.

**Table 7. 28-Pin PDIP/SOIC/SSOP Sequential Pin Identification**

Pin No	Symbol	Function	Signal Direction
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
4	P04	Port 0, bit 4	Input/Output
5	P05	Port 0, bit 5	Input/Output
6	P06	Port 0, bit 6	Input/Output
7	P07	Port 0, bit 7	Input/Output
8	V <sub>DD</sub>	Power supply	Input
9	XTAL2	Crystal oscillator	Output
10	XTAL1	Crystal oscillator	Input
11	P31	Port 3, bit 1	Input
12	P32	Port 3, bit 2	Input
13	P33	Port 3, bit 3	Input
14	P34	Port 3, bit 4	Input/Output
15	P35	Port 3, bit 5	Output
16	P37	Port 3, bit 7	Output
17	P36	Port 3, bit 6	Output
18	P30	Port 3, bit 0; connect to VDD if not used	Input
19	P00	Port 0, bit 0	Input/Output
20	P01	Port 0, bit 1	Input/Output
21	P02	Port 0, bit 2	Input/Output
22	V <sub>SS</sub>	Ground	Input
23	P03	Port 0, bit 3	Input/Output
24	P20	Port 2, bit 0	Input/Output
25	P21	Port 2, bit 1	Input/Output
26	P22	Port 2, bit 2	Input/Output
27	P23	Port 2, bit 3	Input/Output
28	P24	Port 2, bit 4	Input/Output



Table 8 lists the functions and signal directions of each pin within the 28-pin PDIP, SOIC, and SSOP packages by function.

**Table 8. 28-Pin PDIP/SOIC/SSOP Functional Pin Identification**

Pin No	Symbol	Function	Signal Direction
19	P00	Port 0, bit 0	Input/Output
20	P01	Port 0, bit 1	Input/Output
21	P02	Port 0, bit 2	Input/Output
23	P03	Port 0, bit 3	Input/Output
4	P04	Port 0, bit 4	Input/Output
5	P05	Port 0, bit 5	Input/Output
6	P06	Port 0, bit 6	Input/Output
7	P07	Port 0, bit 7	Input/Output
24	P20	Port 2, bit 0	Input/Output
25	P21	Port 2, bit 1	Input/Output
26	P22	Port 2, bit 2	Input/Output
27	P23	Port 2, bit 3	Input/Output
28	P24	Port 2, bit 4	Input/Output
1	P25	Port 2, bit 5	Input/Output
2	P26	Port 2, bit 6	Input/Output
3	P27	Port 2, bit 7	Input/Output
18	P30	Port 3, bit 0; connect to VDD if not used	Input
11	P31	Port 3, bit 1	Input
12	P32	Port 3, bit 2	Input
13	P33	Port 3, bit 3	Input
14	P34	Port 3, bit 4	Input/Output
15	P35	Port 3, bit 5	Output
17	P36	Port 3, bit 6	Output
16	P37	Port 3, bit 7	Output
8	V <sub>DD</sub>	Power supply	Input
22	V <sub>SS</sub>	Ground	Input
10	XTAL1	Crystal oscillator	Input
9	XTAL2	Crystal oscillator	Output

Figure 5 displays the pin configuration of the ZLF645 MCU within the 48-pin SSOP package.

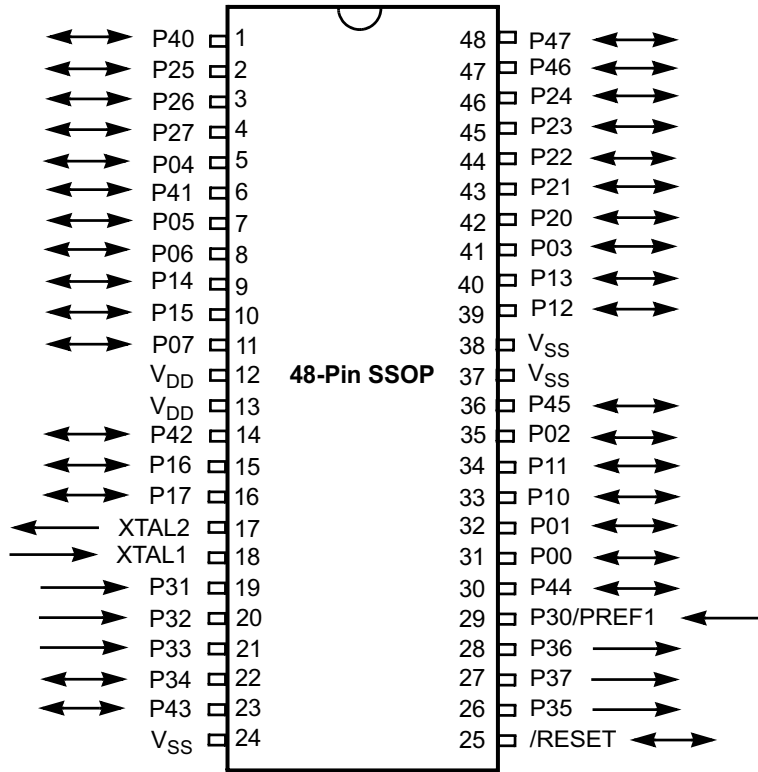


Figure 5. 48-Pin SSOP Pin Configuration

Table 9 lists the functions and signal directions of each pin within the 48-pin SSOP package sequentially by pin number.

Table 9. 48-Pin SSOP Sequential Pin Identification

Pin No	Symbol	Function	Signal Direction
1	P40	Port 4, bit 0	Input/Output
2	P25	Port 2, bit 5	Input/Output
3	P26	Port 2, bit 6	Input/Output
4	P27	Port 2, bit 7	Input/Output
5	P04	Port 0, bit 4	Input/Output
6	P41	Port 4, bit 1	Input/Output
7	P05	Port 0, bit 5	Input/Output



**Table 9. 48-Pin SSOP Sequential Pin Identification (Continued)**

Pin No	Symbol	Function	Signal Direction
8	P06	Port 0, bit 6	Input/Output
9	P14	Port 1, bit 4	Input/Output
10	P15	Port 1, bit 5	Input/Output
11	P07	Port 0, bit 7	Input/Output
12	V <sub>DD</sub>	Power Supply	Input
13	V <sub>DD</sub>	Power Supply	Input
14	P42	Port 4, bit 2	Input/Output
15	P16	Port 1, bit 6	Input/Output
16	P17	Port 1, bit 7	Input/Output
17	XTAL2	Crystal oscillator	Output
18	XTAL1	Crystal oscillator	Input
19	P31	Port 3, bit 1	Input
20	P32	Port 3, bit 2	Input
21	P33	Port 3, bit 3	Input
22	P34	Port 3, bit 4	Input/Output
23	P43	Port 4, bit 3	Input/Output
24	V <sub>SS</sub>	Ground	Input
25	/RESET	Bidirectional reset signal	Input/Output
26	P35	Port 3, bit 5	Output
27	P37	Port 3, bit 7	Output
28	P36	Port 3, bit 6	Output
29	P30/PREF1	Port 3, bit 0	Input
30	P44	Port 4, bit 4	Input/Output
31	P00	Port 0, bit 0	Input/Output
32	P01	Port 0, bit 1	Input/Output
33	P10	Port 1, bit 0	Input/Output
34	P11	Port 1, bit 1	Input/Output
35	P02	Port 0, bit 2	Input/Output
36	P45	Port 4, bit 5	Input/Output
37	V <sub>SS</sub>	Ground	Input
38	V <sub>SS</sub>	Ground	Input
39	P12	Port 1, bit 2	Input/Output
40	P13	Port 1, bit 3	Input/Output



**Table 9. 48-Pin SSOP Sequential Pin Identification (Continued)**

Pin No	Symbol	Function	Signal Direction
41	P03	Port 0, bit 3	Input/Output
42	P20	Port 2, bit 0	Input/Output
43	P21	Port 2, bit 1	Input/Output
44	P22	Port 2, bit 2	Input/Output
45	P23	Port 2, bit 3	Input/Output
46	P24	Port 2, bit 4	Input/Output
47	P46	Port 4, bit 6	Input/Output
48	P47	Port 4, bit 7	Input/Output

Table 10 lists the functions and signal directions of each pin within the 48-pin SSOP package by function.

**Table 10. 48-Pin SSOP Functional Pin Identification**

Pin No	Symbol	Function	Signal Direction
31	P00	Port 0, bit 0	Input/Output
32	P01	Port 0, bit 1	Input/Output
35	P02	Port 0, bit 2	Input/Output
41	P03	Port 0, bit 3	Input/Output
5	P04	Port 0, bit 4	Input/Output
7	P05	Port 0, bit 5	Input/Output
8	P06	Port 0, bit 6	Input/Output
11	P07	Port 0, bit 7	Input/Output
33	P10	Port 1, bit 0	Input/Output
34	P11	Port 1, bit 1	Input/Output
39	P12	Port 1, bit 2	Input/Output
40	P13	Port 1, bit 3	Input/Output
9	P14	Port 1, bit 4	Input/Output
10	P15	Port 1, bit 5	Input/Output
15	P16	Port 1, bit 6	Input/Output
16	P17	Port 1, bit 7	Input/Output
42	P20	Port 2, bit 0	Input/Output
43	P21	Port 2, bit 1	Input/Output
44	P22	Port 2, bit 2	Input/Output



**Table 10. 48-Pin SSOP Functional Pin Identification (Continued)**

Pin No	Symbol	Function	Signal Direction
45	P23	Port 2, bit 3	Input/Output
46	P24	Port 2, bit 4	Input/Output
2	P25	Port 2, bit 5	Input/Output
3	P26	Port 2, bit 6	Input/Output
4	P27	Port 2, bit 7	Input/Output
29	P30	Port 3, bit 0; connect to VDD if not used	Input
19	P31	Port 3, bit 1	Input
20	P32	Port 3, bit 2	Input
21	P33	Port 3, bit 3	Input
22	P34	Port 3, bit 4	Input/Output
26	P35	Port 3, bit 5	Output
28	P36	Port 3, bit 6	Output
27	P37	Port 3, bit 7	Output
1	P40	Port 4, bit 0	Input/Output
6	P41	Port 4, bit 1	Input/Output
14	P42	Port 4, bit 2	Input/Output
23	P43	Port 4, bit 3	Input/Output
30	P44	Port 4, bit 4	Input/Output
36	P45	Port 4, bit 5	Input/Output
47	P46	Port 4, bit 6	Input/Output
48	P47	Port 4, bit 7	Input/Output
12	V <sub>DD</sub>	Power Supply	Input
13	V <sub>DD</sub>	Power Supply	Input
24	V <sub>SS</sub>	Ground	Input
37	V <sub>SS</sub>	Ground	Input
38	V <sub>SS</sub>	Ground	Input
18	XTAL1	Crystal oscillator	Input
17	XTAL2	Crystal oscillator	Output
25	/RESET	Bidirectional reset signal	Input/Output

# I/O Port Pin Functions

The ZLF645 MCU features up to five 8-bit ports which are described below:

1. Port 0 is nibble-programmable as either input or output.
2. Port 1 is byte-programmable as either input or output.
3. Port 2 is bit-programmable as either input or output.
4. Port 3 features four inputs on the lower nibble and four outputs on the upper nibble.
5. Port 4 is bit-programmable as either input or output.

► **Note:** *Port 0, Port 1, Port 2, and Port 4 internal pull-ups are disabled on any pin or group of pins when programmed into output mode.*



**Caution:** *The CMOS input buffer for each Port 0, Port 1, Port 2, or Port 4 pin are always connected to the pin, even when the pin is configured as an output. If the pin is configured as an open-drain output and no external signal is applied, a High output state can cause the CMOS input buffer to float. This may lead to excessive leakage current of more than 100  $\mu$ A. To prevent this leakage, connect the pin to an external signal with a defined logic level or ensure that its output state is Low, especially during STOP mode.*

*Port 0, Port 1, Port 2, and Port 4 have both input and output capability. The input logic is always present no matter whether the port is configured as input or output. When executing a READ instruction, the MCU reads the actual value at the input logic but not from the output buffer. In addition, the instructions of OR, AND, and XOR have the Read-Modify-Write sequence. The MCU first reads the port, then modifies the value, and loads back to the port.*

*Precaution must be taken, if the port is configured as an open-drain output or if the port is driving any circuit that makes the voltage different from the appropriate output logic. If it is configured as open-drain output with output logic as ONE, it is a floating port and reads back as ZERO. The following instruction sets P00–P07 all Low:*

```
AND P0, #%F0
```

## RESET (Input, Active Low)

Reset initializes the MCU and is accomplished either through Power-On Reset (POR), [Watchdog Timer](#) (WDT), Stop Mode Recovery, Low-Voltage detection, or through the external reset pin in the case of 48-pin packaged products.

During POR and WDT Reset, the internally generated reset drives the reset pin Low for the POR time. Any device driving the external reset line must be open-drain to avoid damage from a possible conflict during reset conditions. A pull-up is provided internally for the reset pin, if available. When the ZLF645 MCU asserts (Low) the RESET pin, the internal pull-up is disabled. The ZLF645 MCU does not assert the RESET pin when the VDD voltage is below the VBO trip point level (for more details, see [Reset and Power Management](#) on page 132).

► **Note:** *The external reset does not initiate an exit from STOP mode.*

[Table 11](#) lists the registers used to control I/O ports. Some port pin functions can also be affected by control registers for other peripheral functions.

**Table 11. I/O Port Control Registers**

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
000	0–3	00	Port 0 Register	P0	XXh	<a href="#">32</a>
001	0–3	01	Port 1 Register	P1	XXh	<a href="#">33</a>
002	0–3	02	Port 2 Register	P2	XXh	<a href="#">35</a>
003	0–3	03	Port 3 Register	P3	0Xh	<a href="#">37</a>
F08	0–3	08	Port 4 Register	P4	XXh	<a href="#">40</a>
F09	F	09	Port 4 Mode Register	P4M	FFh	<a href="#">39</a>
0F6	All	F6	Port 2 Mode Register	P2M	FFh	<a href="#">34</a>
0F7	All	F7	Port 3 Mode Register	P3M	XXXX_X000b	<a href="#">36</a>
0F8	All	F8	Port 0/1 Mode Register	P01M	X1XX_XXX1b	<a href="#">31</a>
F00	F	00	Port Configuration Register	PCON	XXXX_1110b	<a href="#">30</a>

## Port 0

Port 0 is an 8-bit bidirectional CMOS-compatible port. Its eight I/O lines are configured under software control to create a nibble I/O port. The output drivers are push/pull or open-drain, controlled by bit 2 of the [Port Configuration Register](#).

If one or both nibbles are required for I/O operation, they must be configured by writing to the [Port 0/1 Mode Register](#). After a hardware reset or a Stop Mode Recovery, Port 0 is configured as an input port.

Port 0, bit 7 is used as the transmit output of the UART when UART Tx is enabled. The I/O function of Port 0, bit 7 is overridden by the UART serial output (TxD) when UART Tx is enabled ( $UCTL[7] = 1$ ). The pin must be configured as an output for TxD data to reach the pin ( $P01M[6] = 0$ ).

An optional pull-up transistor is available as a user-selectable flash programming option on all Port 0 bits with nibble select. [Figure 6](#) displays the Port 0 configuration.

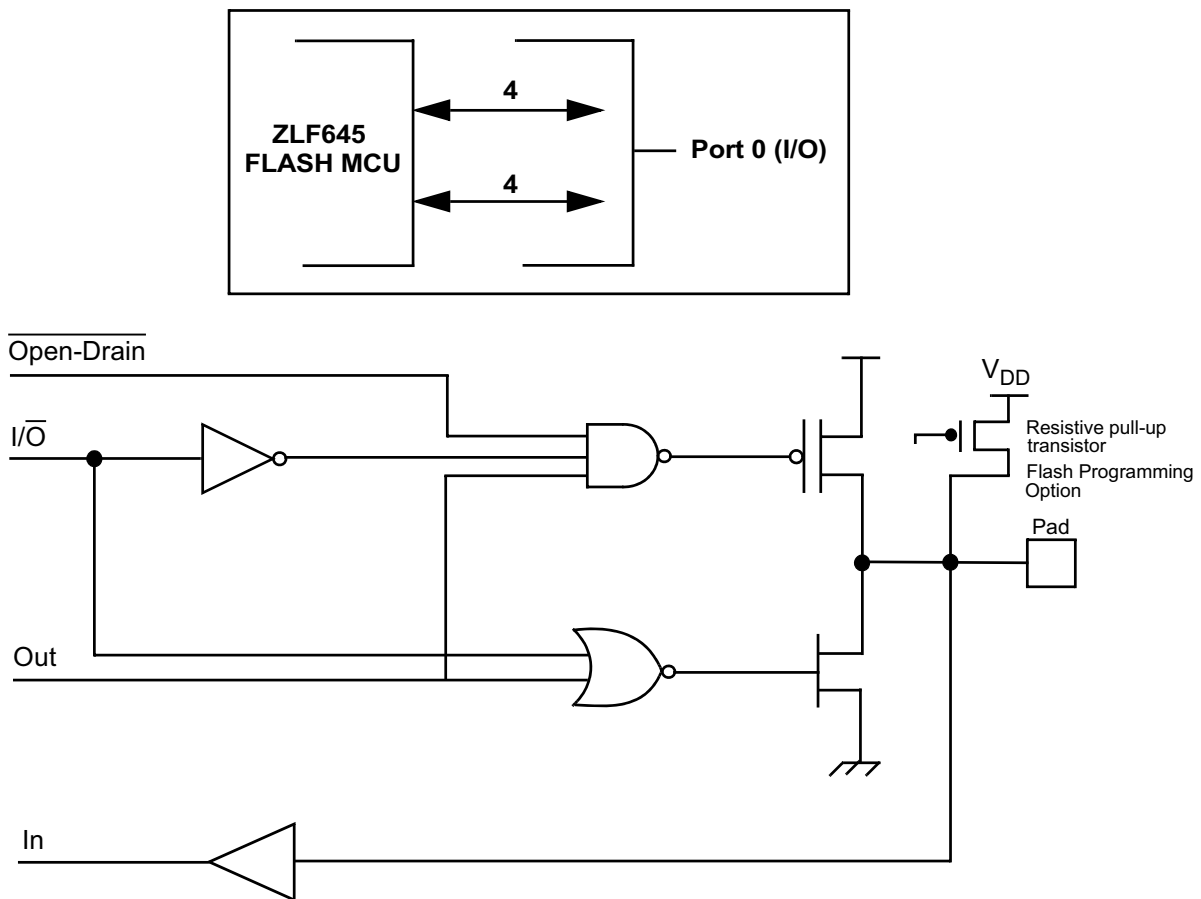


Figure 6. Port 0 Configuration

## Port 1

Port 1 is an 8-bit bidirectional CMOS-compatible I/O port. It can be configured under software control as inputs or outputs. A flash programming option bit is available to connect eight pull-up transistors on this port. Bits programmed as output are globally programmed as either push/pull or open-drain. The power-on reset function resets with the eight bits of Port 1 [P17:10] configured as inputs. [Figure 7](#) displays the Port 1 configuration.

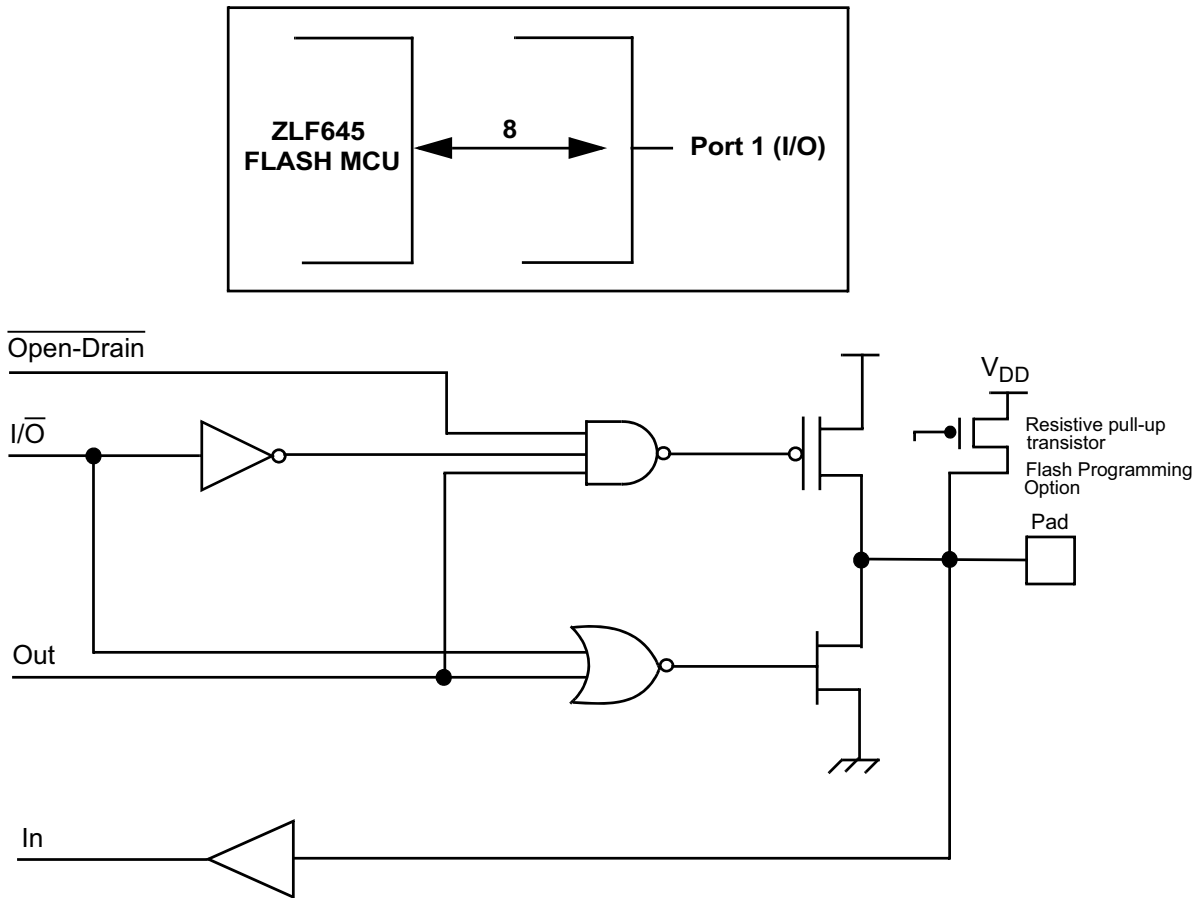


Figure 7. Port 1 Configuration

## Port 2

Port 2 is an 8-bit bidirectional CMOS-compatible I/O port. Its eight I/O lines can be independently configured under software control as inputs or outputs. Port 2 is always available for I/O operation. A flash programming option bit is available to connect eight pull-up transistors on this port. Bits programmed as outputs are globally programmed as either push/pull or open-drain. The Power-On Reset function resets with the eight bits of Port 2 [P27:20] configured as inputs.

Port 2 also has an 8-bit input OR and AND gate and edge detection circuitry, which can be used to recover from the STOP mode. P20 can be programmed to access the edge-detection circuitry in DEMODULATION mode. Figure 8 displays the Port 2 configuration.

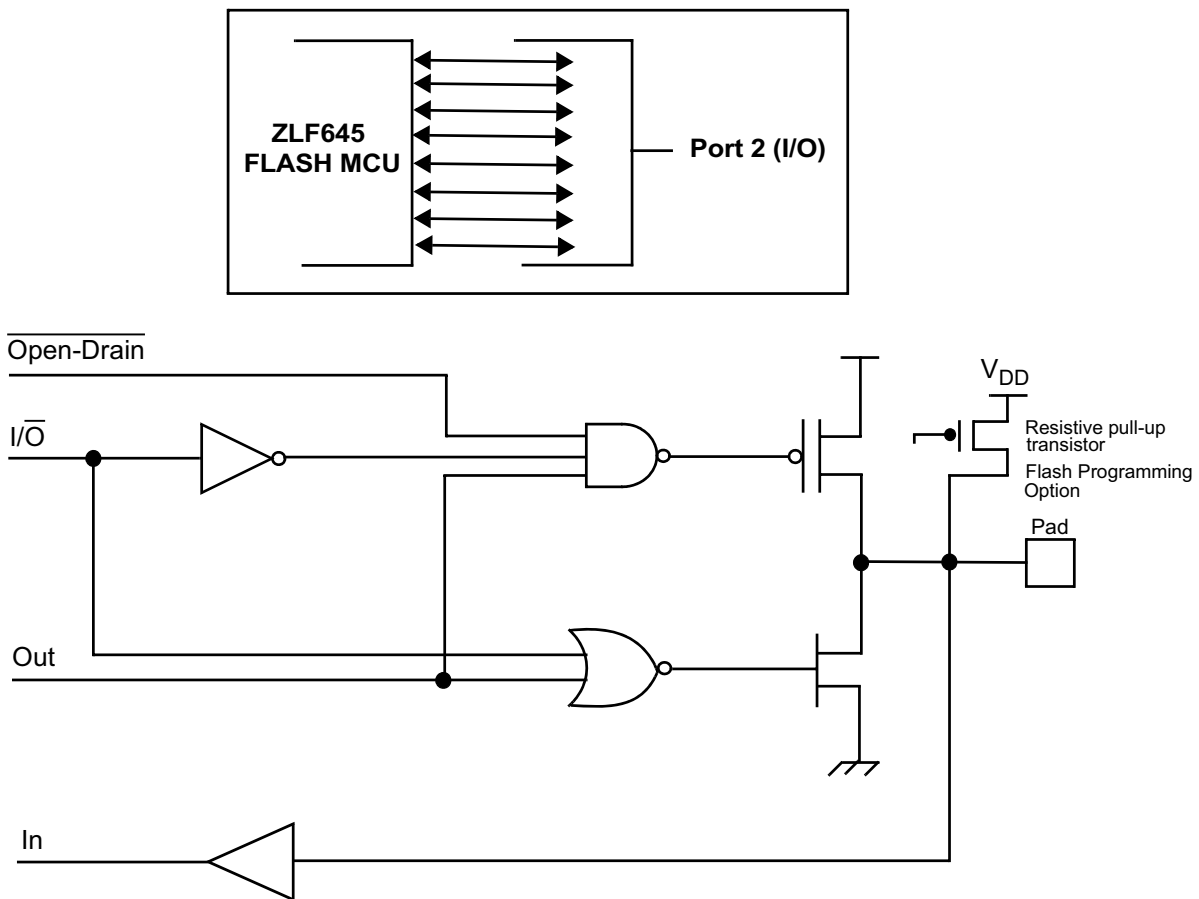


Figure 8. Port 2 Configuration

## Port 3

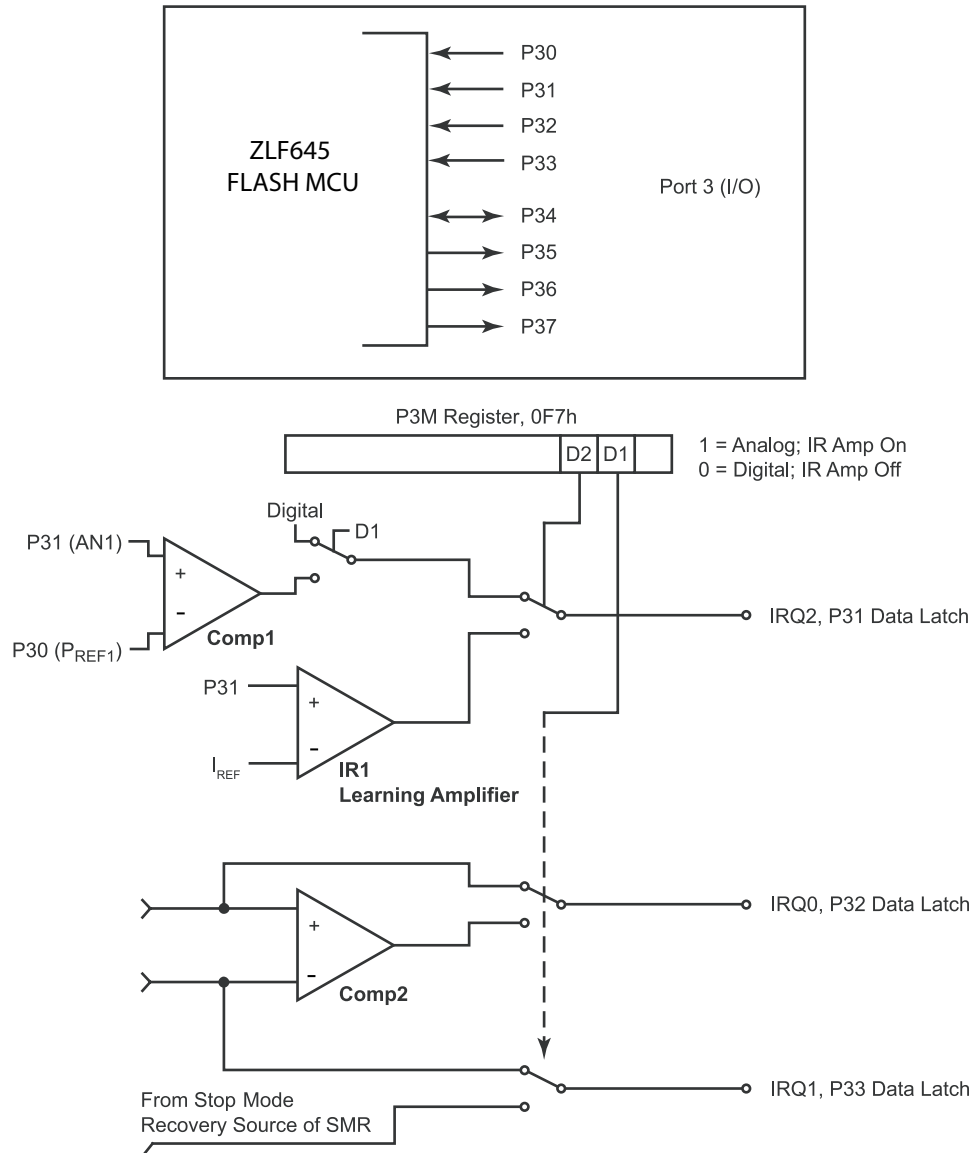
Port 3 is an 8-bit CMOS-compatible I/O port. Port 3 consists of four fixed inputs (P33:P30), three fixed outputs (P37:P36:P35), and one multi-functioned pin (P34) that can function as an output only or as a bidirectional open-drain I/O depending on whether the ZLF645 MCU is in ICP mode.

P30, P31, P32, and P33 are standard CMOS inputs with option enabled pull-up transistors and can be configured under software as interrupts, as received data input to the UART block, as input to comparator circuits, or as input to the IR learning amplifier.

P37, P36, and P35 are push/pull outputs and can be configured as outputs from counter/timers and/or comparator circuits.

During the ZLF645's POR time, P34 is configured as an input pin with pull-up enabled. If after completing its POR period, the ZLF645 has not detected this pin LOW and been put into ICP mode, this pin will revert back to being a push/pull output only. For more details on the function of pin P34, see [ICP Interface](#) on page 53.

Figure 9 displays the Port 3 configuration.



**Figure 9. Port 3 Configuration**

P31 can be used as an interrupt, analog comparator input, infrared learning amplifier input, normal digital input pin, and as a Stop Mode Recovery source. When bit 2 of the Port 3 Mode register (P3M) is set, P31 is used as the infrared learning amplifier, IR1. The reference source for IR1 is GND. The infrared learning amplifier is disabled during STOP mode. When bit 1 of P3M is set, the part is in ANALOG mode and the analog comparator,



COMP1 is used. The reference voltage for COMP1 is P30 ( $P_{REF1}$ ). When in ANALOG mode, P30 cannot be read as a digital input when the CPU reads bit 0 of the Port 3 register; such reads always return a value of 1.

Also, when in ANALOG mode, P31 cannot be used as a Stop Mode Recovery source, as in STOP mode the comparator is disabled and its output will not toggle. The programming of bit 2 of the P3M register takes precedence over the programming of Bit 1 in determining the function of P31. If both bits are set, P31 functions as an IR learning amplifier instead of an analog comparator. As displayed in [Figure 9](#) the output of the function selected for P31 can be used as a source for IRQ2 interrupt assertion. The IRQ2 interrupt can be configured based upon detecting a rising, falling, or edge-triggered input change using Bits 6 and 7 of the IRQ register. The P31 output stage signal also goes to the Counter/Timer edge detection circuitry in the same way that P20 does.

P32 can be used as an interrupt, analog comparator, UART receiver, normal digital input and as Stop Mode Recovery source. When bit 6 of UCTL register is set, P32 functions as a receive input for the UART. When bit 1 of the P3M register is set, thereby placing Port 3 into ANALOG mode, P32 functions as an analog comparator, COMP2. The reference voltage for COMP2 is P33 ( $P_{REF2}$ ). P32 can be used as a rising, falling or edge-triggered interrupt, IRQ0, using IRQ register bits 6 and 7. If UART receiver interrupts are not enabled, the UART receive interrupt is used as the source of interrupts for IRQ0 instead of P32. When in ANALOG mode P32 cannot be used as SMR source because the comparators are turned OFF in STOP mode.

When in ANALOG mode, P33 cannot be read by the CPU as a digital input through bit 3 of the Port 3 register. In this case, a read of bit 3 of the Port 3 register indicates whether Stop Mode Recovery condition exists. Reading a value of 0 indicates an SMR condition; if the ZLF645 MCU is in STOP mode, it will exit STOP mode. Reading a value of 1 indicates that no condition exists to exit the ZLF645 MCU from STOP mode.

Additionally, when in ANALOG mode, P33 cannot be used as an interrupt source. Instead, the existence of a SMR condition can generate an interrupt, if enabled. P33 can be used as a falling-edge interrupt, IRQ1, when not in ANALOG mode. IRQ1 is also used as the UART  $T_X$  interrupt and the UART BRG interrupt. Only one source is active at a time. If bit 7 and bit 5 of UCTL are set to 1, IRQ1 will transmit an interrupt when the Transmit Shift register is empty. If bits 0 and 5 of UCTL are set to 1 and bit 6 of UCTL is cleared to 0, the BRG interrupts will activate IRQ1.

► **Note:** *Comparators and the IR amplifier are powered down by entering STOP mode. For P30:P33 to be used as a Stop Mode Recovery source during STOP mode, these inputs must be placed into DIGITAL mode. When in ANALOG mode, do not configure any Port 3 input as a SMR source. The configuration of these inputs must be re-initialized after Stop Mode Recovery or POR.*

**Table 12. Summary of Port 3 Pin Functions**

Pin	I/O	In-Circuit Programmer	Counter/Timers	Comparator	Interrupt	IRAMP	UART
P30	IN			REF1			
P31	IN		IN	AN1	IRQ2	IR1	
P32	IN			AN2	IRQ0		UART Rx
P33	IN			REF2	IRQ1		
P34	IN/OUT	ICP	T8	AO1		IROUT	
P35	OUT		T16				
P36	OUT		T8/T16				
P37	OUT			AO2			

Port 3 also provides output for each of the counter/timers and AND/OR Logic (see [Figure 10](#)). Control is performed by programming CTR1 bit 5 and bit 4, CTR0 bit 0, and CTR2 bit 0.

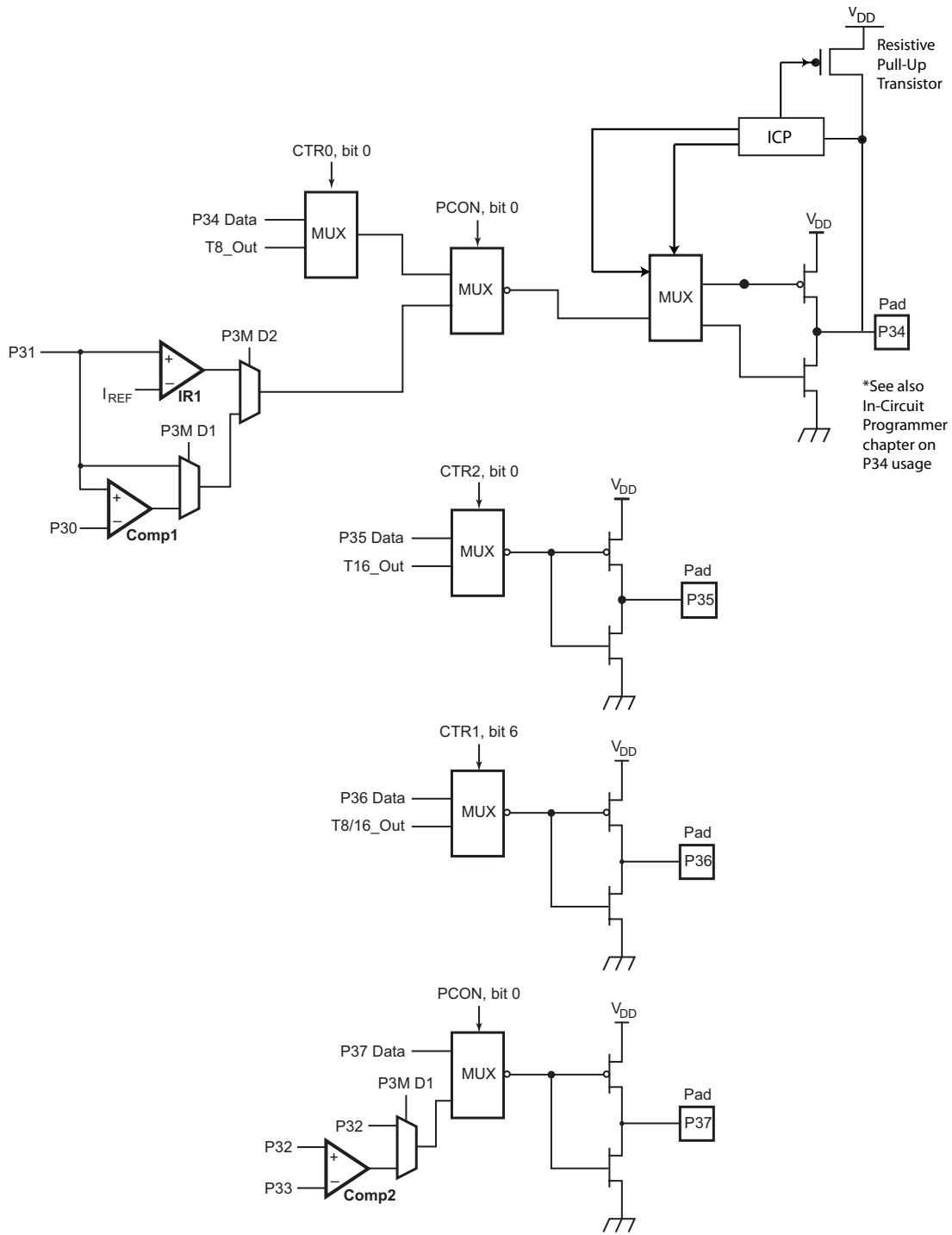


Figure 10. Port 3 Counter/Timer Output Configuration

## Comparator Inputs

In ANALOG mode, P31 and P32 have a comparator front end. The comparator reference is supplied by P33 and P<sub>REF1</sub>. In ANALOG mode, the P33 internal data latch and its corresponding IRQ1 are diverted to the Stop Mode Recovery sources (excluding P31, P32, and P33) as displayed in [Figure 9](#) on page 24. In DIGITAL mode, P33 is used as bit 3 of the Port 3 input register, which then generates IRQ1.

- **Note:** *Comparators are powered down by entering STOP mode. For P30:P33 to be used as an SMR source, these inputs must be placed into DIGITAL mode.*

## Comparator Outputs

The comparators can be programmed to output on P34 and P37 by setting bit 0 of the PCON register.

## Port 4

Port 4 is an 8-bit bidirectional CMOS-compatible I/O port. Its eight I/O lines can be independently configured under software control as inputs or outputs. Port 4 is always available for I/O operation. A flash programming option bit is available to connect eight pull-up transistors on this port. Bits programmed as outputs are globally programmed as either push/pull or open-drain. The POR function resets with the eight bits of Port 4 [P47:40] configured as inputs. [Figure 11](#) on page 29 displays the Port 4 configuration.

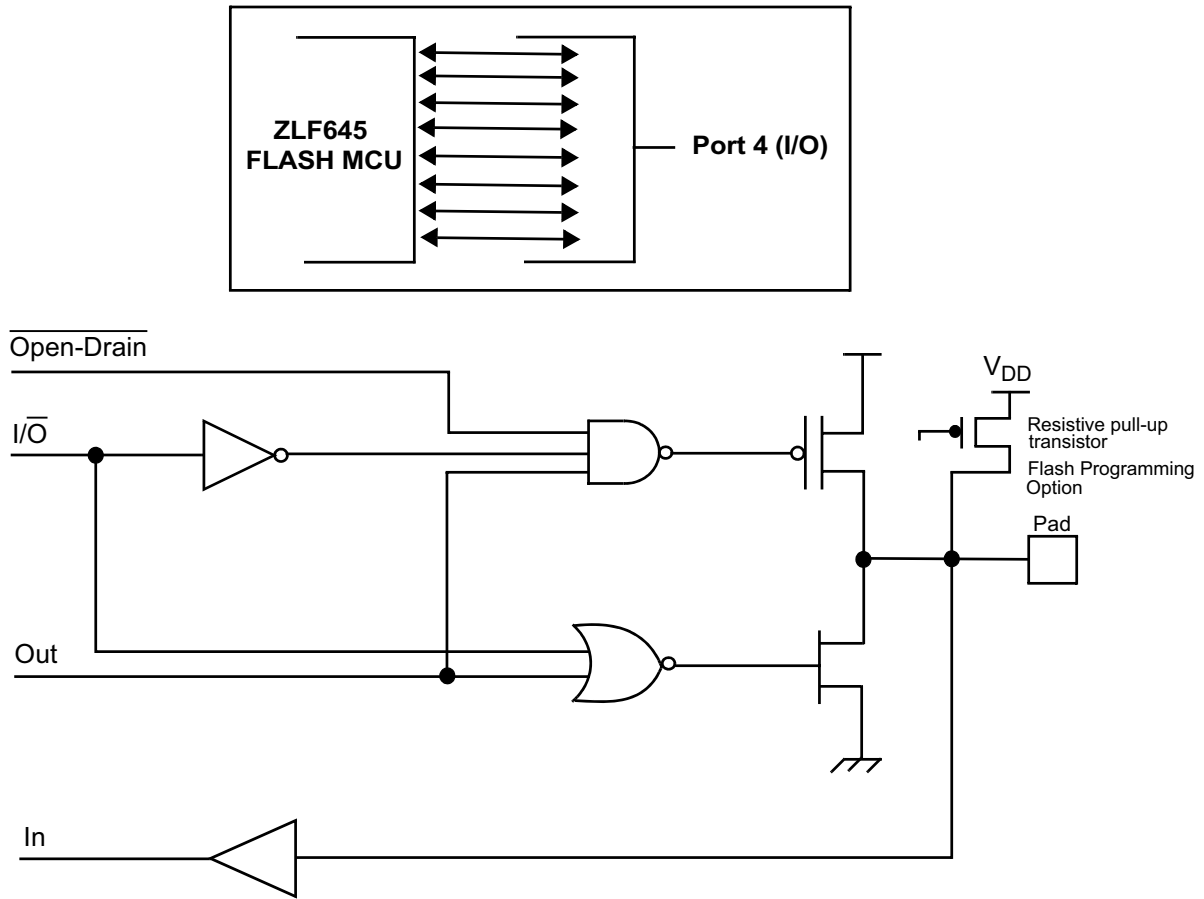


Figure 11. Port 4 Configuration

## Port Configuration Register

The Port Configuration register (see [Table 13](#)) configures the Port 0 output mode and the comparator output on Port 3. The PCON register is located in expanded register Bank F, address 00h.

**Table 13. Port Configuration Register (PCON)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved				Port 4 Output Mode	Port 0 Output Mode	Port 1 Output Mode	Comp/IR Amp Output Port 3
Reset	X	X	X	X	1	1	1	0
R/W	—	—	—	—	W	W	W	W
Address	Bank F: 00h; Linear: F00h							

Bit Position	Value	Description
[7:4]	—	<b>Reserved</b> —Must be written to 1; reads 11111b.
[3]	0 1	<b>Port 4 Output Mode</b> —Controls the output mode of Port 4. Open-drain Push/pull
[2]	0 1	<b>Port 0 Output Mode</b> —Controls the output mode of Port 0. Write only; read returns 1. Open-drain Push/pull
[1]	0 1	<b>Port 1 Output Mode</b> —Controls the output mode of Port 1. Write only; read returns 1 Open-drain Push/pull
[0]	0 1	<b>Comparator or IR Amplifier Output Port 3</b> —Select digital outputs or comparator, and IR amplifier outputs on P34 and P37. Write only; read returns 1. 0 P34 and P37 outputs are digital. 1 P34 is Comparator 1 or IR Amplifier output, P37 is Comparator 2 output.

► **Note:** *PCON register is not reset after a Stop Mode Recovery. Also, for package types other than the 48-pin package, writes to bit 3 and bit 1 have no effect.*

## Port 0/1 Mode Register

The Port 0/1 Mode register (see [Table 14](#)) determines the I/O direction of Port 0 and Port 1. The Port 0 direction is nibble-programmable. Bit 6 controls the upper nibble of Port 0, bits [7:4]. Bit 0 controls the lower nibble of Port 0, bits [3:0]. The Port 1 direction is byte programmable.

**Table 14. Port 0/1 Mode Register (P01M)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved	P07:P04 Mode	Reserved		Port 1 Mode	Reserved		P03:P00 Mode
Reset	X	1	X	X	1	X	X	1
R/W	—	W	—	—	W	—	—	W
Address	Bank Independent: F8h; Linear: 0F8h							

Bit Position	Value	Description
[7]	0	<b>Reserved</b> —Must be written to 1. Reads 1b.
[6]	0 1	<b>P07:P04 Mode</b> Output Input
[5:4]*	—	<b>Reserved</b> —Must be written to 1. Reads 1's.
[3]*	0 1	<b>Port 1 Mode</b> Output Input
[0]	0 1	<b>P00:P03 Mode</b> Output Input

\*For package types other than the 48-pin package, writes to bit 3 have no effect.

► **Note:** Only P00, P01, and P07 are available for ZLF645 Flash MCU 20-pin configuration.

## Port 0 Register

The Port 0 register (see [Table 15](#)) allows read and write access to the Port 0 pins.

**Table 15. Port 0 Register (P0)**

Bit	7	6	5	4	3	2	1	0
Field	P07	P06	P05	P04	P03	P02	P01	P00
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank 0–3: 00h; Linear: 000h							

Bit Position	R/W	Description
[7]	<p>Read</p> <p>0 Pin level is Low.</p> <p>1 Pin level is High.</p> <p>Write</p> <p>0 Assert pin Low.</p> <p>1 Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.</p>	<p><b>Port 0 Pin 7</b>—Available for I/O if UART Tx is disabled.</p>
[6:0]	<p>Read</p> <p>0 Pin level is Low.</p> <p>1 Pin level is High.</p> <p>Write</p> <p>0 Assert pin Low.</p> <p>1 Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.</p>	<p><b>Port 0 Pins 6–0</b>—Each bit provides access to the corresponding Port 0 pin.</p>

► **Note:** *Only P00, P01, and P07 are available for ZLF645 Flash MCU 20-pin configuration.*



## Port 1 Register

The Port 1 register (see [Table 16](#)) allows read and write access to the Port 1 pins.

**Table 16. Port 1 Register (P1)**

Bit	7	6	5	4	3	2	1	0
Field	P17	P16	P15	P14	P13	P12	P11	P10
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank 0–3: 01h; Linear: 001h							
<b>Note:</b> For package types other than the 48-pin package, this register is available as a general-purpose register.								

Bit Position	Value	Description	
[7:0]	Read	<b>Port 1 Pins 7–0</b> —Each bit provides access to the corresponding Port 1 pin.	
		Pin configured as input or output in P01M register.	
	Write	0	Pin level is Low.
		1	Pin level is High.
		0	Pin configured as output in P01M register.
		1	Assert pin Low.
	0	Assert pin High, if configured as push-pull; make pin high-impedance if it is open-drain.	

**Note:** For packages other than 48-pin package, this register is available as general-purpose register.

## Port 2 Mode Register

The Port 2 Mode register (see [Table 17](#)) determines the I/O direction of each bit on Port 2. Bit 0 of the Port 3 Mode register determines whether the output drive is push/pull or open-drain.

**Table 17. Port 2 Mode Register (P2M)**

Bit	7	6	5	4	3	2	1	0
Field	P27 I/O Definition	P26 I/O Definition	P25 I/O Definition	P24 I/O Definition	P23 I/O Definition	P22 I/O Definition	P21 I/O Definition	P20 I/O Definition
Reset	1	1	1	1	1	1	1	1
R/W	W	W	W	W	W	W	W	W
Address	Bank Independent: F6h; Linear: 0F6h							

Bit Position	Value	Description
[7]	0	Defines P27 as output.
	1	Defines P27 as input.
[6]	0	Defines P26 as output.
	1	Defines P26 as input.
[5]	0	Defines P25 as output.
	1	Defines P25 as input.
[4]	0	Defines P24 as output.
	1	Defines P24 as input.
[3]	0	Defines P23 as output.
	1	Defines P23 as input.
[2]	0	Defines P22 as output.
	1	Defines P22 as input.
[1]	0	Defines P21 as output.
	1	Defines P21 as input.
[0]	0	Defines P20 as output.
	1	Defines P20 as input.

► **Note:** *Port 2 Mode register is not reset after a Stop Mode Recovery.*

## Port 2 Register

The Port 2 register (see [Table 18](#)) allows read and write access to the Port 2 pins.

**Table 18. Port 2 Register (P2)**

Bit	7	6	5	4	3	2	1	0
Field	P27	P26	P25	P24	P23	P22	P21	P20
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank 0–3: 02h; Linear: 002h							

Bit Position	Value	Description	
[7:0]	Read	<b>Port 2 Pins 7–0</b> —Each bit provides access to the corresponding Port 2 pin.	
		Pin configured as input or output in P2M register.	
	Write	0	Pin level is Low.
		1	Pin level is High.
		0	Pin configured as output in P2M register.
		1	Assert pin Low.
	1	Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.	

## Port 3 Mode Register

The Port 3 Mode register (see Table 20) is used to configure the functionality of Port 3 inputs and the output mode of Port 2. When bit 2 is set, the IR Learning Amplifier is used instead of the COMP1 comparator, regardless of the value of bit 1.

**Table 20. Port 3 Mode Register (P3M)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved					IR Learning Amplifier	DIGITAL/ ANALOG Mode	Port 2 Open-Drain
Reset	X	X	X	X	X	0	0	0
R/W	—	—	—	—	—	W	W	W
Address	Bank Independent: F7h; Linear: 0F7h							

Bit Position	R/W	Value	Description
[7:3]	—	—	<b>Reserved</b> —Must be written to 1. Reads return 11111b.
[2]	W	0 1	IR Learning Amplifier disabled. IR Learning Amplifier enabled with P31 configured as amplifier input.
[1]	W	0 1	<b>DIGITAL/ANALOG Mode</b> 0 P30, P31, P32, P33 are digital inputs. 1 P30, P32, and P33 are comparator inputs. If P3M[2]=0, P31 also function as a comparator input. If P3M[2]=1, P31 is the IR amplifier input.
[0]	W	0 1	Port 2 open-drain. Port 2 push/pull.

► **Note:** *Port 3 Mode register is not reset after a Stop Mode Recovery.*

## Port 3 Register

The Port 3 register (see [Table 21](#)) allows read access to port pins P33 through P30 and write access to the port pins P37 through P34.

**Table 21. Port 3 Register (P3)**

Bit	7	6	5	4	3	2	1	0
Field	P37	P36	P35	P34	P33	P32	P31	P30
Reset	0	0	0	0	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Banks 0–3: 03h; Linear: 003h							

Bit Position	Value	Description
[7]	Write	<b>Port 3, Pin 7 Output</b> —Writes to this bit do not affect the pin state if write-only register bit PCON[0] is set to 1, which configures P37 as the Comparator 1 or IR Amplifier output. 0 P37 asserted Low if PCON[0]=0. 1 P37 asserted High if PCON[0]=0. A read returns the last value written to this bit.
[6]	Write	<b>Port 3, Pin 6 Output</b> —Writes to this bit do not affect the pin state if register bits CTR1[7:6]=01, which configures P36 as the Timer 8 and Timer 16 combined logic output. 0 P36 asserted Low. 1 P36 asserted High. A read returns the last value written to this bit.
[5]	Write	<b>Port 3, Pin 5 Output</b> —Writes to this bit do not affect the pin state if register bit CTR2[0]=1, which configures P35 as the Timer 16 output. 0 P35 asserted Low. 1 P35 asserted High. A read returns the last value written to this bit.
[4]	Write	<b>Port 3, Pin 4 Output</b> —Writes to this bit do not affect the pin state if write only register bit PCON[0]=1 which configures P34 as a Comparator 2 output, register bit CTR0[0]=1 which configures P34 as Timer 8 output, or if the device is in ICP mode as described in the <a href="#">ICP Interface</a> on page 53. 0 P34 asserted Low. 1 P34 asserted High. A read returns the last value written to this bit.

Bit Position	Value	Description
[3]	Read	<b>Port 3, Pin 3 Input</b> —Writing this bit has no effect.
	0	If P3M[1]=0: P33 is Low.
	1	P33 is High.
	0	If P3M[1]=1 or SMR4[4]=1: SMR condition exists.
	1	SMR condition does not exist.
[2]	Read	<b>Port 3, Pin 2 Input</b> —Writing this bit has no effect.
	0	If P3M[1]=0: P32 input is Low.
	1	P32 input is High.
	0	If P3M[1]=1: Comparator 2 output is Low.
	1	Comparator 2 output is High.
[1]	Read	<b>Port 3, Pin 1 Input</b> —Writing this bit has no effect.
	0	If P3M[2:1]=00: P31 input is Low.
	1	P31 input is High.
	0	If P3M[2:1]=01: Comparator 1 output is Low.
	1	Comparator 1 output is High.
	0	If P3M[2:1]=10 or 11: IR amplifier output is Low.
	1	IR amplifier output is High.
[0]	Read	<b>Port 3, Pin 0 Input</b> —Writing this bit has no effect.
	0	If P3M[1]=0: P30 input is Low.
	1	P30 input is High.
	1	If P3M[1]=1: Reads as 1.

► **Note:** *Port 3 register is not reset after a Stop Mode Recovery.*

## Port 4 Mode Register

The Port 4 Mode register (see [Table 22](#)) determines the I/O direction of each bit on Port 4. Bit 3 of the Port Configuration register (PCON) determines whether the output drive is push/pull or open-drain.

**Table 22. Port 4 Mode Register (P4M)**

Bit	7	6	5	4	3	2	1	0
Field	P47 I/O Definition	P46 I/O Definition	P45 I/O Definition	P44 I/O Definition	P43 I/O Definition	P42 I/O Definition	P41 I/O Definition	P40 I/O Definition
Reset	1	1	1	1	1	1	1	1
R/W	W	W	W	W	W	W	W	W
Address	Bank F: 09h; Linear: F09h							

Bit Position	Value	Description
[7]	0	Defines P47 as output.
	1	Defines P47 as input.
[6]	0	Defines P46 as output.
	1	Defines P46 as input.
[5]	0	Defines P45 as output.
	1	Defines P45 as input.
[4]	0	Defines P44 as output.
	1	Defines P44 as input.
[3]	0	Defines P43 as output.
	1	Defines P43 as input.
[2]	0	Defines P42 as output.
	1	Defines P42 as input.
[1]	0	Defines P41 as output.
	1	Defines P41 as input.
[0]	0	Defines P40 as output.
	1	Defines P40 as input.

► **Note:** *Port 4 Mode register is not reset after a Stop Mode Recovery.*

## Port 4 Register

The Port 4 register (see [Table 23](#)) allows read and write access to the Port 4 pins.

**Table 23. Port 4 Register (P4)**

Bit	7	6	5	4	3	2	1	0
Field	P47	P46	P45	P44	P43	P42	P41	P40
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Banks 0-3: 04h; Linear: 004h							

Bit Position	Value	Description
[7:0]		<b>Port 4 Pins 7–0</b> —Each bit provides access to the corresponding Port 4 pin.
	Read	Pin configured as input or output in P4M register.
	0	Pin level is Low.
	1	Pin level is High.
	Write	Pin configured as output in P4M register.
	0	Assert pin Low.
	1	Assert pin High if configured as push-pull; make pin high-impedance if it is open-drain.



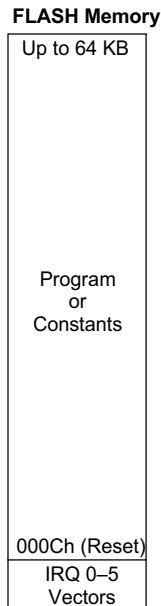
# Memory and Registers

The Z8<sup>®</sup> LXMC CPU used in the ZLF645 Series of Flash MCUs incorporates special features to extend the available memory space while maintaining the benefits of a Z8<sup>®</sup> CPU core in battery-operated applications.

## Flash Program/Constant Memory

The ZLF645 Series of Flash MCUs can address up to 64 KB of Flash memory for object code (program instructions and immediate data) and constant data (ROM tables and data constants). The first 12 bytes of the memory are reserved for the six available 16-bit interrupt request (IRQ) vectors. On reset, program execution begins at address 000Ch in the memory. Execution rolls over to the beginning of the memory if the program counter address exceeds the Flash memory size.

The entire Flash memory is available for either program code or constant data. Outside of normal instruction fetches, the CPU can access the Flash memory by using LDC and LDCI instructions. The LDC and LDCI instructions use 16-bit addresses to access the memory. [Figure 12](#) displays the Program/Constant memory map for the device.



0000h = 16-bit Address

(Not to Scale)

**Figure 12. Program/Constant Memory Map**

## Register File

The ZLF645 Series of Flash MCUs features up to 1024 bytes of register file space, organized in 256-byte banks. Bank 0 contains 235 or 237 bytes of RAM addressed as general purpose registers, 5 or 3 port addresses, and 16 control register addresses. For 20- or 28-pin packages, Port 1 and Port 4 registers of Bank 0 are not implemented and these locations are available as general-purpose registers. Bank 1, Bank 2, and Bank 3; each contain 256 general-purpose register bytes. Bank D and Bank F; each contain 16 addresses for control registers. All other banks are reserved and must not be selected.

The current bank is selected for 8-bit direct or indirect addressing by writing Register Pointer bits RP[3:0]. In the current bank, a 16-byte working register group (addressed as R0–R15) is selected by writing RP[7:4]. A working register operand requires only 4 bits of Program Memory. There are 16 working register groups per bank (see [Figure 13](#) and [Figure 14](#)).

The 8-bit addresses in the range F0h–FFh (and the equivalent 4-bit addresses) are bank-independent, meaning they always access the control registers in Bank 0, regardless of the RP[3:0] value. Addresses in the range 00h–03h always access the Bank 0 Port registers unless Bank D or Bank F is selected (Port 01h is not implemented in this device). When Bank D or Bank F is selected, addresses 10h–EFh access the Bank 0 general-purpose registers.

The LDX and LDXI instructions or indirect addressing is used to access the Bank 1–3 registers not accessible by 8-bit or working register addresses (12-bit addresses—100h–103h, 1F0h–1FFh, 200h–203h, 2F0h–2FFh, 300h–303h, and 3F0h–3FFh). See [Linear Memory Addressing](#) on page 45.

## Stack

The Stack Pointer register provides either 16-bit or 8-bit of stack pointer addressability depending upon the programming of bit 3 of User Option Byte 1 (for more details, see [Flash Option Bits](#) on page 166).

### 16-bit Stack Addressability

When programmed for 16-bit stack addressability, the stack address is formed as a combination of the SPL and SPH registers located at addresses FFh and FEh. For 1K and 512 B RAM products, the most significant 6 or 5 bits, respectively of the SPH register are ignored. The stack address is mapped to a particular RAM memory location by the following formula:

$$\text{Bank} = \{2'b0, \text{SPH}[1:0]\}$$

$$\text{Group} = \text{SPL}[7:4]$$

$$\text{Register number} = \text{SPL}[3:0]$$

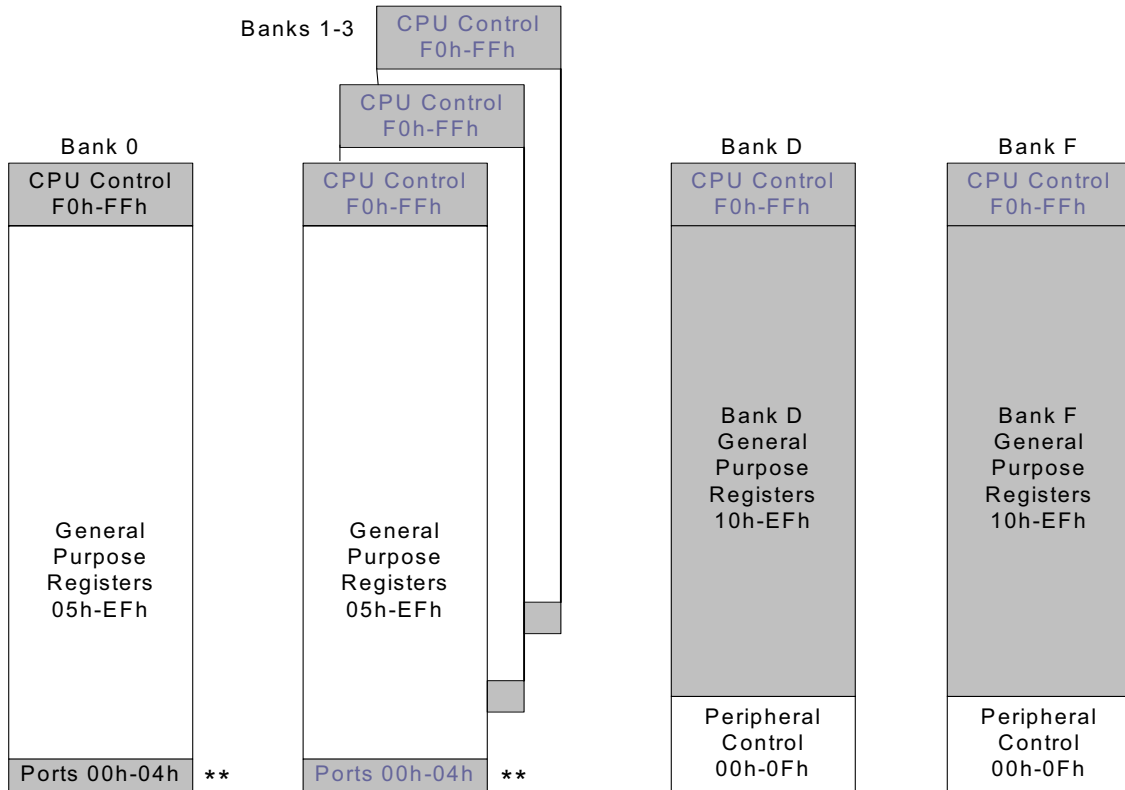
With the ZLF645 MCU configured for 16-bit stack addressability, stack reads or writes to Bank 3, 2, 1, or 0 Group F Registers or to any of the Port registers actually accesses

shadow registers implemented within the RAM memory. This enables the entire 1K or 512 B, depending on the product, of the RAM memory to be used for the stack.

**8-bit Stack Addressability**

For 8-bit stack addressability, only the SPL register is used for stack addressing and stack operations that use the stack pointer always address Bank 0, independent of the RP[3:0] setting. For more details on the stack, refer to *Z8<sup>®</sup> LXMC CPU Core User Manual (UM0215)*.

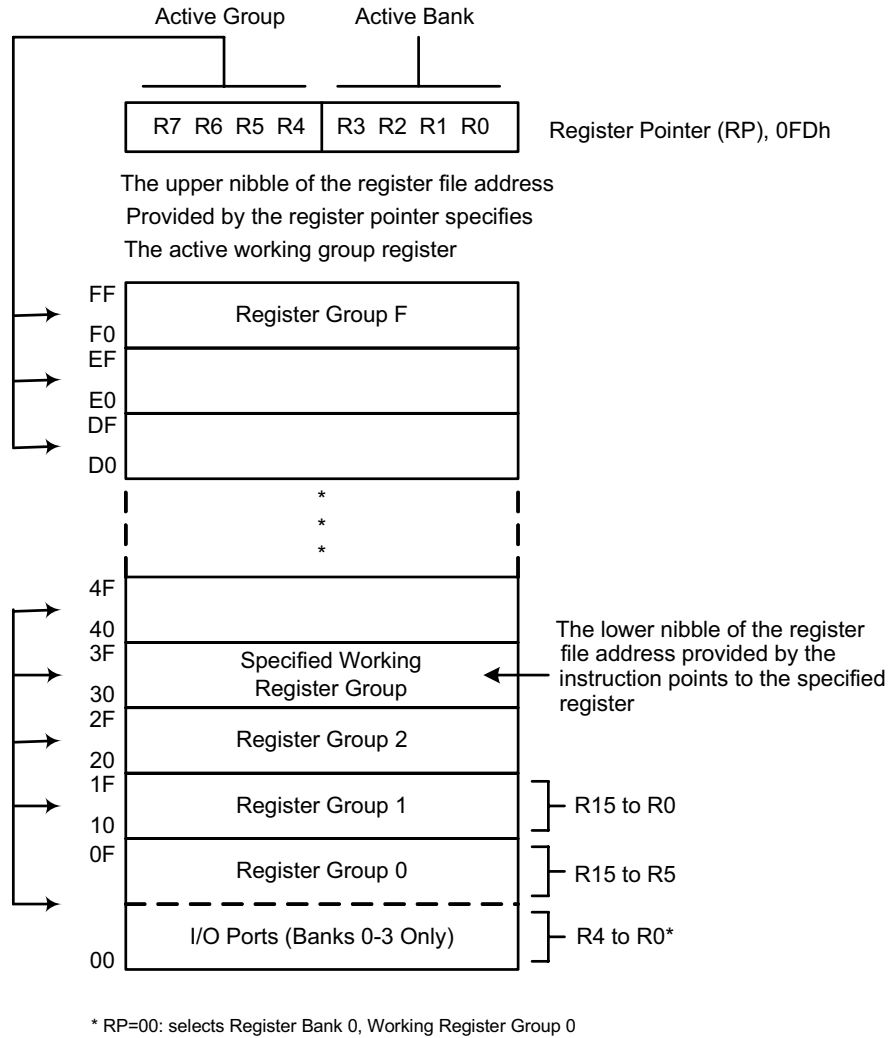
When in 8-bit stack addressability mode, the Bank 0 register FEh can be used to store user data. See [Stack Pointer Register](#) on page 48.



Grey shaded area = Bank-Independent Address (Always Accesses Bank 0)

\*\* For 20 and 28 pin parts, the Port01 and Port04 locations become available for use as general purpose registers

**Figure 13. Register File 8-Bit Banked Address Map**



**Figure 14. Register Pointer—Detail**

## Register Pointer Example

```
R253 RP = 00h
R0 = Port 0
R1 = Port 1
R2 = Port 2
R3 = Port 3
R4 = Port 4
```

But if:

```
R253 RP = 0Dh
R0 = CTR0
R1 = CTR1
R2 = CTR2
R3 = CTR3
R4 = TC8L
```

The counter/timers are mapped into ERF Group D. Access is easily performed using the following code segment:

```
LD RP, #0Dh      ; Select ERF D for access to Bank D
                  ; (working register group 0)
LD R0, #xx       ; load CTR0
LD 1, #xx        ; load CTR1
LD R1, 2         ; CTR2 → CTR1

LD RP, #7Dh      ; Select Expanded Register Bank D and working
                  ; register group 7 of Bank 0 for access.
LD 71h, 2        ; CTR2 → register 71h
LD R1, 2         ; CTR2 → register 71h
```

## Linear Memory Addressing

In addition to using the RP register to designate a bank and working register group for 8-bit or 4-bit addressing, programs can use 12-bit linear addressing to load a register in any other bank to or from a register in the current bank. Linear addressing is implemented through the LDX and LDXI instructions only. Linear addressing treats the register file as if all the registers are logically ordered end-to-end, as opposed to being grouped into banks and working register groups, as displayed in [Figure 15](#) on page 47. For linear addressing, register file addresses are numbered sequentially from Bank 0, register 00h to Bank 0, register FFh, then continuing with Bank 1, register 00h, and so on up to Bank F, register FFh.

Using the LDX and/or the LDXI instructions, either the target or destination register location can be addressed through a 12-bit linear address value stored in a general-purpose register pair.

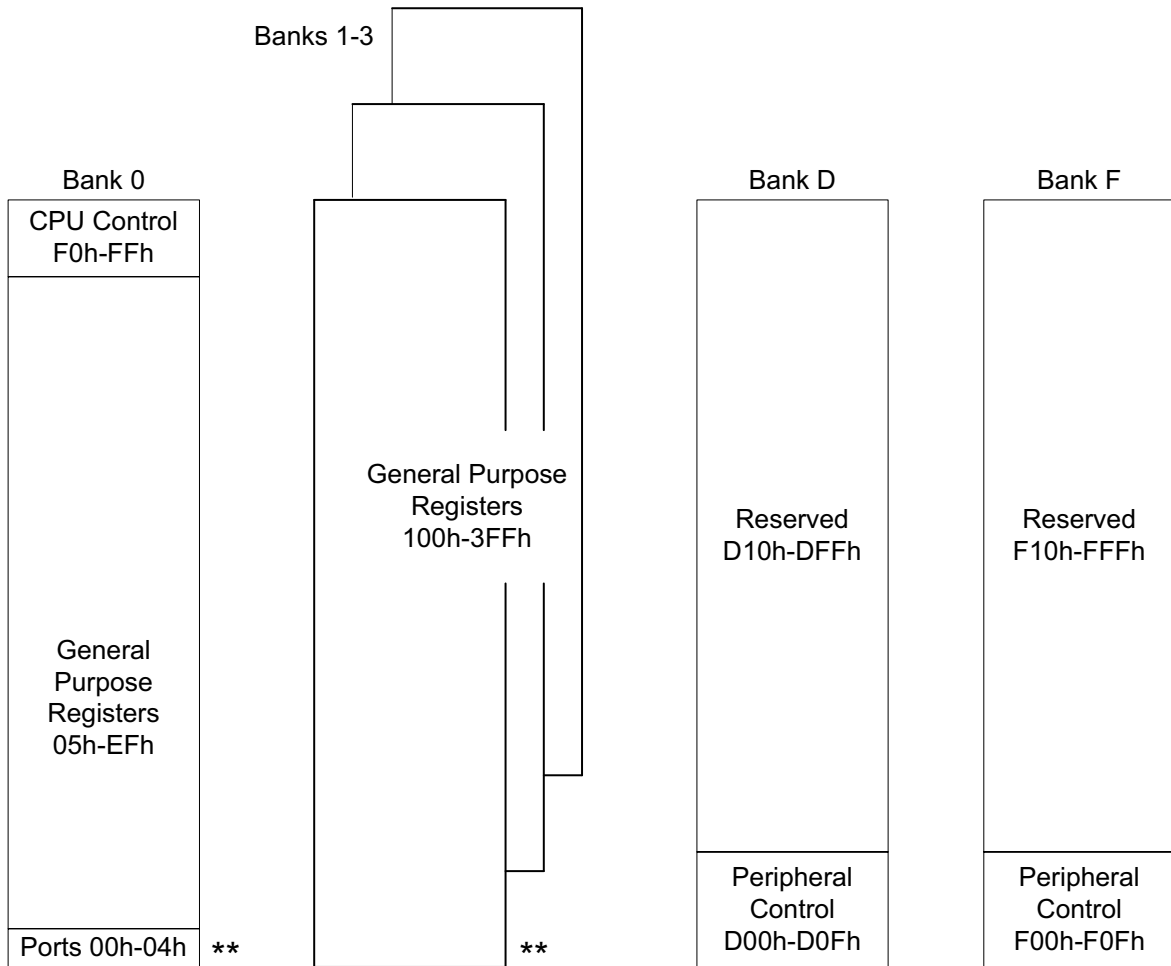
### Example

For example, the following code uses linear addressing for the source of a register transfer operation and uses a working register address for the target:

```
SRP #%23          ;Set working register group 2 in bank 3
LD R0, #%55       ;Load 55 into working register R0 in the current
                  ;group and bank (linear address 320h)
SRP #%12          ;Set working register group 1 in bank 2
LD R6, #%03       ;Load high byte of source linear address (0320h)
LD R7, #%20       ;Load low byte of source linear address (0320h)
LD R0, @RR6       ;Load linear address 320h contents (55h) into
                  ;working register R0 in the current group and
                  ;bank (linear address 210h)
```

In the above code, the source register is referred through a linear address value contained within registers R6 and R7, whereas the destination is referenced via the SRP setting and a working register. For more information about instructions on the usage of LDX and LDXI instructions, refer to Z8<sup>®</sup> LXMC CPU Core User Manual (UM0215).

- **Note:** *The LDE and LDEI instructions that existed in the Z8 CPU are no longer valid; they have been replaced by the LDX and LDXI instructions.*



\*\* For 20 and 28 pin parts, the Port01 and Port04 locations become available for use as general purpose registers

**Figure 15. Register File LDX, LDXI Linear 12-Bit Address Map**



## Register Pointer Register

The upper nibble of the Register Pointer register (see [Table 24](#)) selects which working register group is accessed. A working register group consists of 16 bytes. The lower nibble selects the expanded register file bank; for ZLF645 MCU, Banks 0, 1, 2, 3, F, and D are implemented. A 0h in the lower nibble allows the normal register file (Bank 0) to be addressed. Any other value from 01h to 0Fh exchanges the lower 16 registers to an expanded register bank.

**Table 24. Register Pointer Register (RP)**

Bit	7	6	5	4	3	2	1	0
Field	Working Register Group Pointer				Register Bank Pointer			
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FDh; Linear: 0FDh							

Bit Position	Value	Description
[7:4]	0h–Fh	<b>Working Register Group Pointer</b> Determines which 16-byte working group is addressed.
[3:0]	0h–Fh	<b>Register Bank Pointer</b> Determines which bank is active.

## Stack Pointer Register

Through a Flash programmable option bit, the Stack Pointer register of the ZLF645 MCU is either one or two bytes providing either 8-bit or 16-bit of stack addressing. When not enabled through the option bit for 16-bit stack addressability, the SPH register can be used as a User Data register (USER). The stack pointer resides in the RAM and when the ZLF645 MCU is programmed for 8-bit addressing, this stack pointer resides in Bank 0 of the RAM only. With 16-bit addressing, the entire RAM’s address space is available for use as the stack.

The stack address is decremented prior to a PUSH operation and incremented after a POP operation. The stack address always points to the data stored at the ‘top’ of the stack (the lowest stack address). During a call instruction, the contents of the Program Counter are saved on the stack. Interrupts cause the contents of the Program Counter and Flags registers to be saved on the stack. An overflow or underflow can occur when the stack address is incremented or decremented during normal operations. You must prevent this occurrence or unpredictable operations may result (see [Table 25](#) on page 49).



**Table 25. Stack Pointer Register Low Byte (SPL)**

Bit	7	6	5	4	3	2	1	0
Field	Stack Pointer							
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FFh; Linear: 0FFh							

Bit Position	Value	Description
--------------	-------	-------------

[7:0]	00-FF	Stack Pointer
-------	-------	---------------

**Table 26. Stack Pointer Register High Byte (SPH) or User Data Register (USER)**

Bit	7	6	5	4	3	2	1	0
Field	Stack Pointer							
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FEh; Linear: 0FEh							

- **Notes:**
1. For devices with 1K bytes of RAM and with 16-bit stack pointer mode enabled, the upper 6 bits of this register are unused for stack addressing. For devices with 512 bytes of RAM and with 16-bit stack pointer mode enabled, the upper 7 bits of this register are unused for stack addressing.
  2. When ZLF645 MCU is not in 16-bit stack pointer mode, this register is available to store use user data and its functionality is identical to other Zilog® Crimson products such as the ZLP12840 and ZLR64400 MCUs. When available for user data, this register must not be used as a counter for the DJNZ instruction.

# Register File Summary

Table 27 lists each linear (12-bit) register file address to the associated register, mnemonic, and reset value. The table also lists the register bank (or banks) and corresponding 8-bit address (if any) for each register and a page link to the detailed register table.

Throughout this document, an ‘X’ denotes an undefined digit. A ‘—’ (dash) in a table cell indicates that the corresponding attribute does not apply to the listed item. Reset value digits (highlighted in grey) are not reset by a Stop Mode Recovery. Register bit SMR[7] (shown in **boldface**) is set to 1 instead of reset by a Stop Mode recovery.

**Table 27. Register File Address Summary**

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
000	0–3	00	Port 0 Register	P0	XXh	<a href="#">32</a>
001	0–3	01	Port 1 Register	P1	XXh	<a href="#">33</a>
002	0–3	02	Port 2 Register	P2	XXh	<a href="#">35</a>
003	0–3	03	Port 3 Register	P3	0Xh	<a href="#">37</a>
004	0–3	04	Port 4 Register	P4	XXh	<a href="#">37</a>
005–00F	0	05–0F	General-Purpose Registers (Bank 0 Only)	—	XXh	—
010–0EF	0,D,F	10–EF	General-Purpose Registers (Banks 0, D, F)	—	XXh	—
0F0	All	F0	Reserved	—	—	—
0F1	All	F1	UART Receive/Transmit Data Register	URDATA/ UTDATA	XXh	<a href="#">89</a>
0F2	All	F2	UART Status Register	UST	0000_0010b	<a href="#">89</a>
0F3	All	F3	UART Control Register	UCTL	00h	<a href="#">91</a>
0F4	All	F4	UART Baud Rate Generator Constant Register	BCNST	FFh	<a href="#">92</a>
0F5	All	F5	Reserved	—	—	—
0F6	All	F6	Port 2 Mode Register	P2M	FFh	<a href="#">34</a>
0F7	All	F7	Port 3 Mode Register	P3M	XXXX_X000b	<a href="#">36</a>
0F8	All	F8	Port 0/1 Mode Register	P01M	X1XX_1XX1b	<a href="#">31</a>



**Table 27. Register File Address Summary (Continued)**

Address (Hex)						Page No
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	
0F9	All	F9	Interrupt Priority Register	IPR	XXh	124
0FA	All	FA	Interrupt Request Register	IRQ	00h	125
0FB	All	FB	Interrupt Mask Register	IMR	0XXX_XXXXb	127
0FC	All	FC	Flags Register	FLAGS	XXh	155
0FD	All	FD	Register Pointer Register	RP	00h	48
0FE	All	FE	User Data Register/Stack Pointer Register High Byte <sup>1</sup>	USER/SPH	XXh	49
0FF	All	FF	Stack Pointer Register Low Byte	SPL	XXh	49
100–103	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
104–1EF	1	04–EF	General-Purpose Registers	—	XXh	—
1F0–203	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
204–2EF	2	04–EF	General-Purpose Registers	—	XXh	—
2F0–303	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
304–3EF	3	04–EF	General-Purpose Registers	—	XXh	—
3F0–3FF	—	—	General-Purpose Registers (12-Bit Only)	—	XXh	—
400–CFF	—	—	Reserved	—	—	—
D00	D	00	Counter/Timer 8 Control Register	CTR0	0000_0000b	113
D01	D	01	Timer 8 and Timer 16 Common Functions Register	CTR1	0000_0000b	115
D02	D	02	Counter/Timer 16 Control Register	CTR2	0000_0000b	118
D03	D	03	Timer 8/Timer 16 Control Register	CTR3	0000_0XXXb	119
D04	D	04	Counter/Timer 8 Low Hold Register	TC8L	00h	112
D05	D	05	Counter/Timer 8 High Hold Register	TC8H	00h	111
D06	D	06	Counter/Timer 16 Low Hold Register	TC16L	00h	111
D07	D	07	Counter/Timer 16 High Hold Register	TC16H	00h	110
D08	D	08	Timer 16 Capture Low Register	LO16	00h	110

**Table 27. Register File Address Summary (Continued)**

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
D09	D	09	Timer 16 Capture High Register	HI16	00h	109
D0A	D	0A	Timer 8 Capture Low Register	LO8	00h	109
D0B	D	0B	Timer 8 Capture High Register	HI8	00h	108
D0C	D	0C	Low-Voltage Detection Register	LVD	1 1 1 1_1 0 0 0 b	135
D0D	—	—	Reserved	—	—	—
D0E	D	0E	User Option Byte 0	OPT0	FFh	167
D0F	D	0F	User Option Byte 1	OPT1	FFh	169
D10–DFF	—	—	Reserved (8-Bit access goes to Bank 0)	—	—	—
F00	F	00	Port Configuration Register	PCON	XXXX_1 1 1 0 b	30
F01	F	01	Flash Control and Flash Status Register	FCTL/FSTAT	0 0 0 0_0 0 0 0 b	71/72
F02	F	02	Flash Page Select and Sector Protect Register	FPS/FSEC	0 0 0 0_0 0 0 0 b	73/73
F03	F	03	Flash Frequency High Byte Register	FFREQH	0 0 0 0_0 0 0 0 b	74
F04	F	04	Flash Frequency Low Byte Register	FFREQL	0 0 0 0_0 0 0 0 b	75
F05–F08	—	—	Reserved	—	—	—
F09	F	09	Port 4 Mode Register	P4M	FFh	39
F0A	F	0A	Stop Mode Recovery Register 4	SMR4	XXX0_0 0 0 0 b	151
F0B	F	0B	Stop Mode Recovery Register	SMR	0 0 1 0_0 0 0 0 b	141
F0C	F	0C	Stop Mode Recovery Register 1	SMR1	00h	145
F0D	F	0D	Stop Mode Recovery Register 2	SMR2	X0X0_0 0 XXb	147
F0E	F	0E	Stop Mode Recovery Register 3	SMR3	X0h	150
F0F	F	0F	Watchdog Timer Mode Register	WDTMR	0 0 0 0_1 1 0 1 b	137
F10–FFF	—	—	Reserved (8-Bit access goes to Bank 0)	—	—	—

<sup>1</sup>When ZLF645 is programmed for 16-bit stack addressability, the value in this register is used as the high byte of a 16-bit stack pointer.

# ICP Interface

The ICP interface of the ZLF645 is a single pin RS-232 like interface for performing programming, reads, and memory erasures to the ZLF645's Flash memory. For enabling the ZLF645 into ICP mode and for performing ICP operations, the ZLF645's P34 pin which normally functions as an output only is used.

## Enabling ICP Mode

As mentioned previously, the ZLF645's GPIO pin P34 is multi-functioned to be used for putting the ZLF645 into ICP mode and for ICP communications once it is in that mode. Entry into ICP mode takes place during the ZLF645's power on reset period. During the ZLF645's power on reset period, the P34 pin which normally is an output only pin is configured by the ZLF645 as an input with pull-up enabled. If during this time this pin is driven LOW and held LOW until the end of the power on reset period, the ZLF645 will be put into ICP mode. Once in ICP mode, the P34 pin operates as an open-drain output bidirectional pin with pull-up enabled. The power on reset period as can be seen from the electrical specs section of this document can have a duration range of between 2.5 ms and 10 ms. To ensure proper entry into ICP mode, the P34 pin should be driven LOW and held low a minimum of 10 ms after power up.

If during the ZLF645's power on reset period, the P34 pin is never driven LOW, pin p34 will be pulled HIGH through its pull-up device. In this case, if P34 remains HIGH until the end of the power on reset period, the ZLF645 will go into normal user mode and P34 will revert back to being an output pin only. To ensure proper entry into user mode when it is not intended to put the ZLF645 into ICP mode, it is important that in the customer application P34 only be connected to capacitive loads. This is due to the weak nature of its pull-up device, which can have a resistance ranging between 100 k $\Omega$  up to 600 k $\Omega$  depending on voltage, temperature, and process.

## State of ZLF645 in ICP Mode

The operating characteristics of the device in ICP mode are:

- The CPU stops executing instructions.
- All on-chip peripherals are disabled.
- The ZLF645 constantly refreshes the Watchdog Timer, if enabled.
- The P34 pin is configured as a bidirectional pin with pull-up enabled and with the output stage configured as open-drain. The bidirectional control of the pins comes from the ICP Tx/Rx logic.

## Enabling Flash Accesses Through the ICP

After the ZLF645 is in ICP mode, the `FLASHCTL` bit of the ICP Control register must be programmed to 1 before Flash accesses are enabled through the ICP interface.

## ICP Interface Logic Architecture

The ICP logic within the ZLF645 MCU consists of four primary functional blocks: transmitter, receiver, auto-baud detector/generator, and Flash Controller interface. Figure 16 displays the architecture of the ICP.

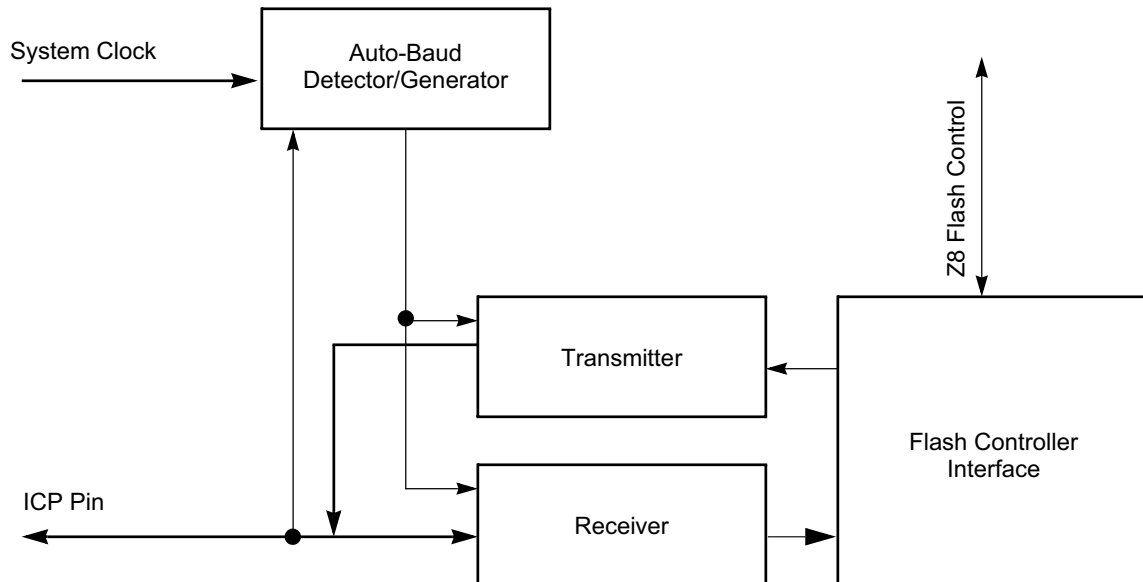


Figure 16. In-Circuit Programmer Block Diagram

## ICP Interface Operation

After the ZLF645 MCU is in ICP mode, pin P34 acts a bidirectional open-drain interface with internal pull-ups used for transmitting and receiving the data. Data transmission is half-duplex, in that transmit and receive cannot occur simultaneously. Serial data on P34 is sent using the standard asynchronous data format defined in RS-232. This pin creates an interface from the ZLF645 MCU to the serial port of a host PC using minimal external hardware. Figure 17 displays the recommended method of connecting P34 pin to an RS-232 connection using an open-drain buffer. The ICP pin must always be connected to  $V_{DD}$  through an external pull-up resistor.



**Caution:** For operation of the ICP, all power pins ( $V_{DD}$  and  $AV_{DD}$ ) must be supplied with power and all ground pins ( $V_{SS}$  and  $AV_{SS}$ ) must be properly grounded.

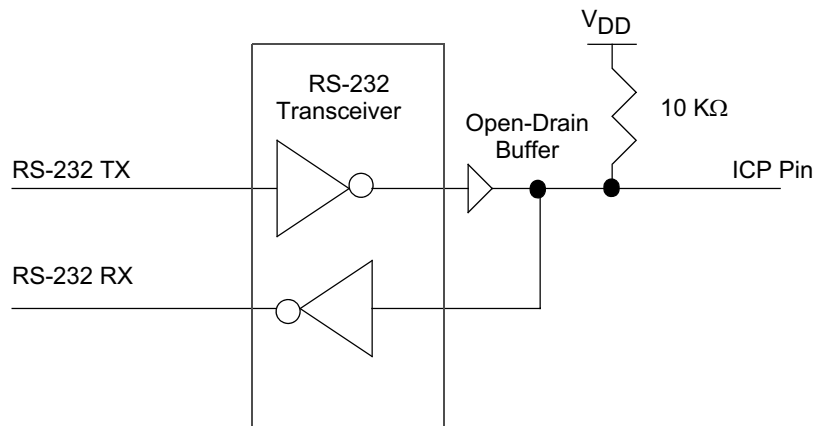


Figure 17. Interfacing the In-Circuit Programming Pin P34 with an RS-232 Interface (2)

### ICP Data Format

The ICP interface uses the asynchronous data format defined for RS-232. Each character is transmitted as 1 Start bit, 8 data bits (least significant bit first), and 1.5 Stop bits (see [Figure 18](#)).

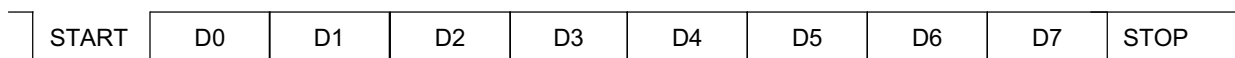


Figure 18. ICP Data Format

### ICP Auto-Baud Detector/Generator

To run over a range of baud rates (data bits per second) with various system clock frequencies, the ICP contains an Auto-Baud Detector/Generator. After a reset, the ICP is non-active until it receives data. The ICP requires that the first character sent from the host is character 80H. The character 80H has eight continuous bits Low (one Start bit plus 7 data bits), framed between High bits. The Auto-Baud Detector measures this period and sets the ICP Baud Rate Generator accordingly.

The Auto-Baud Detector/Generator is clocked by the system clock. The minimum baud rate is the system clock frequency divided by 512. If the datastream can be synchronized with the system clock, the auto-baud generator can run as high as the system clock frequency divided by 2.

For optimal operation with asynchronous datastreams, the maximum recommended baud rate is the system clock frequency divided by 8. The maximum possible baud rate for asynchronous datastreams is the system clock frequency divided by 4, but this theoretical maximum is possible only for low noise designs with clean signals. [Table 28](#) lists minimum and recommended maximum baud rates for sample crystal frequencies.

**Table 28. ICP Baud-Rate Limits**

System Clock Frequency (MHz)	Recommended Maximum Baud Rate (Kbps)	Recommended Standard PC Baud Rate (bps)	Minimum Baud Rate (Kbps)
8.0	1000.0	737280	15.6
1.0	125.0	115,200	1.95
0.032768 (32 kHz)	4.096	2400	0.064

If the ICP receives a Serial Break (nine or more continuous bits Low) the Auto-Baud Detector/Generator resets. You can reconfigure the Auto-Baud Detector/Generator by sending character 80H.

## ICP Serial Errors

The ICP can detect any of the following error conditions on the P34 pin when in ICP mode:

- Serial Break (a minimum of nine continuous bits Low).
- Framing Error (received stop bit is Low).
- Transmit Collision (ICP and host simultaneous transmission detected by the ICP).

When the ICP detects one of these errors, it aborts any command currently in progress, transmits a four character long Serial Break back to the host, and resets the Auto-Baud Detector/Generator. A Framing Error or Transmit Collision can be caused by the host sending a Serial Break to the ICP. Because of the open-drain nature of the interface, returning a Serial Break back to the host only extends the length of the Serial Break, if the host releases the Serial Break early.

The host transmits a Serial Break on the ICP pin when first connecting to the device or recovering from an error. A Serial Break from the host resets the Auto-Baud Generator/Detector but does not resets the [ICP Control Register](#). A Serial Break leaves the device in DEBUG mode if that is the current mode. The ICP is held in Reset until the end of the Serial Break when the ICP pin returns High. Because of the open-drain nature of the ICP pin, the host can send a Serial Break to the ICP even if the ICP is transmitting a character.



As the ICP interface uses a single pin for both receive and transmit, it can only receive or transmit at a given time. For the most part, this is not a problem, as the ICP uses a host driven protocol (Z8<sup>®</sup> does not send any data without the host asking for it).

To aid the ICP in avoiding collisions, the transmitter waits an additional 1/2 bit times after a Stop bit is fully received or transmitted before it starts transmission of a character. On the other hand, the receiver starts searching for a Start bit as soon as the middle of the Stop bit has been sampled and is valid. The transmitter does not start if another character is being received.

## ICP In-Circuit Programming Commands

The host communicates to the ICP by sending ICP commands using the ICP interface. During normal operation, only a subset of the ICP commands are available. In FLASH CONTROL mode, all ICP commands are available, but for few commands their access to the Flash is qualified based upon the programming of the Flash Read/Write Protect Option bit (FLRWP) or the Lower Half Flash Read/Write Protect Option bit (FLPROT1). When either of these bits is enabled, some of the ICP commands will have reduced Flash memory access or will be disabled completely.

Table 29 is a summary of the ICP commands. Each ICP command is described in further detail in the bulleted list following this table. Table 29 also indicates those commands that operate when the device is not in FLASH CONTROL mode (normal operation) and how those commands are effected by programming of the FLRWP and FLPROT1 Option bits.

**Table 29. In-Circuit Programmer Commands**

ICP Command	Command Byte	Enabled when NOT in FLASH CONTROL mode?	Disabled by Flash Read/Write Protect Option Bits (FLRWP and/or FLPROT1)
Read ICP Revision	00H	Yes	—
Reserved	01H	—	—
Read ICP Status Register	02H	Yes	—
Reserved	03H	No	—
Write ICP Control Register	04H	Yes	—
Read ICP Control Register	05H	Yes	—
Reserved	06H – 07H	No	—
Write Flash Controller Registers	08H	No	—

**Table 29. In-Circuit Programmer Commands (Continued)**

ICP Command	Command Byte	Enabled when NOT in FLASH CONTROL mode?	Disabled by Flash Read/Write Protect Option Bits (FLRWP and/or FLPROT1)
Read Flash Controller Registers	09H	No	—
Write Flash Memory	0AH	No	If FLRWP enabled, command is disabled for entire Flash main memory and page 3 of the Information Area. If FLPROT1 enabled, command disabled for page 3 of the Information Area and lower half of main memory only.
Read Flash Memory	0BH	No	If FLRWP enabled, command is disabled for the Flash main memory. If FLPROT1 enabled, command disabled for the lower half of main memory only.
Reserved	0CH – 0DH	—	Disabled
Read Program Memory CRC	0EH	No	—
Reserved	0FH – 1AH	—	—
Read ICP Autobaud Register	1BH	Yes	—
Reserved	1CH – EFH	—	—
Write Test Mode Register	F0H	Yes	—
Read Test Mode Register	F1H	Yes	—
Reserved	F2H – FFH	—	—

In the following bulleted list of ICP commands, data and commands sent from the host to the ICP are identified by ‘ICP ← Command/Data’. Data sent from the ICP back to the host is identified by ‘ICP → Data’:

- **Read ICP Revision (00H)**—The Read ICP Revision command determines the version of the ICP. If ICP commands are added, removed, or changed, the revision number changes.

ICP ← 00H

ICP → ICPRev[15:8] (Major revision number)

ICP → ICPRev[7:0] (Minor revision number)

This command when executed returns a value of 0132H which is the revision ID assigned for the ZLF645 MCU.

- **Read ICP Status Register (02H)**—The Read ICP Status register command reads the ICPSTAT register.

```
ICP ← 02H
ICP → ICPSTAT[7:0]
```

- **Write ICP Control Register (04H)**—The Write ICP Control register command writes the data that follows the command to the ICPCTL register.

```
ICP ← 04H
ICP ← ICPCTL[7:0]
```

- **Read ICP Control Register (05H)**—The Read ICP Control register command reads the value of the ICPCTL register.

```
ICP ← 05H
ICP → ICPCTL[7:0]
```

- **Write Flash Controller Registers (08H)**—The Write Flash Controller register command allows writes to the Flash Controller registers. This command configures the Flash Controller for Flash memory accesses through the Write Flash Memory and Read Flash Memory commands. If the device is not in FLASH CONTROL mode, the register address and data values are discarded.

```
ICP ← 08H
ICP ← {8'hFF}
ICP ← Register Address[7:0]
ICP ← Size[7:0]
ICP ← 1-256 data bytes
```

- **Read Flash Controller Registers (09H)**—The Read Flash Controller command allows reads of the Flash Controller registers. If the device is not in FLASH CONTROL mode this command returns FFH for all the register values.

```
ICP ← 09H
ICP ← {8'HFF}
ICP ← Register Address[7:0]
ICP ← Size[7:0]
ICP → 1-256 data bytes
```

- **Write Flash Memory (0AH)**—The Write Flash Memory command is used to write data to the main memory area or Information Area of the Flash memory. The command has equivalent functionality to the CPU writing the memory through the LDC and LDCI instructions. Data can be written 1 to memsize bytes at a time where memsize represents the size (32 KB or 64 KB) of the Flash memory for the product option chosen (The memsize number of bytes can be written by setting the size to 0). Should a size value greater than the maximum memory size be given by the user, the actual size value

for the command will default to the maximum memory size. The on-chip Flash Controller must be written to and unlocked for the programming operation to occur. If the Flash Controller is not unlocked, the data is discarded. Also, data is discarded for writes to protected areas of the Flash's main or information Page 3 areas based upon the settings of the read/write protect option bits in [User Option Byte 1 \(OPT1\)](#) register.

```
ICP ← 0AH
ICP ← Flash Memory Address[15:8]
ICP ← Flash Memory Address[7:0]
ICP ← Size[15:8]
ICP ← Size[7:0]
ICP ← 1-memsize data bytes
```

- **Read Flash Memory (0BH)**—The Read Flash Memory command is used to read data from the Flash's main memory area or Information Area. This command is equivalent to the CPU reading the memory through the LDC and LDCI instructions. Data can be read 1 to 'memsize' bytes at a time where memsize represents the size (32 KB or 64 KB) of the Flash memory for the product option chosen (The memsize number of bytes can be written by setting the size to 0). Depending on the settings of the read/write protect option bits in User Option Byte 1 register, reads to protected areas of the Flash's main memory area will return FFH for the data.

```
ICP ← 0BH
ICP ← Flash Memory Address[15:8]
ICP ← Flash Memory Address[7:0]
ICP ← Size[15:8]
ICP ← Size[7:0]
ICP → 1-65536 data bytes
```

- **Read Flash Main Memory CRC (0EH)**—The Read Flash Main Memory CRC command computes and returns the Cyclic Redundancy Check (CRC) of the Flash's Main Memory using the 16-bit CRC-CCITT polynomial. If the device is not in ICP mode, this command returns FFFFH for the CRC value. Unlike most other ICP Read commands, there is a delay from issuing of the command until the ICP returns the data. The ICP reads the Main Memory, calculates the CRC value, and returns the result. The delay is a function of the Flash main memory size and is approximately equal to the system clock period multiplied by the number of bytes in the Flash main memory.

```
ICP ← 0EH
ICP → CRC[15:8]
ICP → CRC[7:0]
```

- **Read ICP Autobaud Register (1BH)**— The Read ICP Autobaud register command reads the 12-bit ICP autobaud value set during autobaud detection.

```
ICP ← 1BH
ICP → (4'b0000, Autobaud[11:8])
```



ICP → Autobaud [7:0]

- **Write Test Mode Register (F0H)**— The Write Test Mode Register command writes the data that follows the command to the [TEST Mode Register](#) (TESTMODE).

ICP ← F0H

ICP ← TESTMODE [7:0]

- **Read Test Mode Register (F1H)**— The Read Test Mode register command reads the value of the TESTMODE register.

ICP ← F1H

ICP → TESTMODE [7:0]

ICP → Autobaud [7:0]

## In-Circuit Programming Control Register Definitions

### ICP Control Register

The ICP Control register (see [Table 30](#)) controls the state of the ICP interface. This register is used to enter or exit FLASH CONTROL mode. It can also reset the device.

**Table 30. ICP Control Register (ICPCTL)**

Bits	7	6	5	4	3	2	1	0
Field	FLASHCTL	Reserved						
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R	R	R	R

Bit Position	Value	Description
FLASHCTL [7]		<b>FLASH CONTROL Mode</b> When this bit is programmed to 1, the device enters FLASH CONTROL mode. When programmed to 1, this bit enables the ICP to perform Flash memory accesses through the devices Flash Controller.
	0	The device is operating in NORMAL mode.
	1	The device is in FLASH CONTROL mode.
[6:0]	—	<b>Reserved</b> —Must be written to 1.



## ICP Status Register

The ICP Status register (see [Table 31](#)) reports status information about the current state of the ICP and the device.

**Table 31. ICP Status Register (ICPSTAT)**

Bits	7	6	5	4	3	2	1	0
Field	FLASHCTL	FLPROT1	FLRWP	Reserved	FLWAIT	Reserved		
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit Position	Value	Description
[7]		<b>FLASHCTL</b> —When read, this bit indicates whether the device is in FLASH CONTROL mode.
	0	The device is operating in NORMAL mode.
	1	The device is in FLASH CONTROL mode.
[6]		<b>FLPROT1</b> —When read, this bit indicates the value of the devices FLPROT1 option bit as read from the <a href="#">User Option Byte 1 Shadow Register (OPT1SR)</a> on page 170.
	0	FLPROT1 mode is enabled.
	1	FLPROT1 mode is disabled.
[5]		<b>FLRWP</b> —When read, this bit indicates the value of the devices FLRWP option bit as read from the User Option Byte 1 shadow register.
	0	FLRWP mode is enabled.
	1	FLRWP mode is disabled.
[4]	—	<b>Reserved</b> —Must be written to 1.
[3]		<b>FLWAIT</b> —When read, this bit indicates whether an ICP initiated Flash program, page erase, or mass erase operation is completed or not.
	0	An initiated Flash programming, page erase, or mass erase operation is now complete.
	1	A Flash programming, page erase, or mass erase operation is still in progress and has not yet completed. No new Flash operations must be started until this bit reads as a 0.
[2:0]	—	<b>Reserved</b> —Must be written to 1.



## TEST Mode Register

The TEST Mode register is used to enable various device test or Flash memory access modes. At present this register only provides configuration for a single mode where, once programmed, Flash memory accesses bypass the devices Flash Controller and are done through the devices I/O pins. A complete description of this mode is available in the Flash Byte Programming section. This register can only be read or written using the ICP Read/Write Test Mode Register commands.

**Table 32. TEST Mode Register (TESTMODE)**

Bits	7	6	5	4	3	2	1	0
Field	Reserved					Flash Controller Bypass Mode	Reserved	
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R	R

Bit Position	Value	Description
[7:3]	—	<b>Reserved</b> — Must be written to 1. Reads return 0.
[2]		<b>Flash Controller Bypass Mode</b>
	0	The device is not in Flash Controller Bypass Mode.
	1	The device is in Flash Controller Bypass mode.
[1:0]	—	<b>Reserved</b> —Must be written to 1. Reads return 0.

## Exiting ICP Mode

The ZLF645 MCU is taken out of ICP mode under any of the following conditions:

- Initiating a POR with P36 held High during the entire reset period.
- Lowering  $V_{DD}$  until the ZLF645 MCU reaches a Voltage Brownout reset state.

# Flash Controller

## Flash Memory Overview

The ZLF645 products feature either 32 KB or 64 KB of non-volatile Flash memory with read/write/erase capability. The Flash memory provides a 16-bit data interface but supports both 16-bit and 8-bit programming and read operations. The Flash memory can be programmed, read, or erased by the Flash Controller directed by either user code or the In-Circuit Programmer (ICP) interface pin. All user code or ICP Flash Accesses use the Flash's byte access mode where programs and reads occur 8 bits at a time. A Flash Byte Programming interface, as described in the [Flash Byte Programming Interface](#) on page 76, is also available for Flash accesses through the devices GPIO pins and bypassing the Flash Controller. When the Flash Byte Programming interface is used, Flash programming and reads can be done either 8-bits or 16-bits at a time, depending on the package type of the device.

The Flash memory consists of two blocks, the **Main Memory** and the **Information Block**. The Flash main memory is arranged in pages with 512 bytes per page. The data interface to the Flash memory supports both 16-bit and 8-bit data programming and data reads. Although 512-byte page is the minimum Flash block size that can be erased. Each page is divided into 8 rows of 64 bytes.

- ▶ **Note:** *The term 'page' in the context of the Flash Controller is not equivalent to the Z8<sup>®</sup> LXMC CPU architecture's Program Memory page. For Flash contents protection, the Flash main memory is also divided into sectors, each sector containing 16 consecutive pages.*

In addition to the Flash main memory, there is a 256-byte Information block, arranged as 4 rows of 64 bytes. Each row is defined as a page. User access is only allowed to Page 3, where user definable Option bits reside. Pages 2-0 are for Zilog<sup>®</sup> internal use.

- ▶ **Note:** *Information block does not have a Flash contents sector protection mechanism.*

[Table 33](#) lists the Flash main memory configuration for each device in the family of ZLF645 products. The size and configuration of the Information block is the same for all devices. [Figure 19](#) displays the Flash memory arrangement.

**Table 33. ZLF645 Products Flash Memory Configurations**

Part Number	Flash Size KBytes	Flash Pages	Program Memory Addresses	Flash Sector Size
ZLF645xxxxx32	32 KB	64	0000H–7FFFH	8 KB
ZLF645xxxxx64	64 KB	128	0000H–FFFFH	8 KB



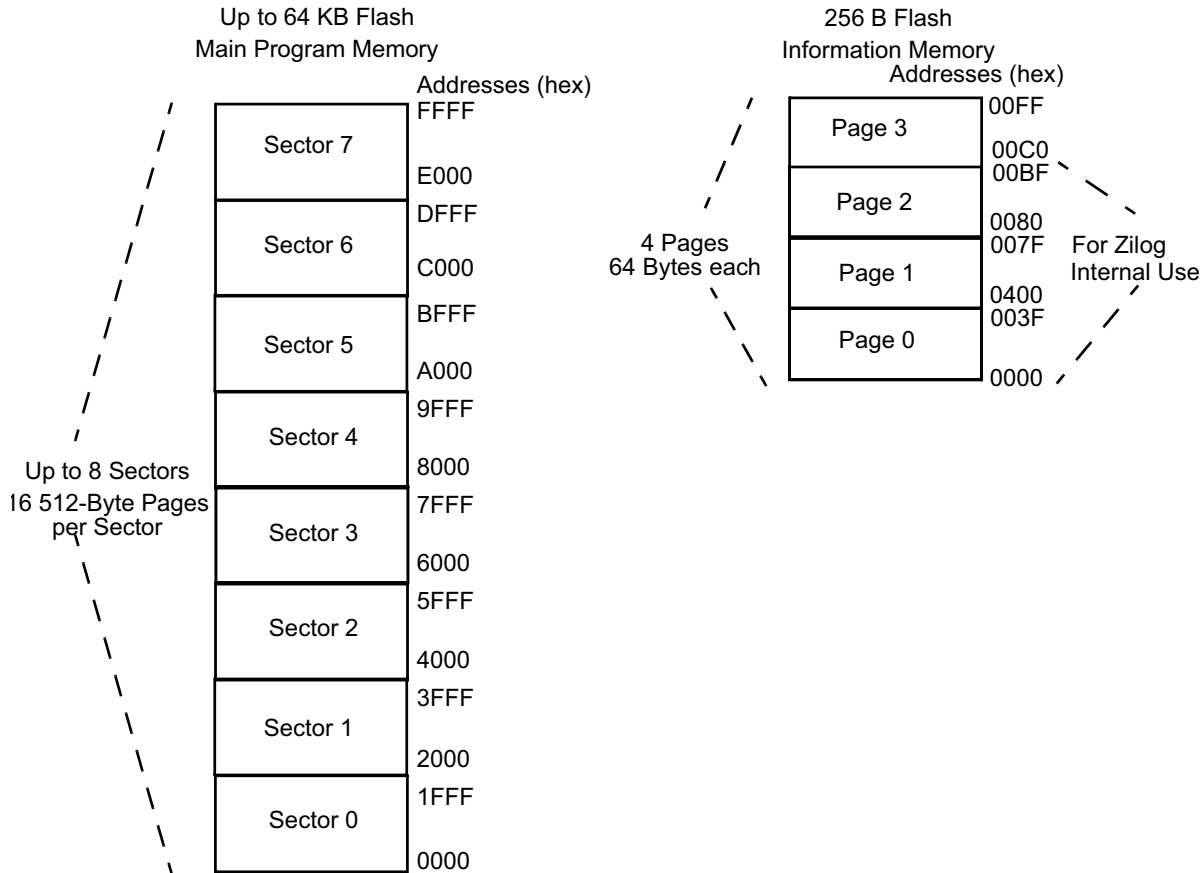


Figure 19. Flash Memory Arrangement

## Flash Information Block

The Flash Information Block of Flash memory is divided into two sections. Page 3 of the Information Block is accessible to the user or Flash programmer vendor for programming, reading, or erasure through ICP or Flash Byte Programming Interface as described in the [Flash Byte Programming Interface](#) on page 76. User Option Bytes 0 and 1 use addresses 00FE and 00FF respectively of the Page 3 area and contain programmable bits with pre-defined functions. Address locations 00C0 and 00C1 of Page 3 contain a Flash Test pass code of 5Ah. This pass code is programmed by Zilog® on completing its Flash testing and is an indication that the Flash memory passed Flash testing.

The Flash read/write protect bits in Byte 1 control the level of Page 3 access allowed to the User along with the User's level of access to the Flash's main memory. Bytes 00C2 through 00FD of Page 3 have no pre-defined function and are available to the User for other operations.

Pages 0 through 2 (addresses 0000 through 00BF), of the Information Area are reserved for Zilog® internal use and are inaccessible by user or programmer vendor, either through the ICP interface or by using the Flash Byte Programming interface.

## Flash Controller Operation

The Flash Controller provides the appropriate Flash controls and timing for byte/word programming, Page Erase, Mass Erase, and reading of the Flash memory for Flash accesses made by either the CPU or through the ICP interface.

The Flash Controller contains several protection mechanisms to prevent accidental programming or erasure. These mechanism operate on the page, sector and full-memory levels. [Figure 20](#) displays basic Flash Controller operation.

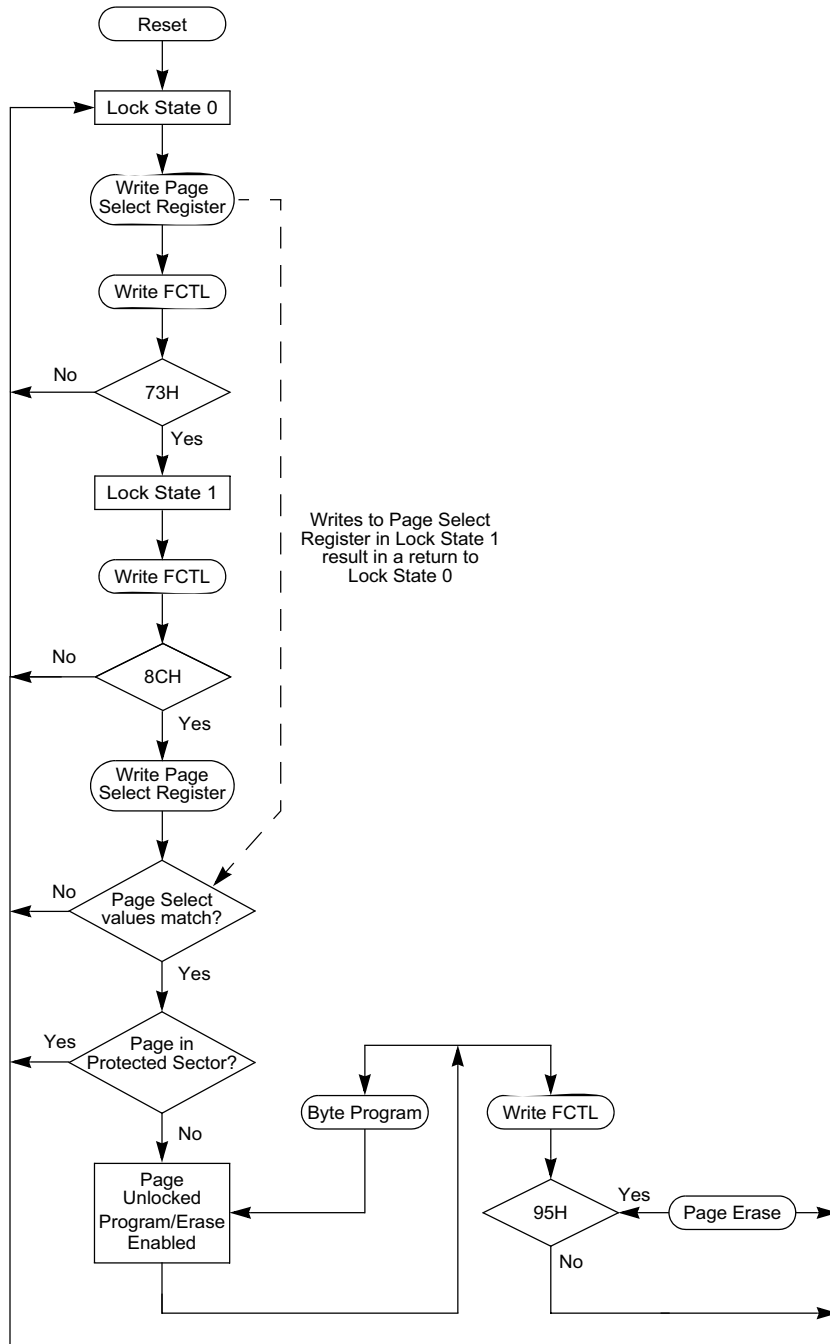


Figure 20. Flash Controller Operation Flow Chart

## Flash Operation Timing Using Flash Frequency Registers

Before a program or erase operation on Flash memory, you must first configure the Flash frequency High and Low Byte registers. The Flash frequency registers allow programming and erasing of the Flash with CPU clock frequencies ranging from 1 MHz to 8 MHz. The Flash Frequency High and Low Byte registers combine to form a 16-bit value (FFREQ) to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain two times the system clock frequency (in kHz).

This 16-bit binary Flash Frequency value is calculated using the following equation:

$$\text{FFREQ}[15:0] = \frac{2 \times \text{System Clock Frequency (Hz)}}{1000}$$



**Caution:** *The System Clock Frequency depends on the Flash memory programming of bit 2 of the User Option Byte 1 and on the register programming of bit 0 of the SMR register and can be equal to the clock input frequency on the XTAL1 pin, a divide by 2 of that input, a divide by 16 of that input, or a divide by 32 of that input. Flash programming and erasure are not supported for CPU clock frequencies below 1 MHz or above 8 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct values.*

## Flash Code Protection Against External Access

User code within the Flash main memory can be protected from external access through the ICP or Flash Byte Programming interfaces. By use of the Flash Protect Option bits in User Option byte 1 of the Flash Information Area, read access to the Flash's main memory can be blocked. For more information, see [Table 86](#) on page 169.

## Flash Code Protection Against Accidental Program and Erasure

The ZLF645 products provide several levels of protection against accidental program and erasure of the Flash main memory contents. This protection is provided by a combination of the register locking mechanism, the page select redundancy, and the sector level protection control of the Flash Controller. Similar levels of protection are in place for the Flash Information Area, minus the sector level protection.

### Flash Code Protection Using Flash Controller

On Reset, the Flash Controller locks to prevent accidental program or erasure of the Flash memory. To program or erase the Flash memory, first write the Page Select register with the target page. Unlock the Flash Controller by making two consecutive writes to the Flash Control register with the values 73H and 8CH, sequentially. The Page Select register must be rewritten with the same page previously stored there. If the two Page Select writes

do not match, the controller reverts to a locked state. If the two writes match, the selected page becomes active. See [Figure 20](#) on page 67 for details.

After unlocking a specific page, you can enable either Page Program or Page Erase. For Flash main memory, writing the value 95H causes a Page Erase only if the active page resides in a sector that is not protected. For Flash Information Area, without sector level protection, writing the value 95H causes a Page Erase. Any other value written to the Flash Control register locks the Flash Controller.

After unlocking a specific page, you can write to any byte on that page. After a byte is written the page remains unlocked, allowing for subsequent writes to other bytes on the same page. Further writes to the Flash Control register causes the active page to revert to a locked state.

### Sector Based Flash Protection

The final protection mechanism for Flash main memory is implemented on a per-sector basis. For all ZLF645 products, each sector contains 16 pages (of 512 bytes each).

Part Number	Number of Sectors
ZLF645xxxxx32	4
ZLF645xxxxx64	8

The Sector Protect register controls the protection state of each Flash sector. This register is address-shared with the Page Select register. It is accessed by unlocking the Flash Controller by writing the command byte 5EH. The next write to Bank F, Register Address 02H targets the Sector Protect register.

The Sector Protect register is initialized to 0 on reset, putting each sector into an unprotected state. When a bit in the Sector Protect register is written to 1, the corresponding sector can no longer be written or erased. After a bit of the Sector Protect register has been set, it cannot be cleared except by powering down the device. As mentioned earlier, sector-based protection is not applicable to the Flash Information Area.

### Byte Programming

As mentioned previously, all Flash accesses either through user code or through the ICP interface occur using the Flash memory byte mode of operation. The Flash memory is enabled for byte programming after unlocking the Flash Controller and enabling either Mass Erase or Page Erase. When the Flash Controller is unlocked and Mass Erase is enabled, all Program Memory locations are available for byte programming. In contrast, when the Flash Controller is unlocked and Page Erase is enabled, only the locations of the selected page are available for byte programming. An erased Flash byte contains all 1's (FFH).

The programming operation can only be used to change bits from 1 to 0. To change a Flash bit (or multiple bits) from 0 to 1 requires execution of either the Page Erase or Mass Erase commands.

Byte Programming can be accomplished using the In-Circuit Programmer's Write Memory command or Z8 LXMC CPU execution of the LDC or LDCI instructions. Refer to the *Z8<sup>®</sup> LXMC CPU User Manual (UM0215)* for a description of the LDC and LDCI instructions. While the Flash Controller programs the Flash memory, the Z8 LXMC CPU is non-active but the system clock and on-chip peripherals continue to operate. To exit programming mode and lock the Flash, write any value to the Flash Control register, except the Mass Erase or Page Erase commands.

## Page Erase

The Flash main memory can be erased one page (512 bytes) at a time. Page Erasing the Flash memory sets all bytes in that page to the value FFH. The Flash Page Select register identifies the page to be erased. Only a page residing in an unprotected sector can be erased. With the Flash Controller unlocked and the active page set, writing the value 95H to the Flash Control register initiates the Page Erase operation. While the Flash Controller executes the Page Erase operation, the Z8 LXMC CPU is non-active but the system clock and on-chip peripherals continue to operate. The Z8 LXMC CPU resumes operation after the Page Erase operation completes.

If a Page Erase operation is performed using the ICP, bit 3 of the ICP Status register can be polled to determine when the operation is complete. When the Page Erase is complete, the Flash Controller returns to its locked state. You can erase Page 3 of Information Area by similar procedure, when Flash Page Select register bit 7 is at logic High. See [Table 35](#) on page 72 for details.

## Mass Erase

The Flash main memory can also be Mass Erased using the Flash Controller, but only through the ICP interface and not by the CPU. Mass Erasing the Flash memory sets all bytes to the value FFH. With the Flash Controller unlocked, writing the value 63H to the Flash Control register initiates the Mass Erase operation. If a Mass Erase operation is performed using the ICP, bit 3 of the ICP Status register can be polled to determine when the operation is complete. When the Mass Erase is complete, the Flash Controller returns to its locked state. You cannot mass erase Information Area.



**Caution:** *If either of the Flash Memory Protect Option Bits are set as defined in the Flash Option Bits section, a mass erase of the Flash's main memory must be performed before Page 3 of the Flash's Information Area can be erased. These two operations must be done when the device is at operating voltage. That is, if a mass erase is followed with a power-down then power-up sequence, performing an Information Area Page 3 erase will not erase its contents.*

## Flash Control Register Definitions

### Flash Control Register

The Flash Controller must be unlocked using the Flash Control register (see [Table 34](#)) before programming or erasing the Flash memory. Writing 73H 8CH, sequentially, to the Flash Control register unlocks the Flash Controller. When the Flash Controller is unlocked, the Flash memory can be enabled for Mass Erase or Page Erase by writing the appropriate enable command to the FCTL. Page Erase applies only to the active page selected in Flash Page Select register. Mass Erase is enabled only through the ICP. Writing an invalid value or an invalid sequence returns the Flash Controller to its locked state. The Write-only Flash Control register shares its Register File address with the Read-only Flash Status register.

**Table 34. Flash Control Register (FCTL)**

Bits	7	6	5	4	3	2	1	0
Field	FCMD							
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	Bank F, Register address: 01H							

Bit Position	Value	Description
[7:0]		<b>FCMD—Flash Command</b>
	73H	First unlock command.
	8CH	Second unlock command.
	95H	Page Erase command (must be third command in sequence to initiate Page Erase).
	63H	Mass Erase command (must be third command in sequence to initiate Mass Erase).
	5EH	Enable Flash Sector Protect Register Access.

## Flash Status Register

The Flash Status register (see [Table 35](#)) indicates the current state of the Flash Controller. This register can be read any time. The read-only Flash Status register shares its Register File address with the Write-only Flash Control register.

**Table 35. Flash Status Register (FSTAT)**

Bits	7	6	5	4	3	2	1	0
Field	Reserved		FSTAT					
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank F, Register address: 01H							

Bit Position	Value	Description
[7:6]	—	<b>Reserved</b> —Reads as 0's.
[5:0]		<b>FSTAT</b> —Flash Controller Status
	000000	Flash Controller locked.
	000001	First unlock command received (73H written).
	000010	Second unlock command received (8CH written).
	000011	Flash Controller unlocked.
	000100	Sector protect register selected.
	001xxx	Program operation in progress.
	010xxx	Page erase operation in progress.
	100xxx	Mass erase operation in progress.

## Flash Page Select Register

The Flash Page Select register (see [Table 36](#)) shares address space with the Flash Sector Protect register. Unless the Flash Controller is unlocked and written with 5EH, writes to this address target the Flash Page Select register.

The register is used to select one Flash main memory page to be programmed or erased when IFEN bit is conditioned logic Low. Each Flash Page contains 512 bytes of Flash memory. During a Page Erase operation, all Flash memory having addresses with the most significant 7-bits given by FPS[6:0] are chosen for Program/Erase operation. This register is also used to select one Flash Information Area page to be programmed or erased, when IFEN bit is conditioned logic High. Each Flash Information Page contains 64 bytes.



**Table 36. Flash Page Select Register (FPS)**

Bits	7	6	5	4	3	2	1	0
Field	IFEN	PAGE						
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank F, Register address: 02H							

Bit Position	Value	Description
[7]	0 1	<b>IFEN</b> —Information Area Enable Operation to be performed on Flash main memory. Operation to be performed on Flash Information Area.
[6:0]	0 1	<b>PAGE</b> —Page Select This 7-bit field identifies the Flash main memory page for Page Erase and Page unlocking. Program Memory Address[15:9] = PAGE[6:0]. The least significant 2 bits of Page identifies the Flash Information page for Page Erase and Page unlocking. The upper significant bits must be logic 0's.

### Flash Sector Protect Register

The Flash Sector Protect register (see [Table 37](#)) address is shared with the Flash Page Select register. When the Flash Control register is unlocked and written with 5EH, the next write to this address targets the Flash Sector Protect register. In all other cases, it targets the Flash Page Select register.

This register selects which of the 8 available Flash memory sectors to be protected. The reset state of each Sector Protect bit is an unprotected state. After a sector is protected by setting its corresponding register bit, it cannot be unprotected (the register bit cannot be cleared) without powering down the device.

**Table 37. Flash Sector Protect Register (FSEC)**

Bits	7	6	5	4	3	2	1	0
Field	SPROT7	SPROT6	SPROT5	SPROT4	SPROT3	SPROT2	SPROT1	SPROT0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank F, Register address 02H							

Bit Position	Value	Description
[7:0]		<b>SPROT7-SPROT0</b> —Sector Protection Each bit corresponds to an 16-page Flash sector. For the ZLF645xxxxx64, all bits are used. Only bits 3-0 are used in the ZLF645xxxxx32.

### Flash Frequency High and Low Byte Registers

The Flash Frequency High and Low Byte registers (Table 38 and Table 39) combine to form a 16-bit value, FFREQ, to control timing for Flash program and erase operations. The 16-bit binary Flash Frequency value must contain two times the system clock frequency (in kHz) and is calculated using the following equation:

$$FFREQ[15:0] = \{FFREQH[7:0], FFREQL[7:0]\} = \frac{2 \times \text{System Clock Frequency (Hz)}}{1000}$$

Programming the Flash Frequency High and Low Byte Registers as per the formula given provides a Flash programming time of approximately 55 μs and an erase time of approximately 10 ms.



**Caution:** *Flash programming and erasure is not supported for system clock frequencies below 1 MHz or above 8 MHz. The Flash Frequency High and Low Byte registers must be loaded with the correct value to ensure proper operation of the device.*

**Table 38. Flash Frequency High Byte Register (FFREQH)**

Bits	7	6	5	4	3	2	1	0
Field	FFREQH							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank F, Register address: 03H							

Bit Position	Value	Description
[7:0]		<b>FFREQH</b> —Flash Frequency High Byte High byte of the 16-bit Flash Frequency value.

**Table 39. Flash Frequency Low Byte Register (FFREQ\_L)**

Bits	7	6	5	4	3	2	1	0
Field	FFREQ_L							
Reset	0							
R/W	R/W							
Address	Bank F, Register address: 04H							

Bit Position	Value	Description
[7:0]		<b>FFREQ_L</b> —Flash Frequency Low Byte Low byte of the 16-bit Flash Frequency value.

### Flash Controller Functions Summary

The Flash Controller performs its functions, directed by either the ICP interface or by instruction codes. [Table 40](#) lists the functions that will or will not be performed, according to the control source and whether operation is performed on Flash main memory or Information Area.

**Table 40. Flash Controller Functions Summary**

Control Source	Flash Memory Block	Program	Read	Page Erase	Mass Erase	Flash Protect
ICP	Main Memory	Yes	Yes	Yes	Yes	Yes <sup>1</sup>
	Information Area	Page 3 Only	Page 3 Only	Page 3 Only	No	Yes <sup>2</sup>
Instruction Codes	Main Memory	Yes	Yes	Yes	No	No <sup>3</sup>
	Information Area	No	No	No	No	No <sup>3</sup>

**Notes**

1. FLPROT1 = 0, cannot read or write lowest half of memory. FLRWP = 0, cannot read or write entire main memory.
2. FLRWP = 0, cannot write or erase Page 3.
3. FLPROT1 = 0, no effect. FLRWP = 0, no effect.

# Flash Byte Programming Interface

Using the ZLF645's Flash Byte Programming interface, the on-chip Flash controller can be bypassed, allowing direct control of the Flash signals through registered values of certain of the ZLF645's GPIO pins. Bypassing the Flash controller allows faster row programming algorithms to be used by controlling the Flash programming signals directly. This method is beneficial when programming a large number of devices and can be used for Flash programming by third party vendors who manufacture gang programmers. For more information on how to use this interface, refer to *Third-Party Flash Programming Support for Z8 Crimson Flash Parts*, available for download at [www.zilog.com](http://www.zilog.com).

## Enabling The Flash Byte Programming Interface

The Flash Byte Programming Interface is enabled by writing three bytes to the ICP interface:

1. 80H — initiates auto-baud calculation of the ICP interface data and clock rate.
2. F0H — ICP Write Test Mode Register command.
3. 04H — Data to be written to the Test Mode Register. This enables the Flash Byte Programming interface.

► **Note:** *Since Flash Byte Programming Interface is enabled with the ZLF645 MCU in ICP mode, the CPU clock will stop and no CPU accesses to the Flash memory will occur.*

## Flash Byte Programming Interface Flash Access Restrictions

The types of Flash access allowed to the Flash memory through the Flash Byte Programming interface is qualified similar to the ICP, by the settings of the Flash Memory Protection Bits in User Option Byte 1. If either of the Flash protect bits are set, the program memory has to be mass erased before full read/program access is allowed to either the main memory or Information area page 3 sections of the Flash memory, respectively. Flash memory access allowed through the Flash Byte Programming interface is summarized in [Table 41](#).

**Table 41. Flash Byte Programming Functions Summary**

Flash Memory Block	Program	Read	Page Erase	Mass Erase	Flash Protect Option Bits
Main Memory	Yes	Yes	Yes	Yes	FLRWP=1, FLPROT1=1
Main Memory	No	No	No	Yes	FLRWP=0, FLPROT1=X
Main Memory	Yes <sup>1</sup>	Yes <sup>1</sup>	Yes <sup>1</sup>	Yes	FLRWP=1, FLPROT1=0
Information Area	Yes <sup>2</sup>	Yes <sup>2</sup>	Yes <sup>2</sup>	Yes	FLRWP=1, FLPROT1=1
Information Area	No	Yes <sup>2</sup>	No	Yes	FLRWP=1, FLPROT1=0
Information Area	No	Yes <sup>2</sup>	No	Yes	FLRWP=0, FLPROT1=1

**Notes**

1. Program, Read, and Page Erase access is limited to the upper half address space of the main memory only.
2. Only Page 3 of the Information Area is accessible for Program, Read, and Page Erase operations.

# Infrared Learning Amplifier

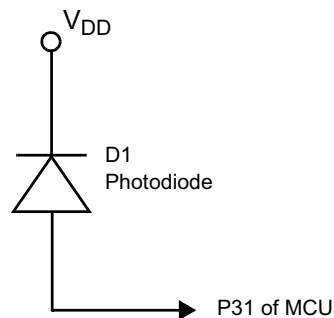
The ZLF645 MCU's infrared learning amplifier allows you to detect and decode infrared transmissions directly from the output of the receiving diode without the need for external circuitry (see [Port 3](#) on page 23).

An IR diode can be connected to the IR amplifier as displayed in [Figure 21](#). When the IR amplifier is enabled and an input current is detected on Port 3, Pin 1 (P31), the IR amplifier outputs a logical High value. When the input current is below the switching threshold of the IR amplifier, the amplifier outputs a logical Low value.

Within the MCU, the IR amp output goes to the capture/timer logic, which can be programmed to demodulate the IR signal. The IR amplifier output can also be read by the CPU, or drive the Port 3, Pin 4 (P34) output if write-only register bit PCON[0] is set to 1.

For the maximum current input that is clearly recognized by the ZLF645 as a 0 and the minimum current input that is clearly recognized as a 1, see  $I_{DETLO}$  and  $I_{DETHI}$  parameters, respectively, in [Table 81](#) on page 160.

The IR learning amplifier can demodulate signals up to a frequency of 500 kHz. A special mode exists that allows you to capture the third, fourth, and fifth edges of the IR amplifier output and generate an interrupt.



**Figure 21. Learning Amplification Circuitry within the ZLF645 Flash MCU**

For details on programming the timers to demodulate a received signal, see [Timers](#) on page 93.

# Universal Asynchronous Receiver/ Transmitter

The Universal Asynchronous Receiver/Transmitter (UART) is a full-duplex communication channel capable of handling asynchronous data transfers. The two UARTs use a single 8-bit data mode with selectable parity.

The UART interface when enabled uses the GPIO pins P07 for the UART transmit and P32 for the UART receive.

The features of the UART include:

- 8-bit asynchronous data transfer
- Selectable even- and odd-parity generation and checking
- One or two Stop bits
- Separate transmit and receive interrupts
- Framing, overrun, and break detection
- Separate transmit and receive enables
- 8-bit Baud Rate Generator
- Baud Rate Generator timer mode
- UART operational during HALT mode

**Table 42. UART Control Registers**

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
0F1	All	F1	UART Receive/Transmit Data Register	URDATA/ UTDATA	XXh	89
0F2	All	F2	UART Status Register	UST	0000_0010b	89
0F3	All	F3	UART Control Register	UCTL	00h	91
0F4	All	F4	UART Baud Rate Generator Constant Register	BCNST	FFh	92

## Architecture

The UARTs consist of three primary functional blocks: **transmitter**, **receiver**, and **Baud Rate Generator**. The UART transmitter and receiver function independently, but employ the same baud rate and data format. [Figure 22](#) displays the UART architecture.

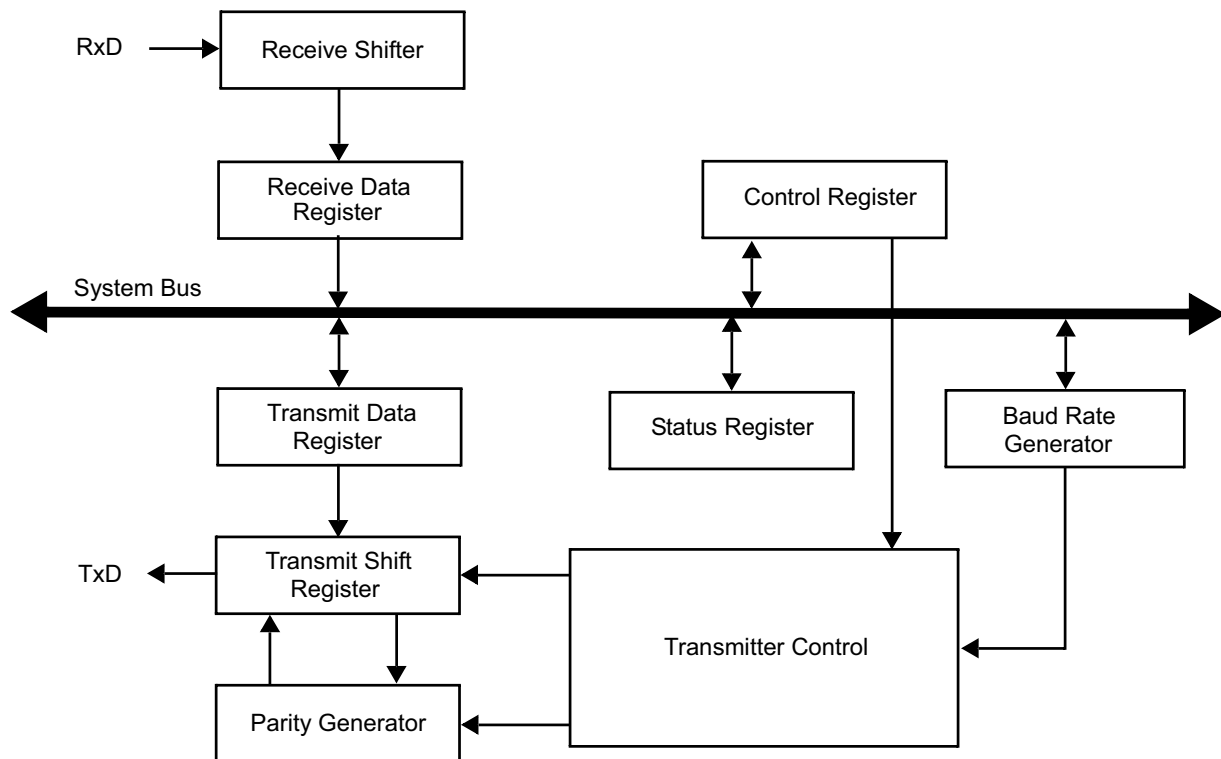


Figure 22. UART Block Diagram

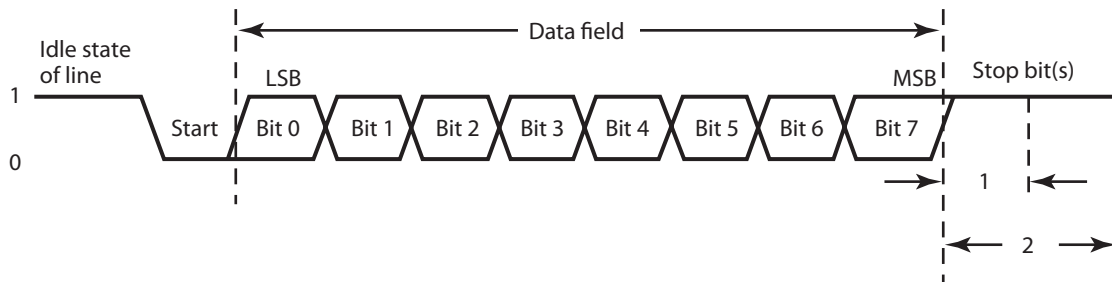
## Operation

The UART channel can be used to communicate with a master microprocessor or a slave microprocessor, both of which exhibit transmit and receive functionality. You can either operate the UART channel by polling the UART Status register or via interrupts. The UART remains active during HALT mode. If neither the transmitter nor the receiver is enabled, the UART baud rate generator can be used as an additional timer. The UART contains a noise filter for the receiver that can be enabled by the user.

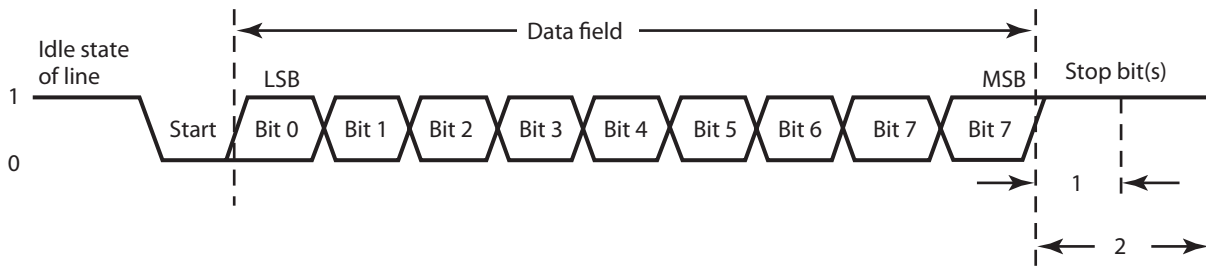


## Data Format

The UART transmits and receives data in an 8-bit data format, with the least significant bit (lsb) occurring first. An even- or odd-parity bit can be optionally added to the data stream. Each character begins with an active Low Start bit and ends with either 1 or 2 active High Stop bits. [Figure 23](#) and [Figure 24](#) display the asynchronous data format employed by the UARTs with or without parity, respectively.



**Figure 23. UART Asynchronous Data Format without Parity**



**Figure 24. UART Asynchronous Data Format with Parity**

## Transmitting Data Using Polled Method

Follow the steps below to transmit data using the polled method of operation:

1. Write to the Baud Rate Generator Constant (BCNST) register, address 0F4h, to set the appropriate baud rate.
2. Write 0 to bit 6 of the P01M register.
3. Write to the UART Control register (UCTL) to:
  - (a) Set the transmit enable bit, UCTL[7], to enable the UART for data transmission.
  - (b) If parity is appropriate, set the parity enable bit, UCTL[4] to 1 and select either even- or odd-parity (UCTL[3]).

4. Check the Transmit Status register bit, UST[2], to determine if the Transmit Data register is empty (indicated by 1). If empty, continue to [Step 6](#). If the Transmit Data register is full (indicated by 0), continue to monitor the UST[2] bit until the Transmit Data register is available to receive new data.
5. Write the data byte to the UART Transmit Data register, 0F1h. The transmitter automatically transfers the data to the internal transmit shift register and transmits the data.
6. To transmit additional bytes, return to [Step 4](#).
7. Before disabling the transmitter, read the transmit completion status bit, UST[1]. If UST[1]=0, continue to monitor the bit until it changes to 1, which indicates that all data in the Transmit Data and internal shift registers has been transmitted.

**Caution:**

*Data written while the transmit enable bit is clear (UCTL[7]=0) will not be transmitted. Data written while the transmit data status bit is clear (UST[2]=0) overwrites the previous value written, so the previous written value will not be transmitted. Disabling the UART transmitter while the transmit completion status bit is clear (UST[1]=0) can corrupt the byte being transmitted.*

## 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 steps below to configure the UART for interrupt-driven data transmission:

1. Write to the BCNST register to set the appropriate baud rate.
2. Write 0 to bit 6 of the P01M register.
3. Execute DI instruction to disable interrupts.
4. Write to the Interrupt Control registers to enable the UART Transmitter interrupt and set the appropriate priority.
5. Write to the UART Control register to:
  - (a) Set the transmit enable bit (UCTL bit 7) to enable the UART for data transmission.
  - (b) Enable parity, if appropriate, and select either even- or odd-parity.
6. Execute an EI instruction to enable interrupts as the transmit buffer is empty, an interrupt is immediately executed.
7. Write the data byte to the UART Transmit Data register. The transmitter automatically transfers the data to the Internal Transmit Shift register and transmits the data.
8. Execute the IRET instruction to return from the interrupt service routine (ISR) and wait for the Transmit Data register to again become empty.

9. Before disabling the transmitter, read the transmit completion status bit, UST[1]. If UST[1]=0, continue to monitor the bit until it changes to 1, which indicates that all data in the Transmit Data and Internal Shift registers has been transmitted.

**Caution:**

*Data written while the transmit enable bit is clear (UCTL[7]=0) will not be transmitted. Data written while the transmit data status bit is clear (UST[2]=0) overwrites the previous value written, so the previous written value will not be transmitted. Disabling the UART transmitter while the transmit completion status bit is clear (UST[1]=0) can corrupt the byte being transmitted.*

## Receiving Data Using the Polled Method

Follow the steps below to configure the UART for polled data reception:

1. Write to the BCNST register to set the appropriate baud rate.
2. Write to the UART Control register (UCTL) to:
  - (a) Set the receive enable bit (UCTL[6]) to enable the UART for data reception
  - (b) Enable parity (if appropriate) and select either even- or odd-parity
3. Check the receive status bit in the UART Status register, bit UST[7], to determine if the Receive Data register contains a valid data byte (indicated by a 1). If UST[7] is set to 1 to indicate available data, continue to [Step 4](#). If the Receive Data register is empty (indicated by a 0), continue to monitor the UST[7] bit awaiting reception of the valid data.
4. Read data from the UART Receive Data register.
5. Return to [Step 3](#) to receive additional data.

## Receiving Data Using the Interrupt-Driven Method

The UART Receiver interrupt indicates the availability of new data (as well as error conditions).

Follow the steps below to configure the UART receiver for interrupt-driven operation:

1. Write to the UART BRG Constant registers to set the appropriate baud rate.
2. Execute DI instruction to disable interrupts.
3. Write to the Interrupt Control registers to enable the UART receiver interrupt and set the appropriate priority.
4. Clear the UART Receiver interrupt in the applicable Interrupt Request register.
5. Write to the UART Control register (UCTL) to:
  - (a) Set the receive enable bit (UCTL[6]) to enable the UART for data reception
  - (b) Enable parity, if appropriate, and select either even- or odd-parity
6. Execute an EI instruction to enable interrupts.

The UART is now configured for interrupt-driven data reception. When the UART Receiver interrupt is detected, the associated ISR performs the following:

1. Checks the UART Status register to determine the source of the interrupt, whether it is an error, break, or received data.
2. Reads the data from the UART Receive Data register, if the interrupt was caused by data available.
3. Clears the UART receiver interrupt in the applicable Interrupt Request register.
4. Executes the IRET instruction to return from the ISR and await more data.

## UART Interrupts

The UART features separate interrupts for the transmitter and the receiver. In addition, when the UART primary functionality is disabled, the BRG can also function as a basic timer with interrupt capability.

- **Note:** *When the UART is set to run at higher baud rates, the UART receiver's service routine may not have enough time to read and manipulate all bits in the UART Status register (especially bits generating error conditions) for a received byte before the next byte is received. You can devise your own hand-shaking protocol to prevent the transmitter from transmitting more data while current data is being serviced.*

### Transmitter Interrupts

The transmitter generates a single interrupt when the Transmit Status bit, UST[2], is set to 1. This indicates that the transmitter is ready to accept new data for transmission. The Transmit Status interrupt occurs after the internal transmit shift register has shifted the first bit of data out. At this point, the Transmit Data register can be written with the next character to send. This provides 7 bit periods of latency to load the Transmit Data register before the transmit shift register completes shifting the current character. Writing to the UART Transmit Data register clears the UST[2] bit to 0. The interrupt is cleared by writing a 0 to the Transmit Data register.

### Receiver Interrupts

The receiver generates an interrupt when any of the following occurs:

- **A data byte is received and available in the UART Receive Data register**—This interrupt can be disabled independent of the other receiver interrupt sources. The received data interrupt occurs once the receive character has been received and placed in the Receive Data register. Software must respond to this received data available condition before the next character is completely received to avoid an overrun error. The interrupt is cleared by reading from the UART Receive Data register.
- **A break is received**—A break is detected when a 0 is sent to the receiver for the full byte plus the parity and stop bits. After a break is detected, it will interrupt

immediately if there is no valid data in the Receive Data register. If data is present in the Receive Data register, an interrupt will occur after the UART Receive Data register is read.

- **An overrun is detected**—An overrun occurs when a byte of data is received while there is valid data in the UART Receive Data register that has not been read by the user. The interrupt will be generated when the user reads the UART Receive Data register. The interrupt is cleared by reading the UART Receive Data register. When an overrun error occurs, the additional data byte will not overwrite the data currently stored in the UART Receive Data register.
- **A data framing error is detected**—A data framing error is detected when the first stop bit is 0 instead of 1. When configured for 2 stop bits, a data framing error is only detected when the first stop bit is 0. A framing error interrupt is generated when the framing error is detected. Reading the UART Receive Data register clears the interrupt.

► **Note:** *Ensure that the transmitter uses the same stop bit configuration as the receiver.*

### UART Overrun Errors

When an overrun error condition occurs the UART prevents overwriting of the valid data currently in the Receive Data register. The Break Detect and Overrun status bits are not displayed until after the valid data has been read.

After the valid data has been read, the UART Status (UST) register is updated to indicate the overrun condition (and Break Detect, if applicable). The UST[7] bit is set to 1 to indicate that the Receive Data register contains a data byte. However, because the overrun error occurred, this byte may not contain valid data and must be ignored. The Break Detect bit, UST[3], indicates if the overrun was caused by a break condition on the line. After reading the status byte indicating an overrun error, the Receive Data register must be read again to clear the error bits in the UART Status 0 register. Updates to the Receive Data register occur only when the next data word is received.

### UART Data and Error Handling Procedure

Figure 25 on page 86 displays the recommended procedure for use in UART receiver interrupt service routine.

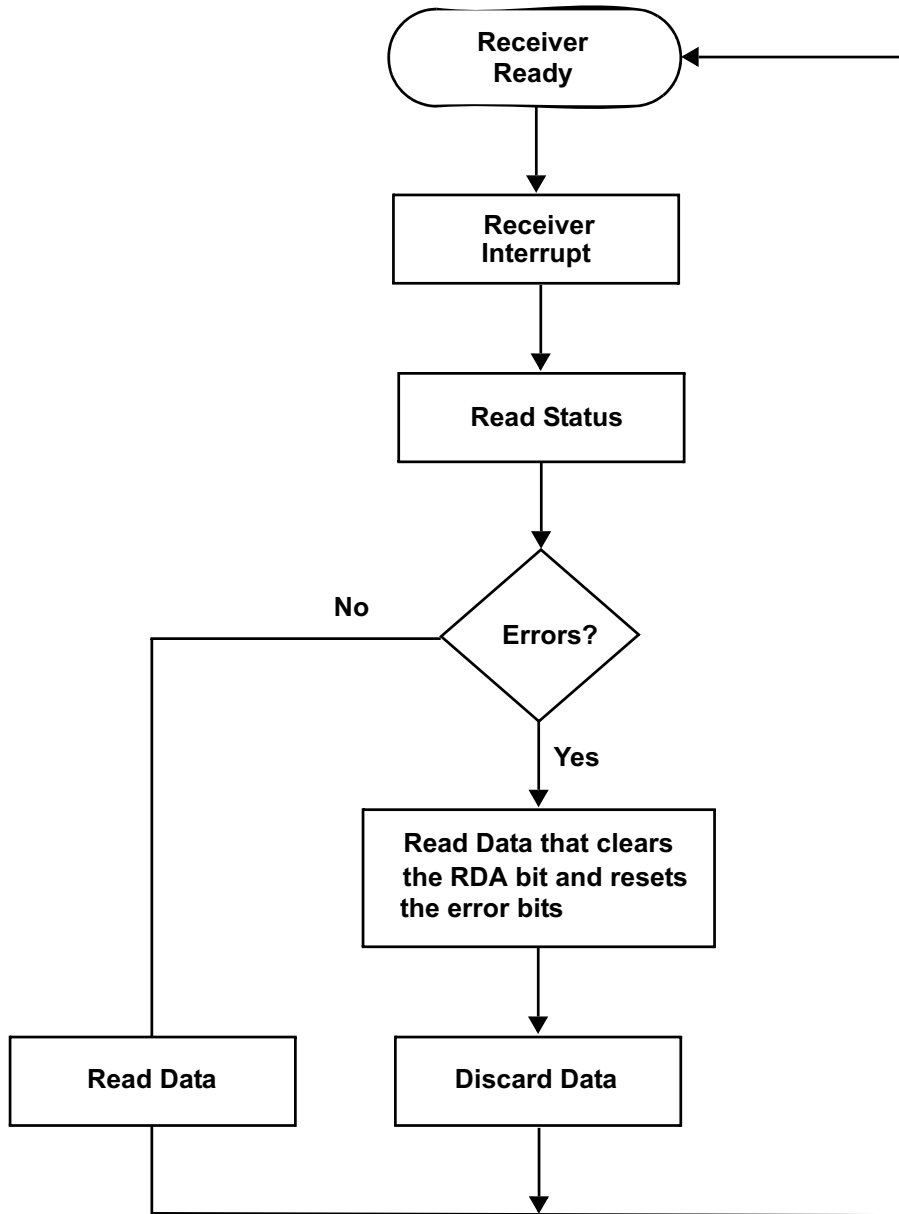


Figure 25. UART Receiver Interrupt Service Routine Flow

### Baud Rate Generator Interrupts

If the BRG interrupt enable is set, the UART Receiver interrupt asserts when the UART Baud Rate Generator reloads. This action allows the BRG to function as an additional counter if the UART functionality is not employed.

### UART Baud Rate Generator

The UART Baud Rate Generator creates a lower frequency baud rate clock for data transmission. The input to the Baud Rate Generator is the system clock. The UART Baud Rate Constant register contains an 8-bit baud rate divisor value (BCNST[7:0]) that sets the data transmission rate (baud rate) of the UART. For programmed register values other than 00h, the UART data rate is calculated using the below equation:

$$\text{UART Data Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value (BCNST)}}$$

When the UART Baud Rate Low register is programmed to 00h, the UART data rate is calculated as follows:

$$\text{UART Data Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{4096}$$

When the UART Baud Rate Generator is used as a general-purpose counter, the counters time-out period can be computed as follows based upon the counters clock input being a divide by 16 of the system clock and the maximum count value being 255:

$$\text{Time-Out Period (us)} = \frac{16 \times \text{UART Baud Rate Divisor Value (BCNST)}}{\text{System Clock Frequency (MHz)}}$$

► **Note:** *The relationship between the XTAL1 clock frequency and the system clock frequency must be considered before making this computation and is dependent upon the programming of bit 2 of User Option Byte 1 as well as the programming of bit 0 of the SMR register. Depending on the programmed values, the system clock frequency can be a divide by 1, a divide by 2, or a divide by 16 of the XTAL1 clock.*

When the UART is disabled, the BRG can function as a basic 8-bit timer with interrupt on time-out.

Follow the steps below to configure the BRG as a timer with interrupt on time-out:

1. Disable the UART by clearing the receive and transmit enable bits, UCTL[7:6] to 0.
2. Load the appropriate 8-bit count value into the UART Baud Rate Generator Constant register. The count frequency is the system clock frequency (in Hz) divided by 16.
3. Enable the Baud Rate Generator timer function and associated interrupt by setting the Baud Rate Generator bit (UCTL bit 0) in the UART Control register to 1. When configured as an 8-bit timer, the count value, instead of the reload value, is read, and the counter begins counting down from its initial programmed value. On timing out (reaching a value of 1), if the time-out interrupt is enabled, an interrupt will be produced. The counter will then reload its programmed start value and begin counting down again.

Table 43 lists a number of BCNST register settings at various baud rates and system clock frequencies.

**Table 43. BCNST Register Settings Examples**

Target UART Data Rate (baud)	System Clock = 4 MHz, Crystal Clock = 8 MHz	System Clock = 3 MHz, Crystal Clock = 6 MHz
2400	BCNST = 01101000 Actual baud rate = 2403	BCNST = 01001110 Actual baud rate = 2403
4800	BCNST = 00110100 Actual baud rate = 4807	BCNST = 00100111 Actual baud rate = 4807
9600	BCNST = 00011010 Actual baud rate = 9615	BCNST = 00010100 Actual baud rate = 9375
19200	BCNST = 00001101 Actual baud rate = 19230	BCNST = 00001010 Actual baud rate = 18750



## UART Receive Data Register/UART Transmit Data Register

The UART Receive/Transmit Data register (see [Table 44](#)) is used to send and retrieve data from the UART channel. When the UART receives a data byte, it can be read from this register. The UART receive interrupt is cleared when this register is used. Data written to this register is transmitted by the UART.

**Table 44. UART Receive/Transmit Data Register (URDATA/UTDATA)**

Bit	7	6	5	4	3	2	1	0
Field	UART Receive/Transmit							
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F1h; Linear: 0F1h							

Bit Position	Description
--------------	-------------

[7:0]	<p><b>UART Receive/Transmit</b> When read, returns received data. When written, transmits written data.</p>
-------	---

## UART Status Register

The UART Status register (see [Table 45](#)) displays the status of the UART. Bits [6:3] are cleared by reading the UART Receive/Transmit register (F1h).

**Table 45. UART Status Register (UST)**

Bit	7	6	5	4	3	2	1	0
Field	Receive Status	Parity Error	Overrun Error	Framing Error	Break	Transmit Data	Transmit Complete	Noise Filter
Reset	0	0	0	0	0	0	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F2h; Linear: 0F2h							

Bit Position	Value	Description
--------------	-------	-------------

[7]		<b>Receive Status</b> —Set when data is received; cleared when URDATA is read.
	0	UART Receive Data register empty.
	1	UART Receive Data register full.

Bit Position	Value	Description
[6]	0	<b>Parity Error</b> —Set when a parity error occurs; cleared when URDATA is read. No parity error occurs.
	1	Parity error occurs.
[5]	0	<b>Overrun Error</b> —Set when an overrun error occurs; cleared when URDATA is read. No overrun error occurs.
	1	Overrun error occurs.
[4]	0	<b>Framing Error</b> —Set when a framing error occurs; cleared when URDATA is read. No framing error occurs.
	1	Framing error occurs.
[3]	0	<b>Break</b> —Set when a break is detected; cleared when URDATA is read. No break occurs.
	1	Break occurs.
[2]	0	<b>Transmit Data Status</b> —Set when the UART is ready to transmit; cleared when TRDATA is written. Do not write to the UART Transmit Data register.
	1	UART Transmit Data register ready to receive additional data.
[1]	0	<b>Transmit Completion Status</b> Data is currently transmitting.
	1	Transmission is complete.
[0]	Read	<b>Noise Filter</b> —Detects noise during data reception.
	0	No noise detected.
	1	Noise detected.
	Write	
	0	Turn off noise filter.
	1	Turn on noise filter.

## UART Control Register

The UART Control register controls the UART. In addition to setting bit 5, you must also set appropriate bit in the Interrupt Mask register (see [Table 66](#) on page 127).

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 46. UART Control Register (UCTL)**

Bit	7	6	5	4	3	2	1	0
Field	Transmitter Enable	Receiver Enable	UART Interrupts Enable	Parity Enable	Parity Select	Send Break	Stop Bits	Baud Rate Generator
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F3h; Linear: 0F3h							

Bit Position	Value	Description
[7]	0 1	Transmitter disabled. Transmitter enabled.
[6]	0 1	Receiver disabled. Receiver enabled.
[5]	0 1	UART Interrupts disabled. UART Interrupts enabled.
[4]	0 1	Parity disabled. Parity enabled.
[3]	0 1	Even parity selected. Odd parity selected.
[2]	0 1	No break is sent. Send Break (force Tx output to 0).
[1]	0 1	One stop bit. Two stop bits.
[0]	0 1	<b>Baud Rate Generator</b> —When the transmitter and receiver are disabled, the BRG can be used as an additional timer. When setting this bit, clear bits [7:6] in this register. Also set bit [5] if an interrupt is required when the BRG is reloaded. 0 BRG used as Baud Rate Generator for UART. 1 BRG used as timer.

## UART Baud Rate Generator Constant Register

The UART baud rate generator determines the frequency at which UART data is received and transmitted. This baud rate is determined by the following equation:

$$\text{UART Data Rate (bps)} = \frac{\text{System Clock Frequency (Hz)}}{16 \times \text{UART Baud Rate Divisor Value (BCNST)}}$$

The system clock is usually the crystal clock divided by 2. When the UART baud rate generator is used as an additional timer, a Read from this register returns the actual value of the count of the BRG in progress and not the reload value. See [Table 47](#).

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 47. UART Baud Rate Generator Constant Register (BCNST)**

Bit	7	6	5	4	3	2	1	0
Field	Baud Rate Generator Constant							
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: F4h; Linear: 0F4h							

**Bit Position Description**

[7:0]	<p><b>Baud Rate Generator Constant</b> When read, returns the actual timer count value (when UCTL[0]=1). When written, sets the Baud Rate Generator Constant. The actual baud rate frequency = XTAL ÷ (32 x BCNST).</p>
-------	---

# Timers

The ZLF645 MCU infrared timer features a 16-bit and an 8-bit counter/timer, each of which can be used simultaneously for transmitting. Both timers can be used for demodulating an input carrier wave and share a single input pin.

Figure 26 displays the counter/timer architecture, which is designed to help unburden the program from coping with real-time problems like generating complex waveforms or receiving and demodulating complex waveforms and pulses.

In addition to the 16-bit and 8-bit timers, the UART's baud rate generator can be used as an additional 8-bit timer when the UART receiver is not in use (for more details, see [Universal Asynchronous Receiver/Transmitter](#) on page 79).

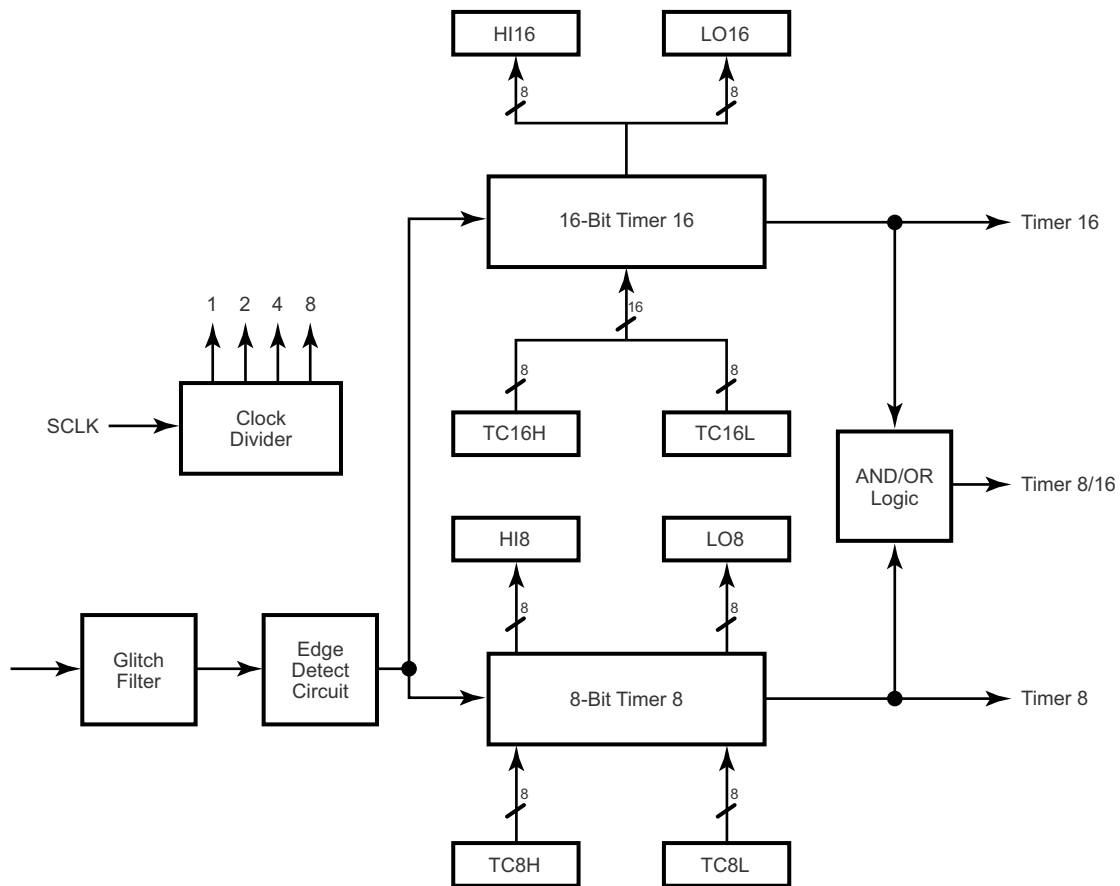


Figure 26. Counter/Timers Block Diagram

Table 48 summarizes the timer control registers. Some timer functions can also be affected by control registers for other peripheral functions.

**Table 48. Timer Control Registers**

Address (Hex)							Page No
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset		
D00	D	00	Counter/Timer 8 Control Register	CTR0	0000_0000b		113
D01	D	01	Timer 8 and Timer 16 Common Functions	CTR1	0000_0000b		115
D02	D	02	Counter/Timer 16 Control Register	CTR2	0000_0000b		118
D03	D	03	Timer 8/Timer 16 Control Register	CTR3	0000_0XXXb		119
D04	D	04	Counter/Timer 8 Low Hold Register	TC8L	00h		112
D05	D	05	Counter/Timer 8 High Hold Register	TC8H	00h		111
D06	D	06	Counter/Timer 16 Low Hold Register	TC16L	00h		111
D07	D	07	Counter/Timer 16 High Hold Register	TC16H	00h		110
D08	D	08	Timer 16 Capture Low Register	LO16	00h		110
D09	D	09	Timer 16 Capture High Register	HI16	00h		109
D0A	D	0A	Timer 8 Capture Low Register	LO8	00h		109
D0B	D	0B	Timer 8 Capture High Register	HI8	00h		108

## Counter/Timer Functional Blocks

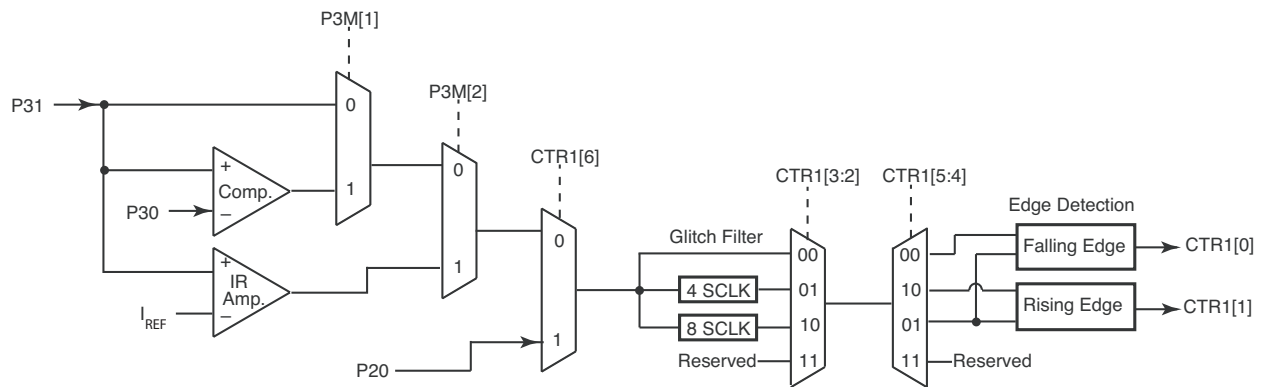
The ZLF645 MCU infrared timer contains a glitch filter for removing noise from the input when demodulating an input carrier. Each timer features its own demodulating mode and can be simultaneously used to generate a signal output. The T8 timer has the ability to capture only one cycle of a carrier wave of a high-frequency waveform.

### Input Circuit

Depending on the setting of register bits P3M[2:1] and CTR1[6], the timer/counter input circuit monitors one of the following conditions:

- The P31 digital signal, if CTR1[6]=0 and P3M[2:1]=00.
- The P31 analog comparator output, if CTR1[6]=0 and P3M[2:1]=01.
- The P31 IR amplifier output, if CTR1[6]=0 and P3M[2]=1.
- The P20 digital signal, if CTR1[6]=1.

Based on register bits CTR1[5:4], a pulse is generated at when a rising edge, falling edge, or any edge is detected. Glitches in the input signal are filtered out if they are shorter than the glitch filter width specified in register bits CTR1[3:2]. [Figure 27](#) displays the input circuit.



**Figure 27. Counter/Timer Input Circuit**

The timers can be configured to operate in following modes:

- [T8 TRANSMIT Mode](#)
- [T8 DEMODULATION Mode](#)
- [T16 TRANSMIT Mode](#)
- [T16 DEMODULATION Mode](#)
- [PING-PONG Mode](#)

### T8 TRANSMIT Mode

Before T8 is enabled, the output of T8 depends on CTR1, bit 1. If the bit is 0, T8\_OUT is 1; if it is 1, T8\_OUT is 0. See [Figure 28](#).

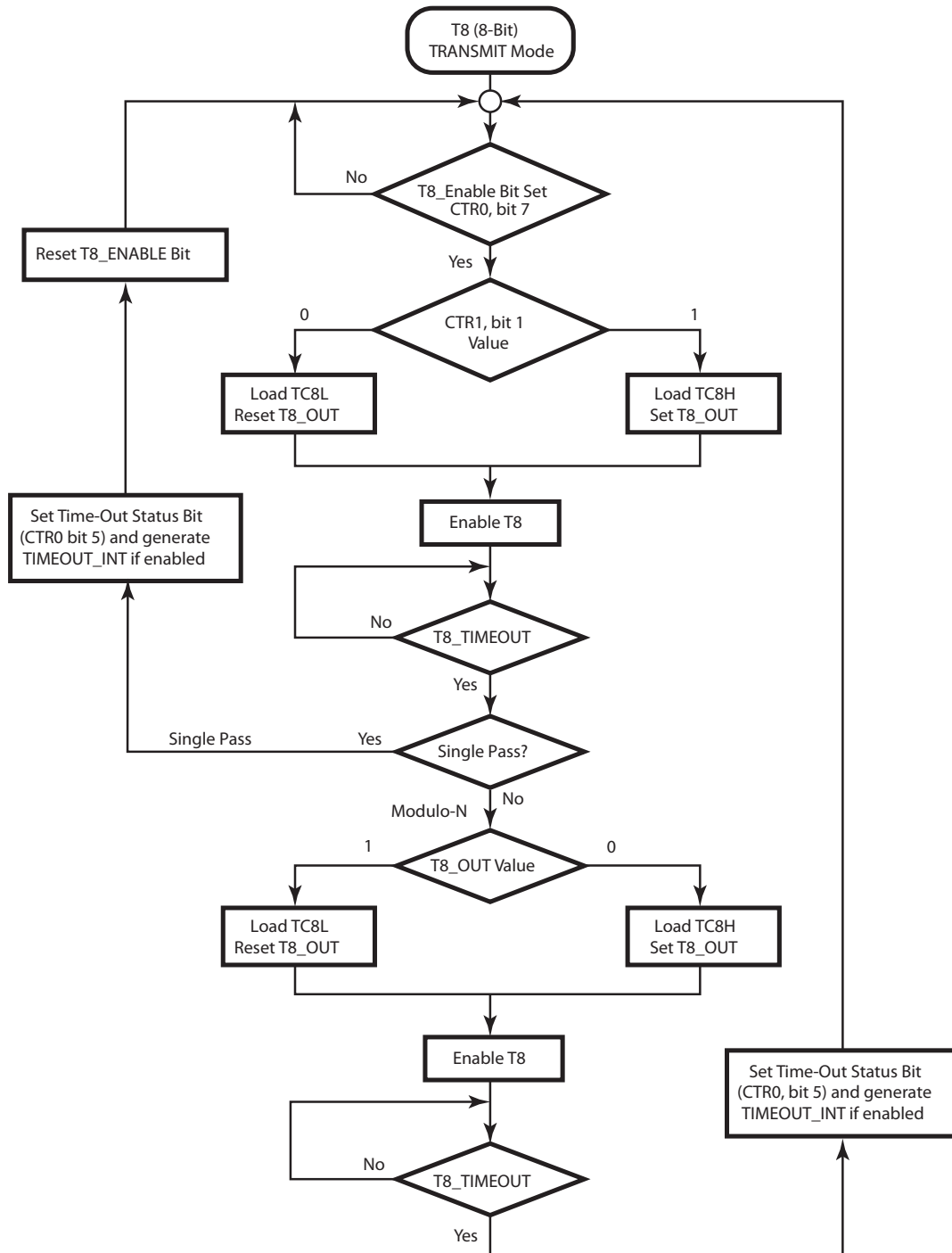


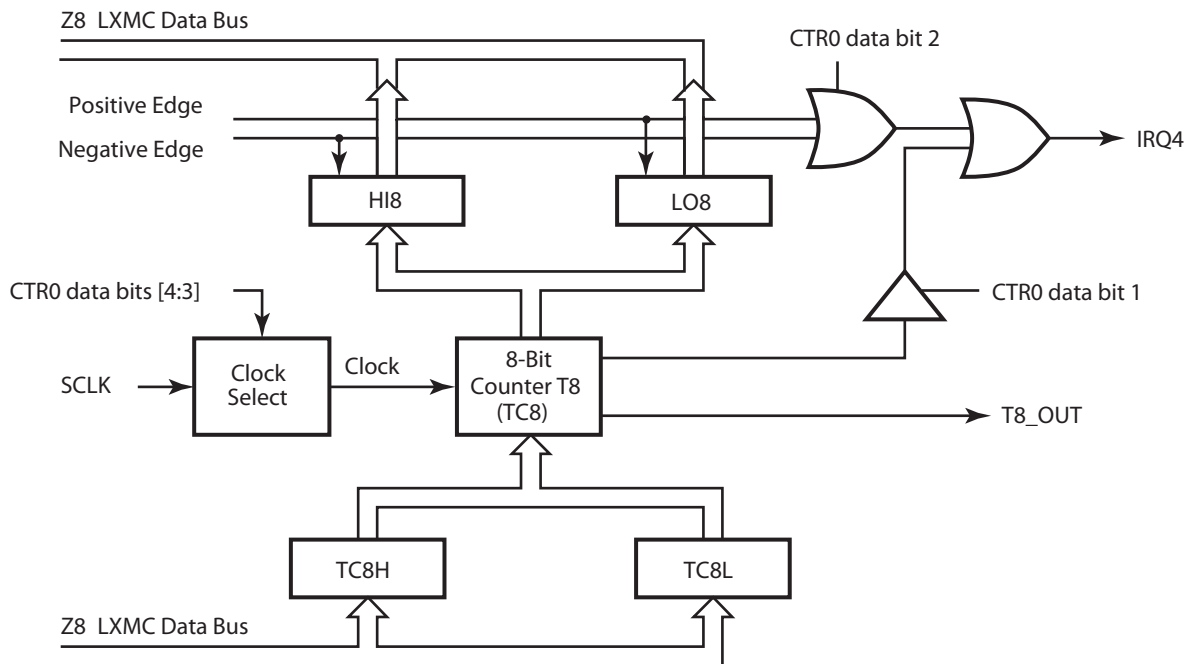
Figure 28. TRANSMIT Mode Flowchart



When T8 is enabled, the output T8\_OUT switches to the initial value (CTR1, bit 1). If the initial value (CTR1, bit 1) is 0, TC8L is loaded; otherwise, TC8H is loaded into the counter.

In SINGLE-PASS mode (CTR0, bit 6), T8 counts down to 0 and stops, T8\_OUT toggles, the time-out status bit (CTR0, bit 5) is set, and a time-out interrupt can be generated if it is enabled (CTR0, bit 1).

In MODULO-N mode, on reaching terminal count, T8\_OUT is toggled, but no interrupt is generated. From that point, T8 loads a new count (if the T8\_OUT level now is 0), TC8L is loaded; if it is 1, TC8H is loaded. T8 counts down to 0, toggles T8\_OUT, and sets the time-out status bit (CTR0, bit 5), thereby generating an interrupt if enabled (CTR0, bit 1). One cycle is thus completed. T8 then loads from TC8H or TC8L according to the T8\_OUT level and repeats the cycle. See [Figure 29](#).



**Figure 29. 8-Bit Counter/Timer Circuits**

You can modify the values in TC8H or TC8L at any time. The new values take effect when they are loaded.



**Caution:** *An initial count of 1 is not allowed (a non-function occurs). An initial count of 0 causes TC8 to count from 0 to FFh to FEh.*

- **Notes:**
1. The “h” suffix denotes hexadecimal values.
  2. Transition from 0 to FFh is not a time-out condition.

⚠ **Caution:** Using the same instructions for stopping the counter/timers and setting the status bits is not recommended.

Two successive commands are necessary. First, the counter/timers must be stopped. Second, the status bits must be reset. These commands are required as it takes one counter/timer clock interval for the initiated event to actually occur. See [Figure 30](#) and [Figure 31](#).

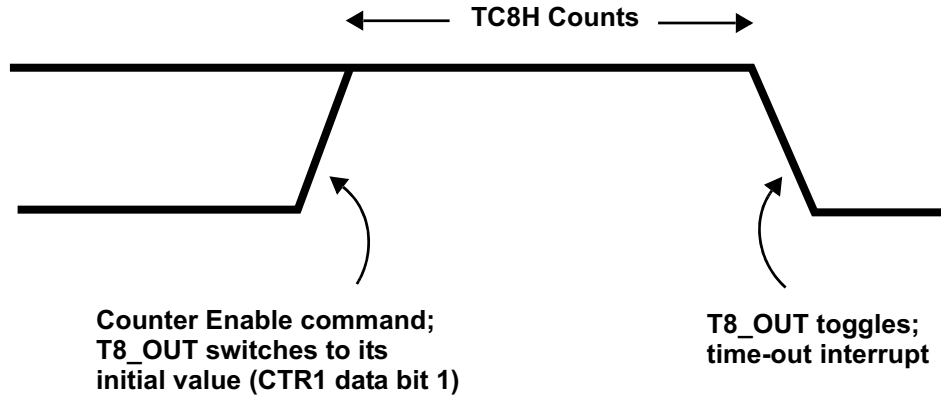


Figure 30. T8\_OUT in SINGLE-PASS Mode

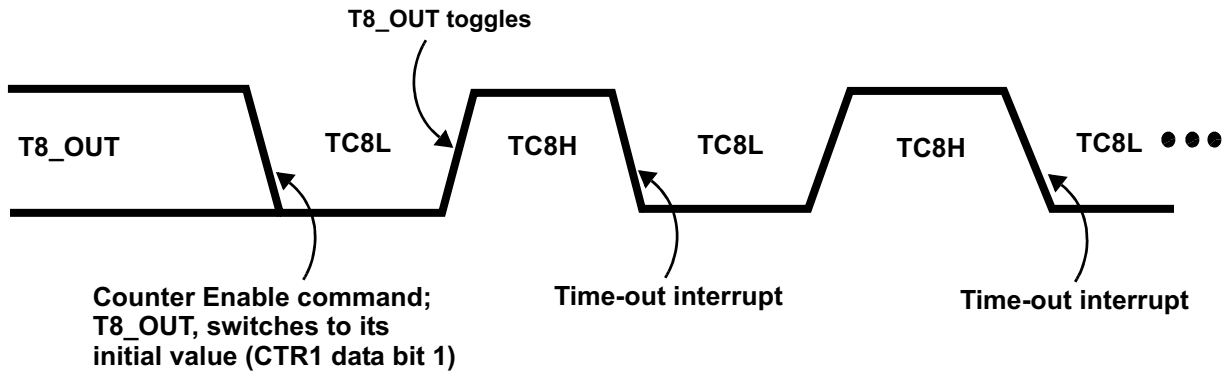
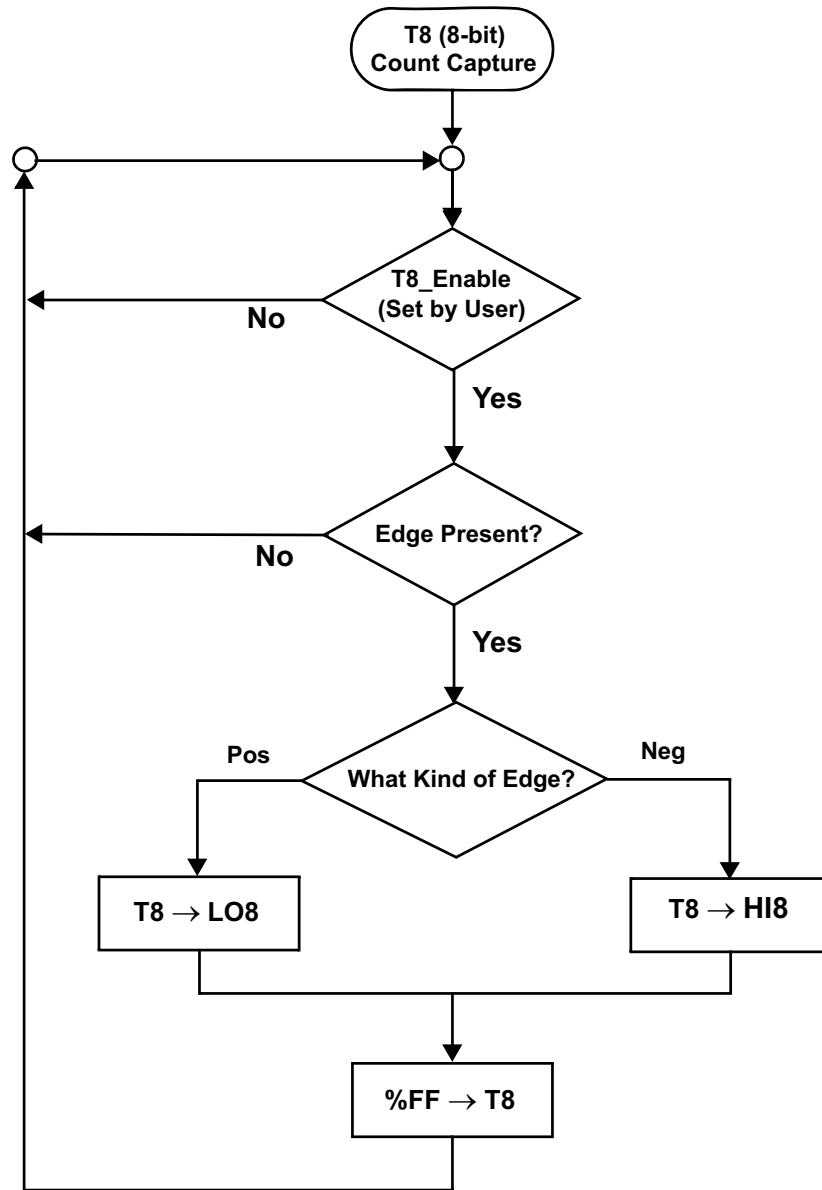


Figure 31. T8\_OUT in MODULO-N Mode

## T8 DEMODULATION Mode

You must program TC8L and TC8H to FFh. After T8 is enabled, when the first edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected, it starts to count down. When a subsequent edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected during counting, the current value of T8 is complemented and put into one of the capture registers.

If it is a positive edge, data is put into LO8; if it is a negative edge, data is put into HI8. From that point, one of the edge detect status bits (CTR1, bits [1:0]) is set, and an interrupt can be generated if enabled (CTR0, bit 2). Meanwhile, T8 is loaded with FFh and starts counting again. If T8 reaches 0, the time-out status bit (CTR0, bit 5) is set, and an interrupt can be generated if enabled (CTR0, bit 1). T8 then continues counting from FFh (see [Figure 32](#) on page 100).



**Figure 32. DEMODULATION Mode Count Capture Flowchart**

When bit 4 of CTR3 is enabled, the flow of the demodulation sequence is altered. The third edge makes T8 active, and the fourth and fifth edges are captured. The capture interrupt is activated after the fifth event occurs. This mode is useful for capturing the carrier duty cycle as well as the frequency at which the first cycle is corrupted. See [Figure 33](#) and [Figure 34](#).

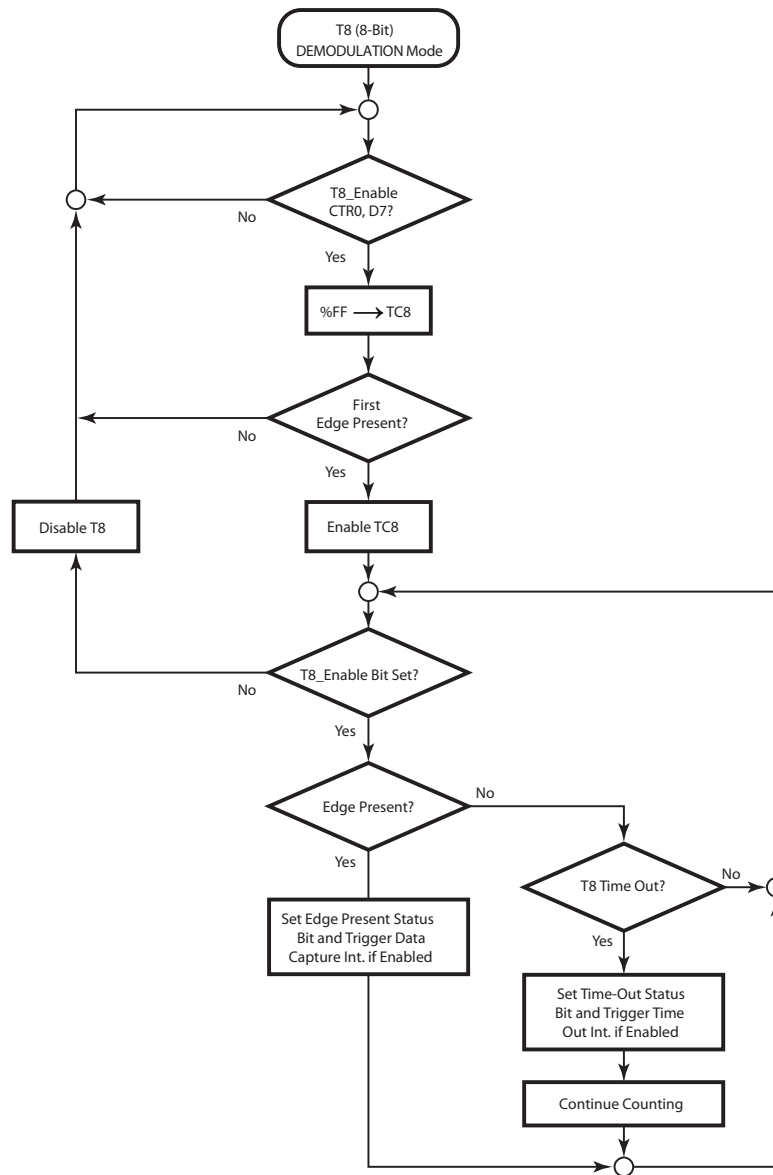


Figure 33. DEMODULATION Mode Flowchart

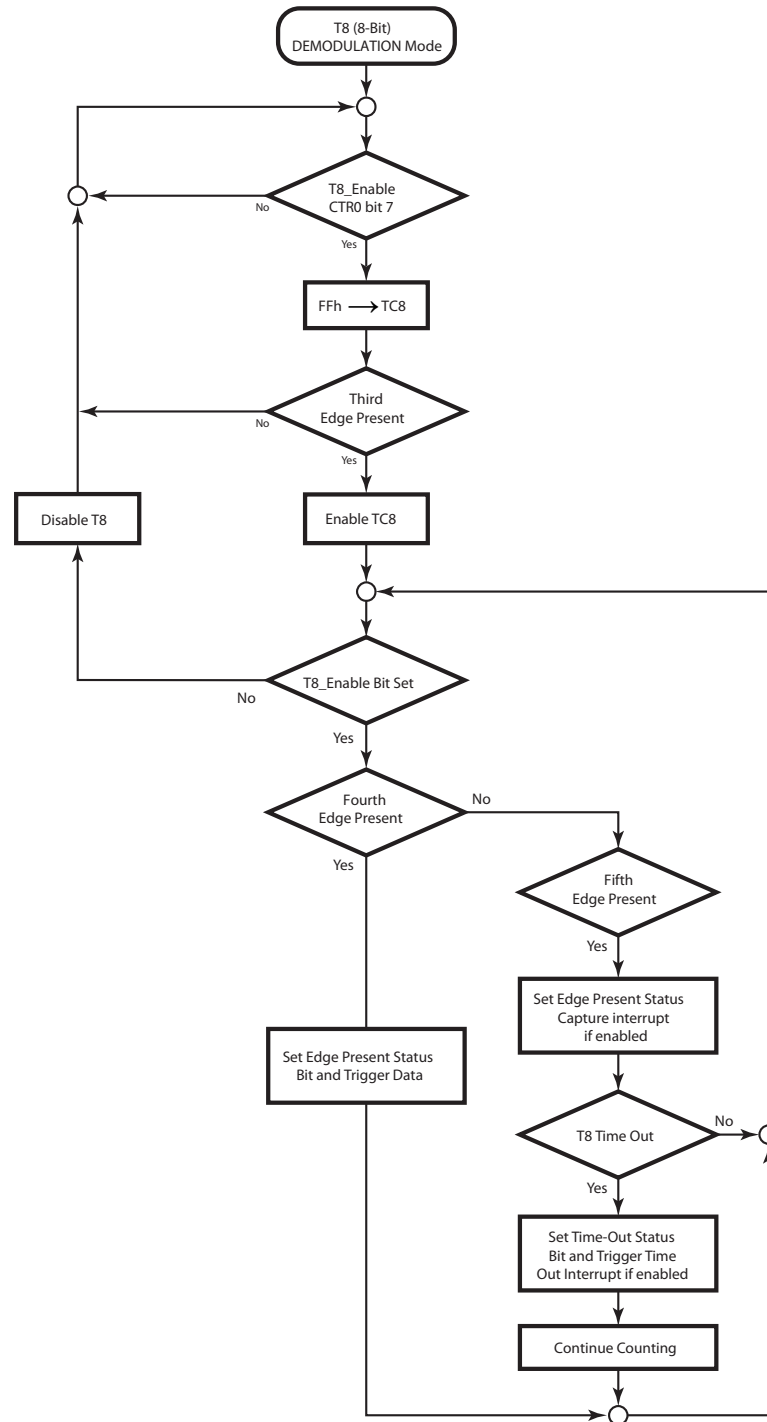


Figure 34. DEMODULATION Mode Flowchart with Bit 4 of CTR3 Set

## T16 TRANSMIT Mode

In NORMAL or PING-PONG mode, the output of T16 when not enabled depends on CTR1, bit 0. If this bit is set to 0, T16\_OUT is a 1; if set to 1, T16\_OUT is 0. You can force the output of T16 to either 0 or 1 whether it is enabled or not by programming CTR1 bits [3:2] to a 10 or 11.

When bit 4 of CTR3 is set, the T16 output does not update. However, time-out interrupts (flags) are still updated. In addition, the T8 carrier is not disrupted by timing out of the T16 timer.

When T16 is enabled, a value of  $(TC16H * 256) + TC16L$  is loaded, and T16\_OUT is switched to its initial value (CTR1, bit 0). When T16 counts down to 0, T16\_OUT is toggled (in NORMAL or PING-PONG mode), an interrupt (CTR2, bit 1) is generated (if enabled), and a status bit (CTR2, bit 5) is set. See [Figure 35](#).

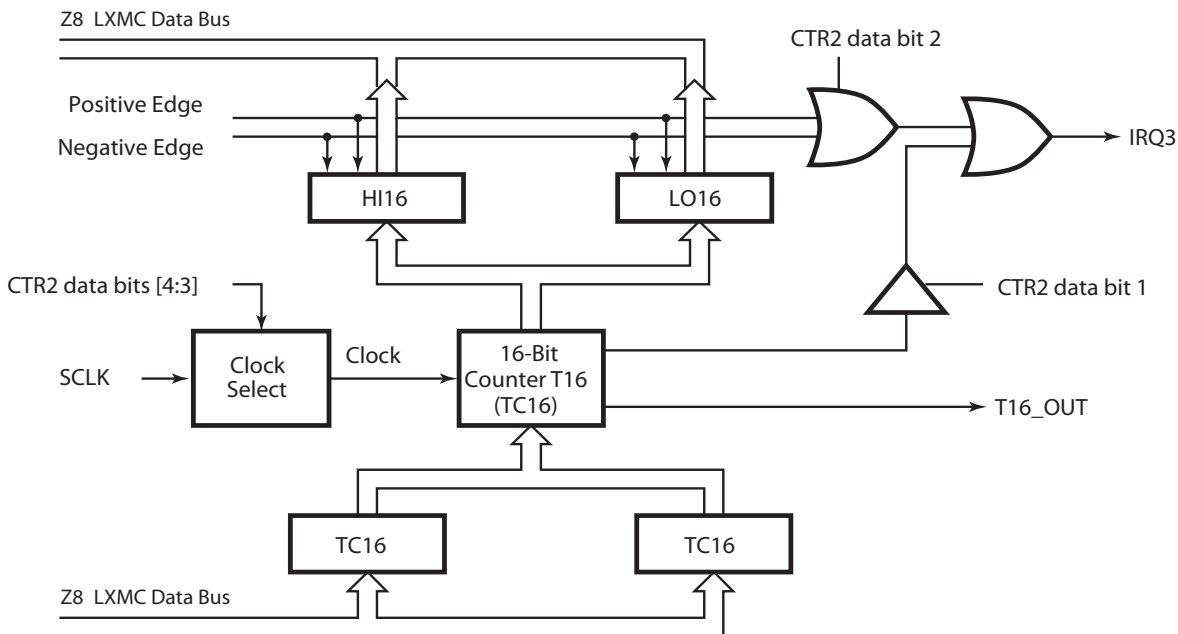


Figure 35. 16-Bit Counter/Timer Circuits

► **Note:** *Global interrupts override this function as described in the [Interrupts](#) on page 121.*

If T16 is in SINGLE-PASS mode, it is stopped at this point (see [Figure 36](#) on page 104). If it is in MODULO-N mode, it is loaded with  $TC16H * 256 + TC16L$ , and the counting continues (see [Figure 37](#) on page 104). You can modify the values in TC16H and TC16L at any time. The new values take effect when they are loaded.



**Caution:** Do not load these registers at the time the values are to be loaded into the counter/timer to ensure known operation. An initial count of 1 is not allowed. An initial count of 0 causes T16 to count from 0 to FFFEh. Transition from 0 to FFFFh is not a time-out condition.

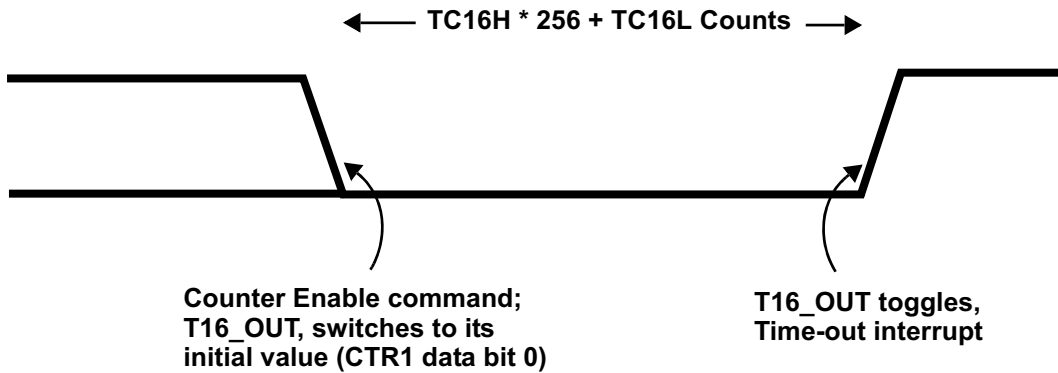


Figure 36. T16\_OUT in SINGLE-PASS Mode

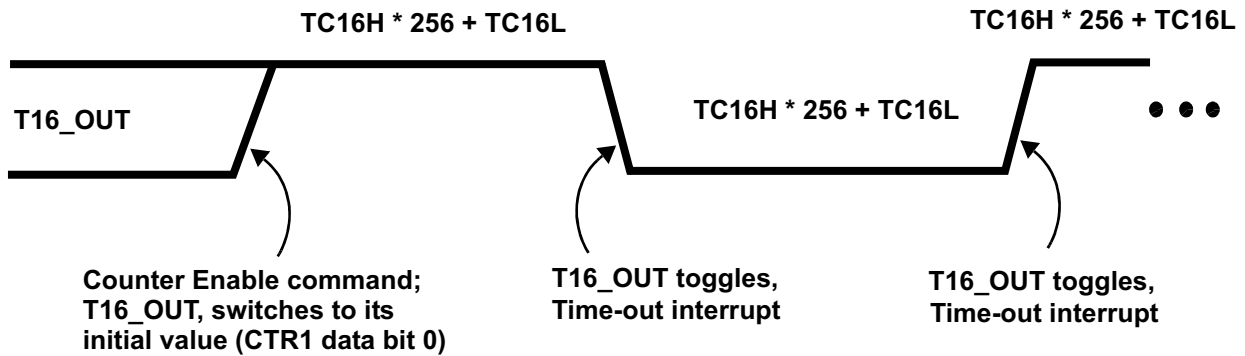


Figure 37. T16\_OUT in MODULO-N Mode

## T16 DEMODULATION Mode

You must program TC16L and TC16H to FFh. After T16 is enabled, and the first edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected, T16 captures HI16 and LO16, reloads, and begins counting.



**If Bit 6 of CTR2 Is 0**—When a subsequent edge (rising, falling, or both depending on CTR1 bits [5:4]) is detected during counting, the current count in T16 is complemented and loaded into HI16 and LO16. When data is captured, one of the edge detect status bits (CTR1, bit 1; bit 0) is set, and an interrupt is generated if enabled (CTR2, Bit 2). T16 is loaded with `FFFFh` and starts again. This T16 mode is generally used to measure space time, the length of time between bursts of carrier signal (marks).

**If Bit 6 of CTR2 Is 1**—T16 ignores the subsequent edges in the input signal and continues counting down. A time-out of T8 causes T16 to capture its current value and generate an interrupt if enabled (CTR2, Bit 2). In this case, T16 does not reload and continues counting. If CTR2 bit 6 is toggled (by writing a 0 then a 1 to it), T16 captures and reloads on the next edge (rising, falling, or both depending on CTR1 bits [5:4]), continuing to ignore subsequent edges. This T16 mode generally measures mark time, the length of an active carrier signal burst.

If T16 reaches 0, T16 continues counting from `FFFFh`. Meanwhile, a status bit (CTR2 bit 5) is set, and an interrupt time-out can be generated, if enabled (CTR2 bit 1).

## PING-PONG Mode

PING-PONG mode is only valid in TRANSMIT mode. T8 and T16 must be programmed in SINGLE-PASS mode (CTR0, bit 6; CTR2, bit 6), and PING-PONG mode must be programmed in CTR1 bits [3:2]. You can begin the operation by enabling either T8 or T16 (CTR0, D7 or CTR2, D7). For example, if T8 is enabled, T8\_OUT is set to this initial value (CTR1, bit 1).

According to T8\_OUT's level, TC8H or TC8L is loaded into T8. After the terminal count is reached, T8 is disabled, and T16 is enabled. T16\_OUT then switches to its initial value (CTR1, bit 0), data from TC16H and TC16L is loaded, and T16 starts to count. After T16 reaches the terminal count, it stops, T8 is enabled again, repeating the entire cycle. Interrupts can be allowed when T8 or T16 reaches terminal control (CTR0, bit 1; CTR2, bit 1). To stop the PING-PONG operation, write 00 to bits CTR1 bits [3:2]. See [Figure 38](#) on page 106.

- **Note:** *Enabling PING-PONG operation while the counter/timers are running may cause intermittent counter/timer function. Disable the counter/timers and reset the status flags before instituting this operation.*

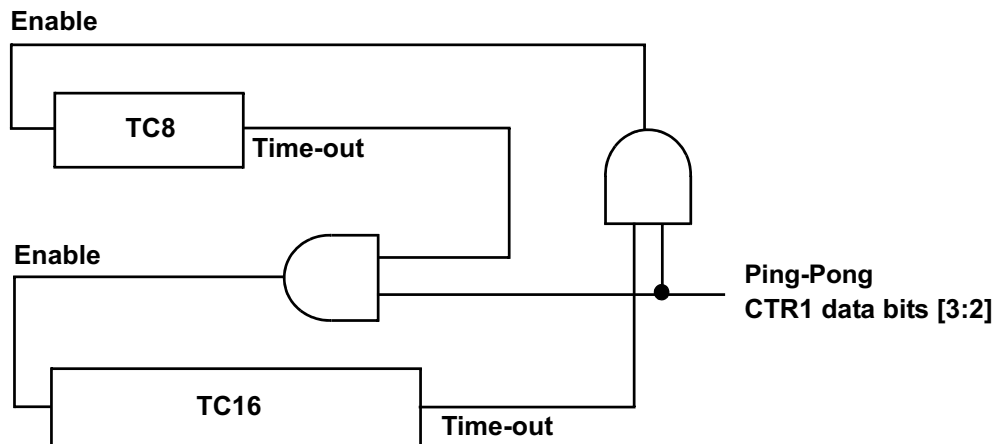


Figure 38. PING-PONG Mode Diagram

### Initiating PING-PONG Mode

First, ensure that both counter/timers are not running.

Follow the steps below to initiate the PING-PONG mode:

1. Set T8 into SINGLE-PASS mode (CTR0, bit 6)
2. Set T16 into SINGLE-PASS mode (CTR2, bit 6)
3. Set the PING-PONG mode (CTR1 bits [3:2])

These instructions are not consecutive and can occur in random order.

4. Finally, start PING-PONG mode by enabling either T8 (CTR0, D7) or T16 (CTR2, D7). The initial value of T8 or T16 must not be 1.

If you stop the timer and restart the timer, reload the initial value to avoid an unknown previous value.

### During PING-PONG Mode

The enable bits of T8 and T16 (CTR0, D7; CTR2, D7) are set and cleared alternately by hardware. The time-out bits (CTR0, bit 5; CTR2, bit 5) are set every time the counter/timers reach the terminal count.

### Timer Output

The output logic for the timers is displayed in [Figure 39](#) on page 107. P34 is used to output T8\_OUT when bit 0 of CTR0 is set. P35 is used to output the value of T16\_OUT when bit

0 of CTR2 is set. When bit 6 of CTR1 is set, P36 outputs the logic combination of T8\_OUT and T16\_OUT via bits [4:5] of CTR1.

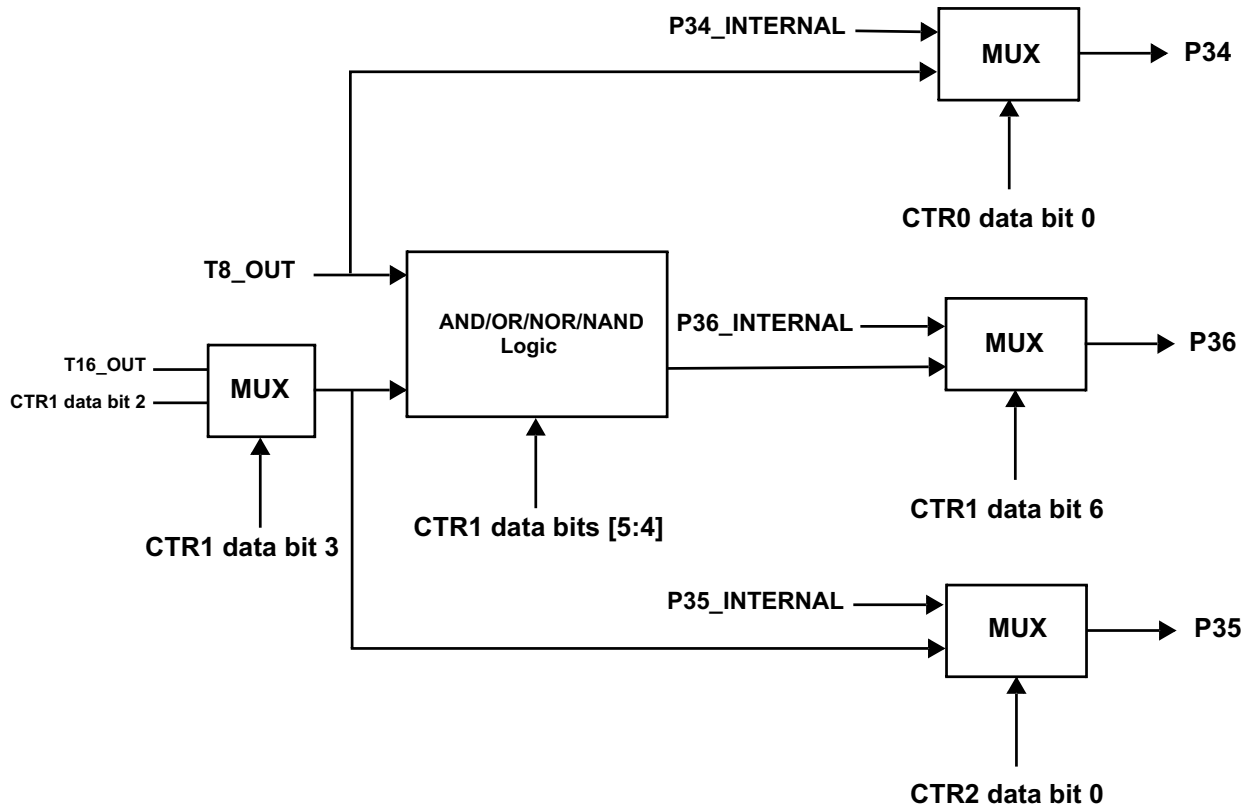


Figure 39. Timer Output Circuit

## Counter/Timer Registers

The following sections describe each of the Timer/Counter registers in detail.

### Timer 8 Capture High Register

The Timer 8 Capture High register (see [Table 49](#)) holds the captured data from the output of the 8-bit Counter/Timer 0. This register contains the number of counts when the input signal is 1.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 49. Timer 8 Capture High Register (HI8)**

Bit	7	6	5	4	3	2	1	0
Field	T8_Capture_HI							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 0Bh; Linear: D0Bh							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T8_Capture_HI</b> —Reads return captured data. Writes have no effect.

### Timer 8 Capture Low Register

The Timer 8 Capture Low register (see [Table 50](#) on page 109) holds the captured data from the output of the 8-bit Counter/Timer 0. Typically, this register contains the number of counts when the input signal is 0.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 50. Timer 8 Capture Low Register (LO8)**

Bit	7	6	5	4	3	2	1	0
Field	T8_Capture_LO							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 0Ah; Linear: D0Ah							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T8_Capture_LO</b> —Read returns captured data. Writes have no effect.

### Timer 16 Capture High Register

The Timer 16 Capture High register (see [Table 51](#)) holds the captured data from the output of the 16-bit Counter/Timer 16. This register contains the most significant byte (MSB) of the data.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 51. Timer 16 Capture High Register (HI16)**

Bit	7	6	5	4	3	2	1	0
Field	T16_Capture_HI							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 09h; Linear: D09h							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T16_Capture_HI</b> —Read returns captured data. Writes have no effect.

## Timer 16 Capture Low Register

The Timer 16 Capture Low register (see [Table 52](#)) holds the captured data from the output of the 16-bit Counter/Timer 16. This register contains the least significant byte (LSB) of the data.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 52. Timer 16 Capture Low Register (LO16)**

Bit	7	6	5	4	3	2	1	0
Field	T16_Capture_LO							
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Address	Bank D: 08h; Linear: D08h							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T16_Capture_LO</b> —Read returns captured data. Writes have no effect.

## Counter/Timer 16 High Hold Register

The Counter/Timer 16 High Hold register (see [Table 53](#)) contains the high byte of the value loaded into the T16 timer.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 53. Counter/Timer 16 High Hold Register (TC16H)**

Bit	7	6	5	4	3	2	1	0
Field	T16_Data_HI							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank D: 07h; Linear: D07h							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T16_Data_HI</b> —Read/Write Data.

## Counter/Timer 16 Low Hold Register

The Counter/Timer 16 Low Hold register (see [Table 54](#)) contains the low byte of the value loaded into the T16 timer.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 54. Counter/Timer 16 Low Hold Register (TC16L)**

Bit	7	6	5	4	3	2	1	0
Field	T16_Data_LO							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank D: 06h; Linear: D06h							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T16_Data_LO</b> —Read/Write Data.

## Counter/Timer 8 High Hold Register

The Counter/Timer 8 High Hold register (see [Table 55](#)) contains the value to be counted while the T8 output is 1.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 55. Counter/Timer 8 High Hold Register (TC8H)**

Bit	7	6	5	4	3	2	1	0
Field	T8_Level_HI							
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank D: 05h; Linear: D05h							

Bit Position	Value	Description
[7:0]	0hh–FFh	<b>T8_Level_HI</b> —Read/Write Data.



### Counter/Timer 8 Low Hold Register

The Counter/Timer 8 Low Hold register (see [Table 56](#)) contains the value to be counted while the T8 output is 0.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 56. Counter/Timer 8 Low Hold Register (TC8L)**

Bit	7	6	5	4	3	2	1	0
Field	T8_Level_LO							
Reset	0	0	0	0	0	0	0	0
R/W	Bank D: 04h; Linear: D04h							
Address	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Position	Value	Description
[7:0]	0hh–FFh	T8_Level_LO—Read/Write Data.



## Counter/Timer 8 Control Register

The Counter/Timer 8 Control register (see [Table 57](#)) controls the timer function of the T8 timer.



**Caution:** *Writing 1 to CTR0[5] is the only way to reset the Terminal Count status condition. Reset this bit before using/enabling the counter/timers.*



**Note:** *You must be careful when using the OR or AND commands to manipulate CTR0, bit 5 and CTR1, bits 0 and 1 (DEMODULATION Mode). These instructions use a Read-Modify-Write sequence in which the current status from the CTR0 and CTR1 registers is ORed or ANDed with the designated value and then written back into the registers.*

**Example:** When the status of bit 5 is 1, a timer reset condition occurs.

**Table 57. Counter/Timer 8 Control Register (CTR0)**

Bit	7	6	5	4	3	2	1	0
Field	T8_Enable	SINGLE-PASS/MODULO-N	Time_Out	T8_Clock		Capture_INT_Mask	Counter_INT_Mask	P34_Out
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank D: 00h; Linear: D00h							

Bit Position	Value	Description
[7]	0	<b>T8_Enable—Disable/enable the T8 counter.</b> Disable counter.
	1	Enable counter. Configure T8 before enabling it.
[6]	0	<b>SINGLE-PASS/MODULO-N</b> MODULO-N mode. Counter reloads the initial value when terminal count is reached
	1	SINGLE-PASS mode. Counter stops when the terminal count is reached
[5]	Read	<b>Time_Out</b> —This bit is set when the T8 terminal count is reached.
	0	No counter time-out occurred.
	1	Counter time-out occurred.
	Write	
	0	No effect.
	1	Reset flag to 0. Software must reset this flag before using counter/timers.



Bit Position	Value	Description
[4:3]		<b>T8_Clock</b> —Select the T8 input clock frequency. These bits are not reset upon Stop Mode Recovery.
	00	SCLK
	01	SCLK ÷ 2
	10	SCLK ÷ 4
	11	SCLK ÷ 8
[2]		<b>Capture_INT_Mask</b> —Disable/enable interrupt when data is captured into either LO8 or HI8 on a positive or negative edge detection in DEMODULATION mode. This bit is not reset upon Stop Mode Recovery.
	0	Disable data capture interrupt.
	1	Enable data capture interrupt.
[1]		<b>Counter_INT_Mask</b> —Disable/enable T8 time-out interrupt. This bit is not reset upon Stop Mode Recovery.
	0	Disable time-out interrupt.
	1	Enable time-out interrupt.
[0]		<b>P34_Out</b> —Select normal I/O or T8 output function for Port 3, pin 4.
	0	P34 as port output.
	1	T8 output on P34.

## T8 and T16 Common Functions Register

The T8 and T16 Common Functions register (CTR1) controls the functions in common with Timer 8 and Timer 16. [Table 58](#) describes the bits for this register.

- **Note:** *Be careful to differentiate TRANSMIT mode from DEMODULATION mode, as set by CTR1[7]. The functions of CTR1[6:0] and CTR2[6] are different depending on which mode is selected. Do not change from one mode to another without first disabling the counter/timers.*

**Table 58. Timer 8 and Timer 16 Common Functions Register (CTR1)**

Bit	7	6	5	4	3	2	1	0
<b>Field</b>	Mode	P36 Out/ Demodulator Input	T8/T16 Logic/ Edge Detect		Transmit Submode/ Glitch Filter		Initial Timer 8 Out/ Rising Edge	Initial Timer 16 Out/ Falling Edge
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Address</b>	Bank D: 01h; Linear: D01h							

Bit Position	Value	Description
[7:0]		<b>Mode</b> —Selects the timer mode for signal transmission or demodulation.
	0	TRANSMIT mode.
	1	DEMODULATION mode.
[6]		<b>TRANSMIT Mode</b>
		P36 Out—Select normal I/O or timer output on Port 3, Pin 6.
	0	P36 acts as normal I/O port output.
	1	P36 acts as combined Timer 8/Timer 16 output.
		<b>DEMODULATION Mode</b>
		Demodulator Input—Select Port 2, Pin 0 or Port 3, Pin 1 as the counter/timer input.
	0	P31 acts as the demodulator input. If IMR[2] = 1, a P31 event can also generate an IRQ1 interrupt. To prevent this, clear IMR[2] or select P20 as input instead.
	1	P20 acts as the demodulator input.



Bit Position	Value	Description
[5:4]	<b>TRANSMIT Mode</b>	
	T8/T16 Logic—Defines how the Timer 8/Timer 16 outputs are combined logically. These bits are not reset upon Stop Mode Recovery.	
	00	Output is T8 AND T16.
	01	Output is T8 OR T16.
	10	Output is T8 NOR T16.
	11	Output is T8 NAND T16.
	<b>DEMODULATION Mode</b>	
	Edge Detect—Define the behavior of the edge detector.	
00	Falling edge detection.	
01	Rising edge detection.	
10	Falling and rising edge detection.	
11	Reserved.	
[3:2]	<b>TRANSMIT Mode</b>	
	Submode Selection—Select NORMAL or PING-PONG mode operation, or force T16 output. When these bits are written to 00b (NORMAL mode) or 01b (PING-PONG mode), T16_OUT assumes the opposite state of bit CTR1[0] until the timer begins counting.	
	00	Normal operation. Writing 00 terminates PING-PONG mode, if it is active.
	01	PING-PONG mode.
	10	Force T16_OUT = 0
	11	Force T16_OUT = 1
	<b>DEMODULATION Mode</b>	
	Glitch Filter—Define the maximum glitch width to be rejected by the counter/timer.	
00	No filter.	
01	4 SCLK cycle filter.	
10	8 SCLK cycle filter.	
11	Reserved.	



Bit Position	Value	Description
[1]	<b>TRANSMIT Mode</b>	
	Initial Timer 8 Out—Select the initial T8_OUT state when Timer 8 is enabled. While the timer is disabled, the opposite state is asserted on the pin to ensure that a transition occurs when the timer is enabled. Changing this bit while the counter is enabled can cause unpredictable output on T8_OUT.	
	0	T8_OUT transitions from High to Low when Timer 8 is enabled.
	1	T8_OUT transitions from Low to High when Timer 8 is enabled.
	<b>DEMODULATION Mode</b>	
	Rising Edge—Indicates whether a rising edge was detected on the input signal. Write 1 to this flag to reset it.	
	Read	
	0	No rising edge detection.
	1	Rising edge detection.
	Write	
0	No effect.	
1	Reset flag to 0.	
[0]	<b>TRANSMIT Mode</b>	
	Initial Timer 16 Out—In NORMAL or PING-PONG mode, this bit selects the initial T16_OUT state when Timer 16 is enabled. While the timer is disabled, the opposite state is asserted on the pin to ensure that a transition occurs when the timer is enabled. Changing this bit while the counter is enabled can cause unpredictable output on T16_OUT.	
	0	If CTR1[3]=0, T16_OUT transitions from High to Low when Timer 16 is enabled.
	1	If CTR1[3]=0, T16_OUT transitions from Low to High when Timer 16 is enabled.
	<b>DEMODULATION Mode</b>	
	Falling Edge—Indicates whether a falling edge was detected on the input signal. Write 1 to this flag to reset it.	
	Read	
	0	No falling edge detection.
	1	Falling edge detection.
	Write	
0	No effect.	
1	Reset flag to 0.	

## Counter/Timer 16 Control Register

Table 59 describes the bits for the Counter/Timer 16 Control register (CTR2).

**Table 59. Counter/Timer 16 Control Register (CTR2)**

Bit	7	6	5	4	3	2	1	0
Field	T16_Enable	Single/ Modulo-N	Time_Out	T16_Clock		Capture_INT _Mask	Counter_INT _Mask	P35_Out
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank D: 02h; Linear: D02h							

Bit Position	Value	Description
[7]	0 1	<b>T16_Enable</b> —Disable/enable the T16 counter. Disable T16 counter. Enable T16 counter.
[6]	0 1	<b>TRANSMIT Mode (CTR1[7]=0)</b> Single/Modulo-N—Selects Timer 16 terminal count action. MODULO-N mode. T16 reloads the initial value when terminal count is reached. SINGLE-PASS mode. T16 stops when the terminal count is reached.
		<b>DEMODULATION Mode (CTR1[7]=1)</b> Enable single-edge capture. See <a href="#">T16 DEMODULATION Mode</a> on page 104.
	0 1	Timer 16 captures and reloads on all edges. Timer 16 captures and reloads on first edge only.
[5]	Read 0 1 Write 0 1	<b>Time_Out</b> —This bit is set when the T16 terminal count is reached. Time_Out—This bit is set when the T16 terminal count is reached. No counter time-out occurs. Counter time-out occurred. No effect. Reset flag to 0. Software must reset this flag before using counter/timers.
[4:3]	00 01 10 11	<b>T16_Clock</b> —Select T16 input clock frequency. These bits are not reset upon Stop Mode Recovery. SCLK SCLK ÷ 2 SCLK ÷ 4 SCLK ÷ 8

Bit Position	Value	Description
[2]		<b>Capture_INT_Mask</b> —Disable/enable interrupt when data is captured into either LO16 or HI16 upon a positive or negative edge detection in DEMODULATION mode. This bit is not reset upon Stop Mode Recovery.
	0	Disable data capture interrupt.
	1	Enable data capture interrupt.
[1]		<b>Counter_INT_Mask</b> —Disable/enable T16 time-out interrupt.
	0	Disable T16 time-out interrupt.
	1	Enable T16 time-out interrupt.
[0]		<b>P35_Out</b> —Select normal I/O or T8 output function for Port 3, Pin 5.
	0	P35 as port output.
	1	P35 is T16 output.

### Timer 8/Timer 16 Control Register

The Timer 8/Timer 16 Counter/Timer register allows the T8 and T16 counters to be synchronized. It also can freeze the T16 output value and change T8 DEMODULATION mode to capture one cycle of a carrier. [Table 60](#) briefly describes the bits for this Bank D register.

**Table 60. Timer 8/Timer 16 Control Register (CTR3)**

Bit	7	6	5	4	3	2	1	0
<b>Field</b>	T16_Enable	T8_Enable	Sync_Mode	T16_Out Disable	T8 Demodulate	Reserved		
<b>Reset</b>	0	0	0	0	0	X	X	X
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	—	—	—
<b>Address</b>	Bank D: 03h; Linear: D03h							

Bit Position	Value	Description
[7]	0	Disable T16 counter.
	1	Enable T16 counter. Configure T16 properly before enabling it.
[6]	0	Disable T8 counter.
	1	Enable T8 counter.
[5]		<b>Sync_Mode</b> —When enabled, the first pulse of Timer 8 (the carrier) is always synchronized with Timer 16 (the demodulated signal). It can always provide a full carrier pulse. This bit is not reset upon Stop Mode Recovery.
	0	Disable SYNC mode.
	1	Enable SYNC mode.



Bit Position	Value	Description
[4]		<b>T16_Out Disable</b> —Set this bit to disable toggling of the Timer 16 output. Time-out interrupts are still generated. This bit is not reset upon Stop Mode Recovery.
	0	T16 toggles normally.
	1	T16 toggle is disabled.
[3]		<b>T8 Demodulate</b> —(Capture one cycle) This bit is not reset upon Stop Mode Recovery.
	0	T8 captures events normally.
	1	T8 becomes active on the third edge, captures events on the fourth and fifth edges, and generates an interrupt on the fifth edge. After a T8 time-out the event count resets to 0 and the fourth and fifth edges are captured again.
[2:0]	—	<b>Reserved</b> —Always reads 111b. Must be written to 1.



# Interrupts

The ZLF645 MCU features six interrupts (see [Table 62](#) on page 123). These interrupts are maskable and prioritized (see [Figure 40](#) on page 122).

The six interrupt sources are divided as follows:

- Three sources are claimed by Port 3 lines P33:P31
- Two by the counter/timers (see [Table 62](#))
- One for low-voltage detection

P32 and UART receiver share the same interrupt. Only one interrupt can be selected as a source. When the UART receiver is enabled, P32 is no longer used as an interrupt source. The UART transmit interrupt and UART baud rate interrupt use the same interrupt as the P33 interrupt. The user selects which source triggers the interrupt. When bit 7 of UCTL is 1, the UART transmit interrupt is the source. When bit 7 of UCTL is 0 and bit 5 of UCTL is 1, the BRG interrupt is selected. The [Interrupt Mask Register](#) (globally or individually) enables or disables the six interrupt requests. The source for IRQ1 is determined by bit 1 of the Port 3 Mode register (P3M) and bit 4 of the SMR4 register.

If P3M[1]=0 (DIGITAL mode) and SMR4[4]=0, pin P33 is the IRQ1 source.

If P3M[1]=1 (ANALOG mode) or SMR4[4]=1 (SMR interrupt enabled), the output of the Stop Mode Recovery source logic is used as the source for the interrupt. For more details, see [Stop Mode Recovery Interrupt](#) on page 138.

**Table 61. Interrupt Control Registers**

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
0F9	All	F9	Interrupt Priority Register	IPR	XXh	<a href="#">124</a>
0FA	All	FA	Interrupt Request Register	IRQ	00h	<a href="#">125</a>
0FB	All	FB	Interrupt Mask Register	IMR	0XXX_XXXb	<a href="#">127</a>

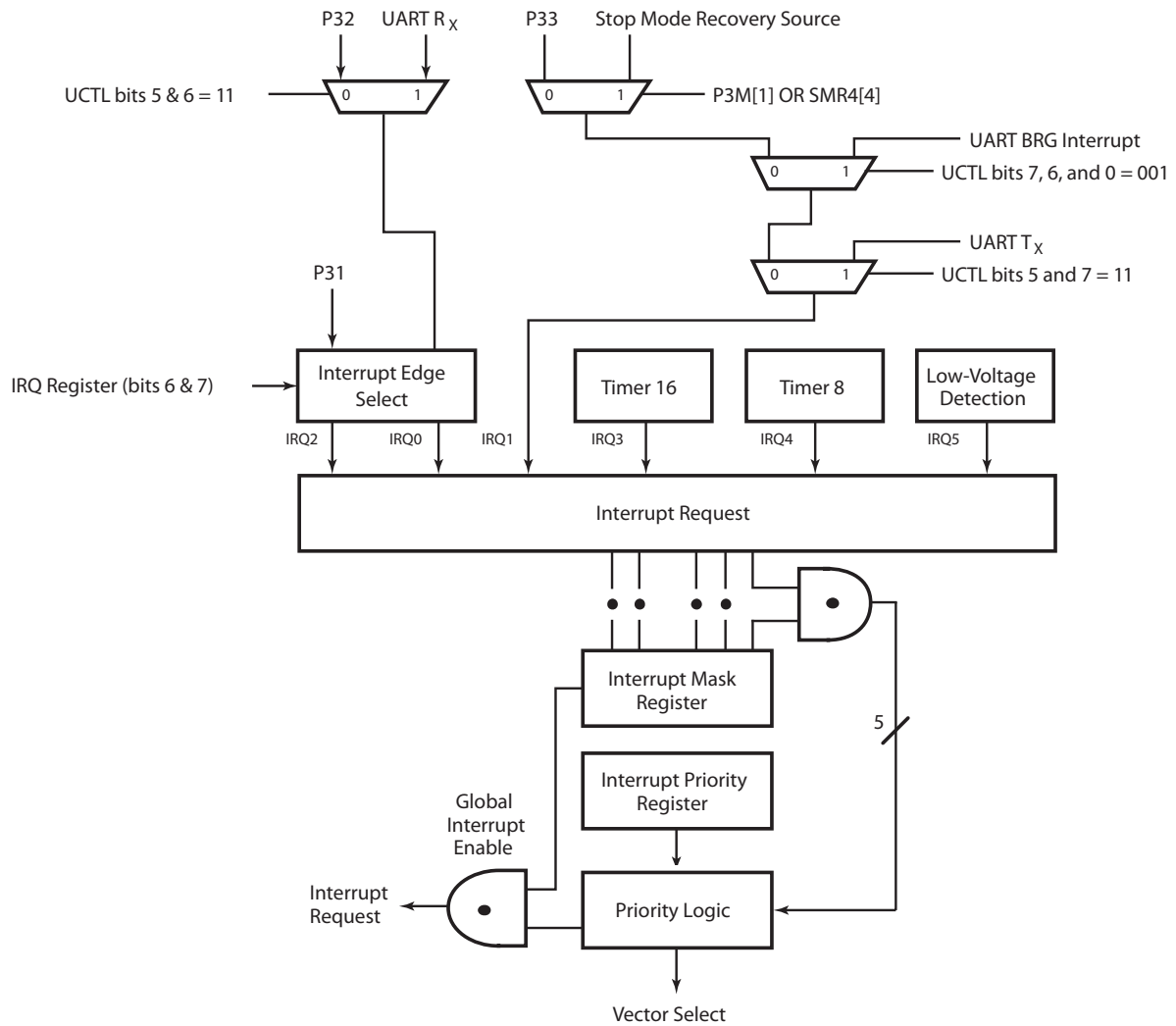


Figure 40. Interrupt Block Diagram

**Table 62. Interrupt Types, Sources, and Vectors**

Name	Source	Vector Location (Program Memory)	Comments
IRQ0	P32, UART Rx	0,1	External (P32), Rising, Falling Edge Triggered
IRQ1	P33, UART Tx, BRG, SMR Event	2, 3	External (P33), Falling Edge Triggered
IRQ2	P31	4, 5	External (P31), Rising, Falling Edge Triggered
IRQ3	Timer 16	6, 7	Internal
IRQ4	Timer 8	8, 9	Internal
IRQ5	Low-Voltage Detection	10, 11	Internal

When more than one interrupt is pending, priorities are resolved by a programmable priority encoder controlled by the [Interrupt Priority Register](#). An interrupt machine cycle activates when an interrupt request is granted. As a result, all subsequent interrupts are disabled, and the Program Counter and Status Flags are saved. The cycle then branches to the Program Memory vector location reserved for that interrupt.

All ZLF645 MCU interrupts are vectored through locations in the Program Memory. This memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request. To accommodate polled interrupt systems, interrupt inputs are masked, and the [Interrupt Request Register](#) is polled to determine which of the interrupt requests require service.

An interrupt resulting from AN1 is mapped into IRQ2, and an interrupt from AN2 is mapped into IRQ0. Interrupts IRQ2 and IRQ0 can be rising, falling, or both edge triggered. These interrupts are user-programmable. The software can poll to identify the state of the pin.



Programming bits for the Interrupt Edge Select are located in the IRQ register (R250), bit 6 and bit 7. [Table 63](#) provides the configuration.

**Table 63. Interrupt Request Register**

IRQ Bit		Interrupt Edge	
7	6	IRQ2 (P31)	IRQ0 (P32)
0	0	F	F
0	1	F	R
1	0	R	F
1	1	R/F	R/F

**Note:** F = Falling Edge; R = Rising Edge.

## Interrupt Priority Register

The Interrupt Priority register (see [Table 64](#)) defines which interrupt holds the highest priority. Interrupts are divided into three groups of two—Group A, Group B, and Group C.

IPR bits 4, 3, and 0 determine which interrupt group has priority. For example, if interrupts IRQ5, IRQ1, and IRQ0 occur simultaneously when IPR[4:3,0]=001b, the interrupts are serviced in the following order: IRQ1, IRQ0, IRQ5.

IPR bits 5, 2, and 1 determine which interrupt within each group has higher priority.

**Table 64. Interrupt Priority Register (IPR)**

Bit	7	6	5	4	3	2	1	0
<b>Field</b>	Reserved		Group A Priority	Group Priority [2:1]		Group B Priority	Group C Priority	Group Priority [0]
<b>Reset</b>	X	X	X	X	X	X	X	X
<b>R/W</b>	—		W	W		W	W	W
<b>Address</b>	Bank Independent: F9h; Linear: 0F9h							

Bit Position	Value	Description
[7:6]	—	<b>Reserved</b> Reads are undefined; writes must be 00b.
[5]	0	<b>Group A Priority (IRQ3, IRQ5)</b> IRQ5 > IRQ3
	1	

Bit Position	Value	Description
[[4:3], [0]]	<b>Group Priority</b>	
	000	Reserved
	001	C > A > B
	010	A > B > C
	011	A > C > B
	100	B > C > A
	101	C > B > A
	110	B > A > C
111	Reserved	
[2]	<b>Group B Priority (IRQ0, IRQ2)</b>	
	0	IRQ2 > IRQ0
	1	IRQ0 > IRQ2
[1]	<b>Group C Priority (IRQ1, IRQ4)</b>	
	0	IRQ1 > IRQ4
	1	IRQ4 > IRQ1

## Interrupt Request Register

Bit 7 and Bit 6 of the Interrupt Request register (see [Table 65](#)) are used to configure the edge detection of the interrupts for Port 3, bit 1 and Port 3, bit 2. The remaining bits (5 through 0) indicate the status of the interrupt. When an interrupt is serviced, the hardware automatically clears the bit to 0. Writing 1 to any of these bits generates an interrupt if the appropriate bits in the Interrupt Mask register are enabled. Writing 0 to these bits clears the interrupts.

**Table 65. Interrupt Request Register (IRQ)**

Bit	7	6	5	4	3	2	1	0
Field	Interrupt Edge		IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FAh; Linear: 0FAh							

Bit Position	Value	Description
[7:6]	<b>Interrupt Edge</b>	
	00	P31↓ P32↓
	01	P31↓ P32↑
	10	P31↑ P32↓
	11	P31↑↓ P32↑↓

Bit Position	Value	Description
[5]	Read	<b>IRQ5 (Low-Voltage Detection)</b>
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
[4]	Read	<b>IRQ4 (T8 Counter)</b>
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
[3]	Read	<b>IRQ3 (T16 Counter)</b>
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
[2]	Read	<b>IRQ2 (Port 3 Bit 1 Input)</b>
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
[1]	Read	<b>IRQ1 (Port 3 Bit 3 Input/SMR Event/UART T<sub>x</sub>/UART BRG)</b>
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.
[0]	Read	<b>IRQ0 (Port 3 Bit 2 Input/UART R<sub>x</sub>)</b>
	0	Interrupt did not occur.
	1	Interrupt occurred.
	Write	
	0	Clear interrupt.

► **Note:** *The IRQ register is protected from change until an EI instruction is executed once.*

## Interrupt Mask Register

Bits [5:0] are used to enable the interrupt. Bit 7 is the status of the master interrupt. When reset, all interrupts are disabled. When writing 1 to bit 7, you must also execute the EI instruction to enable interrupts (see [Table 66](#)).

**Table 66. Interrupt Mask Register (IMR)**

Bit	7	6	5	4	3	2	1	0
Field	Master Interrupt Enable	Reserved	IRQ5 Enable	IRQ4 Enable	IRQ3 Enable	IRQ2 Enable	IRQ1 Enable	IRQ0 Enable
Reset	0	X	X	X	X	X	X	X
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FBh; Linear: 0FBh							

Bit Position	Value	Description
[7]		<b>Master Interrupt Enable</b>
	0	Use only DI and EI instructions to alter this bit. Always disable interrupts (DI instruction) before writing this register. All interrupts are disabled.
	1	Interrupts are enabled/disabled individually in bits [5:0].
[6]	0	Reserved Reads are undefined; Must be written to 1.
[5]	0	Disables IRQ5.
	1	Enables IRQ5.
[4]	0	Disables IRQ4.
	1	Enables IRQ4.
[3]	0	Disables IRQ3.
	1	Enables IRQ3.
[2]	0	Disables IRQ2.
	1	Enables IRQ2.
[1]	0	Disables IRQ1.
	1	Enables IRQ1.
[0]	0	Disables IRQ0.
	1	Enables IRQ0.





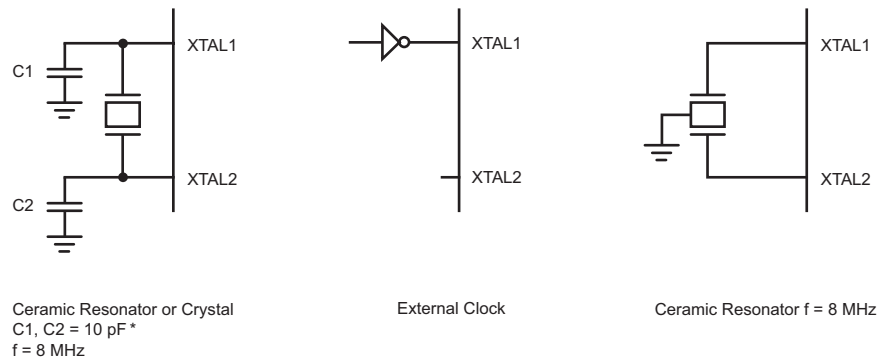
# Clock

ZLF645 MCUs on-chip oscillator has a high-gain, parallel-resonant amplifier for connecting to a crystal, ceramic resonator, or any suitable external clock source (XTAL1 = Input, XTAL2 = Output).

## Crystal Specification

The crystal must be AT cut, 1 MHz to 8 MHz (maximum), with a series resistance ( $R_S$ ) less than or equal to  $100 \Omega$ . The on-chip oscillator can be driven with a suitable external clock source.

The crystal must be connected across XTAL1 and XTAL2 pins using the recommended capacitors from each pin to ground. The typical capacitor value is 10 pF for 8 MHz. Check with the crystal supplier for the optimum capacitance.



\*Note: preliminary value, including pin parasitics.

**Figure 41. Oscillator Configuration**

Zilog's IR MCU supports crystal, resonator, and oscillator. Most resonators have a frequency tolerance of less than  $\pm 0.5\%$ , which is enough for a remote control application. Resonator has a very fast startup time, which is around few hundred microseconds. Most crystals have a frequency tolerance of less than 50 ppm ( $\pm 0.005\%$ ). However, crystal needs longer startup time than the resonator. The large loading capacitance slows down the oscillation startup time. Clock oscillation must be stable before the CPU begins instruction execution. If oscillation is not present or not stable before the chip completes timeout of its Power-On Reset (POR) period, the chip's behavior could be indeterminate.

Zilog<sup>®</sup> recommends not to use more than 10 pF loading capacitor for the crystal. If the stray capacitance of the PCB or the crystal is high, the loading capacitance C1 and C2 must be reduced further to ensure stable oscillation before the  $T_{POR}$  (POR time is typically 5-6 ms. For more details, see [Table 82](#) on page 164.).

For Stop Mode Recovery operation, bit 5 of SMR register allows you to select the Stop Mode Recovery delay, which is the  $T_{POR}$ . If Stop Mode Recovery delay is not selected, the MCU executes instruction immediately after it wakes up from the STOP mode. If resonator or crystal is used as a clock source then STOP mode recovery delay needs to be selected (Bit 5 of SMR = 1).

For both resonator and crystal oscillator, the oscillation ground must go directly to the ground pin of the microcontroller. The oscillation ground must use the shortest distance from the microcontroller ground pin and it must be isolated from other connections.

## Crystal 1 Oscillator Pin (XTAL1)

The Crystal 1 Oscillator time-based input pin connects a parallel-resonant crystal or ceramic resonator to the on-chip oscillator input. Additionally, an optional external single-phase clock can be connected to the on-chip oscillator input.

## Crystal 2 Oscillator Pin (XTAL2)

The Crystal 2 Oscillator time-based output pin connects a parallel-resonant crystal or ceramic resonator to the on-chip oscillator output.

## Internal Clock Signals (SCLK and TCLK)

The CPU and internal peripherals are driven by the internal SCLK signal during normal execution. During HALT mode, the interrupt logic is driven by the internal TCLK signal. The frequency of these signals with respect to the XTAL1 clock input is selectable either by programming bit 2 of the Flash's User Option Byte 1 for no division of the XTAL1 signal input, dividing it by a factor of two, and optionally by applying an additional divide-by-16 prescaler enabled through SMR register bit 0 (see [Table 70 on page 141](#)), as displayed in [Figure 42](#). Selecting the divide-by-16 prescaler reduces device power draw during normal operation and HALT mode. The prescaler is disabled by a POR or Stop Mode Recovery.

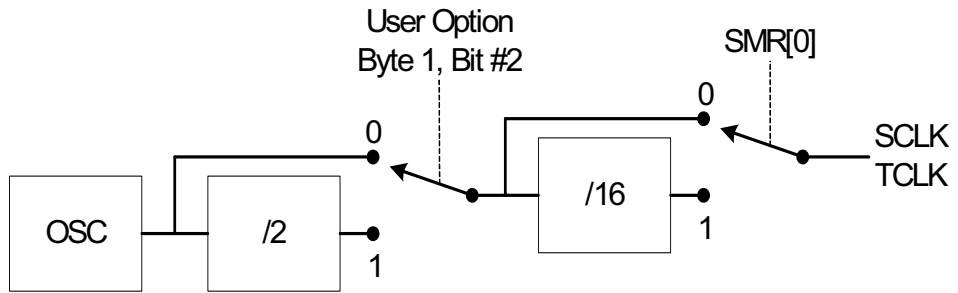


Figure 42. SCLK/TCLK Circuit

# Reset and Power Management

The ZLF645 MCU provides the following reduced-power modes, power monitoring, and reset features:

- **Voltage Brownout Standby**—Stops the oscillator and internal clock when the power level drops below the VBO low voltage detect point. Initiates a power-on reset when power is restored above the VBO detect point.
- **STOP Mode**—Stops the clock and oscillator, reduces the MCU supply current to a very low level until a power-on reset or Stop Mode Recovery occurs.
- **HALT Mode**—Stops the internal clock to the CPU until an enabled interrupt request is received.
- **Voltage Detection**—Optionally sets a flag if a low- or high-voltage condition occurs. The low-voltage detection flag can generate an interrupt request, if enabled.
- **Power-On Reset**—Starts the oscillator and internal clock, and initializes the system to its power-on reset defaults.
- **Watchdog Timer**—Optionally generates a Power-On Reset if the program fails to execute the WDT instruction within a specified time interval.
- **Stop Mode Recovery**—Restarts the oscillator and internal clock, and initializes most of the system to its power-on reset defaults. Some register values are not reset by a Stop Mode Recovery.

► **Note:** For supply current values under various conditions, see [DC Characteristics on page 160](#).

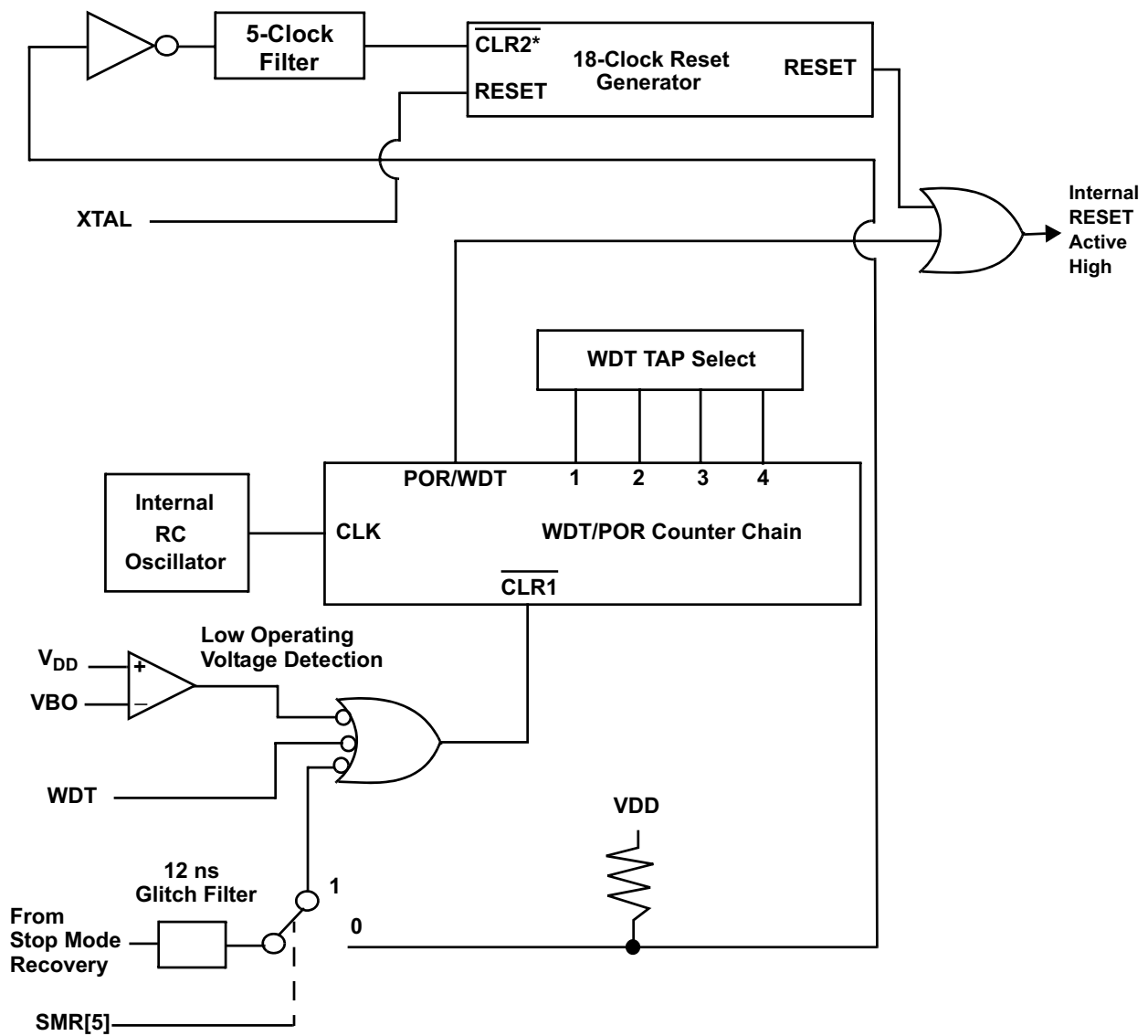
[Figure 43](#) on page 133 displays the Power-On Reset sources. [Table 67](#) lists control registers for reset and power management features. Some features are affected by registers described in other chapters.

**Table 67. Reset and Power Management Registers**

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
D0C	D	0C	Low-Voltage Detection Register	LVD	1 1 1 1 _1000 b	<a href="#">135</a>
F0A	F	0A	Stop Mode Recovery Register 4	SMR4	XXX0 _0000 b	<a href="#">151</a>
F0B	F	0B	Stop Mode Recovery Register	SMR	0 0 1 0 _0000 b	<a href="#">141</a>
F0C	F	0C	Stop Mode Recovery Register 1	SMR1	00h	<a href="#">145</a>
F0D	F	0D	Stop Mode Recovery Register 2	SMR2	X0X0 _00XX b	<a href="#">147</a>

Table 67. Reset and Power Management Registers (Continued)

Address (Hex)						
12-Bit	Bank	8-Bit	Register Description	Mnemonic	Reset	Page No
F0E	F	0E	Stop Mode Recovery Register 3	SMR3	X0h	150
F0F	F	0F	Watchdog Timer Mode Register	WDTMR	0000_1101b	137



\*CLR1 and CLR2 enable the WDT/POR and 18 Clock Reset timers, respectively, on a Low-to-High input transition.

Figure 43. Resets and Watchdog Timer

## Voltage Brownout Standby

An on-chip voltage comparator circuit (VBO) checks that the  $V_{DD}$  is at the required level for correct operation of the device in terms of Flash memory reads. A second on-chip comparator circuit (subVBO) checks that the  $V_{DD}$  level is high enough for proper operation of the VBO circuit. If the  $V_{DD}$  level drops below the VBO trip point, the ZLF645 will be held in a reset state as long as  $V_{DD}$  remains below this trip point value, and the XTAL1 and XTAL2 oscillator circuitry will be disabled thereby stopping the clock input to the ZLF645 and saving power. If the  $V_{DD}$  level continues to drop below the subVBO trip point, the ZLF645 will remain in a reset state and the VBO comparator circuit will be disabled for further power savings. When the power level returns to a value above the VBO trip point, the device performs a power-on reset and functions normally.

## STOP Mode

STOP instruction turns OFF the internal clock and external crystal oscillation, thus reducing the MCU supply current to a very low level. For STOP mode current specifications, see [Table 81](#) on page 160.

To enter STOP mode, first flush the instruction pipeline to avoid suspending execution in mid-instruction. Execute a NOP instruction (OpCode = FFh) immediately before the appropriate sleep instruction, as given below:

```
FF          NOP          ; clear the pipeline
6F          STOP        ; enter STOP mode
```

STOP mode is terminated only by a reset, such as WDT time-out, POR, or one of the Stop Mode Recovery events as described in [Stop Mode Recovery Event Sources](#) on page 138. This condition causes the processor to restart the application program at address 000Ch.

Unlike a normal POR or WDT reset, a Stop Mode Recovery reset does not reset the contents of some registers and bits. Register bits not reset by a Stop Mode Recovery are highlighted in grey in the register tables. Register bit SMR[7] is set to 1 by a Stop Mode Recovery.

## HALT Mode

HALT instruction turns off the internal CPU clock, but not the XTAL oscillation. The counter/timers, UART, and interrupts (IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, and IRQ5) remain active. The devices are recovered by interrupts, either externally or internally generated. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after HALT mode.

To enter HALT mode, first flush the instruction pipeline to avoid suspending execution in mid-instruction. Execute a NOP instruction (OpCode = FFh) immediately before the appropriate sleep instruction, as given below:

```
FF          NOP          ; clear the pipeline
7F          HALT        ; enter HALT mode
```

Power consumption during HALT mode can be reduced by first setting SMR[0]=1 to enable the divide-by-16 clock prescaler.

## Voltage Detection

The Low-Voltage Detection register (LVD, register 0Ch at the expanded register bank 0Dh) provides an option to monitor the  $V_{DD}$  voltage. The voltage detection is enabled when bit 0 of LVD register is set. After voltage detection is enabled, the  $V_{DD}$  level is monitored in real time. The HVD flag (bit 2 of the LVD register) is set only if  $V_{DD}$  is higher than  $V_{HVD}$ . The LVD flag (bit 1 of the LVD register) is set only if  $V_{DD}$  is lower than the  $V_{LVD}$ . When voltage detection is enabled, the LVD flag also triggers IRQ5. The IRQ bit 5 latches the low voltage condition until it is cleared by instructions or reset. The IRQ5 interrupt is served if it is enabled in the IMR register. Otherwise, bit 5 of IRQ register is latched as a flag only.

► **Note:** *Do not modify register P01M while checking a low voltage condition. Switching noise from Port 0 can trigger the LVD flag.*

**Table 68. Low-Voltage Detection Register (LVD)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved					High-Battery Detect	Low-Battery Detect	Voltage Detect Enable
Reset	1	1	1	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Address	Bank D: 0Ch; Linear: D0Ch							

Bit Position	Value	Description
[7:3]	—	Reserved—Reads 11111b. Must be written to 1.
[2]	0	HVD clear.
	1	High-voltage detected ( $V_{DD} > V_{HVD}$ )

Bit Position	Value	Description
[1]	0	LVD clear.
	1	Low-voltage detected ( $V_{DD} < V_{LVD}$ )
[0]	0	Voltage detection disabled.
	1	Voltage detection enabled.

## Power-On Reset Timer

When power is initially applied to the device, a timer circuit clocked by a dedicated on-board RC-oscillator provides the POR timer function. The POR timer circuit is a one-shot timer that keeps the internal reset signal asserted long enough for  $V_{DD}$  and the oscillator circuit to stabilize before instruction execution begins.

The reset timer is triggered by one the following conditions:

- Initial power-on or recovery from a VBO/standby condition.
- Stop Mode Recovery (if register bit SMR[5] = 1)
- Watchdog Timer time-out.

SMR[5] can be cleared to 0 to bypass the POR timer on a Stop Mode Recovery. This must only be done when using an external clock that does not require a startup delay.



**Caution:** *Failure of an application to provide a stable oscillating clock input to XTAL1 before the end of the ZLF645's POR period may result in an indeterminate chip behavior and must be avoided. For details on the POR timing range, see [Table 82](#) on page 164 in the [Electrical Characteristics](#) chapter.*

## Watchdog Timer

The Watchdog Timer (WDT) is a retriggerable one-shot timer that resets the Z8 LXMC CPU if it reaches its terminal count. The WDT must initially be enabled by executing the WDT instruction. On subsequent executions of the WDT instruction, the WDT is refreshed. The WDT circuit is driven by an on-board RC-oscillator. The WDT instruction affects the Zero (Z), Sign (S), and Overflow (V) flags.

The POR clock source is an internal RC-oscillator. Bits 0 and 1 of the WDT register control a tap circuit that determines the minimum time-out period. Bit 2 determines whether the WDT is active during HALT, and bit 3 determines WDT activity during STOP mode. Bits 4 through 7 are reserved (see [Table 69](#) on page 137). This register is accessible only during the first 60 processor cycles (120 XTAL clocks) from the execution of the first instruction after power-on reset, watchdog timer Reset, or a Stop Mode Recovery (see [Fast Stop Mode Recovery](#)). After this point, the register cannot be modified by any means. The



WDTMR register cannot be read. The register is located in Bank F of the Expanded Register Group at address location 0Fh.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

**Table 69. Watchdog Timer Mode Register (WDTMR)**

Bit	7	6	5	4	3	2	1	0
Field	—	Time-Out Select			WDT During STOP Mode	WDT During HALT Mode	Time-Out Select	
Reset	0	0	0	0	1	1	0	1
R/W	W	W	W	W	W	W	W	W
Address	Bank F: 0Fh; Linear: F0Fh							

Bit Position	Value	Description
[7]	—	<b>Reserved</b> —Reads are undefined; must write 0000.
[3]	0 1	<b>WDT During STOP Mode</b> —Determines if WDT is active during STOP mode. Off. WDT active during STOP mode.
[2]	0 1	<b>WDT During HALT Mode</b> —Determines if WDT is active during HALT mode. See <a href="#">Figure 43 on page 133</a> . Off. WDT active during HALT mode.
[6:4], [1:0]	000_00 000_01 000_10 000_11 001_XX 010_XX 100_XX	<b>Time-Out Select</b> —Selects the WDT time period (see Note below). 5 ms minimum 10 ms minimum 20 ms minimum 80 ms minimum 320 ms minimum 1, 280 ms minimum 5,120 ms minimum

► **Note:** *Although not explicitly shown above, if any two bits of bits 6 through 4 are programmed to 1 or if all three bits are programmed to 0, then the time-out period depends on bits 1 and 0 only as shown for the [6:4]=000 case.*

## Reset/Stop Mode Recovery Status

Read-only bit SMR[7]=0, if the previous reset was initiated by a Power-On Reset (including Voltage Brownout or WDT resets). SMR[7]=1, if the previous reset was initiated by a Stop Mode Recovery. A power-on, Voltage Brownout, or WDT reset restores all registers

to their Power-On Reset defaults. A Stop Mode Recovery restores most registers to their Power-On Reset defaults. Register bits not reset by a Stop Mode Recovery are highlighted in grey in the register tables. Register bit SMR[7] is set to 1 instead of reset by a Stop Mode Recovery.

## Fast Stop Mode Recovery

SMR[5] can be cleared to 0 before entering STOP mode to bypass the default  $T_{POR}$  reset timer on Stop Mode Recovery. See [Voltage Brownout Standby](#) on page 134.

If SMR[5]=0, the Stop Mode Recovery source must be kept active for at least 10 input clock periods ( $T_{pC}$ ).

- **Note:** *SMR[5] must be set to 1 if using a crystal or resonator clock source. The  $T_{POR}$  delay allows the clock source to stabilize before executing instructions.*

## Stop Mode Recovery Interrupt

Software can set register bit SMR4[4] = 1 to enable routing of Stop Mode Recovery events to IRQ1 and to Port 3, Pin 3. In this configuration, if an IRQ1 interrupt occurs, register bit P3[3] = 0 indicates that a Stop Mode Recovery event is occurring.

## Stop Mode Recovery Event Sources

Any Port 2 or Port 3 input pin can be configured to generate a Stop Mode Recovery event, either individually or in various logical combinations. The ZLF645 MCU provides the following registers for Stop Mode Recovery source configuration and status:

- **SMR Register**—Selects one Port 3, Pin 1–3 pin state or one of three Port 2 pin logical combinations to generate an event when a defined 0 or 1 level occurs.
- **SMR1 Register**—Configures one or more Port 2 input pins (0–7) to latch the latest read or write value and generate an event when the pin state changes.
- **SMR2 Register**—Selects one of seven Port 2 and 3 pin logical combinations to generate an event when a defined 0 or 1 level occurs.
- **SMR3 Register**—Configures one or more Port 3 input pins (0–3) to latch the latest read or write value and generates an event when the pin state changes.
- **SMR4 Register**—Enables routing of SMR events to IRQ1. Indicates whether port data has been latched for SMR1 or SMR3 event monitoring, and whether the latch was on a port read or write.

A Stop Mode Recovery event occurs if any of the sources defined in the SMR, SMR1, SMR2, and SMR3 registers are active.

## SMR Register Events

The SMR register function is similar to the standard Stop Mode Recovery feature used in previous Z8<sup>®</sup> CPU-compatible parts. Register bits SMR[4:2] are set to select one of six event modes, as displayed in [Figure 44](#) on page 140. The output of the corresponding logic is compared to the state of SMR[6]; when they are the same, a Stop Mode Recovery event is generated. If SMR[4:2]=000, no event source is selected by SMR.

The state SMR[4:2]=001 is reserved and selects no event in this device. The logic configured by the SMR register ignores any port pins that are configured as output or selected as source pins in registers SMR1 or SMR3. The SMR register is summarized in [Table 70 on page 141](#).

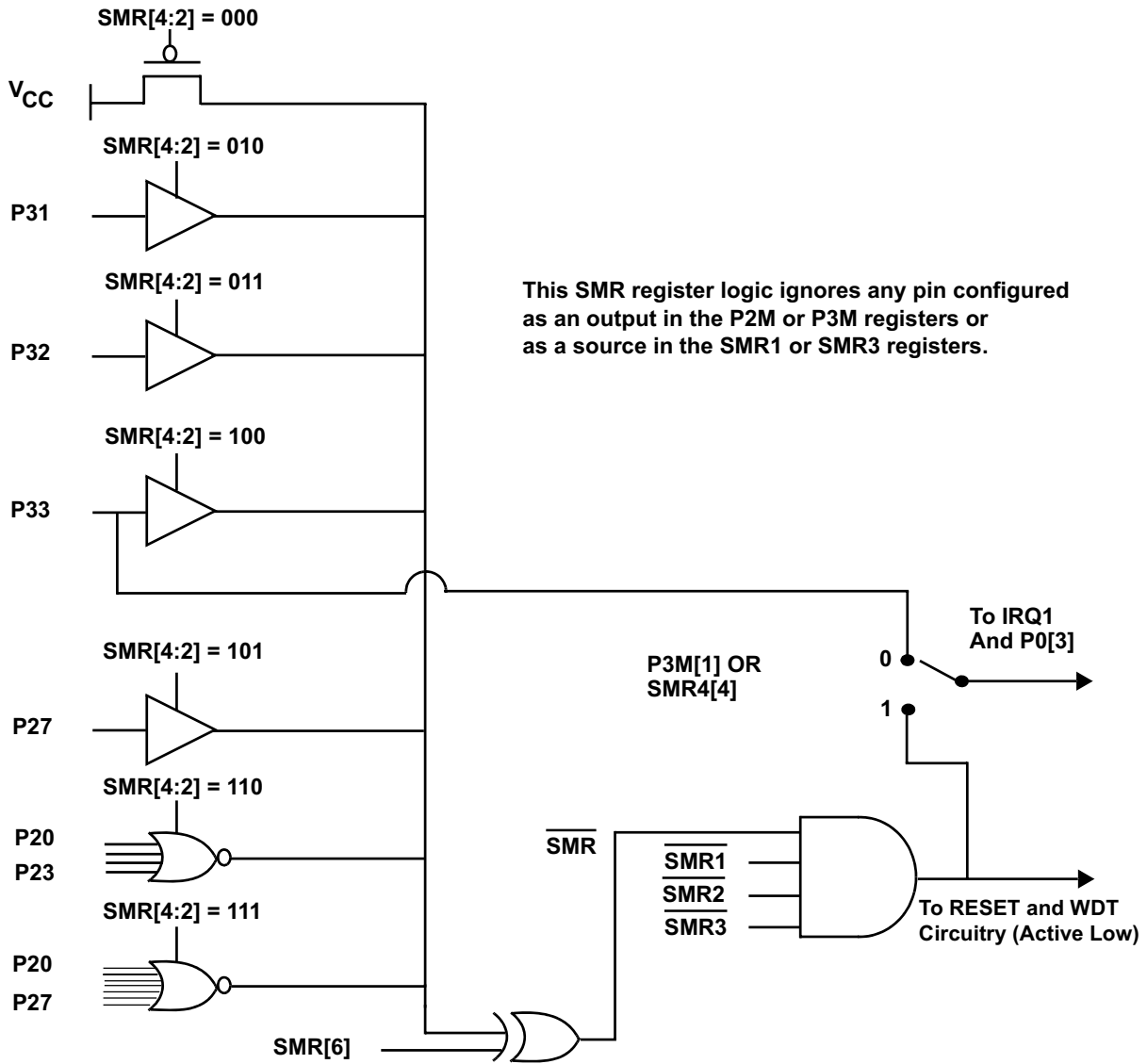


Figure 44. SMR Register-Controlled Event Sources

**Table 70. Stop Mode Recovery Register (SMR)**

Bit	7	6	5	4	3	2	1	0
<b>Field</b>	Stop Flag	Stop Mode Recovery Level	Stop Delay	Stop Mode Recovery Source			Reset Time Reduction	SCLK/TCLK Divide-by-16
<b>Reset</b>	0	0	1	0	0	0	0	0
<b>R/W</b>	R	W	W	W	W	W	W	W
<b>Address</b>	Bank F: 0Bh; Linear: F0Bh							

Bit Position	Value	Description
[7]		<b>Stop Flag</b> —Indicates whether last startup was power-on reset or Stop Mode Recovery. A write to this bit has no effect.
	0	Power-on reset.
	1	Stop Mode Recovery.
[6]		<b>Stop Mode Recovery Level</b> —Selects whether an SMR[4:2]-selected SMR is initiated by a Low or High level at the XOR-gate input (see <a href="#">Figure 44 on page 140</a> ).
	0	Low.
	1	High.
[5]		<b>Stop Delay</b> —Controls the reset delay after recovery. Must be 1 if using a crystal or resonator clock source.
	0	Off.
	1	On.
[4:2]		<b>Stop Mode Recovery Source</b> —Specifies a Stop Mode Recovery wake-up source at the XOR gate input (see <a href="#">Figure 44 on page 140</a> ). This value is not changed by a Stop Mode Recovery. The following equations ignore any Port pin configured as output or selected in SMR1 or SMR3.
	000	No SMR register source selected.
	001	Reserved.
	010	P31.
	011	P32.
	100	P33.
	101	P27.
	110	Port 2 NOR 0–3.
	111	Port 2 NOR 0–7.

Bit Position	Value	Description
[1]		<b>SMR Short Reset Time</b> —Controls whether the devices SMR reset period is equivalent to the RC oscillator based POR reset period or whether it depends on the detection of XTAL1 clock oscillation.
	0	Unless SMR[5]=1, the SMR reset period is equivalent to the devices RC oscillator based POR reset period and falls in the range of 2.5 ms to 10 ms.
	1	Unless SMR[5]=1, the SMR reset period falls in a range of a minimum of 2.5 ms from chip power up or a maximum of 2.5 ms from when the XTAL1 clock reaches a peak-to-peak amplitude of oscillation greater than 250 mV.
[0]		<b>SCLK/TCLK Divide-by-16 Select</b> —Controls a divide-by-16 prescaler of the internal SCLK/TCLK signal (see <a href="#">Internal Clock Signals (SCLK and TCLK)</a> on page 130). A power-on reset or Stop Mode Recovery clears this bit to 0.
	0	Off.
	1	On.

## SMR1 Register Events

The SMR1 register can be used to configure one or more Port 2 pins to be compared with a written or sampled reference value and generate a Stop Mode Recovery event when the pin state differs from the reference value.

To configure a Port 2 pin as an SMR1 event source, ensure it is configured as an input in the P2M register, then set the corresponding SMR1 register bit. By default, a Stop Mode Recovery event occurs when the pin's state is zero.

After a Port 2 pin is configured as an SMR1 source, any subsequent read from or write to the P2 register latches the read or write value for reference. A Stop Mode Recovery event occurs when the pin's state differs from the last reference value latched. The SMR1 source logic is displayed in [Figure 45](#) on page 144.

The program can read register bits SMR4[1:0] to determine whether the Port 2 pins trigger a Stop Mode Recovery on a change from the last read value (SMR4[1:0]=01), or on a change from the last written value (SMR4[1:0]=10). Software can clear SMR4[1:0] to 00 to restore the default behavior (configured pins trigger when their state is 0). The SMR1 register is summarized in [Table 71](#) on page 145.

After the following example code is executed, a 1 on P20 will wake the part from STOP mode:

```
LD P2M, #%FF      ;Set Port 2 to inputs.
SRP #%0F          ;Point to expanded bank F
LD SMR1, #%01     ;Select P20 for SMR1.
SRP #%00          ;Point to bank 0
LD P2, #00        ;Write 00h to Port 2, so the P20 reference
                  ;value is 0, and a 1 on P20 wakes the part.

NOP
STOP
```

After the following example code is executed when the value of P2 is 00h, a 1 on P20 will wake the part from STOP mode:

```
LD P2M, #%FF      ;Set ports to inputs.
SRP #%0F          ;Point to expanded bank F
LD SMR1, #%01     ;Select P20 for SMR1.
SRP #%00          ;Point to bank 0
LD R6, P2         ;If a 0 is read from Port 2, the P20 reference
                  ;value is 0, so a 1 on P20 wakes the part.

NOP
STOP
```

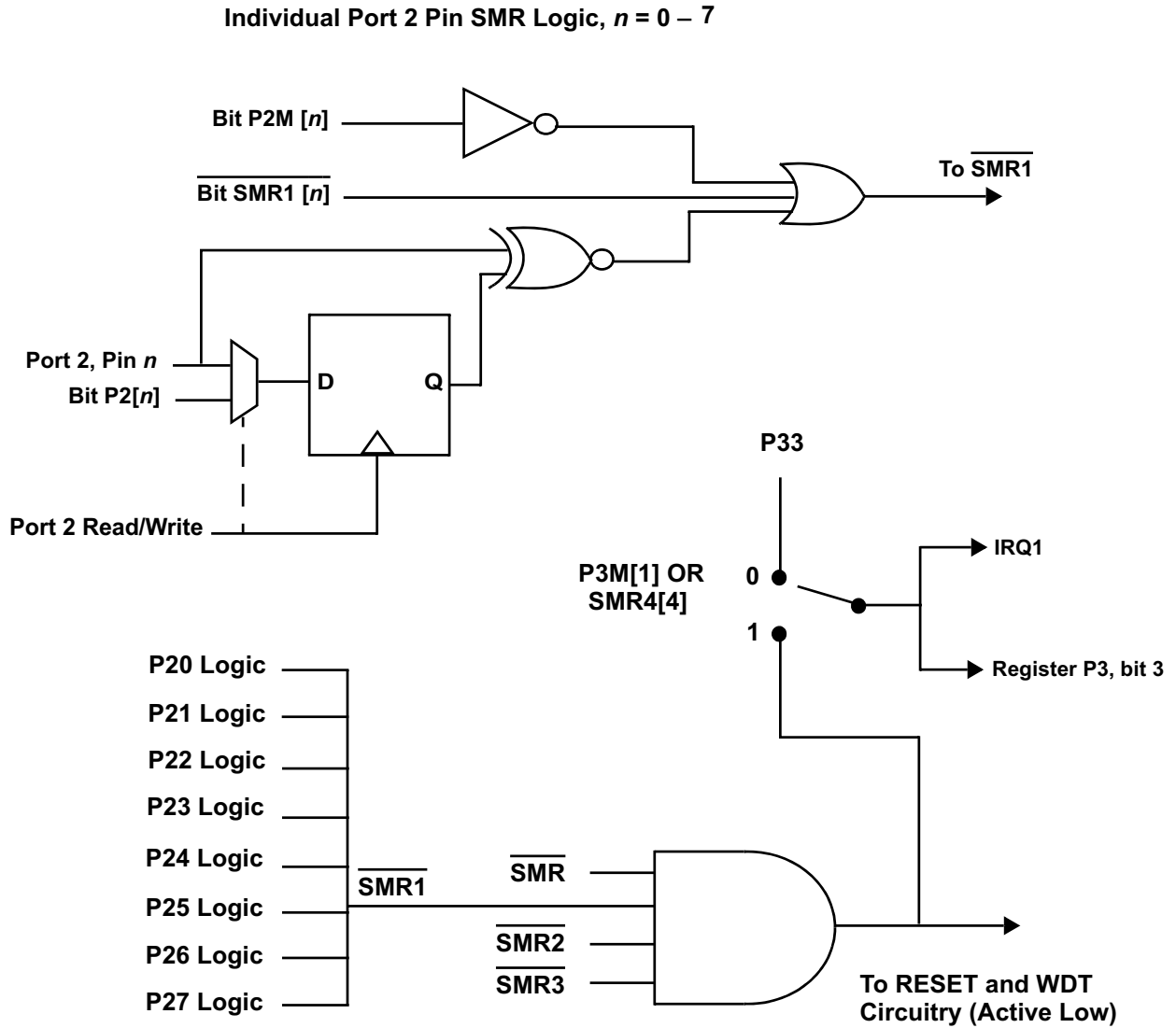


Figure 45. SMR1 Register-Controlled Event Sources



**Table 71. Stop Mode Recovery Register 1 (SMR1)**

Bit	7	6	5	4	3	2	1	0
Field	P27 Stop Select	P26 Stop Select	P25 Stop Select	P24 Stop Select	P23 Stop Select	P22 Stop Select	P21 Stop Select	P20 Stop Select
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Address	Bank F: 0Ch; Linear: F0Ch							

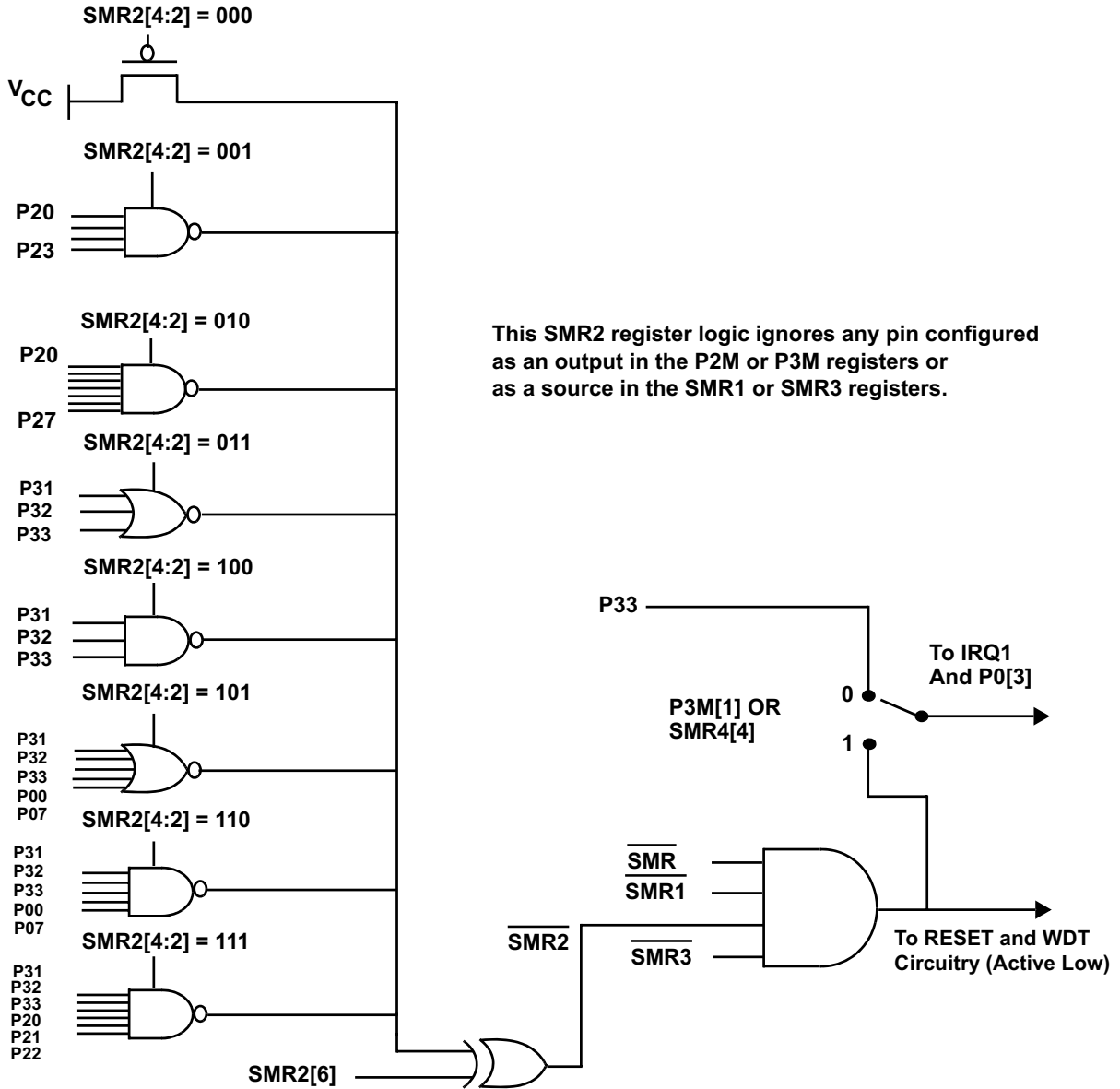
Bit Position	Value	Description
[7]	0	P27 not selected.
	1	P27 selected as an SMR source.
[6]	0	P26 not selected.
	1	P26 selected as an SMR source.
[5]	0	P25 not selected.
	1	P25 selected as an SMR source.
[4]	0	P24 not selected.
	1	P24 selected as an SMR source.
[3]	0	P23 not selected.
	1	P23 selected as an SMR source.
[2]	0	P22 not selected.
	1	P22 selected as an SMR source.
[1]	0	P21 not selected.
	1	P21 selected as an SMR source.
[0]	0	P20 not selected.
	1	P20 selected as an SMR source.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

### SMR2 Register Events

The SMR2 register function is similar to the standard Stop Mode Recovery feature used in previous Z8 CPU-compatible parts. Register bits SMR2[4:2] are set to select one of seven event modes, as displayed in [Figure 46](#). The output of the corresponding logic is compared to the state of SMR2[6]; when they are the same, a Stop Mode Recovery event is generated. If SMR2[4:2]=000, no event source is selected by SMR2.

The logic configured by the SMR2 register ignores any port pins that are configured as an output, or that are selected as source pins in registers SMR1 or SMR3. The SMR2 register is summarized in Table 72 on page 147.



This SMR2 register logic ignores any pin configured as an output in the P2M or P3M registers or as a source in the SMR1 or SMR3 registers.

Figure 46. SMR2 Register-Controlled Event Sources

**Table 72. Stop Mode Recovery Register 2 (SMR2)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Stop Mode Recovery Level 2	Reserved	Stop Mode Recovery Source			Reserved	
Reset	X	0	X	0	0	0	X	X
R/W	—	W	—	W	W	W	—	
Address	Bank F: 0Dh; Linear: F0Dh							

Bit Position	Value	Description
[7]	—	<b>Reserved</b> —Read is undefined; write must be 0.
[6]		<b>Stop Mode Recovery Level 2</b> Selects whether an SMR2[4:2]-selected SMR is initiated by a Low or High level at the XOR-gate input (see <a href="#">Figure 46 on page 146</a> ). 0 Low. 1 High.
[5]	—	<b>Reserved</b> —Read is undefined; Must be written to 1.
[4:2]		<b>Stop Mode Recovery Source</b> Specifies a Stop Mode Recovery wake-up source at the XOR gate input (see <a href="#">Figure 46 on page 146</a> ). Additional sources can be selected by SMR, SMR1, and SMR3 registers. If more than one source is selected, any selected source event causes a Stop Mode Recovery. The following equations ignore any Port pin that is selected in register SMR1 or configured as an output. 000 No SMR2 register source selected. 001 NAND of P23:P20. 010 NAND of P27:P20. 011 NOR of P33:P31. 100 NAND of P33:P31. 101 NOR of P33:P31, P00, P07. 110 NAND of P33:P31, P00, P07. 111 NAND of P33:P31, P22:P20.
[1:0]	—	<b>Reserved</b> —Read is undefined; write must be 00b.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

### SMR3 Register Events

The SMR3 register can be used to configure one or more of Port 3, pins 0–3 to be compared to a written or sampled reference value and generate a Stop Mode Recovery event when the pin state differs from the reference value.

To configure a Port 3 input pin as an SMR3 event source set the corresponding SMR3 register bit. By default, a Stop Mode Recovery event occurs when the pin's state is zero.

After a Port 3 pin is configured as an SMR3 source, any subsequent read from or write to the P2 register latches the read or written value for reference. A Stop Mode Recovery event occurs when the pin's state differs from the last reference value latched. The SMR3 source logic is displayed in [Figure 47](#).

The program can read register bits SMR4[3:2] to determine whether the Port 3 pins trigger a Stop Mode Recovery on a change from the last read value (SMR4[3:2]=01), or on a change from the last written value (SMR4[3:2]=10). Software can clear SMR4[3:2] to 00 to restore the default behavior (configured pins trigger when their state is 0).

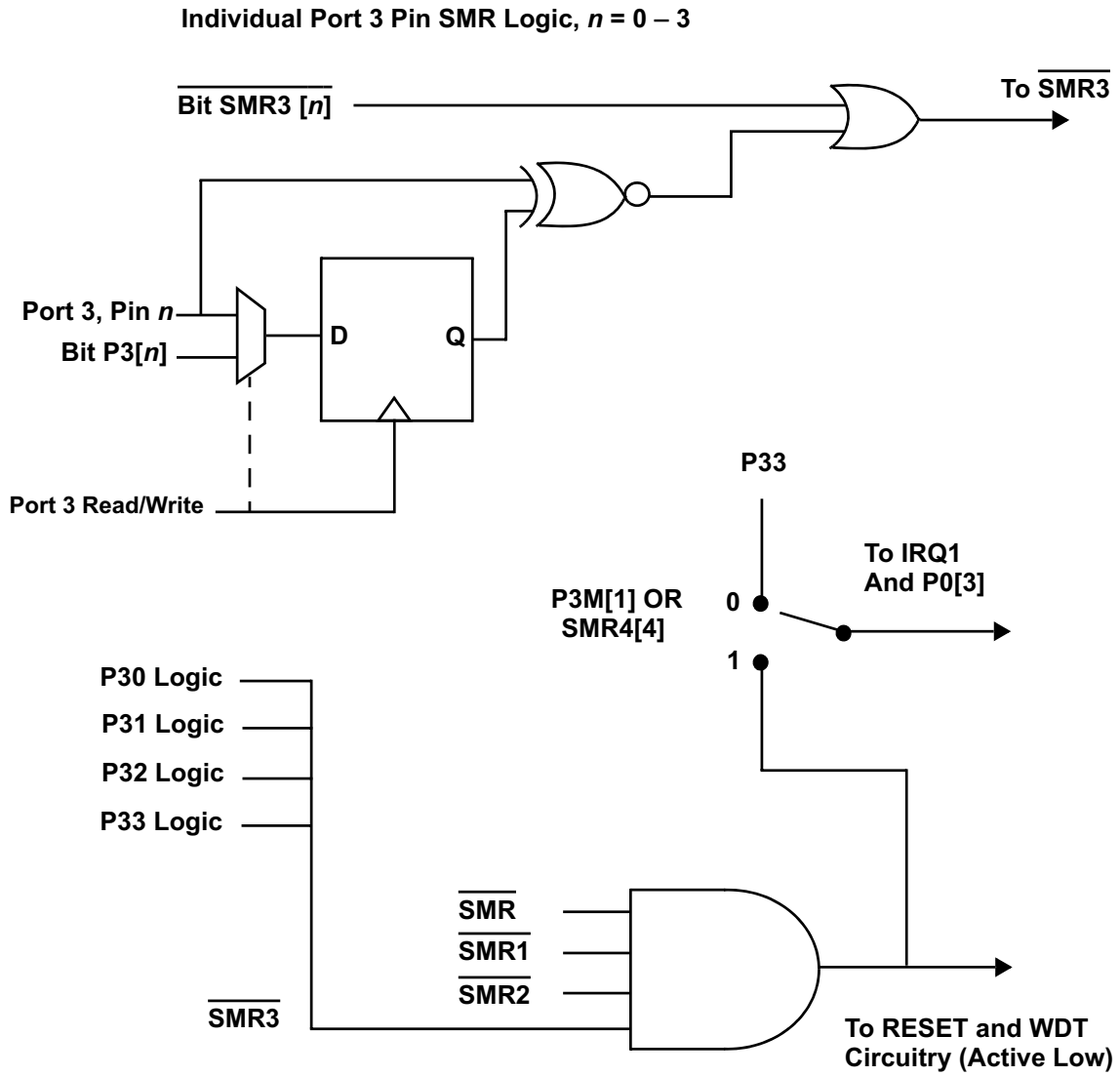
The SMR3 register is summarized in [Table 70 on page 141](#).

After the following example code is executed, a 1 on P30 will wake the part from STOP mode.

```
LD SMR3, #%01    ;Select P30 from SMR3.
LD P3,  #00      ;Write 00h to Port 3, so the P30 reference
                  ;value is 0, and a 1 on P30 wakes the part.
NOP
STOP
```

After the following example code is executed when the value of P3 is 00h, a 1 on P30 will wake the part from STOP mode.

```
LD SMR3, #%01    ;Select P30 for SMR3.
LD R6,  P3       ;If a 0 is read from Port 3, the P30 reference
                  ;value is 0, so a 1 on P30 wakes the part.
NOP
STOP
```



**Figure 47. SMR3 Register-Controlled Event Sources**

**Table 73. Stop Mode Recovery Register 3 (SMR3)**

Bit	7	6	5	4	3	2	1	0
Field	—				P33 SMR Select	P32 SMR Select	P31 SMR Select	P30 SMR Select
Reset	X	X	X	X	0	0	0	0
R/W	—	—	—	—	W	W	W	W
Address	Bank F: 0Eh; Linear: F0Eh							

Bit Position	Value	Description
[7:4]	—	<b>Reserved</b> —Reads undefined; Must be written to 1.
[3]	0	P33 not selected.
	1	P33 SMR source selected.
[2]	0	P32 not selected.
	1	P32 SMR source selected.
[1]	0	P31 not selected.
	1	P31 SMR source selected.
[0]	0	P30 not selected.
	1	P30 SMR source selected.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

## Stop Mode Recovery Register 4

The Stop Mode Recovery register 4 (see [Table 74](#)) enables the SMR interrupt source and indicates the reference value status for registers SMR1 and SMR3.

**Table 74. Stop Mode Recovery Register 4 (SMR4)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved			SMR IRQ Enable	Port 3 SMR Status		Port 2 SMR Status	
Reset	X	X	X	0	0	0	0	0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
Address	Bank F: 0Ah; Linear: F0Ah							

Bit Position	Value	Description
[7:5]	—	<b>Reserved</b> —Reads are undefined; must write 000b.
[4]	0 1	<b>SMR IRQ Enable</b> If P3M[1]=0, SMR events do not generate an interrupt. SMR events generate an interrupt on IRQ1.
[3:2]	00 01 10 11	<b>Port 3 SMR Status</b> No Read or Write of the P3 register occurs. P3 Read occurs; used as SMR3 reference. P3 Write occurs; used as SMR3 reference. Reserved.
[1:0]	00 01 10 11	<b>Port 2 SMR Status</b> No Read or Write of the P2 register occurs. P2 Read occurs; use P2 Read as SMR1 reference. P2 Write occurs; use P2 Write as SMR1 reference. Reserved.

► **Note:** *This register is not reset after a Stop Mode Recovery.*

# Z8 LXMC CPU Programming Summary

The following sections provide a summary of information useful for programming the Z8 LXMC CPU included in this device. For more details on the Z8 LXMC CPU and its instruction set, refer to *Z8<sup>®</sup> LXMC CPU Core User Manual (UM0215)*.

## Addressing Notation

[Table 75](#) summarizes Z8 LXMC CPU addressing modes and symbolic notation. The text variable *n* represents a decimal number; *aa* represents a hexadecimal address; and *LABEL* represents a label defined elsewhere in the assembly source.

In reference notation, *only* lowercase is used to distinguish 4-bit addressed working registers (r1, r2) from 8-bit addressed registers (R1, R2). The numerals 1 and 2, respectively, indicate whether the register is used for destination or source addressing.

**Table 75. Symbolic Notation for Operands**

Symbol	Assembly Operand	Description
cc	–	<b>Condition Code</b> cc represents a condition code mnemonic. See <a href="#">Condition Codes on page 156</a> .
IM	# <i>n</i>	<b>Immediate Data</b> IM represents an Immediate Data value, prefixed by # in assembly language where <i>n</i> = 0 to 255. The immediate value follows the instruction opcode in Program Memory.
r1 r2	R <i>n</i>	<b>Working Register</b> r1 or r2 represents the name, R <i>n</i> , of a working register, where <i>n</i> = 0, 1, 2, ..., 15. The equivalent 12-bit address is {RP[3:0], RP[7:4], <i>n</i> }.
rr1 rr2	RR <i>n</i>	<b>Working Register Pair</b> rr1 or rr2 represents the name, R <i>n</i> , of a working register pair, where <i>n</i> = 0, 2, 4, ..., 14. The equivalent 12-bit address is {RP[3:0], RP[7:4], <i>n</i> }.
R1 R2	% <i>aa</i>	<b>Register</b> R1 or R2 represents an 8-bit register address. For addresses 00h-DFh or F0h-FFh, the equivalent 12-bit address is {RP[3:0], % <i>aa</i> }. For addresses E0h-EFh (escaped mode), the equivalent 12-bit address is {RP[3:0], RP[7:4], % <i>aa</i> [3:0]}.



Table 75. Symbolic Notation for Operands (Continued)

Symbol	Assembly Operand	Description
RR1 RR2	<i>%aa</i>	<b>Register Pair (8-bit Address)</b> RR1 or RR2 represents the 8-bit address of a register pair. For addresses 00h-DFh or F0h-FFh, the equivalent 12-bit address is {RP[3:0], <i>%aa</i> }. For addresses E0h-EFh (escaped mode), the equivalent 12-bit address is {RP[3:0], RP[7:4], <i>%aa</i> [3:0]}.
lr1 lr2	@Rn	<b>Indirect Working Register</b> lr1 or lr2 represents the name a working register, Rn, where n = 0, 1, 2, ..., 15. @ indicates Indirect Working register addressing using an 8-bit effective address contained in the specified working register. The accessed register's equivalent 12-bit address is {RP[3:0], 8-bit effective address}.
lrr1 lrr2	@RRn	<b>Indirect Working Register Pair</b> lrr1 or lrr2 represents the name of a working register pair, RRn, where n = 0, 2, 4, ..., 14. @ indicates Indirect Working register addressing using an effective address in the specified working register pair. Depending on the instruction, the effective address is in the register file (12-bit address) or Program/Constant Memory (16-bit address).
IR1 IR2	@%aa	<b>Indirect Register</b> IR1 or IR2 represents the 8-bit address of a register. @ indicates Indirect register addressing using an 8-bit effective address contained in the specified register. The accessed register's equivalent 12-bit address is {RP[3:0], 8-bit effective address}.
IRR1	@%aa	<b>Indirect Register Pair</b> IRR1 represents the 8-bit address of a register. @ indicates Indirect register addressing with a 16-bit effective address (in Program Memory) contained in the specified register pair.
X(r1) X(r2)	<i>%aa</i> (Rn)	<b>Indexed (X) Addressing</b> X represents the 8-bit base address to which the offset is added. r1 or r2 represents the name, Rn, of a working register containing the 8-bit signed offset. The 8-bit effective address is the sum of X and the contents of working register Rn. The accessed register's equivalent 12-bit address is {RP[3:0], 8-bit effective address}.

**Table 75. Symbolic Notation for Operands (Continued)**

Symbol	Assembly Operand	Description
DA	<i>LABEL</i>	<b>Direct Address (JP, CALL)</b> In a JP or CALL operand, DA is a 16-bit Program Memory address in the range of 0000H to FFFFH. DA replaces the contents of the Program Counter for execution to continue at a new location in Program Memory. In assembly source, the address is represented as a label.
RA	<i>LABEL</i>	<b>Relative Address (JR, DJNZ)</b> RA is a signed 8-bit Program Memory offset in the range +127 to –128, relative to the address of the next instruction in Program Memory. In a JR or DJNZ operation, RA is added to the program counter to cause execution to continue at a new location in Program Memory. In assembly source, the jump address is represented as an absolute label, and the assembler calculates RA.

[Table 76](#) lists additional symbols that are used throughout the instruction set summary.

**Table 76. Additional Symbols**

Symbol	Definition
dst	Destination Operand
src	Source Operand
@	Indirect Address Prefix
C	Carry Flag
SP	Stack Pointer Value
PC	Program Counter
FLAGS	Flags Register
RP	Register Pointer
#	Immediate Operand Prefix
b	Binary Number Suffix
%	Hexadecimal Number Prefix
h	Hexadecimal Number Suffix
←	Assignment of a value. For example, dst ← dst + src indicates the result is stored in the destination

**Table 76. Additional Symbols (Continued)**

Symbol	Definition
↔	Exchange of two values
~	One's complement unary operator

## Flags Register

The Flags register (see [Table 77](#)) informs the current status of the Z8<sup>®</sup> CPU. It contains six bits of status information.

**Table 77. Flags Register (FLAGS)**

Bit	7	6	5	4	3	2	1	0
Field	C	Z	S	O	D	H	F1	F2
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Address	Bank Independent: FCh; Linear: 0FCh							

Bit Position	Value	Description
[7]		<b>Carry Flag (C)</b> Set when the result of an arithmetic operation generates a <i>carry out of</i> or a <i>borrow into</i> the high-order bit (bit 7) of the result. Also used in rotate and shift instructions.
	0	Flag Clear
	1	Flag Set
[6]		<b>Zero Flag (Z)</b> Set when the result of an arithmetic operation is 0.
	0	Flag Clear
	1	Flag Set
[5]		<b>Sign Flag (S)</b> Stores the value of the most significant bit (msb) following an arithmetic, logical, rotate, or shift instruction.
	0	Flag Clear
	1	Flag Set
[4]		<b>Overflow Flag (O)</b> Set when the result of an arithmetic operation is greater than 127.
	0	Flag Clear
	1	Flag Set

Bit Position	Value	Description
[3]		<b>Decimal Adjust Flag (D)</b> Used for binary-coded decimal (BCD) arithmetic.
	0	Flag Clear
	1	Flag Set
[2]		<b>Half Carry Flag (H)</b> Set when a <i>carry out of or borrow into</i> bit 3 of an arithmetic operation occurs.
	0	Flag Clear
	1	Flag Set
[1]		<b>User Flag 1 (F1)</b> Available to software for use as a general-purpose bit.
	0	Bit Clear
	1	Bit Set
[0]		<b>User Flag 2 (F2)</b> Available to software for use as a general-purpose bit.
	0	Bit Clear
	1	Bit Set

## 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). [Table 78](#) summarizes the condition codes. Some binary condition codes can be created using more than one assembly code mnemonic. The result of the flag test operation determines if the conditional jump executes.

**Table 78. Condition Codes**

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	MI	Minus	S = 1
0110	6	Z	Zero	Z = 1
0110	6	EQ	Equal	Z = 1



**Table 78. Condition Codes (Continued)**

Binary	Hex	Assembly Mnemonic	Definition	Flag Test Operation
0111	7	C	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	$(S \text{ XOR } V) = 0$
1010	A	GT	Greater than	$(Z \text{ OR } (S \text{ XOR } V)) = 0$
1011	B	UGT	Unsigned Greater than	$(C = 0 \text{ AND } Z = 0)$
1100	C	NOV	No Overflow	$V = 0$
1101	D	PL	Plus	$S = 0$
1110	E	NZ	Non-Zero	$Z = 0$
1110	E	NE	Not Equal	$Z = 0$
1111	F	NC	No Carry	$C = 0$
1111	F	UGE	Unsigned Greater than or Equal	$C = 0$

# Electrical Characteristics

## Absolute Maximum Ratings

A stress greater than listed in [Table 79](#) may cause permanent damage to the device. Functional operation of the device at any condition outside those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. For improved reliability, unused inputs must be tied to one of the supply voltages ( $V_{DD}$  or  $V_{SS}$ ).

**Table 79. Absolute Maximum Ratings**

Parameter	Min	Max	Units
Ambient temperature under bias	0	+70	C
Storage temperature	-65	+150	C
Voltage on any pin with respect to $V_{SS}$ *	-0.3	+4.0	V
Voltage on $V_{DD}$ pin with respect to $V_{SS}$	-0.3	+3.6	V
Maximum current on input and/or inactive output pin	-5	+5	$\mu$ A
Maximum output current from active output pin	-25	+25	mA
Maximum current into $V_{DD}$ or out of $V_{SS}$	—	75	mA

\*This voltage applies to all pins except  $V_{DD}$ , P32, and P33.

## Standard Test Conditions

The characteristics listed in this product specification apply for standard test conditions. All voltages are referenced to Ground. Positive current flows into the referenced pin (see [Figure 48](#)).

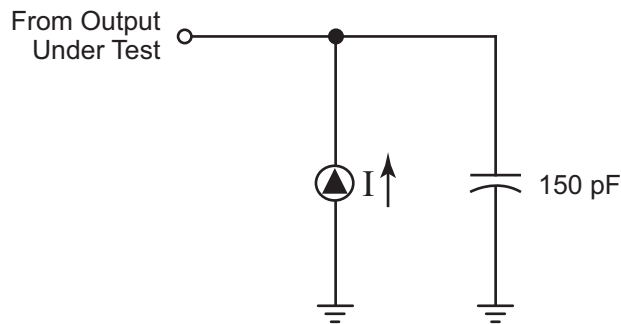


Figure 48. Test Load Diagram

## Capacitance

[Table 80](#) lists the capacitance.

Table 80. Capacitance

Parameter	Maximum
Input capacitance	12 pF
Output capacitance	12 pF
I/O capacitance	12 pF

**Note:**  $T_A = 25\text{ }^\circ\text{C}$ ,  $V_{CC} = \text{GND} = 0\text{ V}$ ,  $f = 1.0\text{ MHz}$ , unmeasured pins return to GND. This voltage applies to all pins except  $V_{DD}$ , P32, and P33.

## DC Characteristics

Table 81 describes the direct current (DC) characteristics of the ZLF645 Flash MCU.

**Table 81. DC Characteristics**

Symbol	Parameter	$T_A = 0\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C}$				Units	Conditions
		$V_{CC}$	Min	Typ	Max		
$V_{CC}$	Supply Voltage <sup>1</sup>	—	1.9	—	3.6	V	See <a href="#">Note 5</a> .
$V_{CH}$	Clock Input High Voltage	1.9–3.6	$0.8 V_{CC}$	—	$V_{CC} + 0.3$	V	Driven by External Clock Generator
$V_{CL}$	Clock Input Low Voltage	1.9–3.6	$V_{SS}-0.3$	—	0.4	V	Driven by External Clock Generator
$V_{IH}$	Input High Voltage	1.9–3.6	$0.7 V_{CC}$	—	$V_{CC} + 0.3$	V	—
$V_{IL}$	Input Low Voltage	1.9–3.6	$V_{SS}-0.3$	—	$0.2 V_{CC}$	V	—
$V_{OH1}$	Output High Voltage	1.9–3.6	$V_{CC}-0.4$	—	—	V	$I_{OH} = -0.5\text{ mA}$
$V_{OH2}$	Output High Voltage (P36, P37, P00, and P01)	1.9–3.6	$V_{CC}-0.8$	—	—	V	$I_{OH} = -7\text{ mA}$
$V_{OL1}$	Output Low Voltage	1.9–3.6	—	—	0.4	V	$I_{OL} = 4.0\text{ mA}$
$V_{OL2}$	Output Low Voltage (P00, P01, P36, and P37)	1.9–3.6	—	—	0.8	V	$I_{OL} = 10\text{ mA}$
$V_{OFFSET}$	Comparator Input Offset Voltage	1.9–3.6	—	—	25	mV	—
$V_{REF}$	Comparator Reference Voltage	1.9–3.6	0	—	$V_{CC} - 1.75$	V	—
$I_{IL}$	Input Leakage	1.9–3.6	-1	—	1	$\mu\text{A}$	$V_{IN} = 0\text{ V}$ , $V_{CC}$ ; pull-ups disabled
$I_{IL1}$	Input Leakage IR Amp (P31)	1.9–3.6	-2.5	—	-12	$\mu\text{A}$	$V_{IN} = 0\text{ V}$ , IR amp enabled
$I_{OL}$	Output Leakage	1.9–3.6	-1	—	1	$\mu\text{A}$	$V_{IN} = 0\text{ V}$ , $V_{CC}$
$I_{CC}$	Supply Current <sup>2, 3</sup>	1.9	—	1	2	mA	See <a href="#">Note 7</a> .
		3.6	—	2	4	mA	
		1.9	—	2	3	mA	See <a href="#">Note 8</a> .
		3.6	—	4	6	mA	



**Table 81. DC Characteristics (Continued)**

Symbol	Parameter	V <sub>CC</sub>	T <sub>A</sub> = 0 °C to +70 °C			Units	Conditions
			Min	Typ	Max		
I <sub>CC1</sub>	Standby Current <sup>2, 3</sup> (HALT mode)	1.9	—	0.7	1.6	mA	V <sub>IN</sub> = 0 V, V <sub>CC</sub> See <a href="#">Note 7</a> .
		3.6	—	1.2	2.0	mA	
		1.9	—	0.9	2	mA	V <sub>IN</sub> = 0 V, V <sub>CC</sub> See <a href="#">Note 8</a> .
		3.6	—	2	3	mA	
I <sub>CC2</sub>	Standby Current <sup>4</sup> (STOP mode)	1.9	—	1.6	8	μA	V <sub>IN</sub> = 0 V, V <sub>CC</sub> WDT is not running
		3.6	—	1.8	10	μA	
		1.9	—	5	20	μA	V <sub>IN</sub> = 0 V, V <sub>CC</sub> WDT is running
		3.6	—	8	30	μA	
I <sub>LV</sub>	Standby Current <sup>5</sup> (Low Voltage)	—	—	1.2	6	μA	Measured at 1.3 V
V <sub>BO</sub>	V <sub>CC</sub> Low Voltage Protection	—	1.8	1.85	1.9	V	—
V <sub>LVD</sub>	V <sub>CC</sub> Low Voltage Detection	—	2.1	2.15	2.2	V	—
V <sub>HVD</sub>	V <sub>CC</sub> High Voltage Detection	—	—	2.7	—	V	—
T <sub>ONIRAMP</sub>	Wake-up time from disabled mode	1.9–3.6	—	—	20	μs	—

**Table 81. DC Characteristics (Continued)**

Symbol	Parameter	V <sub>CC</sub>	T <sub>A</sub> = 0 °C to +70 °C			Units	Conditions
			Min	Typ	Max		
I <sub>DETLO</sub>	IR amp current input guaranteed to be detected as a 0 (see <a href="#">Note 6</a> )	1.9–3.6	—	—	10	μA	IR amp enabled
I <sub>DETHI</sub>	IR amp current input guaranteed to be detected as a 1 (see <a href="#">Note 6</a> )	1.9–3.6	100	—	—	μA	IR amp enabled

**Notes**

1. Zilog® recommends adding a filter capacitor (minimum 0.1 μF), physically close to V<sub>DD</sub> and V<sub>SS</sub> if operating voltage fluctuations are anticipated, such as those resulting from driving an infrared LED.
2. All outputs unloaded, inputs at rail.
3. CL1 = CL2 = 100 pF.
4. Oscillator stopped.
5. Oscillator stops when V<sub>DD</sub> falls below V<sub>BO</sub> limit.
6. For reference, under typical process, 25 °C temperature, and a voltage of 2.8 V, the voltage that would be seen at P31 under an input current of 10 μA or 100 μA would be ~480 mV or .560 mV, respectively.
7. 8.0-MHz XTAL1 input clock frequency with 4-MHz system clock
8. 8.0-MHz XTAL1 input clock frequency with 8-MHz system clock

## AC Characteristics

Figure 49 and Table 82 lists the alternating current (AC) characteristics of ZLF645 Flash MCU.

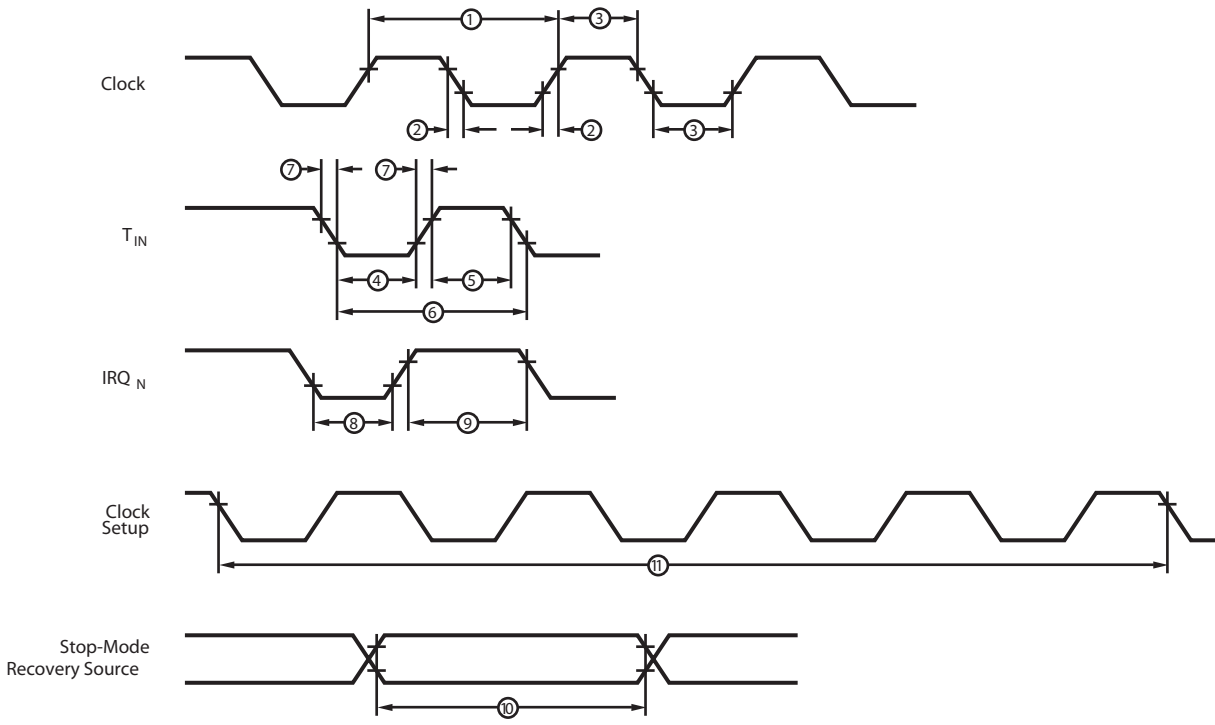


Figure 49. AC Timing Diagram

**Table 82. Clock, Reset, Timers, and SMR Timing**

No	Symbol	Parameter	V <sub>CC</sub>	T <sub>A</sub> = 0 °C to +70 °C 8.0 MHz		Units	WDTMR (Bits 6, 5, 4, 1, 0)
				Min	Max		
1	T <sub>pC</sub>	Input Clock Period <sup>1</sup>	1.9–3.6	121	DC	ns	—
2	T <sub>RC</sub> , T <sub>FC</sub>	Clock Input Rise and Fall Times <sup>1</sup>	1.9–3.6	—	25	ns	—
3	T <sub>WC</sub>	Input Clock Width <sup>1</sup>	1.9–3.6	37	—	ns	—
4	T <sub>WTINL</sub>	Timer Input Low Width <sup>1</sup>	1.9	100	—	ns	—
			3.6	70	—	ns	—
5	T <sub>WTINH</sub>	Timer Input High Width <sup>1</sup>	1.9–3.6	3T <sub>pC</sub>	—	—	—
6	T <sub>PTIN</sub>	Timer Input Period <sup>1</sup>	1.9–3.6	8T <sub>pC</sub>	—	—	—
7	T <sub>RTIN</sub> , T <sub>FTIN</sub>	Timer Input Rise and Fall Timers <sup>1</sup>	1.9–3.6	—	100	ns	—
8	T <sub>WIL</sub>	Interrupt Request Low Time <sup>1,2</sup>	1.9	100	—	ns	—
			3.6	70	—	ns	—
9	T <sub>WIH</sub>	Interrupt Request Input High Time <sup>1,2</sup>	1.9–3.6	5T <sub>pC</sub>	—	—	—
10	T <sub>WSM</sub>	Stop Mode Recovery Width Spec	1.9–3.6	12 <sup>3</sup>	—	ns	—
				10T <sub>pC</sub> <sup>4</sup>	—	—	—
11	T <sub>OST</sub>	Oscillator Startup Time <sup>4</sup>	1.9–3.6	—	5T <sub>pC</sub>	—	—
12	T <sub>WDT</sub>	Watchdog Timer Delay Time	1.9–3.6	5	—	ms	0, 0, 0, 0, 0
			1.9–3.6	10	—	ms	0, 0, 0, 0, 1
			1.9–3.6	20	—	ms	0, 0, 0, 1, 0
			1.9–3.6	80	—	ms	0, 0, 0, 1, 1
			1.9–3.6	320	—	ms	0, 0, 1, X, X
			1.9–3.6	1, 280	—	ms	0, 1, 0, X, X
			1.9–3.6	5, 120	—	ms	1, 0, 0, X, X

**Table 82. Clock, Reset, Timers, and SMR Timing (Continued)**

No	Symbol	Parameter	V <sub>CC</sub>	T <sub>A</sub> = 0 °C to +70 °C 8.0 MHz		Units	WDTMR (Bits 6, 5, 4, 1, 0)
				Min	Max		
13	T <sub>POR</sub>	Power-on reset	1.9–3.6	2.5	10 <sup>5</sup>	ms	—
14	f <sub>iramp</sub>	Frequency of input signal for IR amplifier	—	0	500	kHz	—

**Notes**

1. Timing Reference uses 0.9 V<sub>CC</sub> for a logic 1 and 0.1 V<sub>CC</sub> for a logic 0.
2. Interrupt request through Port 3 (P33:P31).
3. SMR – bit 5 = 1.
4. SMR – bit 5 = 0.
5. If bit 1 of the SMR register is programmed to 1, this value is 2.5 ms as measured from the time the oscillator input to XTAL1 reaches a peak to peak voltage oscillation of at least 300 mV.

**Table 83. Flash Memory Electrical Characteristics and Timing**

No	Symbol	Parameter	T <sub>A</sub> = 0 °C to +70 °C 8.0 MHz		Units	Condition
			Min	Max		
1	I <sub>FLP</sub>	Flash Memory Programming Current	—	10	mA	—
2	I <sub>FLE</sub>	Flash Memory Page/Mass Erase Current	—	6	mA	—
3	I <sub>RD</sub>	Flash Dynamic Read Current	—	420	a	Assumes reads every clock cycle with a 1 MHz clock
4	V <sub>FLPE</sub>	Flash Memory Program/Erase Voltage	2.1	3.6	V	—
5	V <sub>FLR</sub>	Flash Memory Read Voltage	1.8	3.6	V	—
6	T <sub>PROG</sub>	Flash Programming Time	30	60	us	—
7	T <sub>pe</sub>	Flash Page Erase Time	10	—	ms	—
8	T <sub>me</sub>	Flash Mass Erase Time	10	—	ms	—
9	T <sub>dr</sub>	Flash Data Retention Time	10	—	years	Temp=25 °C
10	F <sub>en</sub>	Flash Program/Erase Endurance	20,000	—	cycles	—

# Flash Option Bits

Programmable Flash Option Bits allow user configuration of certain aspects of ZLF645 MCU functionality. This configuration data is stored in the Flash memory Information Block and then read into option byte shadow registers during the last portion of the ZLF645 MCUs reset period.

Features available for control through the Flash Option Bits include:

- Port 0 low nibble pull-ups
- Port 0 high nibble pull-ups
- Port 1 low nibble pull-ups
- Port 1 high nibble pull-ups
- Port 2 pull-ups
- Port 3 low nibble pull-ups
- Port 4 pull-ups
- WDT always enabled
- Flash protect entire main memory
- Flash protect lower half main memory
- XTAL1 to System Clock (no division enable)
- 16-bit Stack addressibility enable

## Operation

### Option Bit Shadow Register Loading By Reset

For each Flash memory option bit, there is an associated option bit shadow register that is used to register the value of the option bit. The output of the option bit shadow registers are used by the ZLF645 MCU to enable various features and functions for the ZLF645 MCU. Each time the Flash Memory Information Block Option Bits are programmed or erased, the device must be reset for the change in ZLF645 configuration to take effect.

A POR or Stop Mode Recovery Reset with SMR bit 5 set to 1, loads the option bits from the Flash memory to the Option Bit Shadow registers during the last few clock cycles of the reset period. In some cases, in order to provide a required value before being loaded, the Option Bit Shadow registers are reset to a predefined value on the start of the reset period.



The Option Bit Shadow registers are part of the ZLF645’s Register File and are accessible for read/write access.

### User Option Bit Locations in Flash Memory

The user option bits are located in the upper two bytes of the Information block, address FFh and FEh.

### User Option Bit Shadow Register Access

Except for bits 0 and 1 of the User Option Byte 1 Shadow register, the CPU has full read/write access to all the User Option Byte Shadow registers at the address locations given in register tables for each register.

### User Option Byte 0 and Option Byte 0 Shadow Register Definitions

This option byte allows user control over the enabling of the ZLF645’s I/O pull-ups and the conditions under which the devices watchdog timer is enabled. For its associated shadow registers, until the registers are loaded with their corresponding option bit values, their outputs will be in an unknown state.

**Table 84. User Option Byte 0 (OPT0)**

Bit	7	6	5	4	3	2	1	0
Field	WDT	P4PU	P3PU	P2PU	P1HPU	P1LPU	P0HPU	P0LPU
Erased State	1	1	1	1	1	1	1	1
Flash Address	Flash Memory Information Area address: FEH							

Bit Position	Value	Description
[7]		<b>WDT—Watchdog Timer Enable</b>
	1	WDT = 1: WDT is enabled by WDT instruction only.
	0	WDT = 0: WDT is always enabled.
[7:0]		<b>P4PU through P0LPU—GPIO Pin Pull-Up Enables</b>
		P4P = 1: Port 4 Pull-ups are not enabled.
		P4P = 0: Port 4 Pull-ups enabled.
		P3PU = 0: Port 3 Pull-ups disabled.
		P3PU = 0: Port 3 Pull-ups enabled.

P2PU = 1: Port 2 Pull-ups disabled.  
P2PU = 0: Port 2 Pull-ups enabled.  
P1HPU = 1: Port 1 high nibble Pull-ups disabled  
P1HPU = 0: Port 1 high nibble Pull-ups enabled.  
P1LPU = 1: Port 1 low nibble Pull-ups disabled.  
P1LPU = 0: Port 1 low nibble Pull-ups enabled.  
P0HPU = 1: Port 0 high nibble Pull-ups disabled.  
P0HPU = 0: Port 0 high nibble Pull-ups enabled.  
P0LPU = 1: Port 0 low nibble Pull-ups disabled.  
P0LPU = 0: Port 0 low nibble Pull-ups enabled.

**Table 85. User Option Byte 0 Shadow Register (OPT0SR)**

Bit	7	6	5	4	3	2	1	0
Field	WDT	P4PU	P3PU	P2PU	P1HPU	P1LPU	P0HPU	P0LPU
Reset State	X	X	X	X	X	X	X	X
CPU Access (R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Register Address (R/W)	Bank D: 0Eh; Linear: D0Eh							

### User Option Byte 1 and Option Byte 1 Shadow Register Definitions

User Option byte allows the enabling of various features including protecting the Flash’s main memory from read operations through either of the ZLF645’s Flash access interfaces. For its associated shadow registers, until the registers are loaded with their corresponding option bit values, their outputs will be in an unknown state.

[Table 86](#) describes User Option byte 1 function.

► **Note:** *This byte can be programmed and erased (by Page 3 erase) only through the ICP.*

During device Power-on Reset, bit 1 and bit 0 value of this Flash Option byte are sampled into flip-flops, whose outputs control the Flash memory protect function. User codes can read the flip-flop values, by reading from a uniquely assigned peripheral register address.

► **Note:** *User codes cannot over-write the flip-flop values to change this Flash memory protect function.*





**Table 86. User Option Byte 1 (OPT1)**

Bit	7	6	5	4	3	2	1	0
Field	Reserved				16BITSTK	DIVBY1	FLPROT1	FLRWP
Erased State	1	1	1	1	1	1	1	1
Flash Address	Flash Memory Information Area address: FFH							

Bit Position	Value	Description
[7:4]	—	<b>Reserved</b> —Must be written 1.
[3]		<b>16BITSTK</b> —16 bit Stack Pointer Addressibility Enable
	1	The ZLF645 is enabled for 8-bits of stack pointer addressibility allowing usage of Bank 0 only of the devices general-purpose RAM space as the CPU stack.
	0	The ZLF645 is enabled for 16-bits of stack pointer addressibility allowing usage of all of the devices general-purpose RAM space as the CPU stack.
[2]		<b>DIVBY1</b> —System Clock Divide By 1 Enable
	1	If SMR register bit 0 is also programmed to 0, the system clock frequency is equal to the external clock frequency input on the XTAL1 pin divided by 2.
	0	If SMR register bit 0 is also programmed to 0, the system clock frequency is equal to the external clock frequency input on the XTAL1 pin.
[1]		<b>FLPROT1</b> —Flash Main Memory Lower Half Protect
	1	The Flash main memory and all of Information Area Page 3 can be read, written, and erased by both the Flash Byte Programming interface or through the ICP interface as long as FLRWP is also 1.
	0	Reads and Writes to the lower half of Flash main memory and writes and erasures to Information Area Page 3, by the ICP or Flash Byte Programming interfaces is disabled unless, with this bit 0, a main memory mass erase is completed first. A main memory mass erase causes resetting of this bit value in the Option Byte 1 shadow register to a 1 but does not effect the corresponding Flash memory bit. Once the Option Byte 1 shadow register bit is reset, the ICP or Flash Byte Programming interface is allowed full read, write, and erase access to the Flash's main memory and to Page 3 of the Information Area and can reset the corresponding Flash memory bit.




---

[0]	<b>FLRWP</b> —Flash Main Memory Protect
1	Flash Main Memory and Information Area Page 3 can be read, programmed, and erased by both the Flash Byte Programming interface or through the ICP interface.
0	Reads and writes to the Flash main memory and writes and erasures to Information Area Page 3 by the ICP or Flash Byte Programming interfaces is disabled unless, with this bit 0, a main memory mass erase is completed first. A main memory mass erase causes resetting of this bit value in the Option Byte 1 shadow register to a 1 but does not effect the corresponding Flash memory bit. Once the Option Byte 1 shadow register bit is reset, the ICP or Flash Byte Programming interface is allowed full read, write, and erase access to the Flash’s main memory and to Page 3 of the Information Area and can reset the corresponding Flash memory bit.

---

**Table 87. User Option Byte 1 Shadow Register (OPT1SR)**

Bit	7	6	5	4	3	2	1	0
Field	RESERVED				16BITSTK	DIVBY1	FLPROT1	FLRWP
Reset State	X	X	X	X	X	1	X	X
CPU Access (R/W)					R/W	R/W	R	R
Register Address (R/W)	Bank D: 0Fh; Linear: D0Fh							

► **Note:** *RESERVED bits when read by the CPU will return 0 and when written have no effect.*

# Packaging

Figure 50 displays the 20-pin quad flat no-lead (QFN) package for the ZLF645 Series of Flash MCUs.

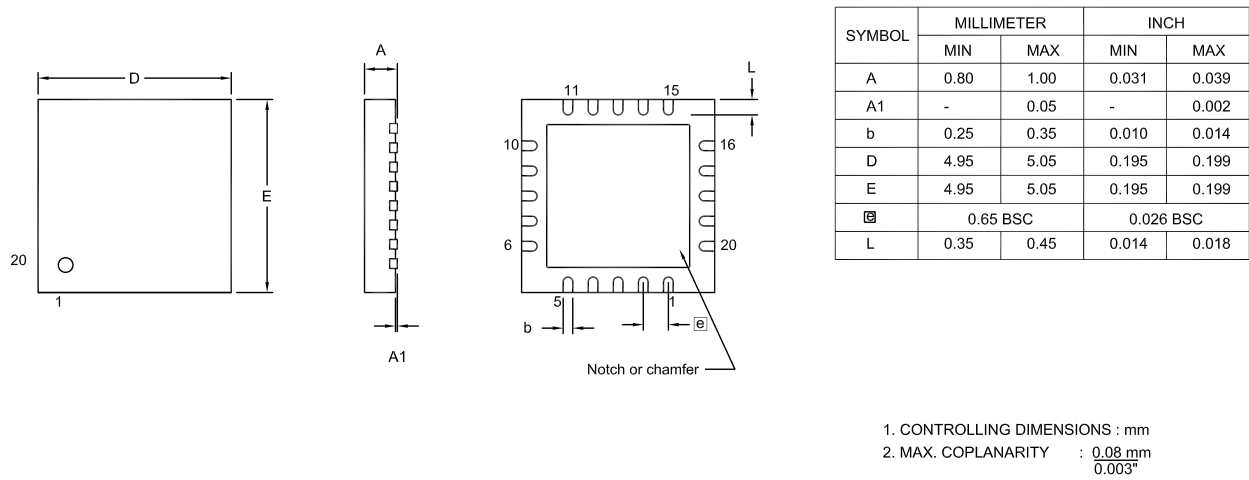


Figure 50. 20-Pin QFN Package Diagram

Figure 51 displays the 28-pin shrink small outline package (SSOP) for the ZLF645 Series of Flash MCUs.

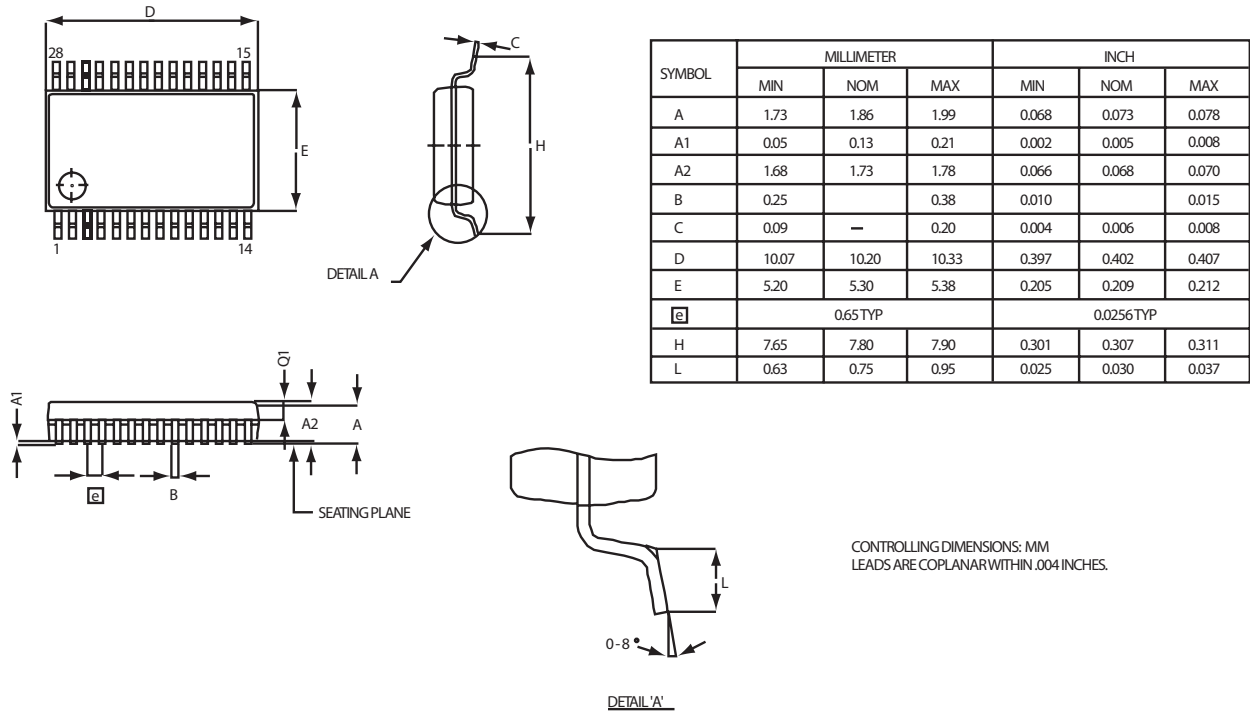


Figure 51. 28-Pin SSOP Package Diagram

Figure 52 displays the 28-pin small outline integrated circuit (SOIC) package for the ZLF645 Series of Flash MCUs.

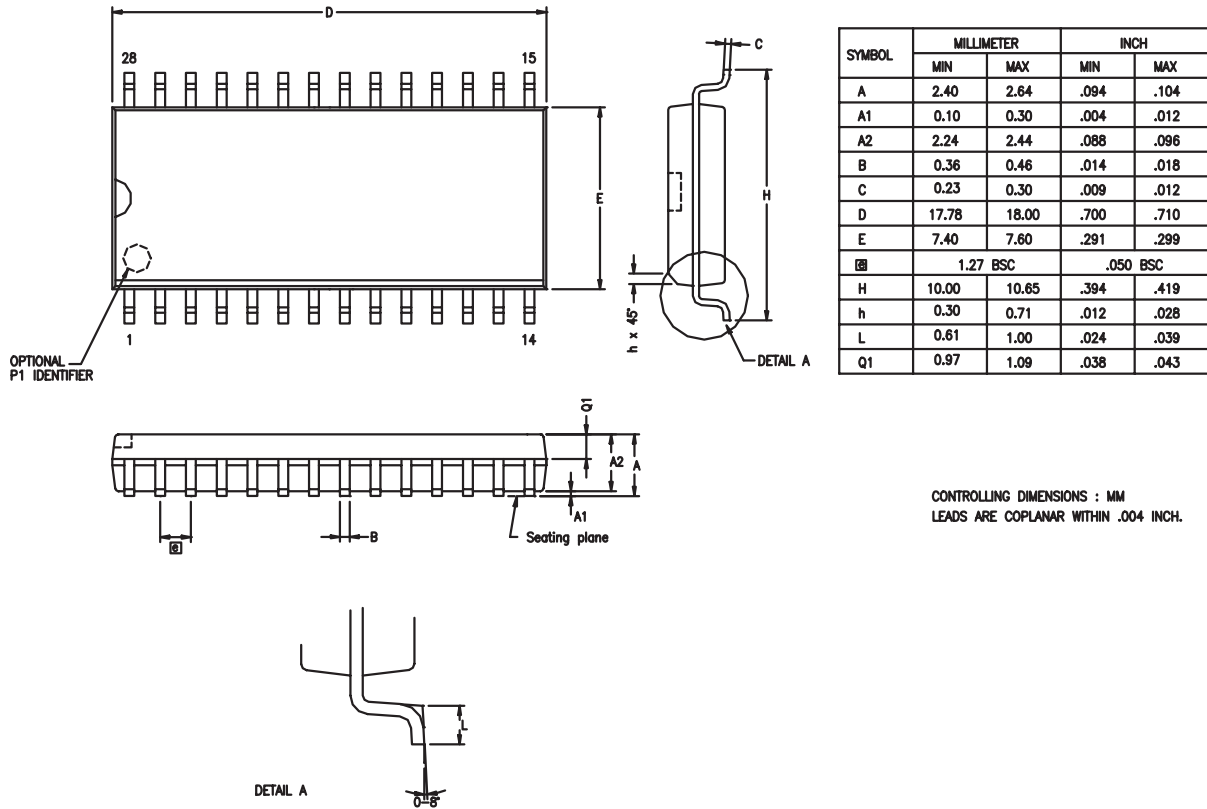
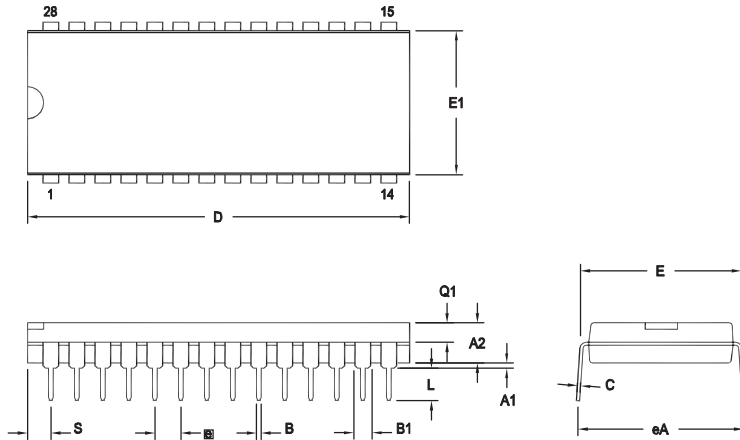


Figure 52. 28-Pin SOIC Package Diagram

Figure 53 displays the 28-pin plastic dual in line package (PDIP) for the ZLF645 Series of Flash MCUs.



SYMBOL	OPT #	MILLIMETER		INCH	
		MIN	MAX	MIN	MAX
A1		0.38	1.02	.015	.040
A2		3.18	4.19	.125	.165
B		0.38	0.53	.015	.021
B1	01	1.40	1.65	.055	.065
	02	1.14	1.40	.045	.055
C		0.23	0.38	.009	.015
D	01	36.58	37.34	1.440	1.470
	02	35.31	35.94	1.390	1.415
E		15.24	15.75	.600	.620
E1	01	13.59	14.10	.535	.555
	02	12.83	13.08	.505	.515
□		2.54 TYP		.100 BSC	
eA		15.49	16.76	.610	.660
L		3.05	3.81	.120	.150
Q1	01	1.40	1.91	.055	.075
	02	1.40	1.78	.055	.070
S	01	1.52	2.29	.060	.090
	02	1.02	1.52	.040	.060

CONTROLLING DIMENSIONS : INCH

OPTION TABLE	
OPTION #	PACKAGE
01	STANDARD
02	IDF

Note: Zilog supplies both options for production. Component layout PCB design should cover bigger option 01.

Figure 53. 28-Pin PDIP Package Diagram

Figure 54 displays the 20-pin shrink small outline package (SSOP) for the ZLF645 Series of Flash MCUs.

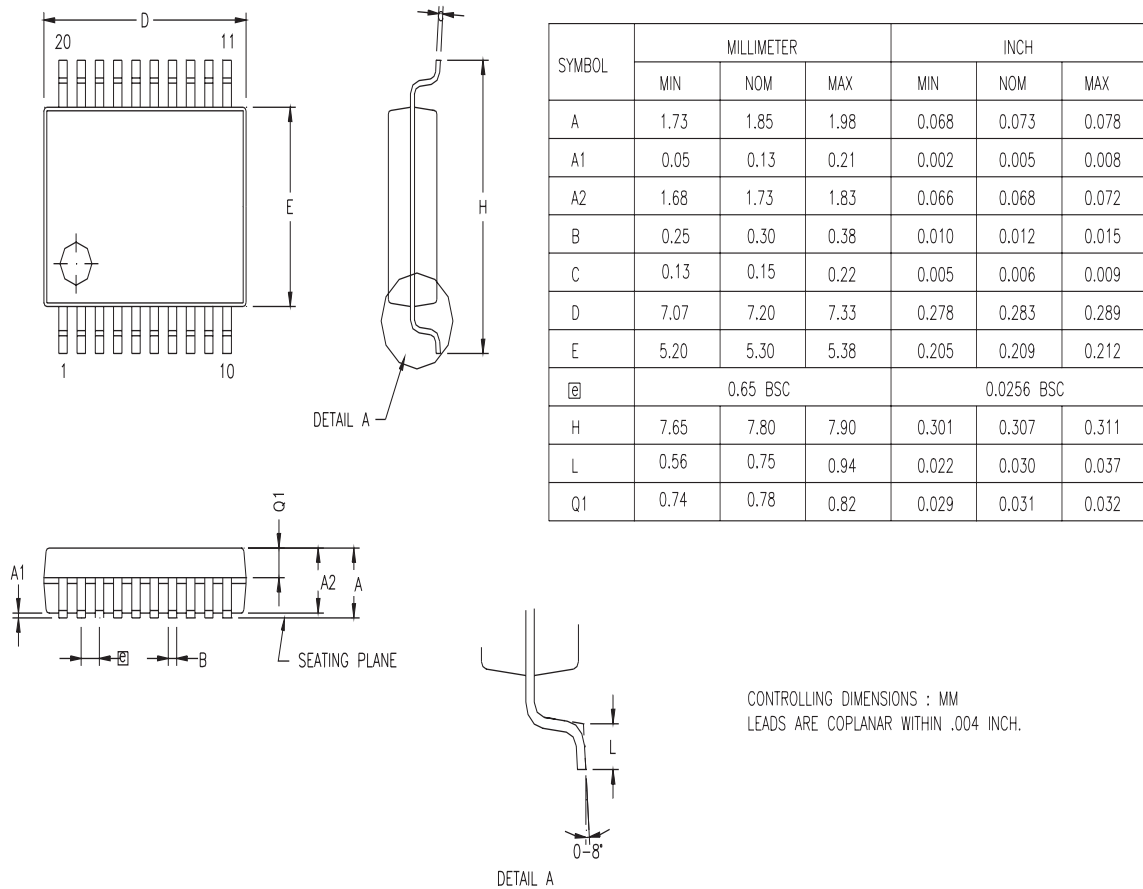


Figure 54. 20-Pin SSOP Package Diagram

Figure 55 displays the 20-pin small outline integrated circuit (SOIC) package for the ZLF645 Series of Flash MCUs.

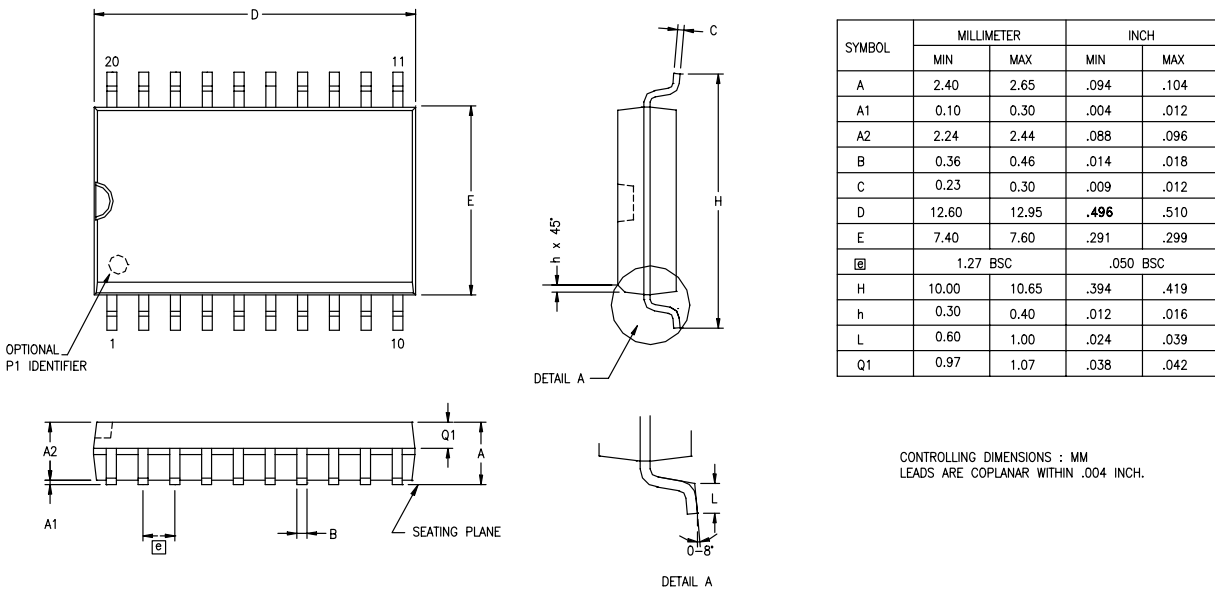


Figure 55. 20-Pin SOIC Package Diagram



Figure 56 displays the 20-pin plastic dual in line package (PDIP) for the ZLF645 Series of Flash MCUs.

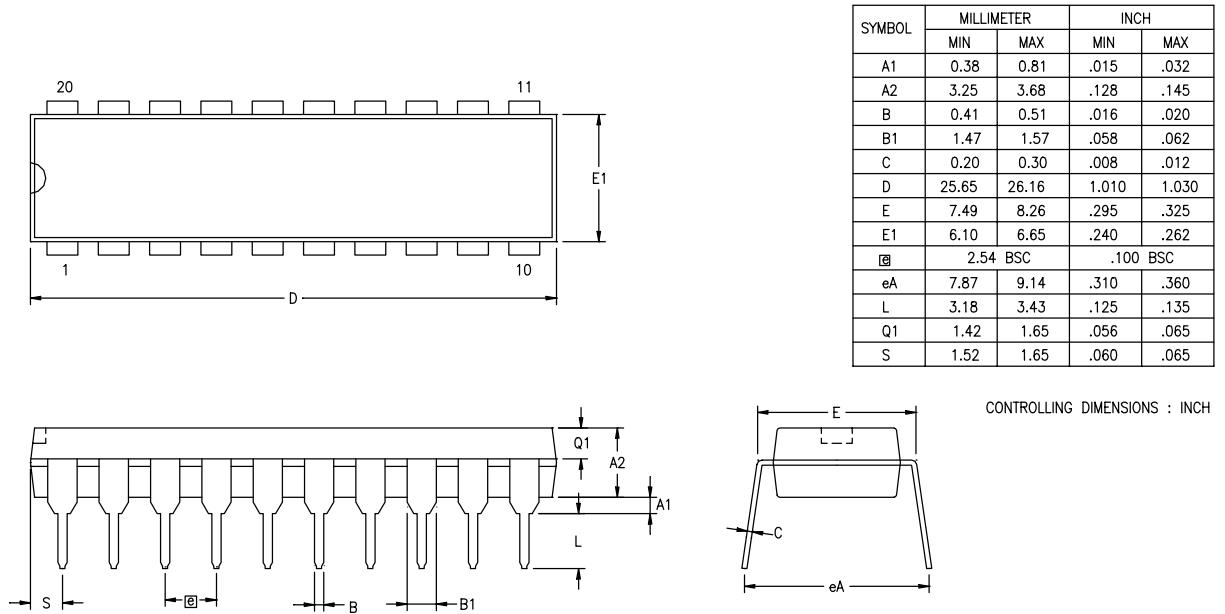


Figure 56. 20-Pin PDIP Package Diagram

Figure 57 displays the 48-pin shrink small outline package (SSOP) for the ZLF645 Series of Flash MCUs.

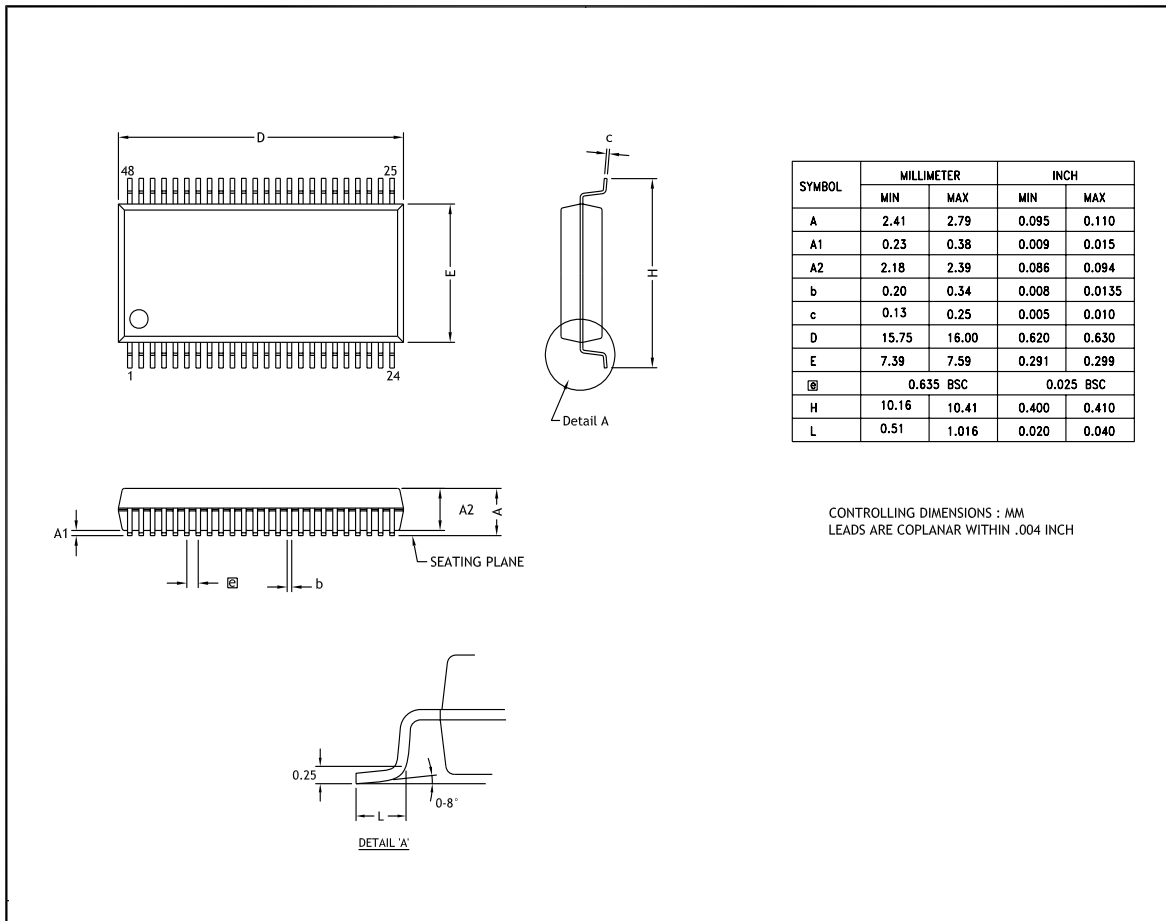


Figure 57. 48-Pin SSOP Package Diagram

Figure 58 displays the 20-pin Quad Flat No Lead (QFN) for the ZLF645 Series of Flash MCUs.

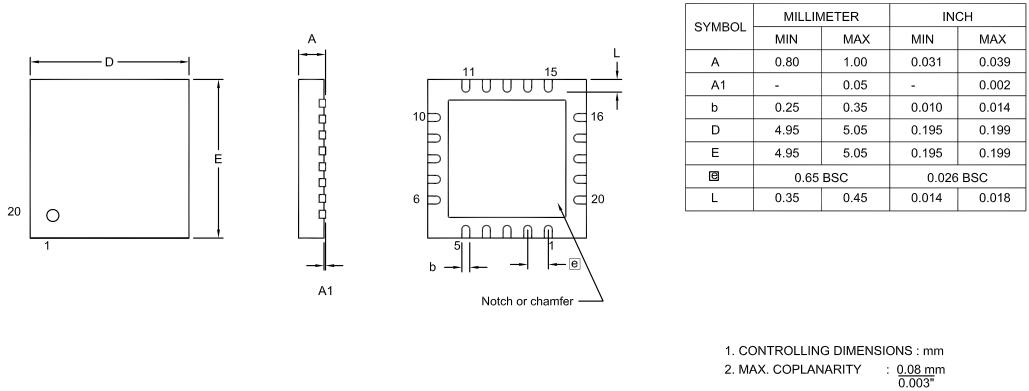


Figure 58. 20-Pin QFN Package Diagram

# Ordering Information

Table 88 lists the part numbers for ZLF645 Series of Flash MCUs and a brief description of each part.

**Table 88. ZLF645 Flash MCU Part Numbers Description**

Part Number	Flash (KB)	RAM	Description
<b>ZLF645 Flash MCU with 512 B RAM</b>			
ZLF645S0H2064G	64	512 B	SSOP 20-pin package
ZLF645S0H2864G	64	512 B	SSOP 28-pin package
ZLF645S0H4864G	64	512 B	SSOP 48-pin package
ZLF645S0P2064G	64	512 B	PDIP 20-pin package
ZLF645S0P2864G	64	512 B	PDIP 28-pin package
ZLF645S0S2064G	64	512 B	SOIC 20-pin package
ZLF645S0S2864G	64	512 B	SOIC 28-pin package
ZLF645S0Q2064G	64	512 B	QFN 20-pin package
ZLF645S0H2032G	32	512 B	SSOP 20-pin package
ZLF645S0H2832G	32	512 B	SSOP 28-pin package
ZLF645S0H4832G	32	512 B	SSOP 48-pin package
ZLF645S0P2032G	32	512 B	PDIP 20-pin package
ZLF645S0P2832G	32	512 B	PDIP 28-pin package
ZLF645S0S2032G	32	512 B	SOIC 20-pin package
ZLF645S0S2832G	32	512 B	SOIC 28-pin package
ZLF645S0Q2032G	32	512B	QFN 20-pin package
<b>ZLF645 Flash MCU with 1K RAM</b>			
ZLF645E0H2064G	64	1 K	SSOP 20-pin package
ZLF645E0H2864G	64	1 K	SSOP 28-pin package
ZLF645E0H4864G	64	1 K	SSOP 48-pin package
ZLF645E0P2064G	64	1 K	PDIP 20-pin package
ZLF645E0P2864G	64	1 K	PDIP 28-pin package
ZLF645E0S2064G	64	1 K	SOIC 20-pin package
ZLF645E0S2864G	64	1 K	SOIC 28-pin package



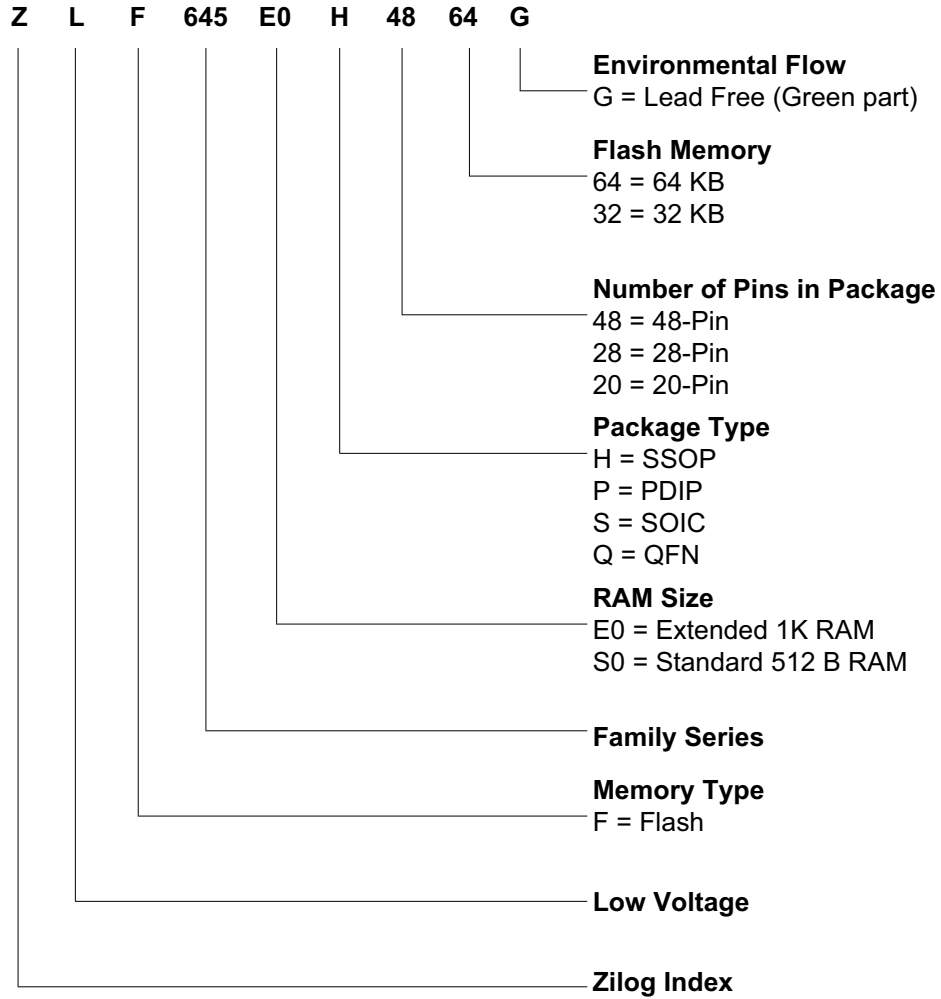
**Table 88. ZLF645 Flash MCU Part Numbers Description (Continued)**

Part Number	Flash (KB)	RAM	Description
ZLF645E0Q2064G	64	1K	QFN 20-pin package
ZLF645E0H2032G	32	1 K	SSOP 20-pin package
ZLF645E0H2832G	32	1 K	SSOP 28-pin package
ZLF645E0H4832G	32	1 K	SSOP 48-pin package
ZLF645E0P2032G	32	1 K	PDIP 20-pin package
ZLF645E0P2832G	32	1 K	PDIP 28-pin package
ZLF645E0S2032G	32	1 K	SOIC 20-pin package
ZLF645E0S2832G	32	1 K	SOIC 28-pin package
ZLF645E0Q2032G	32	1K	QFN 20-pin package
ZCRMZNICE01ZEMG			Crimzon ICE Kit (includes Smart Cable for ZLF645 in-circuit programming)
ZCRMZN00100KITG			Crimzon/ZLF645 IR Development Board Kit
ZCRMZNICE01ZACG			Crimzon/ZLF645 20-Pin Accessory Kit
ZCRMZNICE02ZACG			Crimzon/ZLF645 40-/48-Pin Accessory Kit



## Part Number Description

Zilog<sup>®</sup> part numbers consist of a number of components as shown below:



# Index

## Numerics

- 11890
  - Figure Title
    - Figure 34. Resets and WDT 133
- 12-bit address map 47
- 16-bit counter/timer circuits 103
- 20-pin
  - package pins 6, 8, 9
  - PDIP package 177, 178
  - SOIC package 176
  - SSOP package 175
- 28-pin
  - package pins 11, 13, 16
  - PDIP package 174
  - SOIC package 173
  - SSOP package 171, 172
- 32178
  - Figure Title
    - Figure 17. T8\_OUT in MODULO-N Mode 98
- 40203
  - Figure Title
    - Figure 16. T8\_OUT in SINGLE-PASS Mode 98
- 60053
  - Table Title
    - Table 32. Stop Mode Recovery Register 2 150
- 64329
  - Table Title
    - Table 37. Stop Mode Recovery Register 4 151
- 8-bit counter/timer circuits 97

## A

- absolute maximum ratings 158
- AC characteristics 163, 164
- AC timing 163

- active low notation 3
- address
  - 12-bit linear 47
  - notation 152
- amplifier, infrared 78
- AND caution 113
- architecture
  - UART 80
- asynchronous data 81

## B

- baud rate generator
  - description 87
  - example 88
  - interrupt 87
- Baud Rate Generator Constant register (BCNST) 91, 92
- block diagram
  - counter/timer 93
  - interrupt 122
  - MCU 4
  - reset and watch-dog timer 133
  - UART 80
- brown-out, voltage 138
- brownout, voltage 134

## C

- capacitance 159
- caution
  - open-drain output 18
  - stopping timer 98
  - timer count 97
  - timer modes 115
  - timer registers 104
  - UART transmit 82, 83
- characteristics
  - AC 163, 164
  - DC 160

- clock 129
  - internal signals 130
- CMOS gate, caution 18
- comparator
  - inputs 28
  - outputs 28
- condition codes 156
- conditions, test 159
- connection, power 3
- constant memory 41
- constant, baud rate 91, 92
- counter/timer
  - block diagram 93
  - capture flowchart 100
  - input circuit 95
  - output configuration 27
- crystal 129
- crystal oscillator pins (XTAL1, XTAL2) 129, 130
- Customer Support 189

## D

- data format, UART 81
- data handling, UART 85
- DC characteristics 160
- demodulation
  - changing mode 115
  - flowchart 100, 101
  - timer 99, 104
- demodulation mode flowchart 102
- device
  - block diagram 4
  - features 1
- diagram, package 171, 172, 173, 174, 175, 176, 177, 178
- divisor, baud rate 91, 92

## E

- electrical characteristics 158
- error handling, UART 85
- example
  - BCNST register 88
  - register pointer 45

## F

- fast recovery, stop mode 138
- features, device 1
- flags register 155
- flash
  - option bit configuration - reset 166
- flash memory 64, 76
  - arrangement 65
  - byte programming 69
  - code protection 68
  - configurations 64
  - control register definitions 71
  - flash status register 72
  - flow chart 67
  - frequency high and low byte registers 74
  - mass erase 70
  - operation 66
  - operation timing 68
  - page erase 70
  - page select register 72, 73
- floating CMOS gate, caution 18
- flowchart
  - demodulation mode 100, 101, 102
  - timer transmit 96
  - UART receive 86
- format, UART data 81
- FPS register 72, 73
- FSTAT register 72
- functional block diagram 4
- functions, I/O port pins 18

## H

- h suffix 98
- HALT mode 138
- handshaking, UART 84

## I

- I/O port pin functions 18
- ICP
  - architecture 54
  - auto-baud detector/generator 55



- baud rate limits 56
  - block diagram 54
  - commands 57
  - control register 61
  - data format 55
  - debug mode 54
  - serial errors 56
  - status register 62
  - ICP commands
    - read ICP control register (05H) 59
    - read ICP revision (00H) 58
    - read ICP status register (02H) 59
    - read program memory (0BH) 60
    - read program memory CRC (0EH) 60
    - read register (09H) 59
    - read runtime counter (03H) 59
    - write ICP control register (04H) 59
    - write program memory (0AH) 59
    - write register (08H) 59
  - ICP Interface 53
  - infrared learning amplifier 78
  - input
    - comparator 28
    - counter/timer 95
    - timers 94
  - instruction set summary 157
  - instruction symbols 154
  - internal clock 130
  - interrupt
    - baud rate generator 87
    - block diagram 122
    - description 121
    - mask register 127
    - priority register 124
    - request register 124, 125
    - source 123
    - stop-mode recovery 138
    - type 123
    - UART 84
    - UART receive 83, 84, 86
    - UART transmit 82, 84
    - vector 123
  - Interrupt Mask Register (IMR) 127
  - Interrupt Priority Register (IPR) 124
  - Interrupt Request Register (IRQ) 124, 125
- L**
- LDE and LDEI instructions removed 46
  - LDX, LDXI instruction addresses 47
  - leakage, caution 18
  - learning amplifier, infrared 78
  - linear address 45
  - load, test 159
  - Low-Voltage Detection Register (LVD) 135
- M**
- map
    - program/constant memory 41
    - register 12-bit 47
    - register 8-bit 43
    - register file summary 50
  - maximum ratings 158
  - MCU
    - block diagram 4
    - features 1
  - memory
    - address, linear 45
    - program/constant map 41
    - register 12-bit map 47
    - register file map 43
    - register file summary 50
  - modulo-N mode 98, 104
- N**
- notation
    - addressing 152
    - operand 152
- O**
- open-drain output caution 18
  - operand symbols 152
  - operation, UART 80
  - Operational Description 64, 166

OR caution 113  
oscillator 129  
output  
    comparator 28  
    timer/counter 106  
    timer/counter circuit 107  
    timer/counter configuration 27  
overline, in text 3  
overrun, UART 85

## P

package diagram 171, 172, 173, 174, 175, 176, 177, 178  
package information 171  
parity, UART data 81  
pin description 5  
pin function  
    port 0 20  
    port 2 21, 28  
    port 3 23  
    port 3 summary 26  
ping-pong mode 105, 106  
pins  
    20-pin package 6, 8, 9  
    28-pin package 11, 13, 16  
polled UART receive 83  
polled UART transmit 81  
port 0  
    configuration 20  
    pin function 20  
Port 0 Mode Register (P01M) 31  
Port 0 Register (P0) 32  
port 2  
    configuration 22  
    pin function 21, 28  
Port 2 Mode Register (P2M) 34, 39  
Port 2 Register (P2) 33, 35, 40  
port 3  
    configuration 24  
    counter/timer output 27  
    pin function 23  
    pin function 26  
Port 3 Mode Register (P3M) 36

Port 3 Register (P3) 37  
Port Configuration Register (PCON) 28, 30  
port pin functions 18  
power connection 3  
power management 132  
power-on reset timer 134  
program memory  
    map 41  
programming summary 152  
pull-up, disabled 18

## R

ratings, maximum 158  
register  
    BCNST 92  
    CTR1 115  
    CTR3 119  
    flash high and low byte (FFREQH and FRE-EQL) 74  
    flash page select (FPS) 72, 73  
    flash status (FSTAT) 72  
    HI16 109  
    HI8 108  
    ICP control 61  
    ICP status 62  
    IMR 127  
    IPR 124  
    IRQ 124, 125  
    LO16 110  
    LO8 109  
    LVD 135  
    P0 32  
    P01M 31  
    P2 33, 35, 40  
    P2M 34, 39  
    P3 37  
    P3M 36  
    PCON 28, 30  
    Register Pointer register 48  
    RP 48  
    SMR 141  
    SMR1 145  
    SMR2 147

- SMR3 150
  - SMR4 151
  - SPL 48, 49
  - UCTL 90, 91
  - URDATA 89
  - USER 48
  - UST 89
  - UTDATA 89
  - WDTMR 137
  - register file
    - 12-bit address 47
    - address summary 50
    - description 42
    - memory map 43
  - register pointer
    - detail 44
    - example 45
  - Register Pointer Register 48
  - Register Pointer Register (RP) 48
  - Register Pointer register (RP) 48
  - reset
    - block diagram 133
    - delay bypass 138
    - features 132
    - POR timer 134
    - status 137
    - timer terminal count 113
- S**
- SCLK signal 130
  - single-pass mode 98
  - source
    - interrupt 123
    - stop mode recovery 138
    - stop-mode recovery 140, 144, 146
  - stack 42
  - Stack Pointer Register (SPL) 48, 49
  - standard test conditions 159
  - standby, brown-out 134
  - standby, brownout 138
  - status
    - reset 137
    - UART 89
  - stop bit, UART 85
  - stop mode
    - fast recovery 138
  - Stop Mode Recovery Register (SMR) 141
  - Stop Mode Recovery Register 1 (SMR1) 145
  - Stop Mode Recovery Register 2 (SMR2) 147
  - Stop Mode Recovery Register 3 (SMR3) 150
  - Stop Mode Recovery Register 4 (SMR4) 151
  - stop-mode
    - description 138
    - recovery events 138, 139, 142, 145, 147
    - recovery interrupt 138
    - recovery source 138, 140, 144, 146
    - recovery status 137
  - Stop-Mode Recovery Register 4 (SMR4) 151
  - suffix, h 98
  - symbols
    - address 152
    - instruction 154
    - operand 152
- T**
- T16\_OUT signal
    - modulo-N mode 104
  - T8\_OUT signal
    - modulo-N mode 98
    - single-pass mode 98
  - TCLK signal 130
  - terminal count, reset 113
  - test conditions 159
  - test load 159
  - timer
    - block diagram 93
    - changing mode 115
    - description 93
    - input circuit 94, 95
    - output circuit 107
    - output configuration 27
    - output description 106
    - reset 134
    - starting count caution 97
    - stopping caution 98
    - T16 demodulation 104

- T16 transmit 103
- T16\_OUT signal 104
- T8 demodulation 99
- T8 transmit 95
- T8\_OUT signal 98
- transmit flowchart 96
- transmit versus demodulation mode 115
- Timer 16 Capture High Register (HI16) 109
- Timer 16 Capture Low Register (L016) 110
- Timer 16 Control register (CTR2) 118
- Timer 16 High Hold register (TC16H) 110
- Timer 16 Low Hold Register (TC16L) 111
- Timer 8 and Timer 16 Common Functions Register (CTR1) 115
- Timer 8 Capture High Register (HI8) 108
- Timer 8 Capture Low Register (L08) 109
- Timer 8 Control Register (CTR0) 113
- Timer 8 High Hold Register (TC8H) 111
- Timer 8 Low Hold Register (TC8L) 112
- Timer 8/Timer 16 Control Register (CTR3) 119
- timing, AC 163
- transmit caution, UART 82, 83
- transmit mode
  - caution 115
  - flowchart 96
  - timer 95, 103

## U

### UART

- architecture 80
- baud rate generator 87
- block diagram 80
- data and error handling 85
- data format 81
- interrupts 84
- operation 80
- overrun error 85
- polled receive 83
- polled transmit 81
- receive interrupt 83, 84, 86
- stop bit 85
- transmit caution 82, 83
- transmit interrupt 82, 84

- UART Control Register (UCTL) 90, 91
- UART Receive/Transmit Data Register (URDATA/UTDATA) 89
- UART Status Register (UST) 89
- User Data Register (USER) 48

## V

- vector, interrupt 123
- voltage
  - brown-out 134, 138
  - detection 138
  - detection register 135

## W

- Watchdog Timer
  - description 136
- watchdog timer
  - diagram 133
- Watchdog Timer Mode Register (WDTMR) 137

## X

- XTAL1 pin 129
- XTAL2 pin 130

## Z

- ZLR64400 MCU
  - block diagram 4
  - features 1

# Customer Support

For answers to technical questions about the product, documentation, or any other issues with Zilog's offerings, please visit Zilog's Knowledge Base at <http://www.zilog.com/kb>.

For any comments, detail technical questions, or reporting problems, please visit Zilog's Technical Support at <http://support.zilog.com>.