

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

概述

MAX3108A为小尺寸通用异步收发器(UART), 每路接收和发送FIFO具有128个字, 通过串行I²C或SPI控制器接口控制。自动休眠和关断模式有助于降低待机功耗。500μA (最大)低电源电流、1μA关断电流以及微小的25焊球WLP (2.1mm x 2.1mm)封装使器件非常适合于低功耗便携装置。器件工作在1.71V至3.6V电源电压。

MAX3108A的波特率高达24Mbps, 非常适合于当今的高数据率应用。锁相环(PLL)、预分频器和分数波特率发生器支持高分辨率波特率编程, 将波特率受基准时钟频率的制约降至最小。

4路GPIO可用作输入、输出或中断输入。配置为输出时, 可设置为开漏输出, 并可吸入高达20mA的电流。

器件非常适合于便携和手持装置, 采用25焊球(2.1mm x 2.1mm)、0.4mm焊距、WLP封装, -40°C至+85°C扩展级工作温度范围。

应用

便携式通信设备
移动互联网设备
低功耗手持设备
医疗系统
销售终端(POS)

IrDA是Infrared Data Association Corporation.的服务商标。

特性

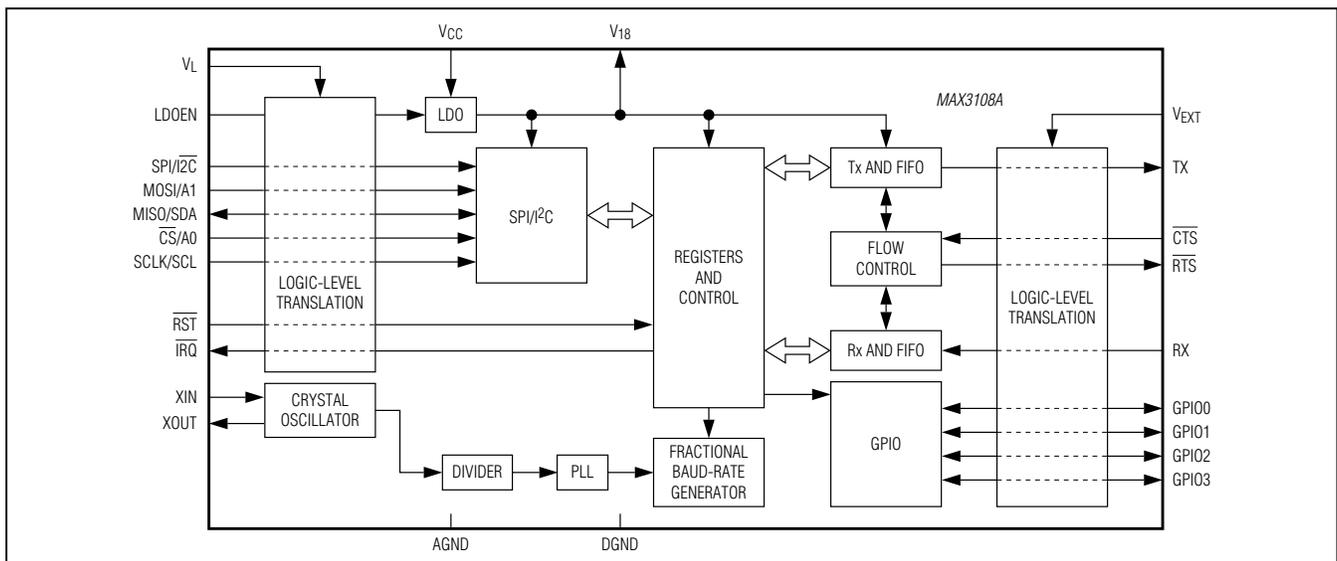
- ◆ 24Mbps (最大)波特率
- ◆ 1μA关断电流
- ◆ 1.71V至3.6V电源范围
- ◆ 支持高分辨率波特率编程
- ◆ SPI时钟速率高达26MHz
- ◆ 快速脉冲模式I²C, 速率高达1MHz
- ◆ $\overline{\text{RTS}}$ 和 $\overline{\text{CTS}}$ 自动硬件流控制
- ◆ XON/XOFF自动软件流控制
- ◆ 特殊字符检测
- ◆ 9位多点模式数据滤波
- ◆ SIR和MIR兼容IrDAS^M编码器/解码器
- ◆ 控制器和收发器接口提供灵活的逻辑电平
- ◆ 4路灵活的GPIO
- ◆ 电源噪声指示
- ◆ 关断和自动休眠模式
- ◆ 集成PLL和分频器
- ◆ 寄存器与MAX3107、MAX3109和MAX14830兼容
- ◆ 25焊球、WLP封装(2.1mm x 2.1mm)

订购信息

器件	温度范围	引脚-封装
MAX3108AEWA+T	-40°C至+85°C	25 WLP

+表示无铅(Pb)/符合RoHS标准的封装。

功能框图



本文是英文数据资料的译文, 文中可能存在翻译上的不准确或错误。如需进一步确认, 请在您的设计中参考英文资料。有关价格、供货及订购信息, 请联络Maxim亚洲销售中心: 10800 852 1249 (北中国区), 10800 152 1249 (南中国区), 或访问Maxim的中文网站: china.maximintegrated.com。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

目录

Absolute Maximum Ratings	6
Package Thermal Characteristics	6
DC Electrical Characteristics	6
AC Electrical Characteristics	9
时序图	11
典型工作特性	12
焊球配置	13
焊球说明	13
详细说明	15
寄存器组	15
接收和发送FIFO	15
发送器工作	15
接收器工作	16
线路噪声指示	17
时钟选择	18
晶振	18
外部时钟源	18
PLL和预分频器	18
分数波特率发生器	18
2倍/4倍速模式	19
多点模式	19
多点模式下的自动数据筛选	19
自动收发器方向控制	20
回波抑制	21
自动控制硬件流	22
AutoRTS控制	22
AutoCTS控制	22
自动软件(XON/XOFF)流控制	22
接收器流控制	22
发送器流控制	23
FIFO中断触发	23
低功耗待机模式	23
强制休眠模式	23
自动休眠模式	23
关断模式	23
上电和 $\overline{\text{IRQ}}$	23
中断结构	24

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

目录(续)

中断使能	24
中断清除	24
寄存器映射	25
寄存器详细说明	26
串行控制器接口	48
SPI接口	48
SPI单周期访问	48
SPI突发存取	48
I ² C接口	49
START、STOP和重复START条件	49
从地址	49
位传输	49
单字节写	50
突发写	50
单字节读	51
突发读	51
应答位	52
应用信息	52
启动和初始化	52
低功耗工作	53
中断和轮询	53
逻辑电平转换	53
电源排序	54
连接器共用	54
RS-232 5 x 3应用	54
典型应用电路	55
芯片信息	55
封装信息	55
修订历史	56

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

图示目录

图1. I ² C时序图.....	11
图2. SPI时序图.....	11
图3. 发送FIFO信号.....	15
图4. 接收数据格式.....	16
图5. 发送FIFO.....	16
图6. 位中间采样.....	17
图7. 时钟选择框图.....	17
图8. 2倍速和4倍速波特率.....	19
图9. 自动收发器方向控制.....	20
图10. 自动收发器方向控制中的建立和保持时间.....	20
图11. 带回波抑制的半双工.....	21
图12. 回波抑制时序.....	21
图13. 中断结构简化图.....	24
图14. PLL信号通路.....	45
图15. 单周期读操作.....	48
图16. 单周期写操作.....	48
图17. I ² C START、STOP和重复START条件.....	49
图18. 写字节序列.....	50
图19. 突发写序列.....	50
图20. 读字节序列.....	51
图21. 突发读序列.....	51
图22. 应答.....	52
图23. 启动和初始化流程图.....	52
图24. 逻辑电平转换.....	53
图25. 与USB收发器共用连接器.....	54
图26. RS-232应用.....	54
图27. RS-485半双工应用.....	55

表格目录

表1. StopBits真值表.....	37
表2. Lengthx真值表.....	37
表3. SwFlow[3:0]真值表.....	42
表4. PLLFactorx选择指南.....	45
表5. I ² C寄存器映射.....	49

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

图示目录

接收保持寄存器(RHR)26
发送保持寄存器(THR)26
IRQ使能寄存器(IRQEn)27
中断状态寄存器(ISR)28
线路状态中断使能寄存器(LSRIntEn)29
线路状态寄存器(LSR)30
特殊字符中断使能寄存器(Spc1ChrIntEn)31
特殊字符中断寄存器(Spc1CharInt)32
STS中断使能寄存器(STSIntEn)33
状态中断寄存器(STSInt)34
MODE1寄存器35
MODE2寄存器36
线路控制寄存器(LCR)37
接收器超时寄存器(RxTimeOut)38
HDplxDelay寄存器38
IrDA寄存器39
流控制门限寄存器(FlowLvl)39
FIFO中断触发点寄存器(FIFOTrgLvl)40
发送FIFO空/满状态寄存器(TxFIFOLvl)40
接收FIFO空/满状态寄存器(RxFIFOLvl)40
流控制寄存器(FlowCtrl)41
XON1寄存器42
XON2寄存器43
XOFF1寄存器43
XOFF2寄存器44
GPIO配置寄存器(GPIOConfig)44
GPIO数据寄存器(GPIOData)45
PLL配置寄存器(PLLConfig)45
波特率发生器配置寄存器(BRGConfig)46
波特率发生器除数LSB寄存器(DIVLSB)46
波特率发生器除数MSB寄存器(DIVMSB)47
时钟源寄存器(CLKSource)47

MAX3108A

帶有128字FIFO的SPI/I²C UART, WLP封裝

ABSOLUTE MAXIMUM RATINGS

(Voltages referenced to AGND.)

V_L, V_{CC}, V_{EXT}, XIN -0.3V to +4.0V
 XOUT -0.3V to (V_{CC} + 0.3V)
 V₁₈ -0.3V to the lesser of (V_{CC} + 0.3V) and 2.0V
 RST, IRQ, MOSI/A1, CS/A0, SCLK/SCL,
 MISO/SDA, LDOEN, SPI/I²C -0.3V to (V_L + 0.3V)
 TX, RX, RTS, CTS, GPIO_ -0.3V to (V_{EXT} + 0.3V)

DGND -0.3V to +0.3V
 Continuous Power Dissipation (T_A = +70°C)
 WLP (derate 19.2mW/°C above +70°C) 1536mW
 Operating Temperature Range -40°C to +85°C
 Maximum Junction Temperature +150°C
 Storage Temperature Range -65°C to +150°C
 Soldering Temperature (reflow) +260°C

PACKAGE THERMAL CHARACTERISTICS (Note 1)

WLP

Junction-to-Ambient Thermal Resistance (θ_{JA}) 52°C/W

Junction-to-Case Thermal Resistance (θ_{JC}) 11°C/W

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to china.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 1.71V to 3.6V, V_L = 1.71V to 3.6V, V_{EXT} = 1.71V to 3.6V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{CC} = 2.8V, V_L = 1.8V, V_{EXT} = 2.5V, T_A = +25°C.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Digital Interface Supply Voltage	V _L		1.71		3.6	V
Analog Supply Voltage	V _{CC}	Internal PLL disabled and bypassed	1.71		3.6	V
		Internal PLL enabled	2.35		3.6	
UART Interface Logic Supply Voltage	V _{EXT}		1.71		3.6	V
Logic Supply Voltage	V ₁₈		1.65		1.95	V
CURRENT CONSUMPTION						
V _{CC} Supply Current	I _{CC}	1.8MHz crystal oscillator active, PLL disabled, SPI/I ² C interface idle, UART interfaces idle, LDOEN = high			500	μA
		Baud rate = 1Mbps, 20MHz external clock, SPI/I ² C interface idle, PLL disabled, UART in loopback mode, LDOEN = low			500	
V _{CC} + V _L + V _A Shutdown Supply Current	I _{SHDN}	RST = low, MISO, SCLK, MOSI, SPI/I ² C, CS = low; LDOEN = low/high, CTS = low/high, RX = high.		0	1	μA
V ₁₈ Input Power-Supply Current in Shutdown Mode	I _{18SHDN}	Shutdown mode, LDOEN = low, RST = low, all inputs and outputs are idle			50	μA

MAX3108A

帶有128字FIFO的SPI/I²C UART, WLP封裝

DC ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 1.71V to 3.6V, V_L = 1.71V to 3.6V, V_{EXT} = 1.71V to 3.6V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{CC} = 2.8V, V_L = 1.8V, V_{EXT} = 2.5V, T_A = +25°C.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V ₁₈ Input Power-Supply Current	I ₁₈	Baud rate = 1Mbps, 20MHz external clock, PLL disabled, UART in loopback mode, LDOEN = low (Note 4)			2	mA
SCLK/SCL, MISO/SDA						
MISO/SDA Output Logic-Low Voltage in I ² C Mode	V _{OLI2C}	Sink current = 3mA, V _L > 2V			0.4	V
		Sink current = 3mA, V _L < 2V			0.2 x V _L	
MISO/SDA Output Logic-Low Voltage in SPI Mode	V _{OLSPI}	Sink current = 2mA			0.4	V
MISO/SDA Output Logic-High Voltage in SPI Mode	V _{OHSPI}	Source current = 2mA	V _L - 0.4			V
Input Logic-Low Voltage	V _{IL}	SPI and I ² C mode			0.3 x V _L	V
Input Logic-High Voltage	V _{IH}	SPI and I ² C mode	0.7 x V _L			V
Input Hysteresis	V _{HYST}	SPI and I ² C mode		0.05 x V _L		V
Input Leakage Current	I _{IL}	V _{IN} = 0 to V _L , SPI and I ² C mode	-1		+1	μA
Input Capacitance	C _{IN}	SPI and I ² C mode		5		pF
SPI/I²C, CS/A0, MOSI/A1 INPUTS						
Input Logic-Low Voltage	V _{IL}	SPI and I ² C mode			0.3 x V _L	V
Input Logic-High Voltage	V _{IH}	SPI and I ² C mode	0.7 x V _L			V
Input Hysteresis	V _{HYST}	SPI and I ² C mode		50		mV
Input Leakage Current	I _{IL}	V _{IN} = 0 to V _L , SPI and I ² C mode	-1		+1	μA
Input Capacitance	C _{IN}	SPI and I ² C mode		5		pF
IRQ OUTPUT (OPEN DRAIN)						
Output Logic-Low Voltage	V _{OL}	Sink current = 2mA			0.4	V
Output Leakage Current	I _{OL}	V _{IRQ} = 0 to V _L , $\overline{\text{IRQ}}$ is not asserted	-1		+1	μA
LDOEN AND RST INPUTS						
Input Logic-Low Voltage	V _{IL}				0.3 x V _L	V
Input Logic-High Voltage	V _{IH}		0.7 x V _L			V
Input Hysteresis	V _{HYST}			50		mV
Input Leakage Current	I _{IL}	V _{IN} = 0 to V _L	-1		+1	μA

MAX3108A

帶有128字FIFO的SPI/I2C UART, WLP封裝

DC ELECTRICAL CHARACTERISTICS (continued)

(VCC = 1.71V to 3.6V, VL = 1.71V to 3.6V, VEXT = 1.71V to 3.6V, TA = -40°C to +85°C, unless otherwise noted. Typical values are at VCC = 2.8V, VL = 1.8V, VEXT = 2.5V, TA = +25°C.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
UART INTERFACE						
RTS, TX OUTPUTS						
Output Logic-Low Voltage	VOL	Sink current = 2mA			0.4	V
Output Logic-High Voltage	VOH	Source current = 2mA	0.7 x VEXT			V
Input Leakage Current	IIL	Output is three-stated, VRTS = 0 to VEXT	-1		+1	μA
Input Capacitance	CIN	High-impedance mode		5		pF
CTS, RX INPUTS						
Input Logic-Low Voltage	VIL				0.3 x VEXT	V
Input Logic-High Voltage	VIH		0.7 x VEXT			V
Input Hysteresis	VHYST			50		mV
CTS Input Leakage Current	IIL	VCTS = 0 to VEXT	-1		+1	μA
RX Pullup Current	IPU	VRX = 0V, VEXT = 3.6V	-7.5	-5.5	-3.5	μA
Input Capacitance	CIN			5		pF
GPIO_ INPUTS/OUTPUTS						
Output Logic-Low Voltage	VOL	Sink current = 20mA, push-pull or open-drain output type, VEXT > 2.3V			0.45	V
		Sink current = 20mA, push-pull or open-drain output type, VEXT < 2.3V			0.55	
Output Logic-High Voltage	VOH	Source current = 5mA, push-pull output type	VEXT - 0.4V			V
Input Logic-Low Voltage	VIL	GPIO_ is configured as an input			0.4	V
Input Logic-High Voltage	VIH	GPIO_ is configured as an input	2/3 x VEXT			V
Pulldown Current	IPD	VGPIO_ = VEXT = 3.6V, GPIO_ is configured as an input	3.5	5.5	7.5	μA
XIN						
Input Logic-Low Voltage	VIL				0.6	V
Input Logic-High Voltage	VIH		1.2			V
Input Capacitance	CXIN			16		pF
XOUT						
Input Capacitance	CXOUT			16		pF

帶有128字FIFO的SPI/I²C UART, WLP封裝

AC ELECTRICAL CHARACTERISTICS

(V_{CC} = 1.71V to 3.6V, V_L = 1.71V to 3.6V, V_{EXT} = 1.71V to 3.6V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{CC} = 2.8V, V_L = 1.8V, V_{EXT} = 2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
External Crystal Frequency	f _{XOSC}		1		4	MHz
External Clock Frequency	f _{CLK}		0.5		35	MHz
External Clock Duty Cycle		(Note 5)	45		55	%
Baud-Rate Generator Clock Input Frequency	f _{REF}				96	MHz
I²C BUS: TIMING CHARACTERISTICS (Figure 1)						
SCL Clock Frequency	f _{SCL}	Standard mode			100	kHz
		Fast mode			400	
		Fast mode plus			1000	
Bus Free Time Between a STOP and START Condition	t _{BUF}	Standard mode	4.7			μs
		Fast mode	1.3			
		Fast mode plus	0.5			
Hold Time for START Condition and Repeated START Condition	t _{HD:STA}	Standard mode	4.0			μs
		Fast mode	0.6			
		Fast mode plus	0.26			
Low Period of the SCL Clock	t _{LOW}	Standard mode	4.7			μs
		Fast mode	1.3			
		Fast mode plus	0.5			
High Period of the SCL Clock	t _{HIGH}	Standard mode	4.0			μs
		Fast mode	0.6			
		Fast mode plus	0.26			
Data Hold Time	t _{HD:DAT}	Standard mode	0		0.9	μs
		Fast mode	0		0.9	
		Fast mode plus	0			
Data Setup Time	t _{SU:DAT}	Standard mode	250			ns
		Fast mode	100			
		Fast mode plus	50			
Setup Time for Repeated START Condition	t _{SU:STA}	Standard mode	4.7			μs
		Fast mode	0.2			
		Fast mode plus	0.26			
Rise Time of Incoming SDA and SCL Signals	t _R	Standard mode (0.3 × V _L to 0.7 × V _L) (Note 6)	20 + 0.1C _B		1000	ns
		Fast mode (0.3 × V _L to 0.7 × V _L) (Note 6)	20 + 0.1C _B		300	
		Fast mode plus			120	
Fall Time of SDA and SCL Signals	t _F	Standard mode (0.3 × V _L to 0.7 × V _L) (Note 6)	20 + 0.1C _B		1000	ns
		Fast mode (0.3 × V _L to 0.7 × V _L) (Note 6)	20 + 0.1C _B		300	
		Fast mode plus			120	

MAX3108A

帶有128字FIFO的SPI/I²C UART, WLP封裝

AC ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 1.71V to 3.6V, V_L = 1.71V to 3.6V, V_{EXT} = 1.71V to 3.6V, T_A = -40°C to +85°C, unless otherwise noted. Typical values are at V_{CC} = 2.8V, V_L = 1.8V, V_{EXT} = 2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Setup Time for STOP Condition	t _{SU:STO}	Standard mode	4.7			μs
		Fast mode	0.6			
		Fast mode plus	0.26			
Capacitive Load for SDA and SCL	C _B	Standard mode (Note 5)			400	pF
		Fast mode (Note 5)			400	
		Fast mode plus (Note 5)			550	
SCL and SDA I/O Capacitance	C _{I/O}	(Note 5)			10	pF
Pulse Width of Spike Suppressed	t _{SP}				50	ns
SPI BUS: TIMING CHARACTERISTICS (Figure 2)						
SCLK Clock Period	t _{CH+tCL}		38.4			ns
SCLK Pulse Width High	t _{CH}		16			ns
SCLK Pulse Width Low	t _{CL}		16			ns
$\overline{\text{CS}}$ Fall to SCLK Rise Time	t _{CSS}		0			ns
MOSI Hold Time	t _{DH}		3			ns
MOSI Setup Time	t _{DS}		5			ns
Output Data Propagation Delay	t _{DO}				20	ns
MISO Rise and Fall Times	t _{FT}				10	ns
$\overline{\text{CS}}$ Hold Time	t _{CSH}		30			ns

Note 2: All units are production tested at T_A = +25°C. Specifications over temperature are guaranteed by design.

Note 3: Currents entering the IC are positive and currents exiting the IC are negative.

Note 4: When V₁₈ is powered by an external voltage supply, it must have current capability above or equal to I₁₈.

Note 5: Guaranteed by design; not production tested.

Note 6: C_B is the total capacitance of either the clock or data line of the synchronous bus in pF.

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

时序图

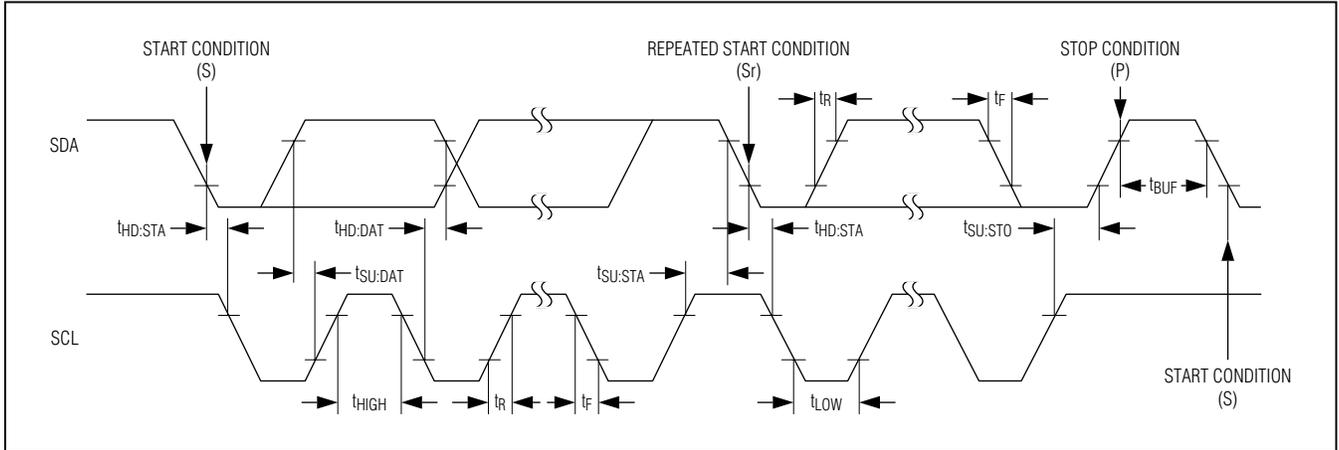


图1. I2C时序图

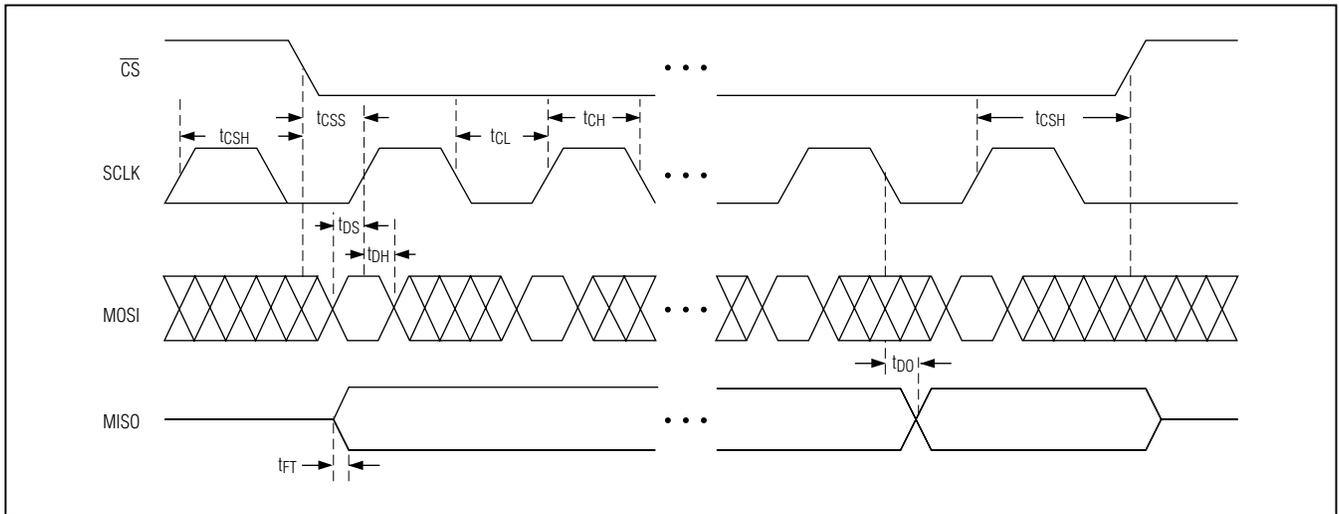


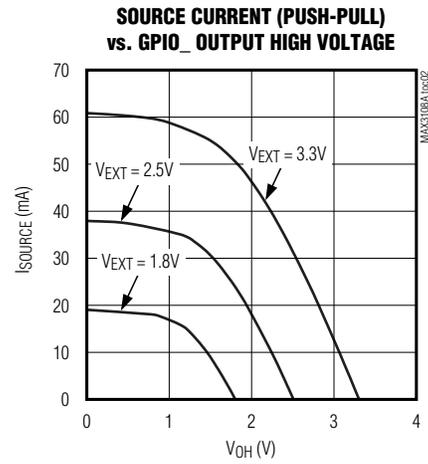
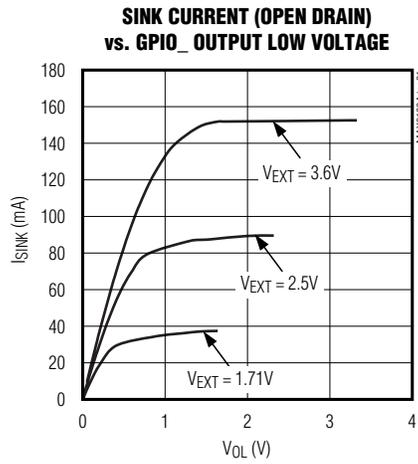
图2. SPI时序图

MAX3108A

帶有128字FIFO的SPI/I2C UART, WLP封裝

典型工作特性

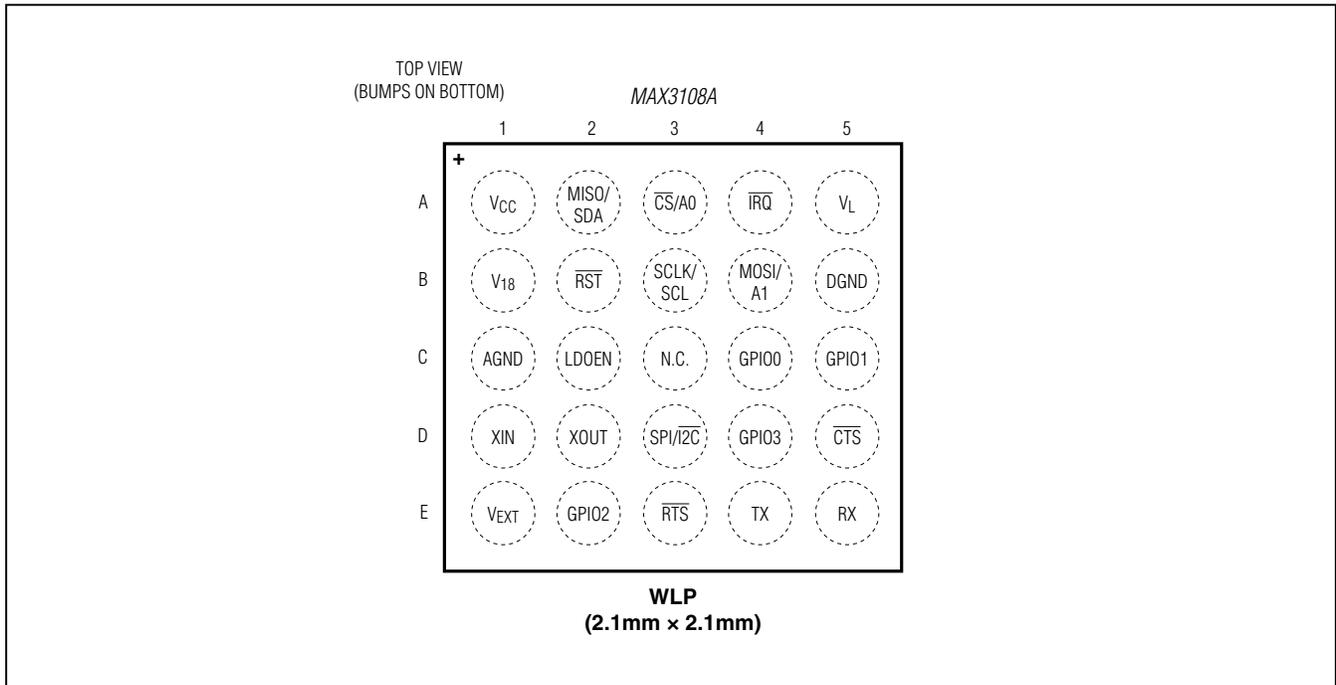
($V_{CC} = 2.5V$, $V_L = 2.5V$, $V_{EXT} = 2.5V$, $V_{LDOEN} = V_L$, $T_A = +25^\circ C$, unless otherwise noted.)



MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

焊球配置



焊球说明

焊球	名称	功能
A1	V _{CC}	模拟电源。V _{CC} 为PLL和内部LDO供电。利用0.1μF陶瓷电容将V _{CC} 旁路至AGND。
A2	MISO/SDA	串行数据输出。SPI/I ² C为高电平时，MISO/SDA作为SPI主机输入、从机输出(MISO)；SPI/I ² C为低电平时，MISO/SDA作为SDA，即I ² C串行数据输入/输出；将RST驱动为低电平，或者关断外部1.8V电源时，MISO/SDA为高阻态。
A3	CS/A0	低电平有效片选和地址0输入。SPI/I ² C为高电平时，CS/A0作为CS，即SPI低电平有效片选；SPI/I ² C为低电平时，CS/A0作为A0，即I ² C器件地址编程输入。SPI/I ² C为低电平时，将CS/A0连接至DGND、V _L 、SCL或SDA。
A4	IRQ	低电平有效中断开漏输出。发生中断时，IRQ变为有效；将RST驱动为低电平时，IRQ为高阻态。
A5	V _L	数字接口电源。V _L 为RST、IRQ、MOSI/A1、CS/A0、SCLK/SCL、MISO/SDA、LDOEN及SPI/I ² C的内部逻辑电平转换器供电。利用0.1μF陶瓷电容将V _L 旁路至DGND。
B1	V ₁₈	内部1.8V LDO输出和1.8V电源输入。利用1μF陶瓷电容将V ₁₈ 旁路至DGND。
B2	RST	低电平有效复位输入。将RST驱动为低电平时，强制UART进入硬件复位模式，也能使低功耗关断模式。RST为低电平时，内部1.8V LDO关断，即使LDOEN输入保持为高电平。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

焊球说明(续)

焊球	名称	功能
B3	SCLK/SCL	串行时钟输入。SPI/I ² C为高电平时, SCLK/SCL作为SCLK, 即SPI串行时钟输入(高达26MHz); SPI/I ² C为低电平时, SCLK/SCL作为SCL, 即I ² C串行时钟输入(快速脉冲模式下高达1MHz)。
B4	MOSI/A1	串行数据输入和地址1输入。SPI/I ² C为高电平时, MOSI/A1作为SPI主机输出、从机输入(MOSI); SPI/I ² C为低电平时, MOSI/A1作为A1, 即I ² C器件地址编程输入。SPI/I ² C为低电平时, 将MOSI/A1连接至DGND、V _L 、SCL或SDA。
B5	DGND	数字地。
C1	AGND	模拟地。
C2	LDOEN	LDO使能输入。将LDOEN驱动为高电平时, 使能内部1.8V LDO。将LDOEN驱动为低电平时, 禁止内部LDO。LDOEN为低电平时, 利用外部电压源为V ₁₈ 供电。
C3	N.C.	空脚。内部无连接。
C4	GPIO0	通用输入/输出0。GPIO0可由用户设置为输入或输出(推挽式或开漏式), 或者作为外部事件驱动中断源。GPIO0配置为输入时, 由弱下拉电阻拉至DGND。
C5	GPIO1	通用输入/输出1。GPIO1可由用户设置为输入或输出(推挽式或开漏式), 或者作为外部事件驱动中断源。GPIO1配置为输入时, 由弱下拉电阻拉至DGND。
D1	XIN	晶振/时钟输入。使用外部晶振时, 将晶振的一端连接至XIN, 另一端连接至XOUT;使用外部时钟源时, 利用单端外部时钟驱动XIN。
D2	XOUT	晶振输出。使用外部晶振时, 将晶振的一端连接至XOUT, 另一端连接至XIN; 使用外部时钟源时, 使XOUT浮空。
D3	SPI/I ² C	SPI选择器输入或低电平有效I ² C。将SPI/I ² C驱动为低电平时, 使能I ² C; 将SPI/I ² C驱动为高电平时, 使能SPI。
D4	GPIO3	通用输入/输出3。GPIO3可由用户设置为输入或输出(推挽式或开漏式), 或者作为外部事件驱动中断源。GPIO3配置为输入时, 由弱下拉电阻拉至DGND。
D5	CTS	低电平有效允许发送输入。CTS为流控制状态输入。
E1	VEXT	收发器接口电源。VEXT为RX、TX、RTS、CTS和GPIO_的内部逻辑电平转换器供电。利用0.1μF陶瓷电容将VEXT旁路至DGND。
E2	GPIO2	通用输入/输出2。GPIO2可由用户设置为输入或输出(推挽式或开漏式), 或者作为外部事件驱动中断源。GPIO2配置为输入时, 由弱下拉电阻拉至DGND。
E3	RTS	低电平有效请求发送输出。RTS可由LCR寄存器设置为高电平或低电平。如果外部供电1.8V未上电, RTS为低电平时, RTS为逻辑高。
E4	TX	串行发送数据输出。RST为低电平或外部供电1.8V未上电时, TX为逻辑高。
E5	RX	串口接收数据输入。RX由内部弱上拉电阻拉至VEXT。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

详细说明

MAX3108A通用异步收发器(UART)将SPI/MICROWIRE™或I²C微处理器总线桥接至异步串行数据通信链路。器件包括一个高级UART、分数波特率发生器和四路GPIO。八位寄存器配置和监测器件,并可通过外部引脚选择的SPI或I²C访问。寄存器按相关功能布置,如寄存器映射部分所示。

主机控制器通过SPI或I²C接口将数据装载至发送保持寄存器(THR),自动将数据推入发送先进先出(FIFO)存储器、格式化,并通过TX送出。以所选波特率发送数据之前,器件为数据添加START、STOP以及奇偶检验位。时钟配置寄存器决定波特率、时钟源和时钟频率预分频。

RX由高电平跳变为低电平时,MAX3108A接收器检测到START位。内部时钟以波特率的16倍对数据进行采样。接收到的数据自动置于接收FIFO,可由主机微控制器通过接收保持寄存器(RHR)进行读取。

MAX3108A的寄存器组与MAX3107兼容。

寄存器组

器件采用平行的寄存器结构,无需寄存器影射,寄存器宽8位。寄存器与16C550的寄存器有些类似。

接收和发送FIFO

UART的接收器和发送器各有128字深的FIFO,减少主机处理器所需的间隔数量,专用于与器件之间的高速、大量数据收发。随着异步Rx/Tx接口数据率增加并接近主控制器的SPI/I²C数据率,UART管理和流控制成为主要的主机活动。通过增加FIFO大小,减少中断主机的次数,可使用数据块与FIFO进行数据传输。

FIFO触发点可向主控制器产生中断,表示已经达到设置的FIFO空满状态。可通过FIFOTrgLvl寄存器设置发送器和接收器的触发点,分辨率为8个FIFO单元。接收器FIFO触发通知主机:接收FIFO中有指定数量的字等待读取,或者有已知数量的空FIFO单元可供使用并做好填充准备。当发送FIFO的空满状态高于设置的触发点时,发送FIFO触发产生中断,主机即可减缓通过THR向发送FIFO写数据。

主机可通过TxFIFOLvl和RxFIFOLvl寄存器读取每个FIFO中的字数。

MODE2[1]: FIFORst位置高时,清除TxFIFO和RxFIFO的内容。

发送器工作

图3所示为带TxFIFO的发送器结构。通过写THR寄存器,发送FIFO可储存多达128字的数据。

主控制器可通过TxFIFOLvl寄存器手动读出TxFIFO中数据字的当前数量。发送FIFO空满状态可通过FIFOTrgLvl寄存器设置,当TxFIFO中的字数超过设置值时产生中断。TxFIFO中断触发点由FIFOTrgLvl[3:0]位选择。发送FIFO空满状态增大至超过设置的触发点时,在ISR[4]: TxTrgInt中产生中断。

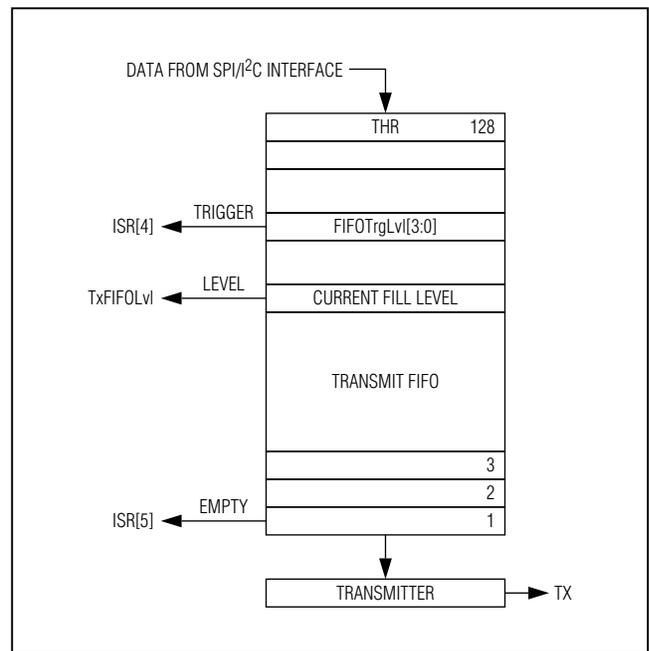


图3. 发送FIFO信号

MICROWIRE是National Semiconductor Corp.的商标。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

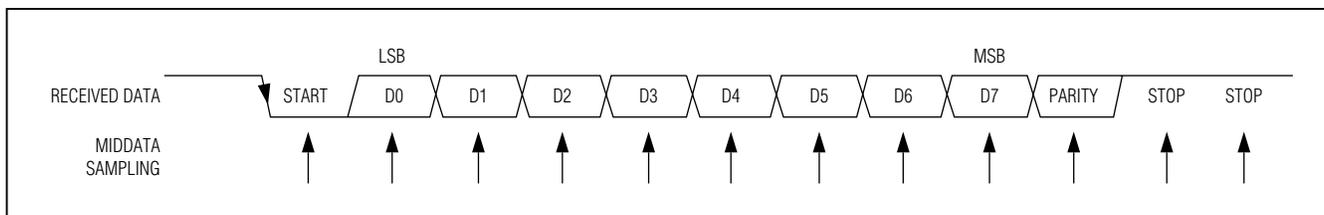


图4. 接收数据格式

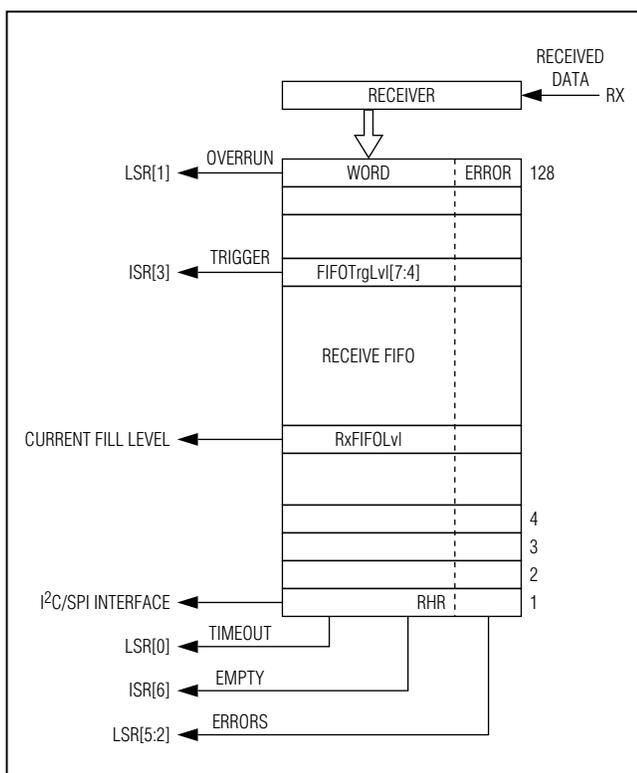


图5. 接收FIFO

发送FIFO为空时，在ISR[5]: TxFifoEmptyInt中产生中断。当发送器开始发送Tx FIFO的最后一个字时，ISR[5]变为逻辑高。当发送器完成最后一个字的发送后，在STSInt[7]: TxEmptyInt产生附加中断。

为停止发送，将MODE1[1]: TxDisabl位置高。TxDisabl置位后，发送器完成当前字符的发送，然后结束发送。使能自动软件流控制和AutoRTS流控制之前，将发送器关闭。

TX输出逻辑可通过IrDA[5]: TxInv位反相。除非另外说明，本数据资料中的全部发送器逻辑说明均假设TxInv为逻辑低。

接收器工作

接收器RX的数据格式如图4所示。静态逻辑状态为高电平，第一位(START位)为逻辑低电平。接收8位数据字，LSB在前。接收器在每一位的中间时刻对数据进行采样(图4)。将接收字及其相关错误置于接收FIFO，存储每个接收字的误码和状态信息(图5)。主机通过读取RHR读出接收FIFO的数据，最早进入的数据排在前面。从RHR最新读取的字节状态和误码信息位于线路状态寄存器(LSR)中。从RHR中读出字后，LSR包含该字的状态信息。

带有128字FIFO的SPI/I2C UART, WLP封装

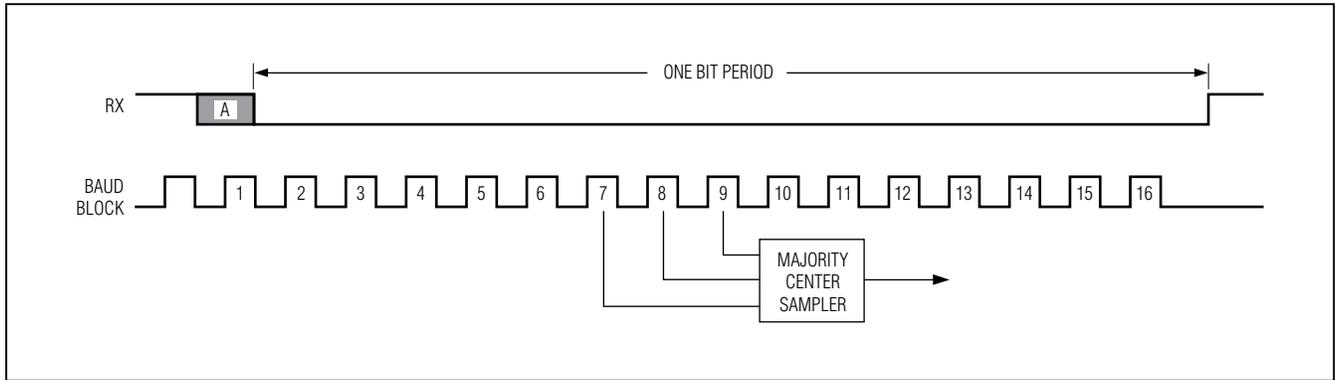


图6. 每位中间采样

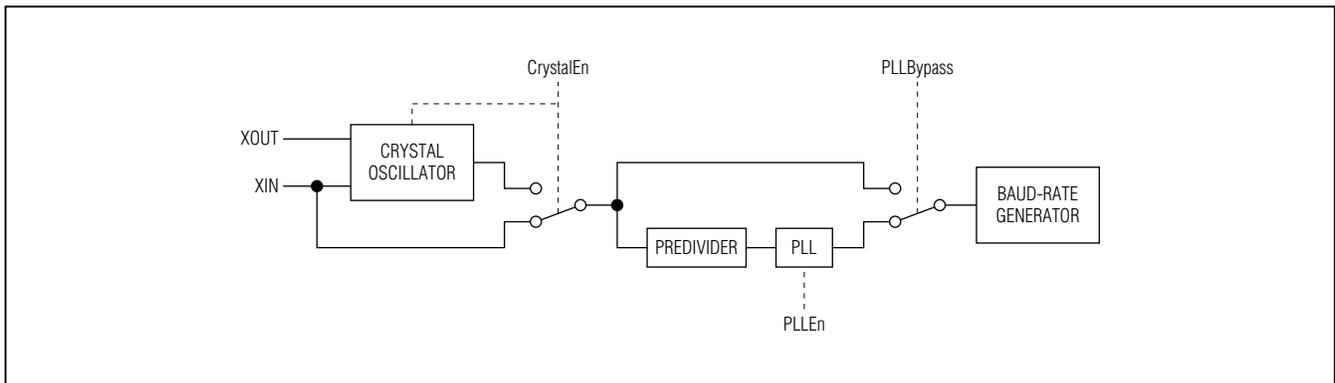


图7. 时钟选择框图

检查每个接收字的以下三种错误条件：奇偶校验错误、帧错误和线上噪声。根据寄存器设置计算接收字的奇、偶校验值，检测奇偶检验错误。当接收到的数据帧与预期帧格式在长度上不匹配时，检测到帧错误。通过检查每一位的三次采样结果的逻辑一致性，检测线噪声(图6)。

通过将MODE1[0]: RxDisabl位置高，可关闭接收器。该位置高后，器件在当前字之后立即关闭接收器，不再接收其它数据。

可通过IrDA[4]: RxInv置高将RX输入逻辑反相。除非另外说明，本数据资料中的全部接收器逻辑说明均假设RxInv为逻辑低。

线路噪声指示

工作在标准或两倍(即非四倍)速率模式时，器件检查每一接收位的三次采样的二进制逻辑电平是否相同。如果每一接收位的三次采样中有任何一次的逻辑电平不同，则说明传输线上的噪声已经影响了接收的数据，可认为噪声较大。每一接收字节的噪声指示体现在LSRI[5]: RxNoise位。奇偶检验错误是另一种噪声指示，但不敏感。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

时钟选择

器件可采用外部晶振或外部时钟源工作。图7所示为时钟选择电路的简化框图。器件采用晶振工作时，STSIInt[5]: ClkReady位表示晶振是否达到稳态以及波特率发生器是否准备好稳定工作。

通过将CLKSource[7]: CLKtoRTS位置高，可将波特率时钟连接至RTS输出。标准工作模式下，时钟速率为波特率的16倍；两倍速率模式下为波特率的8倍；四倍速率模式下为波特率的4倍。如果使用波特率发生器的分数部分，时钟将不稳定并发生抖动。

晶振

器件带有晶振，具有高波特率精度和低功耗。将CLKSource[1]: CrystalEn位置高时，使能选择晶振。片上晶振在XIN和XOUT引脚内集成16pF负载电容。只在XIN和XOUT之间连接外部晶体或陶瓷振荡器。

外部时钟源

不使用晶振时，将外部单端时钟源连接至XIN，使XOUT保持浮空。将CLKSource[1]: CrystalEn位置低时，选择外部时钟。

PLL和预分频器

内部预分频器和PLL允许兼容宽范围外部时钟频率和波特率。利用PLLConfig[7:6]位，可将PLL配置为输入时钟速率乘以6、48、96或144。预分频器位于输入时钟和PLL之间，对输入时钟进行分频，整数分频比为1至63，由PLLConfig[5:0]位设置。更多信息请参见PLLConfig寄存器说明，使用PLL要求V_{CC}高于2.35V。

分数波特率发生器

内部分数波特率发生器提供更高的波特率编程灵活性和分辨率。波特率发生器具有16位整数除数和4位字分数除数。分数波特率发生器支持晶振和外部时钟源。

整数和分数除数由除数D计算：

$$D = \frac{f_{REF} \times RateMode}{16 \times BaudRate}$$

式中，f_{REF}为输入至波特率发生器的基准频率，RateMode为速率模式倍数(默认为1x)，BaudRate为预期波特率，D为理想除数。f_{REF}必须小于96MHz。RateMode在1x速率模式下为1，在两倍速率模式下为2，在四倍速率模式下为4。整数除数部分DIV通过截断D得到：

$$DIV(\text{decimal}) = \text{TRUNC}(D)$$

DIV最大可为16位(65,535)宽，写至两个单字节宽寄存器DIVMSB和DIVLSB。DIVLSB的最小允许值为1。

除数的分数部分FRACT为4位半字节，可写至BRGConfig[3:0]，最大值为15，允许以分辨率0.0625设置除数。FRACT的计算公式为：FRACT = ROUND(16 × (D - DIV))。

以下为计算除数的例子。本例中，要求波特率为190kbaud，基准输入频率为28.23MHz，速率模式为1x(默认值)。

理想除数计算如下：

$$D = 28,230,000 / (16 \times 190,000) = 9.286$$

因此，DIV = 9。

$$FRACT = \text{ROUND}(16 \times 0.286) = 5$$

所以，DIVMSB = 0x00，DIVLSB = 0x09，BRGConfig[3:0] = 0x05。

产生的实际波特率可计算如下：

$$BR_{ACTUAL} = \frac{f_{REF} \times RateMode}{16 \times D_{ACTUAL}}$$

本例中：D_{ACTUAL} = 9 + 5/16 = 9.3125，RateMode = 1，以及BR_{ACTUAL} = 28,230,000 / (16 × 9.3125) = 189,463 baud。

因此，因此实际波特率在理想速率的0.28%范围之内。

带有128字FIFO的SPI/I2C UART, WLP封装

2倍/4倍速模式

器件提供两倍和四倍速率模式，以支持高于采用16倍采样的标准工作模式的波特率。在这些模式下，基准时钟速率只需分别比波特率高8倍或4倍。在四倍速率模式下，仅在每位的中间对其采样一次，而非通常采样三次，从而确定接收位的逻辑值，这样在四倍速率模式下就减少了在接收数据上测试线路噪声的功能。通过BRGConfig[5:4]选择两倍和四倍速率模式。注意，IrDA编码和解码不适用于两倍和四倍速率模式。

选择两倍速率模式时，实际波特率为波特率发生器中所设置速率的两倍。如果使能四倍速率模式，线路上的实际波特率为所设置波特率的四倍(图8)。

多点模式

多点模式(也称为9位模式)下，数据字长度为8位，第9位用于区分地址字和数据字。多点模式由MODE2[6]: MultiDrop位使能。MultiDrop位占用数据字结构中的奇偶检验位的位置。多点模式下，接收到地址(第9位为1)时，禁止奇偶校验并在SpclCharInt[5]: MultiDropInt中产生中断。

由主处理器负责筛选出用于其地址的数据，也可使用自动数据筛选功能自动滤除非用于规定9位模式地址的数据。

多点模式下的自动数据筛选

多点模式下，器件可配置为自动滤除非用于其地址的数据。地址由用户通过设置寄存器值或寄存器值与GPIO硬件输入的组合进行定义，使用整个XOFF2寄存器或XOFF2[7:4]位与GPIO_输入组合来定义地址。

将MODE2[6]: MultiDrop位置高时，使能多点模式；将MODE2[4]: SpecialChr位置高时，使能自动数据筛选功能。利用寄存器与GPIO_输入组合定义地址时，地址的MSB写至XOFF2[7:4]位，地址的LSB由GPIO定义。为使能该地址定义方法以及自动数据筛选，除置位MODE2[4]: SpecialChr和MODE2[6]: MultiDrop位外，将FlowCtrl[2]: GPIAddr位置高。将FlowCtrl[2]: GPIAddr位置高时，自动读取GPIO_输入，并在任意GPIO引脚的逻辑发生变化时自动更新地址。

使用自动数据筛选功能时，器件将接收的地址与设置的地址进行比较。当接收的地址与站点的地址匹配时，将接收的数据储存在RxFIFO中；当接收的地址与站点的地址不匹配时，则废弃接收到的数据。接收到地址时，不将其储存至FIFO，但仍然在SpclCharInt[5]: MultiDropInt中产生中断。当接收到站点地址时，在SpclCharInt[3]: XOFF2Int中产生附加中断。

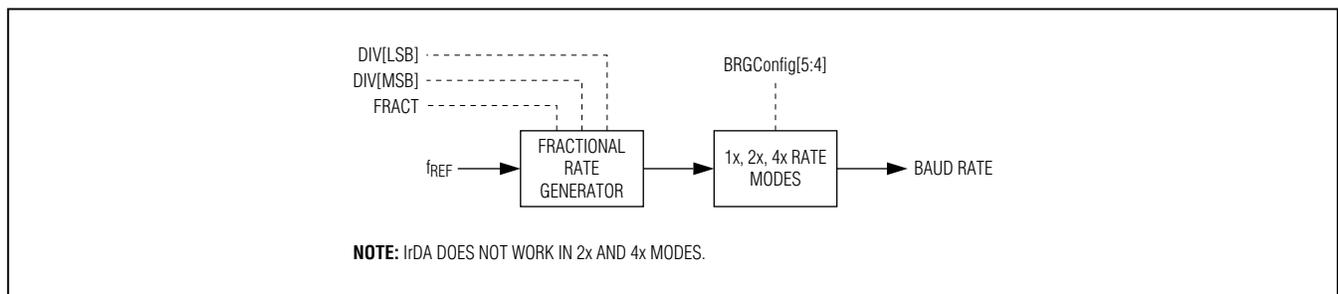


图8. 两倍和四倍波特率

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

自动收发器方向控制

在有些半双工通信系统中，接收数据时必须关闭收发器的发送器，以免增加总线负载。半双工RS-485通信就是这种情况。类似地，全双工多点通信中，例如RS-485或RS-422 V.11，在任意时刻只能使能一个发送器，其它发送器必须禁止。器件可配置为通过控制DE和RE引脚在硬件级自动使能/禁止收发器的发送器和/或接收器，该功能使主处理器不负责该时间关键的任务。

RTS输出用于控制收发器的发射使能输入，并在器件的发送器开始发送时自动设置为高电平，发送FIFO中出现数据

时立即发生这种情况。自动收发器方向控制由MODE1[4]: TrnscvCtrl位使能。图9所示为RS-485应用中使用自动收发器方向控制功能的典型连接。

通过设置称为建立时间的周期，可在Tx发送之前将RTS输出设置为高电平(图10)。建立时间由HDpIxDelay[7:4]: Setupx位设置。类似地，发送器完成发送后，RTS可在设置的时间周期内保持为高电平，该时间周期称为保持时间。保持时间由HDpIxDelay[3:0]: Holdx位设置。

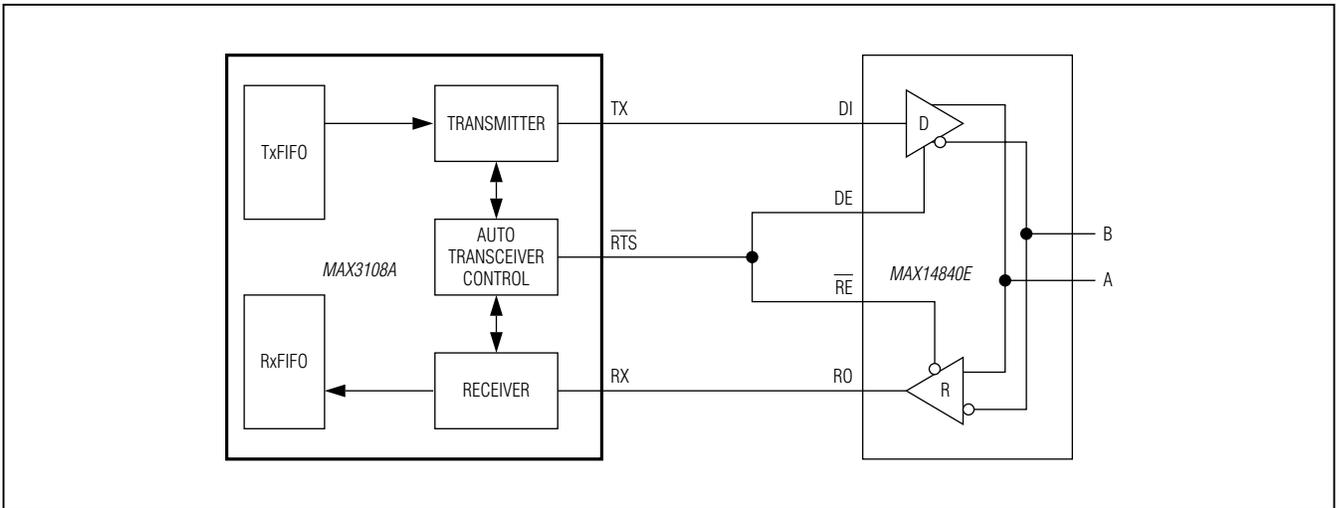


图9. 自动收发器方向控制

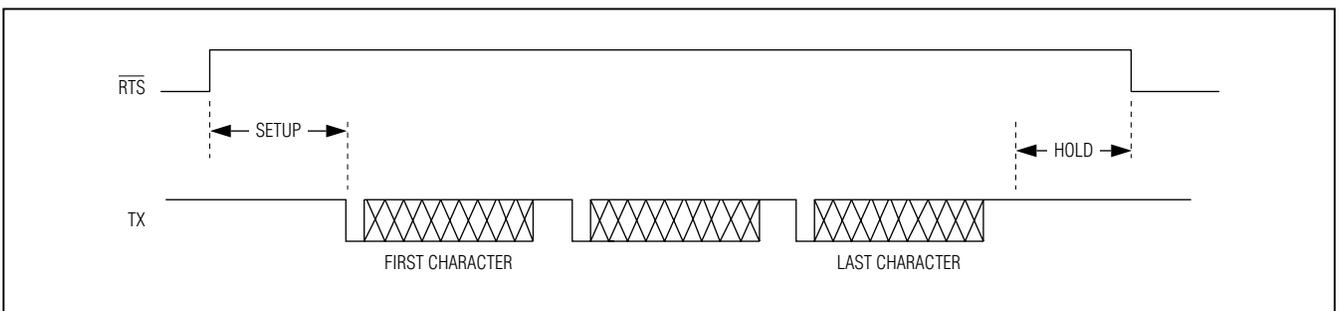


图10. 自动收发器方向控制中的建立和保持时间

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

回波抑制

器件可抑制回波数据，回波常见于半双工通信，例如RS-485和IrDA。收发器正在发送时，如果收发器的接收器未关闭，UART就会接收到发送的数据(回波)。通过使能回波抑制，器件的接收器可防止接收回波数据。图11所示为使用回波抑制功能的典型RS-485应用。将MODE2[7]: EchoSuprs位置位时，使能回波抑制。

器件在发送时，器件也可利用 $\overline{\text{RTS}}$ 输出禁止收发器的接收器，阻止长线延迟的回波。发送器可配置为在发送结束后的可编程时间周期(保持时间延迟)内保持使能(图12)。保持时间延迟由HDpplxDelay[3:0]位设置。更多信息请参见寄存器详细说明部分的HDpplxDelay介绍。

回波抑制功能与自动收发器方向控制功能可同时工作。

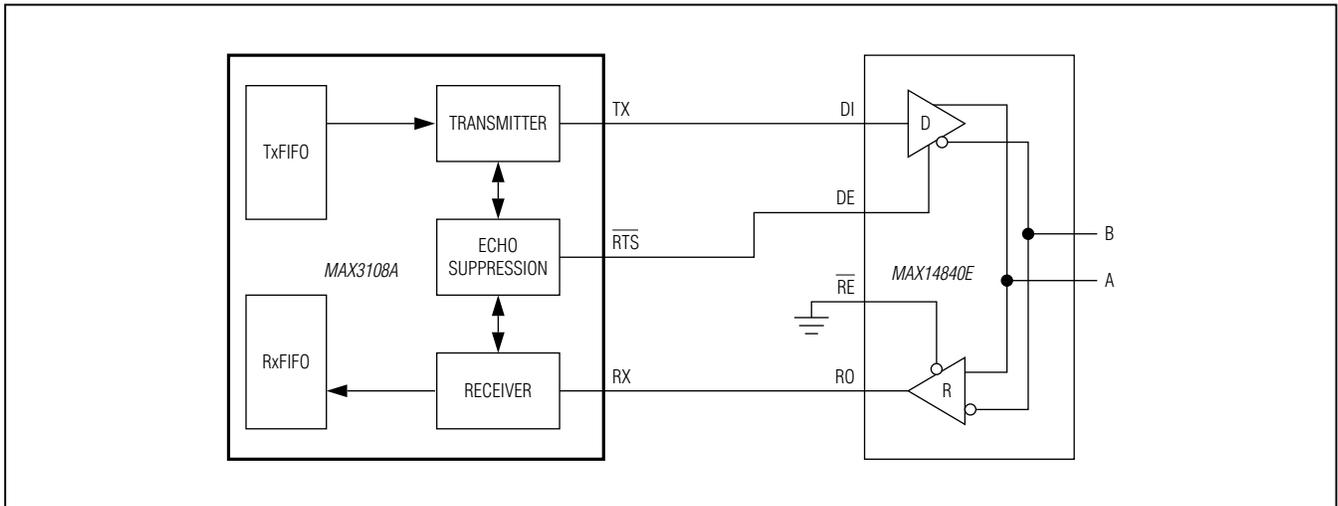


图11. 带回波抑制的半双工

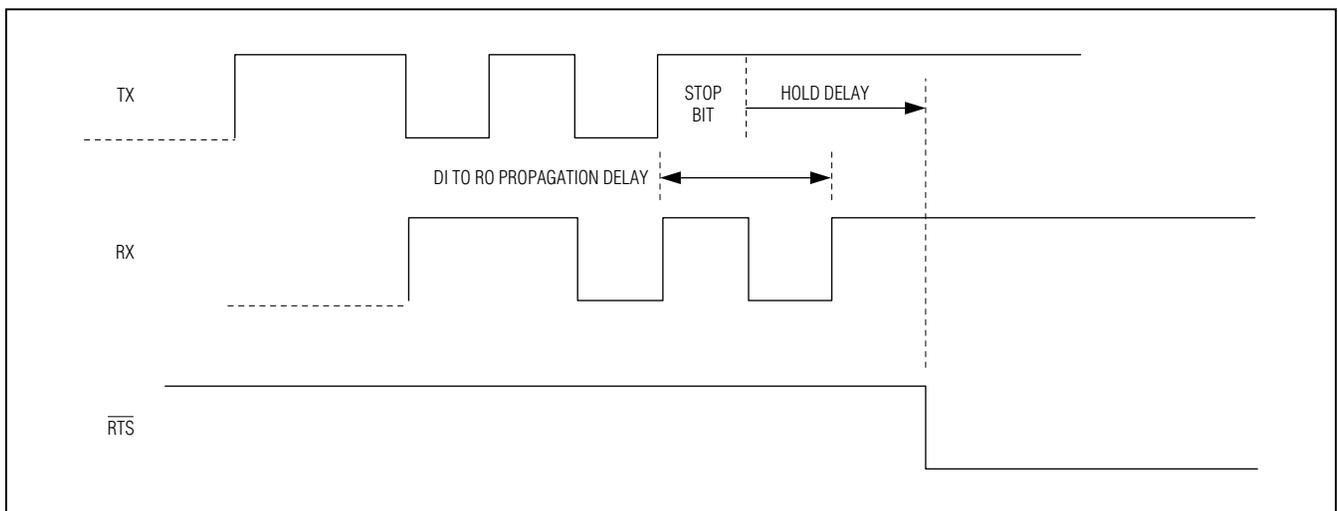


图12. 回波抑制时序

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

自动控制硬件流

器件能够自动控制硬件($\overline{\text{RTS}}$ 和 $\overline{\text{CTS}}$)流, 无需主处理器介入。使能AutoRTS控制时, 器件自动控制 $\overline{\text{RTS}}$ 握手, 无需主处理器介入。AutoCTS流控制根据 $\overline{\text{CTS}}$ 输入独立将器件的发送器打开和关闭。AutoRTS和AutoCTS流控制模式由FlowCtrl[1:0]位独立使能。

AutoRTS控制

AutoRTS流控制通过通知远端UART停止发送数据, 确保接收FIFO不会溢出; 器件通过控制 $\overline{\text{RTS}}$ 输出自动实现该操作。通过将FlowCtrl[0]: AutoRTS位置高使能AutoRTS流控制。HALT和RESUME可编程值决定 $\overline{\text{RTS}}$ 变为有效及无效时的RxFIFO空满状态门限。在FlowLvl寄存器中设置HALT和RESUME空满状态。对于不同的HALT和RESUME空满状态, 可定义RTS跳变的RxFIFO空满状态滞回。

RxFIFO的空满状态高于HALT空满状态时, 器件使 $\overline{\text{RTS}}$ 变为无效, 使远端UART停止进一步发送任何数据。 $\overline{\text{RTS}}$ 保持无效状态, 直到RxFIFO空满状态足以比RESUME空满状态低一定数量的数据字。

达到HALT和RESUME空满状态时, 不产生中断, 这允许主控制器完全不负责 $\overline{\text{RTS}}$ 流控制的管理。

AutoCTS控制

使能AutoCTS流控制时, UART在 $\overline{\text{CTS}}$ 输入为逻辑低时自动开始发送数据, 在 $\overline{\text{CTS}}$ 为逻辑高时停止发送数据, 这使主处理器不负责管理该时间关键的流控制任务。通过将FlowCtrl[1]: AutoCTS位置高使能AutoCTS流控制。AutoCTS流控制期间, ISR[7]: CTSInt中断正常工作。将IRQEn[7]: CTSIntEn位置低时, 禁止将 $\overline{\text{CTS}}$ 中断连接至 $\overline{\text{IRQ}}$, 确保主机不接收 $\overline{\text{CTS}}$ 跳变中断。如果 $\overline{\text{CTS}}$ 在数据字传输期间从低电平跳变为高电平, 器件则完成当前字的发送后停止发送。

使能AutoCTS控制之前, 通过将MODE1[1]: TxDisabl位置高关闭发送器。

自动软件(XON/XOFF)流控制

使能自动软件流控制时, 器件检测和/或发送预定义的XON/XOFF字符, 以控制异步串行链路上的数据流。XON字符表示接收FIFO中具有足够空间, 应继续发送数据; XOFF

字符表示接收FIFO接近溢出, 应停止发送数据。自动软件流控制自主工作, 不需要主机介入, 与自动硬件流控制类似。为降低接收等于单字节XON或XOFF字符的错误数据的几率, 器件允许双宽度(16位)XON/XOFF字符。XON和XOFF字符设置在XON1、XON2和XOFF1、XOFF2寄存器中。

FlowCtrl[7:3]位用于使能和配置自动软件流控制。接收到XON或XOFF字符后, 在ISR[1]: SpCharInt中产生中断, 并将详细信息储存在SpCharInt寄存器中。将IRQEn[1]: SpCharInt位置低时, 禁止将中断连接至 $\overline{\text{IRQ}}$ 。

软件流控制由发送流控制和接收流控制组成, 两者彼此独立工作。

接收器流控制

通过FlowCtrl[7:6]位使能自动接收流控制时, 器件通过发送XOFF和XON控制字符自动控制远端UART的数据发送。HALT和RESUME空满状态决定发送XOFF和XON字符时的RxFIFO空满状态门限。HALT和RESUME在FlowLvl寄存器中设置。对于不同的HALT和RESUME空满状态, 可定义接收器流控制活动的RxFIFO空满状态滞回。

当RxFIFO的空满状态高于HALT空满状态时, 器件发送一个XOFF字符, 以停止数据发送。RxFIFO空满状态足以比RESUME空满状态低一定数量的数据字时, 发送一个XON字符。

如果通过将FlowCtrl[7:6]位设置为11选择双宽度(16字节)XON/XOFF字符, 每次发送控制字符时, 首先发送XON1/XOFF1, 然后再发送XON2/XOFF2。

发送器流控制

如果通过FlowCtrl[5:4]位使能发送控制, 接收器将全部接收到的数据字与XOFF和XON字符进行比较。接收到XOFF字符时, 器件使发送器在发送完当前数据字后停止发送数据。接收器不受影响, 继续接收。接收到XON字符时, 发送器重新开发发送数据。将接收到的XON和XOFF字符滤除, 不储存至接收FIFO。不产生中断。

带有128字FIFO的SPI/I²C UART, WLP封装

如果通过将FlowCtrl[5:4]位设置为11选择双宽度(16位)XON/XOFF字符,则必须先接收到与XON1/XOFF1匹配的字符然后又接收到与XON2/XOFF2匹配的字符,才视为一个控制字符。

使能软件发送器流控制之前,通过将MODE1[1]: TxDisabl位置高关闭发送器。

FIFO中断触发

如果定义了FIFO触发空满状态,将产生与接收和发送FIFO空满状态相关的中断。FIFO中数据字的数量达到或超过FIFOTrgLvl寄存器中设置的触发点时,在ISR[3]或ISR[4]中产生一个中断。AutoRTS或自动软件流控制模式下,中断触发点与HALT及RESUME流控制空满状态相互独立工作。

例如, FIFO中断触发可用于块数据传输。触发点中断为主机提供了在接收FIFO中有给定大小的数据块可供读取或发送FIFO中有数据可供传输的指示。如果HALT和RESUME空满状态超出该范围,在块读/写操作期间, UART继续发送或接收数据,不中断总线上的数据传输。

低功耗待机模式

器件具有休眠和关断模式,降低待机期间的功耗。在休眠和关断模式下, UART禁止特定的功能模块,以降低功耗。

退出休眠或关断模式时,内部时钟启动,时钟需要一定的时间才可达到稳定。STSInt[5]: ClkReady位表示时钟是否达到稳定。使用外部时钟源时, ClkReady位不表示时钟稳定性。

强制休眠模式

强制休眠模式下,与UART相关的片上时钟全部停止。以下电路不活动:晶振、PLL、预分频器、接收器和发送器。I²C/SPI接口和寄存器保持活动,主控制器可对其进行操作。

为强制器件进入休眠模式,将MODE1[5]: ForcedSleep位置高;为退出强制休眠模式,将ForcedSleep位置低。

自动休眠模式

通过将MODE1[6]: AutoSleep位置高,可将器件配置为工作在自动休眠模式。自动休眠模式下,当满足以下条件时,器件自动进入休眠模式:

- 两个FIFO均为空。
- 无等待处理的 \overline{IRQ} 中断。
- 任何输入引脚上均无活动的时间周期等于65,536个UART字符长度。

UART处于自动休眠模式时,不活动的电路与强制休眠模式下相同。

当任意GPIO₊、RX或 \overline{CTS} 输入上检测到活动时,器件立即退出自动休眠模式。

为手动退出自动休眠模式,将MODE1[6]: AutoSleep位置低。

关断模式

将 \overline{RST} 输入驱动为逻辑低时,进入关断模式。关断模式下的耗流小于1 μ A。关断模式下,器件的全部电路完全关闭,包括I²C/SPI接口、寄存器、FIFO和时钟电路。

\overline{RST} 输入从低电平跳变为高电平时,器件退出关断模式,触发硬件复位。当控制器能够从器件读取抑制寄存器内容时,芯片完成初始化,例如,寄存器可以是DIVLSB寄存器。

关断后,器件需要重新编程。

上电和IRQ

\overline{IRQ} 输出只有在全部电源有效时才工作。 \overline{IRQ} 作为低电平有效硬中断输出;有中断未处理时, \overline{IRQ} 变为有效。IRQEn寄存器中至少有一个中断使能位置位时,正常工作期间才有可能发生 \overline{IRQ} 中断。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

轮询模式下，可轮询任意已知复位值的寄存器，以检查器件是否做好工作准备。如果控制器从被轮询寄存器得到有效响应，器件则做好工作准备。

中断结构

图13所示为中断的结构。有四个中断源寄存器：ISR、LSR、STSIInt和SpclCharInt。中断源分为上层和底层中断。上层中断发生的频率通常比较频繁，可由主控制器通过ISR直接读取。底层中断发生的频率不太频繁，可由主控制器通过LSR、STSIInt或SpclCharInt读取其具体来源。ISR的三个LSB指向包含中断源详细信息的底层中断寄存器。

中断使能

四个中断寄存器的每个中断位均可通过相关的中断使能寄存器位使能或屏蔽，分别是IRQEn、LSRIIntEn、SpclChrIntEn和STSIIntEn寄存器。默认设置下，屏蔽全部中断。

中断清除

发生中断(即 \overline{IRQ} 有效)且读取ISR时，则清零ISR位， \overline{IRQ} 输出变为无效。底层中断信息不会因为相同中断使 \overline{IRQ} 重新变为有效，但仍然储存在底层中断寄存器中，直到分别清除每个中断。SpclCharInt和STSIInt为读清除(COR)。LSR位只有在中断源消除后才清除，读取LSR不会将其清除。

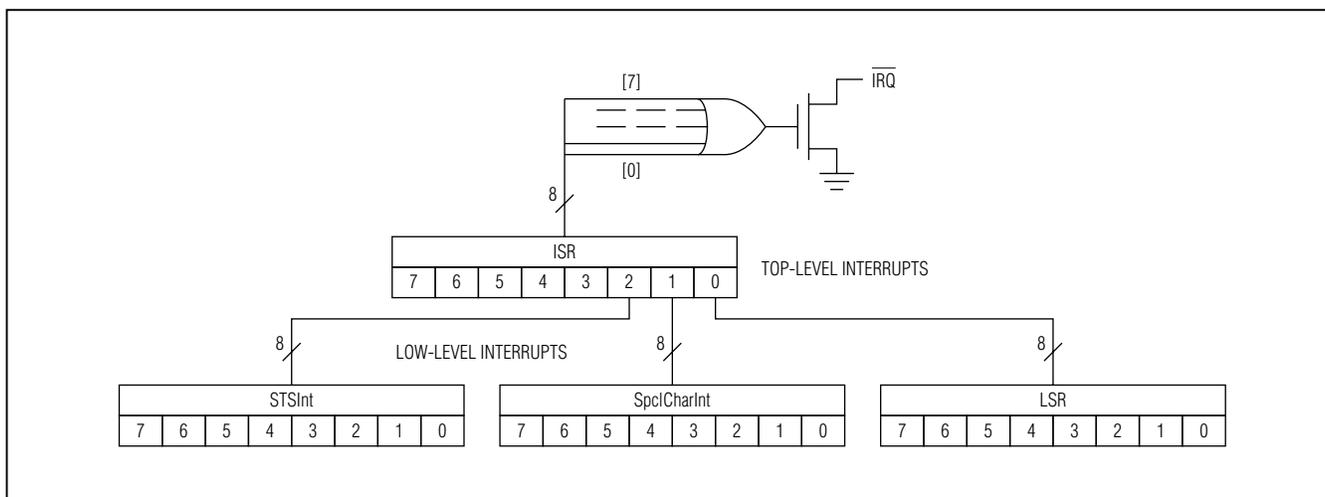


图13. 中断结构简化图

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

寄存器映射

(注: 除非另外说明, 全部默认复位值为0x00。除非另外说明, 全部寄存器均可R/W。)

寄存器	地址	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FIFO DATA									
RHR ¹	0x00	RData7	RData6	RData5	RData4	RData3	RData2	RData1	RData0
THR ¹	0x00	TData7	TData6	TData5	TData4	TData3	TData2	TData1	TData0
INTERRUPTS									
IRQEn	0x01	CTSIEn	RxEmtlyEn	TFifoEmtlyEn	TxTrglEn	RxTrglEn	STSIEn	SpChrlEn	LSRErrEn
ISR ^{1, 2}	0x02	CTSInt	RFifoEmptyInt	TFifoEmptyInt	TxTrglInt	RxTrglInt	STSInt	SpCharInt	LSRErrInt
LSRIntEn	0x03	—	—	NoiseIntEn	RBreaklEn	FrameErrlEn	ParitylEn	ROverrlEn	RTimoutlEn
LSR ¹	0x04	CTSbit	—	RxNoise	RxBreak	FrameErr	RxParityErr	RxOverrun	RTimeout
SpclChrlntEn	0x05	—	—	MltDrplntEn	BREAKlntEn	XOFF2lntEn	XOFF1lntEn	XON2lntEn	XON1lntEn
SpclCharInt ¹	0x06	—	—	MultiDropInt	BREAKInt	XOFF2Int	XOFF1Int	XON2Int	XON1Int
STSIIntEn	0x07	TxEmpyIntEn	SleepIntEn	ClkRdyIntEn	—	GP13IntEn	GP12IntEn	GP11IntEn	GP10IntEn
STSIInt ¹	0x08	TxEmpyInt	SleepInt	ClkReady	—	GP13Int	GP12Int	GP11Int	GP10Int
UART MODES									
MODE1	0x09	—	AutoSleep	ForcedSleep	TrnscvCtrl	RTSHiZ	TXHiZ	TxDisabl	RxDisabl
MODE2	0x0A	EchoSuprs	MultiDrop	Loopback	SpecialChr	RFifoEmptyInv	RxTrglInv	FIFORst	RST
LCR ²	0x0B	RTSbit	TxBreak	ForceParity	EvenParity	ParityEn	StopBits	Length1	Length0
RxTimeOut	0x0C	TimOut7	TimOut6	TimOut5	TimOut4	TimOut3	TimOut2	TimOut1	TimOut0
HDplxDelay	0x0D	Setup3	Setup2	Setup1	Setup0	Hold3	Hold2	Hold1	Hold0
IrDA	0x0E	—	—	TxInv	RxInv	MIR	—	SIR	IrDAEn
FIFOs CONTROL									
FlowLvl	0x0F	Resume3	Resume2	Resume1	Resume0	Halt3	Halt2	Halt1	Halt0
FIFOTrgLvl ²	0x10	RxTrig3	RxTrig2	RxTrig1	RxTrig0	TxTrig3	TxTrig2	TxTrig1	TxTrig0
TxFIFOLvl ¹	0x11	TxFL7	TxFL6	TxFL5	TxFL4	TxFL3	TxFL2	TxFL1	TxFL0
RxFIFOLvl ¹	0x12	RxFL7	RxFL6	RxFL5	RxFL4	RxFL3	RxFL2	RxFL1	RxFL0
FLOW CONTROL									
FlowCtrl	0x13	SwFlow3	SwFlow2	SwFlow1	SwFlow0	SwFlowEn	GP1Addr	AutoCTS	AutoRTS
XON1	0x14	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XON2	0x15	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XOFF1	0x16	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XOFF2	0x17	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIOs									
GPIOConfig	0x18	GP3OD	GP2OD	GP1OD	GP0OD	GP3Out	GP2Out	GP1Out	GP0Out
GPIOData	0x19	GP13Dat	GP12Dat	GP11Dat	GP10Dat	GP03Dat	GP02Dat	GP01Dat	GP00Dat
CLOCK CONFIGURATION									
PLLConfig ²	0x1A	PLLFactor1	PLLFactor0	PreDiv5	PreDiv4	PreDiv3	PreDiv2	PreDiv1	PreDiv0
BRGConfig	0x1B	—	—	4xMode	2xMode	FRACT3	FRACT2	FRACT1	FRACT0
DIVLSB ²	0x1C	Div7	Div6	Div5	Div4	Div3	Div2	Div1	Div0
DIVMSB	0x1D	Div15	Div14	Div13	Div12	Div11	Div10	Div9	Div8
CLKSource ²	0x1E	CLKtoRTS	—	—	—	PLLBypass	PLLEn	CrystalEn	—

¹表示非读/写模式: RHR = R, THR = W, ISR = COR, LSR = R, SpclCharInt = COR, STSIInt = R/COR, TxFIFOLvl = R, RxFIFOLvl = R。

²表示非零默认复位值: ISR = 0x60, LCR = 0x05, FIFOTrgLvl = 0xFF, PLLConfig = 0x01, DIVLSB = 0x01, CLKSource = 0x18。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

寄存器详细说明

MAX3108A采用平行寄存器结构，无需寄存器影射，设置简单、高效。全部寄存器均为8位宽。

接收保持寄存器(RHR)

地址:	0x00							
模式:	R							
位	7	6	5	4	3	2	1	0
名称	RData7	RData6	RData5	RData4	RData3	RData2	RData1	RData0
复位	0	0	0	0	0	0	0	0

第7至0位: RData_x

RHR为接收FIFO的底部，用于读出接收FIFO中的数据，包含接收FIFO中的最早(最先接收)字符。RHR[0]为在RX输入上接收字符的LSB，是接收器接收的串行数据字的第一个数据位。读RHR将从接收FIFO中删除被读取的字，清除空间，以供接收更多数据。

发送保持寄存器(THR)

地址:	0x00							
模式:	W							
位	7	6	5	4	3	2	1	0
名称	TData7	TData6	TData5	TData4	TData3	TData2	TData1	TData0
复位	0	0	0	0	0	0	0	0

第7至0位: TData_x

THR寄存器供主控制器写入数据，以供UART随后发送，该数据置于发送FIFO中。THR[0]为LSB，是发送器在START位之后发送的串行数据字的第一个数据位。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

IRQ使能寄存器(IRQEn)

地址:	0x01							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	CTSIEn	RxEmtlyEn	TFifoEmtlyEn	TxTrglEn	RxTrglEn	STSIEn	SpChrlEn	LSRErrlEn
复位	0	0	0	0	0	0	0	0

IRQEn寄存器用于使能 $\overline{\text{IRQ}}$ 硬中断。可启用八个ISR中断源中的任意中断源，以在 $\overline{\text{IRQ}}$ 上产生中断。IRQEn位只影响 $\overline{\text{IRQ}}$ 输出，对ISR内容或行为无任何影响。IRQEn的每个位都有一个对应的ISR位。

第7位: CTSIEn

CTSIEn位在ISR[7]中的CTSInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将CTSIEn置低时禁止因为CTSInt产生 $\overline{\text{IRQ}}$ 中断。

第6位: RxEmtlyEn

RxEmtlyEn位在ISR[6]中的RFifoEmptyInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将RxEmtlyEn置低时禁止因为RFifoEmptyInt产生 $\overline{\text{IRQ}}$ 中断。

第5位: TFifoEmtlyEn

TFifoEmtlyEn位在ISR[5]中的TFifoEmptyInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将TFifoEmtlyEn置低时禁止因为TFifoEmptyInt产生 $\overline{\text{IRQ}}$ 中断。

第4位: TxTrglEn

TxTrglEn位在ISR[4]中的TxTrglInt interrupt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将TxTrglEn置低时禁止因为TxTrglInt产生 $\overline{\text{IRQ}}$ 中断。

第3位: RxTrglEn

RxTrglEn位在ISR[3]中的RxTrglInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将RxTrglEn置低时禁止因为RxTrglInt产生 $\overline{\text{IRQ}}$ 中断。

第2位: STSIEn

STSIEn位在ISR[2]中的STSIInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将STSIEn置低时禁止因为STSIInt产生 $\overline{\text{IRQ}}$ 中断。

第1位: SpChrlEn

SpChrlEn位在ISR[1]中的SpCharInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将SpChrlEn置低时禁止因为SpCharInt产生 $\overline{\text{IRQ}}$ 中断。

第0位: LSRErrlEn

LSRErrlEn位在ISR[0]中的LSRErrInt中断置位时使能 $\overline{\text{IRQ}}$ 产生中断。将LSRErrlEn置低时禁止因为LSRErrInt产生 $\overline{\text{IRQ}}$ 中断。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

中断状态寄存器(ISR)

地址:	0x02							
模式:	COR							
位	7	6	5	4	3	2	1	0
名称	CTSInt	RFifoEmptyInt	TFifoEmptyInt	TxTrgInt	RxTrgInt	STSInt	SpCharInt	LSRErrInt
复位	0	1	1	0	0	0	0	0

中断状态寄存器提供MAX3108A产生的全部中断的概要信息。读取ISR后, 中断位和 \overline{IRQ} 上的中断均被清除。MAX3108A工作在轮询模式时, 可通过轮询ISR确定UART的状态。中断驱动模式下, \overline{IRQ} 由相应的IRQEn位使能。ISR的值提供关于中断源的直接信息, 或者指向包含更多详细信息的其它寄存器。

第7位: CTSInt

\overline{CTS} 输入的逻辑状态发生跳变时, 产生CTSInt中断。读取ISR后, CTSInt清零。 \overline{CTS} 输入的当前逻辑状态可通过LSR[7]: \overline{CTS} bit位读取。

第6位: RFifoEmptyInt

接收FIFO为空时, 产生RFifoEmptyInt中断。读取ISR后, RFifoEmptyInt清零。其逻辑可通过MODE2[3]: RFifoEmptyInv位反相。

第5位: TFifoEmptyInt

发送FIFO为空及发送器正在发送最后一个字符时, 产生TFifoEmptyInt中断。利用STSInt[7]: TxEmptyInt确定何时完成最后一个字符的发送。读取ISR后, TFifoEmptyInt清零。

第4位: TxTrgInt

发送FIFO中的字符数量等于或大于FIFOTrgLvl[3:0]中定义的发送FIFO触发点时, 产生TxTrgInt中断。发送FIFO空满状态下降至触发点以下或读取ISR后, TxTrgInt清零。TxTrgInt可作为发送FIFO接近溢出的报警。

第3位: RxTrgInt

接收FIFO空满状态达到FIFOTrgLvl[7:4]中定义的接收FIFO触发点时, 产生RxTrgInt中断。RxTrgInt可作为接收FIFO接近溢出的指示, 也可用于指示已知数量的数据字可供在一个数据块中读取。RxTrgInt的逻辑可通过MODE2[2]: RxTrgInv位反相。读取ISR后, RxTrgInt清零。

第2位: STSInt

由STSIntEn位使能的STSInt寄存器中的任意中断为逻辑高时, 产生STSInt中断。读取ISR后, STSInt清零, 但STSInt中引起该中断的位仍然保持置位。关于该中断的详细信息, 请参见STSInt寄存器说明。

第1位: SpCharInt

接收到特殊字符、检测到线路断开或在多点模式下接收到地址字符时, 产生SpCharInt中断。读取ISR后, SpCharInt清零, 但SpclCharInt中引起该中断的位仍然保持置位。关于该中断的详细信息, 请参见SpclCharInt寄存器说明。

第0位: LSRErrInt

当LSR中由LSRIntEn中对应位使能的任意中断置位时, 产生LSRErrInt中断。读取ISR后, 该位清零。关于该中断的详细信息, 请参见LSR寄存器说明。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

线路状态中断使能寄存器(LSRIntEn)

地址:	0x03							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	—	—	NoiseIntEn	RBreakIEn	FrameErrIEn	ParityIEn	ROverrIEn	RTimoutIEn
复位	0	0	0	0	0	0	0	0

LSRIntEn允许将LSR中断连接至ISR[0]。LSRIntEn位只影响ISR[0]: LSRErrInt位, 对LSR内容或行为无任何影响。LSRIntEn寄存器的第5至0位有对应的LSR位, 不使用第7和第6位。

第7位和第6位: 无功能

第5位: NoiseIntEn

将NoiseIntEn位置高时, 使能将LSR[5]: RxNoise中断连接至ISR[0]; 如果NoiseIntEn置低, RxNoise不连接至ISR[0]。

第4位: RBreakIEn

将RBreakIEn位置高时, 使能将LSR[4]: RxBreak中断连接至ISR[0]; 如果RBreakIEn置低, RxBreak不连接至ISR[0]。

第3位: FrameErrIEn

将FrameErrIEn位置高时, 使能将LSR[3]: FrameErr中断连接至ISR[0]; 如果FrameErrIEn置低, FrameErr不连接至ISR[0]。

第2位: ParityIEn

将ParityIEn位置高时, 使能将LSR[2]: RxParityErr中断连接至ISR[0]; 如果ParityIEn置低, RxParityErr不连接至ISR[0]。

第1位: ROverrIEn

将ROverrIEn位置高时, 使能将LSR[1]: RxOverrun中断连接至ISR[0]; 如果ROverrIEn置低, RxOverrun不连接至ISR[0]。

第0位: RTimoutIEn

将RTimoutIEn位置高时, 使能将LSR[0]: RTimeout中断连接至ISR[0]; 如果RTimoutIEn置低, RTimeout不连接至ISR[0]。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

线路状态寄存器(LSR)

地址:	0x04							
模式:	R							
位	7	6	5	4	3	2	1	0
名称	CTSbit	—	RxNoise	RxBreak	FrameErr	RxParityErr	RxOverrun	RTimeout
复位	X	0	0	0	0	0	0	0

LSR包含与最新通过RHR从RxFIFO读取的数据字的全部错误信息。读取LSR后不清零LSR位；从RHR读出下一个字符之前，这些位保持置位；LSR[1]除外，读取RHR或LSR即将其清零。LSR也包含CTS输入的当前逻辑状态。

第7位: CTSbit

CTSbit位表示CTS输入的当前逻辑状态。CTS输入为低电平时，该位清零；CTS输入为高电平时，该位置位。上电或复位后，CTSbit的逻辑状态取决于CTS输入的状态。

第6位: 无功能

第5位: RxNoise

接收字符期间，如果在RX输入上检测到噪声，针对该字符产生RxNoise中断。LSR[5]对应于从RHR最新读取的字符。从RHR读出“噪声字符”之后的字符后，RxNoise清零。如果经LSRIntEn[5]使能，RxNoise在ISR[0]中产生一个中断。

第4位: RxBreak

如果检测到线路断开(RX输入为低电平的时间周期超过设置的字符持续时间)，将一个线路断开字符置于Rx FIFO中，并为该字符产生RxBreak中断。线路断开字符由一个全零数据字符表示。RxBreak中断区分正常的全零字符与线路断开字符。LSR[4]对应于从RHR中最新读取的当前字符。从RHR读出断开字符之后的字符后，RxBreak清零。如果经LSRIntEn[4]使能，RxBreak在ISR[0]中产生一个中断。

第3位: FrameErr

当接收到的数据帧与预期帧格式在长度上不匹配时，产生FrameErr中断。帧错误与预期STOP位中的错误相关。LSR[3]对应于从RHR中最新读取的字符的帧错误。从RHR读出受影响字符之后的字符后，FrameErr清零。如果经LSRIntEn[3]使能，FrameErr在ISR[0]中产生一个中断。

第2位: RxParityErr

当根据接收字符计算的奇偶检验位与接收到的奇偶检验位不匹配时，产生RxParityErr中断。LSR[2]表示从RHR中最新读取的字符的奇偶校验错误。从RHR读出受影响字符之后的字符时，RxParityErr清零。

9位多点模式下(MODE2[6]为逻辑1)，接收器不进行奇偶校验，第9位(地址/数据)储存在LSR[2]中。

如果经LSRIntEn[2]使能，RxParityErr在ISR[0]中产生一个中断。

第1位: RxOverrun

当接收FIFO为满状态，接收的更多数据不能进入接收FIFO时，产生RxOverrun中断。接收FIFO保存已有的数据，丢弃全部新数据。读取LSR或者Rx FIFO空满状态下降至低于其最大值时，RxOverrun清零。如果经LSRIntEn[1]使能，RxOverrun在ISR[0]中产生一个中断。

第0位: RTimeout

RTimeout中断表示接收FIFO中存在旧数据。当Rx FIFO中的所有字符存在的时间至少达到RxTimeOut寄存器设置的时间周期时，RTimeout置位。

带有128字FIFO的SPI/I2C UART, WLP封装

读取RHR或RxFIFO接收到新字符时，超时计数器重新启动。如果RxTimeOut的值为零，禁止RTimeout。从RxFIFO中读出一个字或接收到新字时，RTimeout清零。如果经LSRIntEn[0]使能，RTimeout在ISR[0]中产生一个中断。

特殊字符中断使能寄存器(SpchlChrIntEn)

地址:	0x05							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	—	—	MltDrpIntEn	BREAKIntEn	XOFF2IntEn	XOFF1IntEn	XON2IntEn	XON1IntEn
复位	0	0	0	0	0	0	0	0

SpchlChrIntEn允许将SpchlCharInt中断连接至ISR[1]。SpchlChrIntEn位只影响ISR[1]: SpchlCharInt位，对SpchlCharInt内容或行为无任何影响。

第7位和第6位：无功能

第5位：MltDrpIntEn

将MltDrpIntEn位置高时，使能将SpchlCharInt[5]: MultiDropInt中断连接至ISR[1]；如果MltDrpIntEn置低，MultiDropInt不连接至ISR[1]。

第4位：BREAKIntEn

将BREAKIntEn位置高时，使能将SpchlCharInt[4]: BREAKInt中断连接至ISR[1]；如果BREAKIntEn置低，BREAKInt不连接至ISR[1]。

第3位：XOFF2IntEn

将XOFF2IntEn位置高时，使能将SpchlCharInt[3]: XOFF2Int中断连接至ISR[1]；如果XOFF2IntEn置低，XOFF2Int不连接至ISR[1]。

第2位：XOFF1IntEn

将XOFF1IntEn位置高时，使能将SpchlCharInt[2]: XOFF1Int中断连接至ISR[1]；如果XOFF1IntEn置低，XOFF1Int不连接至ISR[1]。

第1位：XON2IntEn

将XON2IntEn位置高时，使能将SpchlCharInt[1]: XON2Int中断连接至ISR[1]；如果XON2IntEn置低，XON2Int不连接至ISR[1]。

第0位：XON1IntEn

将XON1IntEn位置高时，使能将SpchlCharInt[0]: XON1Int中断连接至ISR[1]；如果XON1IntEn置低，XON1Int不连接至ISR[1]。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

特殊字符中断寄存器(SpchlCharInt)

地址:	0x06							
模式:	COR							
位	7	6	5	4	3	2	1	0
名称	—	—	MultiDroplnt	BREAKInt	XOFF2Int	XOFF1Int	XON2Int	XON1Int
复位	0	0	0	0	0	0	0	0

SpchlCharInt包含接收到特殊字符、多点模式下接收到地址或发生线路断开时产生的中断。

第7位和第6位：无功能

第5位：MultiDroplnt

器件在9位多点模式下接收到地址字符时，产生MultiDroplnt中断，由MODE2[6]使能。读取SpchlCharInt后，MultiDroplnt清零。如果经SpchlCharIntEn[5]使能，MultiDroplnt在ISR[1]中产生一个中断。

第4位：BREAKInt

接收器检测到线路断开时(RX为低电平的时间长于一个字符长度)，产生BREAKInt中断。读取SpchlCharInt后，BREAKInt清零。如果经SpchlCharIntEn[4]使能，BREAKInt在ISR[1]中产生一个中断。

第3位：XOFF2Int

接收到XOFF2特殊字符并且MODE2[4]使能特殊字符检测功能时，产生XOFF2Int中断。读取SpchlCharInt后，XOFF2Int清零。如果经SpchlCharIntEn[3]使能，XOFF2Int在ISR[1]中产生一个中断。

第2位：XOFF1Int

接收到XOFF1特殊字符并且MODE2[4]使能特殊字符检测功能时，产生XOFF1Int中断。读取SpchlCharInt后，XOFF1Int清零。如果经SpchlCharIntEn[2]使能，XOFF1Int在ISR[1]中产生一个中断。

第1位：XON2Int

接收到XON2特殊字符并且MODE2[4]使能特殊字符检测功能时，产生XON2Int中断。读取SpchlCharInt后，XON2Int清零。如果经SpchlCharIntEn[1]使能，XON2Int在ISR[1]中产生一个中断。

第0位：XON1Int

接收到XON1特殊字符并且MODE2[4]使能特殊字符检测功能时，产生XON1Int中断。读取SpchlCharInt后，XON1Int清零。如果经SpchlCharIntEn[0]使能，XON1Int在ISR[1]中产生一个中断。

带有128字FIFO的SPI/I2C UART, WLP封装

STS中断使能寄存器(STSIntEn)

地址:	0x07							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	TxEmptyIntEn	SleepIntEn	ClkRdyIntEn	—	GPI3IntEn	GPI2IntEn	GPI1IntEn	GPI0IntEn
复位	0	0	0	0	0	0	0	0

STSIntEn允许将STSInt中断连接至ISR[2]。STSIntEn位只影响ISR[2]: STSInt位, 对STSInt内容和行为无任何影响; GPIxIntEn中断使能位除外, 该位控制STSInt[3:0]中断的产生。

第7位: TxEmptyIntEn

将TxEmptyIntEn位置高时, 使能将STSInt[7]: TxEmptyInt中断连接至ISR[2]; 如果TxEmptyIntEn置低, TxEmptyInt不连接至ISR[2]。

第6位: SleepIntEn

将SleepIntEn位置高时, 使能将STSInt[6]: SleepInt中断连接至ISR[2]; 如果SleepIntEn置低, SleepInt不连接至ISR[2]。

第5位: ClkRdyIntEn

将ClkRdyIntEn位置高时, 使能将STSInt[6]: ClkReady中断连接至ISR[2]; 如果ClkRdyIntEn置低, ClkReady不连接至ISR[2]。

第4位: 无功能

第3至0位: GPIxIntEn

将GPIxIntEn位置高时, 使能产生STSInt[3:0]: GPIxInt中断; 如果任意GPIxIntEn位置低, 则不产生相关的GPIxInt中断。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

状态中断寄存器(STSInt)

地址:	0x08							
模式:	R/COR							
位	7	6	5	4	3	2	1	0
名称	TxEmptyInt	SleepInt	ClkReady	—	GPI3Int	GPI2Int	GPI1Int	GPI0Int
复位	0	0	0	0	0	0	0	0

第7位: TxEmptyInt

TxFIFO为空且完成最后一个字符的发送时,产生TxEmptyInt中断。读取STSInt后, TxEmptyInt清零。如果经STSIntEn[7]使能, TxEmptyInt在ISR[2]中产生一个中断。

第6位: SleepInt

器件进入休眠模式时,产生SleepInt状态位。器件退出休眠模式时, SleepInt清零。禁用时钟时, 该状态位清零, 不通过读取STSInt清零。如果经STSIntEn[6]使能, SleepInt在ISR[2]中产生一个中断。

第5位: ClkReady

当时钟、预分频器和PLL已经达到稳定时,产生ClkReady状态位,表示器件已经做好数据通信准备。ClkReady位仅适用于使用晶振时,不支持XIN上的外部时钟。

禁用时钟时, ClkReady清零,不通过读取STSInt清零。如果经STSIntEn[5]使能, ClkReady在ISR[2]中产生一个中断。

第4位: 无功能

第3至0位: GPIxInt

当对应GPIO输入的逻辑状态发生变化时,产生GPIxInt中断。读取STSInt后, GPIxInt中断清零。如果经STSIntEn[3:0]中的对应位使能, GPIxInt中断在ISR[2]中产生一个中断。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

MODE1寄存器

地址:	0x09							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	—	AutoSleep	ForcedSleep	TrnscvCtrl	RTSHiZ	TXHiZ	TxDisabl	RxDisabl
复位	0	0	0	0	0	0	0	0

第6位: AutoSleep

将AutoSleep位置高时, 将器件设置为在一定周期内无活动时自动进入低功耗休眠模式(见*自动休眠模式*部分)。器件进入休眠模式时, 在STSInt[6]: SleepInt中产生一个中断。

第5位: ForcedSleep

将ForcedSleep位置高时, 强制器件进入低功耗休眠模式(见*强制休眠模式*部分)。可通过ForcedSleep位读取器件的当前休眠状态, 即使UART处于休眠模式。

第4位: TrnscvCtrl

将TrnscvCtrl位置高时, 进入收发器自动方向控制模式。该模式下, $\overline{\text{RTS}}$ 自动控制收发器的发送/接收使能/禁止输入。 $\overline{\text{RTS}}$ 为逻辑低, 收发器处于接收模式, 禁止发送器, 直到TxFIFO包含可供发送的数据, 此时 $\overline{\text{RTS}}$ 自动设置为逻辑高, 然后发送器发送数据。一旦发送器为空, $\overline{\text{RTS}}$ 再次自动强制为逻辑低。

$\overline{\text{RTS}}$ 相对于TX输出的建立和保持时间可通过HDpIxDelay寄存器定义。TxFIFO为空时, 在ISR[5]中产生发送器空中断。

第3位: RTSHiZ

将RTSHiZ位置高时, 使 $\overline{\text{RTS}}$ 为三态。

第2位: TXHiZ

将TXHiZ位置高时, 使TX输出为三态。

第1位: TxDisabl

将TxDisabl位置高时, 禁止发送。如果在发送期间将TxDisabl位置高, 发送器完成发送当前字符, 然后结束发送。发送FIFO中的数据仍然保留在TxFIFO中。发送之后, TX输出置为逻辑高。

自动收发器方向控制模式下, 发送器完全空时, TxDisabl为逻辑高。

第0位: RxDisabl

将RxDisabl位置高时, 禁止接收器, 所以接收器停止接收数据。接收FIFO中的全部数据仍然保留在Rx FIFO中。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

MODE2寄存器

地址:	0x0A							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	EchoSuprs	MultiDrop	Loopback	SpecialChr	RFifoEmptyInv	RxTrgInv	FIFORst	RST
复位	0	0	0	0	0	0	0	0

第7位: EchoSuprs

将EchoSuprs位置高时, 丢弃器件在发送器发送数据期间接收到的所有数据。半双工模式下, 例如RS-485和IrDA, 这允许防止本地回波数据。通过在HDpIxDelay[3:0]中设置保持时间, 接收器可在发送器结束发送后的一段时间内阻塞数据。

第6位: MultiDrop

将MultiDrop位置高时, 使能9位多点模式。该位置位时, 接收器不进行奇偶校验, 发送器不产生奇偶检验位。地址/数据指示占据接收和发送数据字中奇偶检验位的位置。多点模式下, LSR[2]中的奇偶校验错误中断表示接收的每个9位数据字符的第9位(地址/数据指示)。

第5位: Loopback

将Loopback位置高时, 使能内部本地环回模式, 在内部将TX连接至RX, 将 $\overline{\text{RTS}}$ 连接至 $\overline{\text{CTS}}$ 。本地环回模式下, TX输出和RX输入从内部发送器和接收器断开, TX输出为三态。 $\overline{\text{RTS}}$ 输出保持连接至内部逻辑电路, 表示LCR[7]中设置的逻辑状态。 $\overline{\text{CTS}}$ 输入从 $\overline{\text{RTS}}$ 及内部逻辑电路断开, 因此 $\overline{\text{CTS}}$ 保持为高阻态。

第4位: SpecialChr

将SpecialChr位置高时, 使能特殊字符检测。接收器可检测最多4个特殊字符, 在FlowCtrl[5:4]中选择, 在XON1、XON2、XOFF1和或XOFF2寄存器中定义; 如果经FlowCtrl[2]:GPIOAddr使能, 可选择与GPIOx输入组合。接收到特殊字符时, 将其置于RxFIFO, 并在ISR[1]中产生特殊字符检测中断。

如果经FlowCtrl[3]: SwFlowEn使能, 除自动XON/XOFF流控制外, 特殊字符检测也可用于其它流控制模式。此时, XON/XOFF流控制限于单字节XON和XOFF字符(XON1和XOFF1), 只可定义两个特殊字符(XON2和XOFF2)。

第3位: RFifoEmtyInv

将RFifoEmtyInv位置高时, 将ISR[6]: RFifoEmptyInt中接收器空中断的逻辑反相。如果RFifoEmtyInv置低, 接收FIFO为空时产生RFifoEmptyInt; 如果RFifoEmtyInv置高, 将数据置于空接收FIFO时产生RFifoEmptyInt。

第2位: RxTrgInv

将RxTrgInv位置高时, 将RxFIFO触发的逻辑反相。如果RxTrgInv位置低, RxFIFO空满状态高于FIFOTrgLvl[7:4]中设置的触发点时, 在ISR[3]: RxTrgInt中产生中断; 如果RxTrgInv置高, RxFIFO的空满状态低于FIFOTrgLvl[7:4]中设置的触发点时, 在ISR[3]中产生中断。

第1位: FIFORst

将FIFORst位置高时, 将接收和发送FIFO的全部数据清除。FIFO复位后, 将FIFORst置低, 继续常规工作。

第0位: RST

将RST为置高时, 启动器件的软复位。复位期间, I²C/SPI总线保持活动; RST置位时, 可与器件进行通信。复位期间, 全部寄存器位恢复为其复位状态, 清除全部FIFO。

软复位之后, 将RST置低, 继续常规工作。软件复位后, 需要对器件重新编程。

带有128字FIFO的SPI/I2C UART, WLP封装

线路控制寄存器(LCR)

地址:	0x0B							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	RTSbit	TxBreak	ForceParity	EvenParity	ParityEn	StopBits	Length1	Length0
复位	0	0	0	0	0	1	0	1

第7位: $\overline{\text{RTSbit}}$

$\overline{\text{RTSbit}}$ 位直接控制 $\overline{\text{RTS}}$ 输出逻辑状态。如果 $\overline{\text{RTSbit}}$ 位为逻辑1, $\overline{\text{RTS}}$ 则为逻辑1; 如果 $\overline{\text{RTSbit}}$ 为逻辑0, $\overline{\text{RTS}}$ 则为逻辑0。 $\overline{\text{RTSbit}}$ 仅在CLKSource[7]: CLKtoRTS为逻辑低时工作。

第6位: TxBreak

将TxBreak位置高时, 产生线路断开, TX输出由此保持为低电平。将TxBreak置低之前, TX维持低电平。

第5位: ForceParity

ForceParity位使能强制奇偶校验,屏蔽常规产生奇偶校验。将LCR[3]: ParityEn和ForceParity位置高时,使用强制奇偶校验。强制奇偶校验模式下, 如果LCR[4]: EvenParity位为逻辑低, 发送器强制奇偶校验位为逻辑高; 如果EvenParity为逻辑高, 则强制奇偶校验位为逻辑低。强制奇偶校验模式使能发送器控制9位多点通信中的地址/数据位。

第4位: EvenParity

将EvenParity位置高时, 使能发送器和接收器的偶校验; 如果EvenParity置低, 则采用奇校验。

第3位: ParityEn

将ParityEn位置高时, 使能在Tx和Rx接口上使用奇偶校验位; 将ParityEn位置低时, 禁止使用奇偶校验。

如果ParityEn置低, 发送器则不产生奇偶校验位, 接收器也不检测奇偶校验位; 如果ParityEn置高, 发送器产生奇偶校验位, 其逻辑状态在LCR[4]: EvenParity中定义, 接收器根据相同的极性状态检查奇偶校验位。

第2位: StopBits

StopBits位定义停止位的数量, 与LCR[1:0]中设置的字长度有关(表1)。例如, 当StopBits置高且字长度为5, 发送器产生的字的停止位长度等于1.5倍波特周期。这种条件下, 接收器将长于1位时间的长度作为停止位。

第1位和第0位: Lengthx

Lengthx位配置发送器在异步Tx接口产生以及接收器在异步Rx接口检查的字长度(表2)。

表1. StopBits真值表

StopBits	WORD LENGTH	STOP BIT LENGTH
0	5, 6, 7, 8	1
1	5	1-1.5
1	6, 7, 8	2

表2. Lengthx真值表

Length1	Length0	WORD LENGTH
0	0	5
0	1	6
1	0	7
1	1	8

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

接收器超时寄存器(RxTimeOut)

地址:	0x0C							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	TimOut7	TimOut6	TimOut5	TimOut4	TimOut3	TimOut2	TimOut1	TimOut0
复位	0	0	0	0	0	0	0	0

第7至0位: TimOutx

RxTimeOut寄存器允许设置从接收到接收FIFO中最后(最新)一个字符到LSR[0]中产生接收数据超时中断之间的延时。TimOutx的单位以完整的字符帧表示,取决于字符长度、奇偶校验和STOP位设置,以及波特率。如果RxTimeOut值等于零,则不产生超时中断。

HDplxDelay寄存器

地址:	0x0D							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Setup3	Setup2	Setup1	Setup0	Hold3	Hold2	Hold1	Hold0
复位	0	0	0	0	0	0	0	0

HDplxDelay寄存器允许设置自动收发器方向控制模式下 $\overline{\text{RTS}}$ 跳变与TX输出活动之间的建立和保持时间,通过将MODE1[4]: TrnscvCtrl位置高使能。保持时间也用于确保半双工通信中的回波抑制。HDplxDelay适用于两倍和四倍速率模式。

第7至4位: Setupx

Setupx位定义自动收发器方向控制模式下发送器开始发送其第一个字符之前 $\overline{\text{RTS}}$ 跳变为高电平的建立时间,通过将MODE1[4]: TrnscvCtrl位置高使能。这允许器件计算外部发送器的使能延迟和传输延迟之间的偏移。Setupx也用于在发送开始之前确定发送线路的稳定状态。

HDplxDelay建立时间延迟的分辨率为一个位间隔,即一个波特周期;该延迟与波特率有关。最大延迟为15个位间隔。

第3至0位: Holdx

Holdx位定义自动收发器方向控制模式下发送器结束其最后一个字符发送后 $\overline{\text{RTS}}$ 保持为高电平的保持时间,通过将MODE1[4]: TrnscvCtrl位置高使能。保持时间延迟(从发送最后一个STOP位之后开始)之后, $\overline{\text{RTS}}$ 跳变位低电平。这使外部发送器在保持时间内保持使能。

Holdx位也定义回波抑制模式下的延迟,通过将MODE2[7]: EchoSuprs位置高使能。更多信息请参见回波抑制部分。

HDplxDelay保持时间延迟的分辨率为一个位间隔,即一个波特周期,因此该延迟与波特率有关。最大延迟为15个位间隔。

带有128字FIFO的SPI/I2C UART, WLP封装

IrDA寄存器

地址:	0x0E							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	—	—	TxInv	RxInv	MIR	—	SIR	IrDAEn
复位	0	0	0	0	0	0	0	0

IrDA寄存器允许选择Tx和Rx接口上的IrDA SIR和MIR兼容脉冲整形，也允许将TX和RX的逻辑反相，与是否使能IrDA脉冲整形无关。

第7、6、2位：无功能

第5位：TxInv

将TxInv位置高时，将TX输出的逻辑反相。该功能与IrDA工作无关。

第4位：RxInv

将RxInv位置高时，将RX输入的逻辑反相。该功能与IrDA工作无关。

第3位：MIR

将MIR和IrDAEn位置高时，选择IrDA 1.1 (MIR)的宽度为1/4周期脉宽。

第1位：SIR

将SIR和IrDAEn位置高时，选择IrDA 1.0脉冲(SIR)的宽度为3/16周期脉宽。

第0位：IrDAEn

将IrDAEn位置高时，将器件设置为在TX输出产生IrDA兼容脉冲，以及在TX输入检测IrDA兼容脉冲；如果IrDAEn置低，发送器产生以及接收器检测常规(非IrDA)脉冲。使用IrDAEn与SIR或MIR位组合，选择脉宽。

流控制门限寄存器(FlowLvl)

地址:	0x0F							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Resume3	Resume2	Resume1	Resume0	Halt3	Halt2	Halt1	Halt0
复位	0	0	0	0	0	0	0	0

FlowLvl寄存器用于选择自动软件(XON/XOFF)和硬件($\overline{RTS}/\overline{CTS}$)流控制模式下的RxFIFO门限。

第7至4位：Resumex

Resumex位设置自动软件流控制模式下自动发送XON字符或AutoRTS模式下 \overline{RTS} 自动变为有效的接收FIFO门限。一旦RxFIFO空满状态低于Resumex中的值，则发生这些流控制动作，通知远端站点恢复发送。门限值计算公式为 $8 \times \text{Resumex}$ 。最终可用的门限值范围为0至120 (十进制)。

第3至0位：Haltx

Haltx位设置自动软件流控制模式下自动发送XOFF字符或AutoRTS模式下 \overline{RTS} 自动变为无线的接收FIFO门限。一旦RxFIFO空满状态高于Haltx中的值，则发生这些流控制动作，通知远端站点暂停发送。门限值计算公式为 $8 \times \text{Haltx}$ 。最终可用的门限值范围为0至120 (十进制)。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

FIFO中断触发点寄存器(FIFOTrgLvl)

地址:	0x10							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	RxTrig3	RxTrig2	RxTrig1	RxTrig0	TxTrig3	TxTrig2	TxTrig1	TxTrig0
复位	1	1	1	1	1	1	1	1

第7至4位: RxTrigx

RxTrigx位允许定义器件在ISR[3]中产生中断的接收FIFO门限值, 该中断可用于表示接收FIFO接近溢出或预定义数量的FIFO单元可用于作为一个数据块进行读取, 取决于MODE2[2]: RxTrgInvl的状态。

可选的门限分辨率为8个FIFO单元, 所以实际的FIFO触发点计算公式为 $8 \times \text{RxTrigx}$ 。最终可用的触发点范围为0至120 (十进制)。

第3至0位: TxTrigx

TxTrigx位允许定义器件在ISR[4]中产生中断的发送FIFO门限值, 该中断可用于管理发送至发送FIFO的数据流。例如, 如果定义的触发点接近TxFIFO的底部, 主机就知道预定义数量的FIFO单元可供作为一个数据块进行写入; 如果设置的触发点接近FIFO的顶部, 当发送FIFO接近溢出时, 主机就会报警。

可选的门限分辨率为8个FIFO单元, 所以实际的FIFO触发点计算公式为 $8 \times \text{TxTrigx}$ 。最终可用的触发点范围为0至120 (十进制)。

发送FIFO空满状态寄存器(TxFIFOLvl)

地址:	0x11							
模式:	R							
位	7	6	5	4	3	2	1	0
名称	TxFL7	TxFL6	TxFL5	TxFL4	TxFL3	TxFL2	TxFL1	TxFL0
复位	0	0	0	0	0	0	0	0

第7至0位: TxFLx

TxFIFOLvl寄存器表示发送FIFO中当前的字数量。

接收FIFO空满状态寄存器(RxFIFOLvl)

地址:	0x12							
模式:	R							
位	7	6	5	4	3	2	1	0
名称	RxFL7	RxFL6	RxFL5	RxFL4	RxFL3	RxFL2	RxFL1	RxFL0
复位	0	0	0	0	0	0	0	0

第7至0位: RxFLx

RxFIFOLvl寄存器表示接收FIFO中当前的字数量。

带有128字FIFO的SPI/I2C UART, WLP封装

流控制寄存器(FlowCtrl)

地址:	0x13							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	SwFlow3	SwFlow2	SwFlow1	SwFlow0	SwFlowEn	GPIAddr	AutoCTS	AutoRTS
复位	0	0	0	0	0	0	0	0

FlowCtrl寄存器配置硬件(RTS/CTS)和软件(XON/XOFF)流控制, 以及特殊字符检测。

第7至4位: SwFlowx

SwFlowx位选择用于自动软件流控制和/或特殊字符检测的XON和XOFF字符, 与XON1、XON2、XOFF1和/或XOFF2寄存器中设置的字符组合使用。请参见表3。

如果使能自动软件流控制(通过FlowCtrl[3]: SwFlowEn)且没有使能特殊字符检测, SwFlowx允许选择单或双XON/XOFF字符流控制。使能双字符流控制时, 接收流控制期间, 发送器首先发送XON1/XOFF1, 然后发送XON2/XOFF2。对于发送流控制, 接收器将接收的XON1/XOFF1和后边接收的XON2/XOFF2作为有效的控制序列, 以恢复/暂停发送。

如果只使能特殊字符检测(通过MODE2[4]: SpecialChr)而禁止自动软件流控制时, SwFlowx允许选择单或双字符检测。单字符检测允许检测两个字符: XON1或XON2和XOFF1或XOFF2。双字符检测不区分是序列还是两个接收的XON1/XON2或XOFF1/XOFF2字符。两个字符必须连续接收, 但是先接收哪个无关紧要。特殊字符置于接收FIFO中。接收到特殊字符时, 产生ISR[1]: SpCharInt中断。

自动软件流控制和特殊字符检测可同时工作。如果将两项都使能, XON1和XOFF1定义自动流控制字符, XON2和XOFF2构成特殊字符检测的字符。

第3位: SwFlowEn

将SwFlowEn位置高时, 使能自动软件流控制模式。自动软件流控制使用的字符由SwFlowx选择。如果除了自动软件流控制之外, 还通过将MODE2[4]: SpecialChr位置高使能了特殊字符检测, 则XON1和XOFF1用于流控制, XON2和XOFF2定义特殊字符。

第2位: GPIAddr

将GPIAddr位置高时, 使能四路GPIO输入与XOFF2组合定义特殊字符, 例如可用于通过硬件定义RS-485从设备的地址。GPIOx输入的逻辑电平定义特殊字符的四个LSB, 四个MSB由XOFF2[7:4]位定义。GPIO输入用于特殊字符定义时, 忽略XOFF2[3:0]位的值。该模式下, 读取XOFF2寄存器不表示GPIO的逻辑。

第1位: AutoCTS

将AutoCTS位置高时, 使能AutoCTS流控制模式。该模式下, 发送器根据 $\overline{\text{CTS}}$ 输入的逻辑状态在Tx接口停止和开始发送数据。关于AutoCTS流控制的更多信息, 请参见*自动硬件流控制*部分。CTS输入的逻辑变化在ISR[7]: CTSInt中产生中断。使能AutoCTS模式之前, 必须通过将MODE1[1]: TxDisabl位置高来关闭发送器。

第0位: AutoRTS

将AutoRTS位置高时, 使能AutoRTS流控制模式。该模式下, $\overline{\text{RTS}}$ 输出的逻辑状态取决于接收FIFO空满状态。使 $\overline{\text{RTS}}$ 改变状态的FIFO门限在FlowLvl中设置。关于AutoRTS流控制的更多信息, 请参见*自动硬件流控制*部分。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

表3. SwFlow[3:0]真值表

RECEIVE FLOW CONTROL		TRANSMIT FLOW CONTROL/SPECIAL CHARACTER DETECTION		DESCRIPTION
SwFlow3	SwFlow2	SwFlow1	SwFlow0	
0	0	0	0	No flow control/no special-character detection.
0	0	X	X	No receive flow control.
1	0	X	X	Transmitter generates XON1, XOFF1.
0	1	X	X	Transmitter generates XON2, XOFF2.
1	1	X	X	Transmitter generates XON1, XON2, XOFF1, and XOFF2.
X	X	0	0	No transmit flow control.
X	X	1	0	Receiver compares XON1 and XOFF1 and controls the transmitter accordingly. XON1 and XOFF1 special-character detection.
X	X	0	1	Receiver compares XON2 and XOFF2 and controls the transmitter accordingly. XON2 and XOFF2 special-character detection.
X	X	1	1	Receiver compares XON1, XON2, XOFF1, and XOFF2 and controls the transmitter accordingly. XON1, XON2, XOFF1, and XOFF2 special-character detection.

X = 无关。

XON1寄存器

地址:	0x14							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
复位	0	0	0	0	0	0	0	0

XON1和XON2寄存器值定义用于自动XON/XOFF流控制的XON字符和/或用于特殊字符检测的特殊字符。更多信息请参见FlowCtrl寄存器说明。

第7至0位: Bitx

如果在FlowCtrl[7:4]中使能单字符XON自动软件流控制, 这些位定义XON1字符; 如果在FlowCtrl[7:4]中选择双字符流控制, 这些位构成双字节XON字符的最低字节; 如果在MODE2[4]中使能特殊字符检测且没有使能自动流控制, 这些位定义一个特殊字符。

如果同时使能特殊字符检测和自动软件流控制, XON1定义XON流控制字符。

带有128字FIFO的SPI/I2C UART, WLP封装

XON2寄存器

地址:	0x15							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
复位	0	0	0	0	0	0	0	0

XON1和XON2寄存器值定义用于自动XON/XOFF流控制的XON字符和/或用于特殊字符检测的特殊字符。更多信息请参见FlowCtrl寄存器说明。

第7至0位: Bitx

如果在FlowCntrl[7:4]中使能单字符自动软件流控制, 这些位定义XON2字符; 如果在FlowCntrl[7:4]中选择双字符流控制, 这些位构成双字节XON字符的最高字节; 如果在MODE2[4]中使能特殊字符检测且没有使能自动软件流控制, 这些位定义一个特殊字符。

如果同时使能特殊字符检测和自动软件流控制, XON2定义特殊字符。

XOFF1寄存器

地址:	0x16							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
复位	0	0	0	0	0	0	0	0

XOFF1和XOFF2寄存器值定义用于自动XON/XOFF流控制的XOFF字符和/或用于特殊字符检测的特殊字符。更多信息请参见FlowCtrl寄存器说明。

第7至0位: Bitx

如果在FlowCntrl[7:4]中使能单字符XOFF自动软件流控制, 这些位定义XOFF1字符; 如果在FlowCntrl[7:4]中选择双字符流控制, 这些位构成双字节XOFF字符的最低字节; 如果在MODE2[4]中使能特殊字符检测且没有使能自动软件流控制, 这些位定义一个特殊字符。

如果同时使能特殊字符检测和自动软件流控制, XOFF1定义XOFF流控制字符。

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

XOFF2寄存器

地址:	0x17							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
复位	0	0	0	0	0	0	0	0

XOFF1和XOFF2寄存器值定义用于自动XON/XOFF流控制的XOFF字符和/或用于特殊字符检测的特殊字符。更多信息请参见FlowCtrl寄存器说明。

第7至0位: Bitx

如果在FlowCntrl[7:4]中使能单字符XOFF自动软件流控制, 这些位定义XOFF1字符; 如果在FlowCntrl[7:4]中选择双字符流控制, 这些位构成双字节XOFF字符的最低字节; 如果在MODE2[4]中使能特殊字符检测且没有使能自动软件流控制, 这些位定义一个特殊字符。

如果同时使能特殊字符检测和自动软件流控制, XOFF2定义特殊字符。

GPIO配置寄存器(GPIOConf)

地址:	0x18							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	GP3OD	GP2OD	GP1OD	GP0OD	GP3Out	GP2Out	GP1Out	GP0Out
复位	0	0	0	0	0	0	0	0

四路GPIO可配置为输入或输出, 可工作在推挽或开漏模式。基准时钟需要有效, GPIO才可工作。

第7至4位: GPxOD

将GPxOD位置高时, 将对应GPIO配置为开漏输出; 将GPxOD位置低时, 将对应GPIO配置为推挽式输出。

在GPxOut中配置为输入时, GPIO为高阻态输入, 具有弱下拉电阻, 与GPxOD的状态无关。

第3至0位: GPxOut

GPxOut位将对应GPIO配置为输入或输出。将GPxOut位置高时, 将对应GPIO配置为输出; 将GPxOut位置低时, 将对应GPIO配置为输入。

带有128字FIFO的SPI/I2C UART, WLP封装

GPIO数据寄存器(GPIOData)

地址:	0x19							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	GPI3Dat	GPI2Dat	GPI1Dat	GPI0Dat	GPO3Dat	GPO2Dat	GPO1Dat	GPO0Dat
复位	0	0	0	0	0	0	0	0

第7至4位: GPIxDat

GPIxDat位表示对应GPIO的输入逻辑。注意, GPIO作为中断源或定时器时, 需要晶振或外部振荡器。

第3至0位: GPOxDat

GPOxDat位允许在GPIOConfig[3:0]中将GPIO配置为输出时设置其逻辑状态。对于开漏工作, GPIO上需要上拉电阻。

PLL配置寄存器(PLLConfig)

地址:	0x1A							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	PLLFactor1	PLLFactor0	PreDiv5	PreDiv4	PreDiv3	PreDiv2	PreDiv1	PreDiv0
复位	0	0	0	0	0	0	0	1

第7至6位: PLLFactorx

PLLFactorx位允许设置PLL乘数。PLL的输入和输出频率必须在表4所示的范围内。在CLKSource[2]中使能PLL。

第5至0位: PreDivx

PreDivx位允许设置PLL的预分频器除数。选择的除数必须使预分频器的输出频率(即PLL输入频率)限制在表4所示的范围内。PLL输入频率计算如下:

$$f_{PLLIN} = f_{CLK} / PreDiv$$

式中, f_{CLK} 为晶振或外部时钟源的输入频率(图14), PreDiv为1至63范围内的整数。

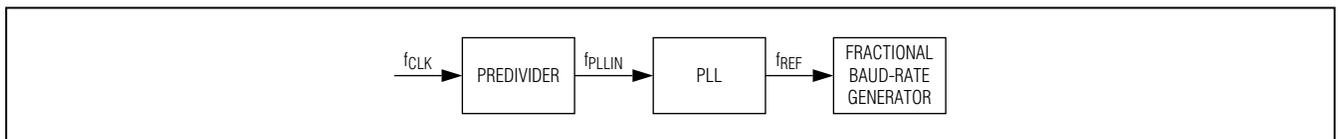


图14. PLL信号通路

表4. PLLFactorx选择指南

PLLFactor1	PLLFactor0	MULTIPLICATION FACTOR	f _{PLLIN}		f _{REF}	
			MIN	MAX	MIN	MAX
0	0	6	500kHz	800kHz	3MHz	4.8MHz
0	1	48	850kHz	1.2MHz	40.8MHz	56MHz
1	0	96	425kHz	1MHz	40.8MHz	96MHz
1	1	144	390kHz	667kHz	56MHz	96MHz

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

波特率发生器除数MSB寄存器(DIVMSB)

地址:	0x1B							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	—	—	4xMode	2xMode	FRACT3	FRACT2	FRACT1	FRACT0
复位	0	0	0	0	0	0	0	0

第7位和第6位：无功能

第5位：4xMode

将4xMode位置高时，将常规(16倍采样)波特率提高为四倍。使能4xMode时，将2xMode位置低。更多信息请参见 *两倍和四倍速率模式* 部分。

第4位：2xMode

将2xMode位置高时，将常规(16倍采样)波特率提高为两倍。使能2xMode时，将4xMode位置低。更多信息请参见 *两倍和四倍速率模式* 部分。

第3至0位：FRACTx

FRACTx位为波特率发生器除数的分数部分。不使用时，将FRACTx设置为0000b。关于如何设置该值以选择波特率的计算方法，请参见 *分数波特率发生器* 部分。

波特率发生器除数LSB寄存器(DIVLSB)

地址:	0x1C							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Div7	Div6	Div5	Div4	Div3	Div2	Div1	Div0
复位	0	0	0	0	0	0	0	1

DIVLSB和DIVMSB定义波特率发生器的整数除数。DIVLSB的最小值为1。更多信息请参见 *分数波特率发生器* 部分。

第7至0位：Divx

Divx位是波特率发生器整数除数部分(DIV)的8个LSB。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

波特率发生器除数MSB寄存器(DIVMSB)

地址:	0x1D							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	Div15	Div14	Div13	Div12	Div11	Div10	Div9	Div8
复位	0	0	0	0	0	0	0	0

DIVLSB和DIVMSB定义波特率发生器的整数除数。DIVLSB的最小值为1。更多信息请参见分数波特率发生器部分。

第7至0位: Divx

Divx位是波特率发生器整数除数部分(DIV)的8个MSB。

时钟源寄存器(CLKSource)

地址:	0x1E							
模式:	R/W							
位	7	6	5	4	3	2	1	0
名称	CLKtoRTS	—	—	—	PLLBypass	PLLEn	CrystalEn	—
复位	0	0	0	1	1	0	0	0

第7位: CLKtoRTS

将CLKtoRTS位置高时, 将波特率发生器(16x波特率)输出时钟连接至RTS。在1倍、2倍和4倍速率模式下, RTS时钟频率分别为波特率的16倍、8倍或4倍。

第6、5、4、0位: 无功能

第3位: PLLBypass

将PLLBypass位置高时, 旁路内部PLL和预分频器。

第2位: PLLEn

将PLLEn位置高时, 使能内部PLL; 将PLLEn置低时, 禁止内部PLL。

第1位: CrystalEn

将CrystalEn位置高时, 使能晶振。使用XIN上的外部时钟源时, 将CrystalEn置低。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

串行控制器接口

可通过I²C或SPI控制器件，由SPI/I²C的逻辑状态定义。更多详细信息请参见焊球配置部分。

SPI接口

SPI支持单周期和突发读/写访问。SPI主机必须在SPI MODE0 (即时钟极性CPOL = 0, 时钟相位CPHA = 0)下产生时钟和数据信号。

SPI单周期访问

图15所示为单周期读操作，图16所示为单周期写操作。

SPI突发存取

突发存取允许通过在SPI命令字节中只定义初始寄存器地址将多个数据字节作为一个数据块进行读和写操作。使用THR (0x00)作为初始突发写地址，可将多个字符装载

至TxFIFO。类似地，使用RHR (0x00)作为SPI突发读地址，可读出Rx FIFO中的多个字符。如果SPI突发地址不是0x00，器件在每个SPI数据字节之后自动递增寄存器地址。因此，就能够高效率设置多个连续寄存器。整个周期期间，片选输入CS/A0必须保持为低电平。突发存取期间，SCLK/SCL时钟连续工作。SPI主机将CS/A0拉高时，突发周期结束。

例如，为了将128个字节写入至Tx FIFO，可采用以下步骤通过突发写操作实现：

- 1) 将CS/A0拉低。
- 2) 向地址0x00发送SPI写命令。
- 3) 发送128个字节。
- 4) 释放CS/A0。

以上过程总占用(1 + 128) × 8个时钟周期。

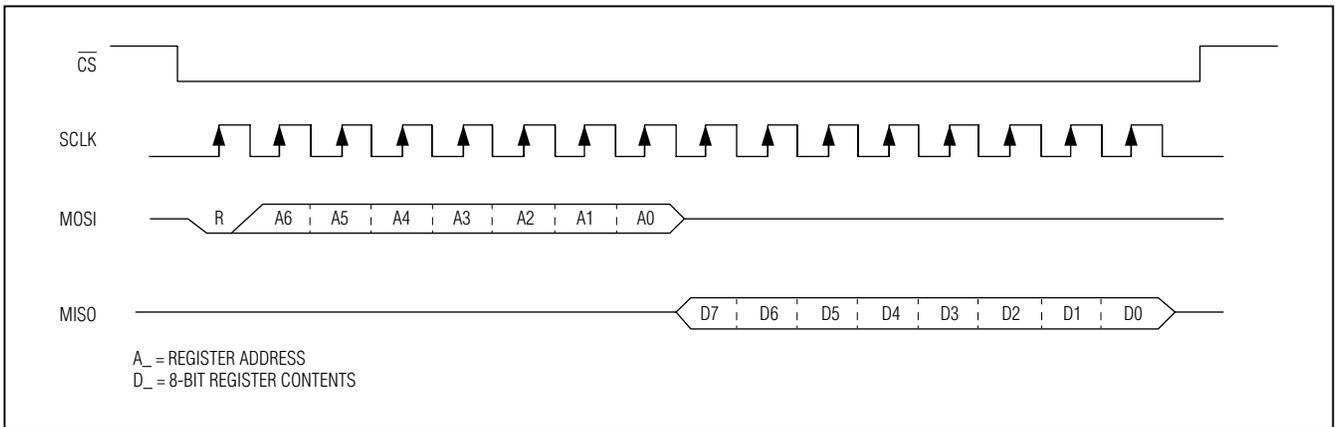


图15. 单周期读操作

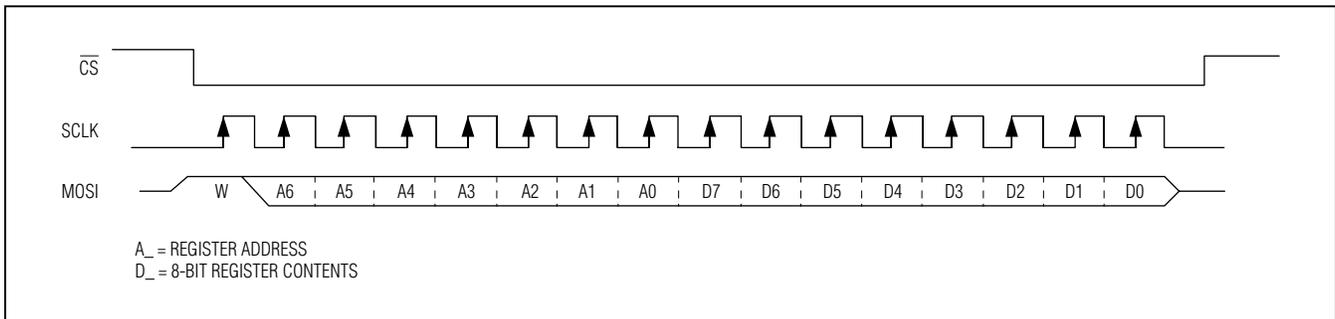


图16. 单周期写操作

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

I²C接口

器件包括I²C兼容接口，用于与主处理器进行数据通信(SCL和SDA)。接口支持高达1MHz时钟频率。SCL和SDA需要连接至正电源的上拉电阻。

表5. I²C地址映射

MOSI/A1	$\overline{CS}/A0$	I ² C WRITE ADDRESS	I ² C READ ADDRESS
DGND	DGND	0xD8	0xD9
DGND	V _L	0xC2	0xC3
DGND	SCL	0xC4	0xC5
DGND	SDA	0xC6	0xC7
V _L	DGND	0xC8	0xC9
V _L	V _L	0xCA	0xCB
V _L	SCL	0xCC	0xCD
V _L	SDA	0xCE	0xCF
SCL	DGND	0xD0	0xD1
SCL	V _L	0xD2	0xD3
SCL	SCL	0xD4	0xD5
SCL	SDA	0xD6	0xD7
SDA	DGND	0xC0	0xC1
SDA	V _L	0xDA	0xDB
SDA	SCL	0xDC	0xDD
SDA	SDA	0xDE	0xDF

START、STOP和重复START条件

利用I²C写器件时，主机发送START条件(S)，接着为器件的MAX3108A的I²C地址。地址之后，主机发送被编程寄存器的地址。然后主机通过发送STOP条件(P)释放对总线的控制，结束通信；或者发送重复START条件(Sr)，与另一个I²C从机通信(图2)。请参见图17。

从地址

器件包括可配置7位I²C从地址，允许最多16个器件共用I²C总线。通过将MOSI/A1和 $\overline{CS}/A0$ 输入连接至DGND、V_L、SCL或SDA定义地址(表5)。将R/ \overline{W} 位置高时，将器件配置为读模式；将R/ \overline{W} 位置低时，将器件配置为写模式。该地址是在START条件后发送到MAX3108A的第一个字节。

位传输

在每个SCL时钟周期的上升沿传输一个数据位。在SCL时钟脉冲为高电平期间，SDA数据必须保持稳定。当SCL为高电平且稳定时，SDA上的变化表示控制信号(参见START、STOP和重复START条件部分)。总线空闲时，SDA和SCL保持高电平。

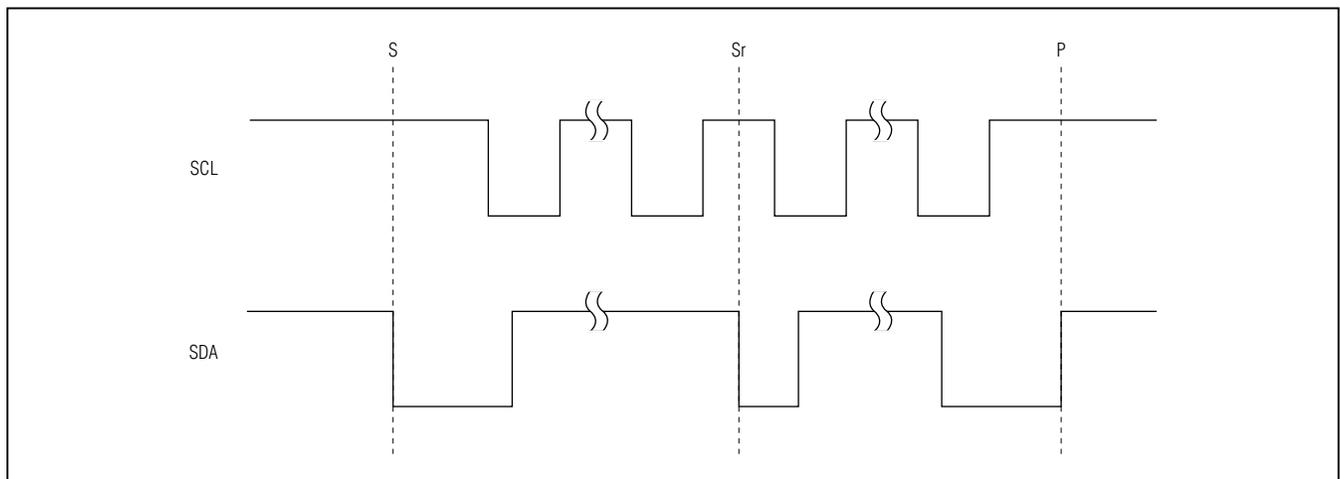


图17. I²C START、STOP和重复START条件

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

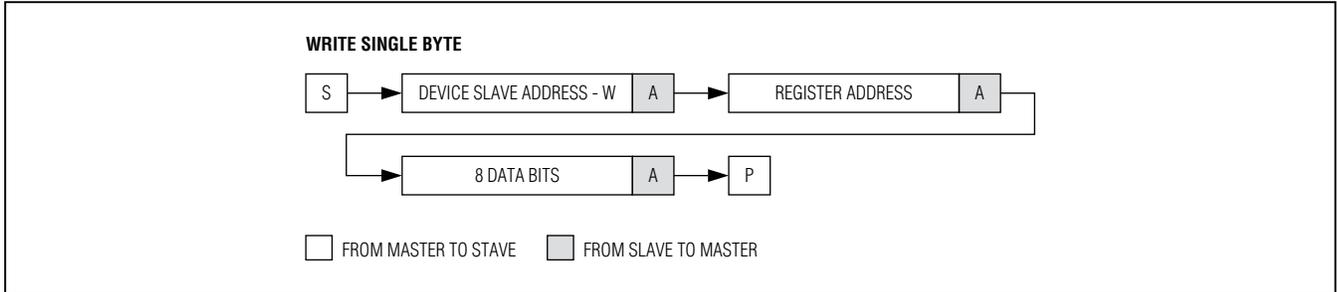


图18. 写字节序列

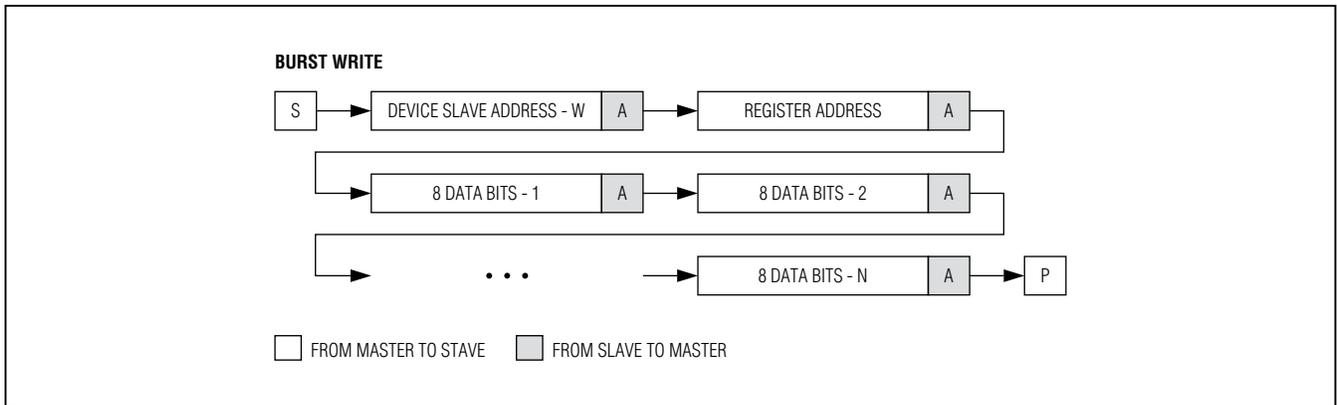


图19. 突发写序列

单字节写

该项操作期间，主机向从机发送地址和两个数据字节(图18)。单字节写操作的步骤如下：

- 1) 主机发送START条件。
- 2) 主机发送7位从地址加一个写标识位(逻辑低)。
- 3) 被寻址从机在数据线上发送ACK。
- 4) 主机发送8位寄存器地址。
- 5) 如果地址有效，从机在数据线上发送ACK (若地址无效，则发送NACK)。
- 6) 主机发送8个数据位。
- 7) 从机在数据线上发送ACK。
- 8) 主机产生STOP条件。

突发写

该项操作期间，主机向从机发送地址和多个数据字节(图19)。发送每个数据字节后，从器件自动递增寄存器地址，除非访问的寄存器是0x00，此时寄存器地址保持不变。突发写操作的步骤如下：

- 1) 主机发送START条件。
- 2) 主机发送7位从地址加一个写标识位(逻辑低)。
- 3) 被寻址从机在数据线上发送ACK。
- 4) 主机发送8位寄存器地址。
- 5) 如果地址有效，从机在数据线上发送ACK (若地址无效，则发送NACK)。
- 6) 主机发送8个数据位。
- 7) 从机在数据线上发送ACK。
- 8) 将步骤6和7重复N-1次。
- 9) 主机产生STOP条件。

带有128字FIFO的SPI/I2C UART, WLP封装

单字节读

该项操作期间, 主机发送地址和两个数据字节并从从器件接收一个数据字节(图20)。单字节读操作的步骤如下:

- 1) 主机发送START条件。
- 2) 主机发送7位从地址加一个写标识位(逻辑低)。
- 3) 被寻址从机在数据线上发送ACK。
- 4) 主机发送8位寄存器地址。
- 5) 如果地址有效, 从机在数据线上发送ACK (若地址无效, 则发送NACK)。
- 6) 主机发送重复START条件。
- 7) 主机发送7位从地址加一个读标识位(逻辑高)。
- 8) 被寻址从机在数据线上发送ACK。
- 9) 从机发送8个数据位。

10) 主机在数据线上发送NACK。

11) 主机产生STOP条件。

突发读

该项操作期间, 主机发送地址和两个数据字节并从从器件接收多个数据字节(图21)。发送每个数据字节后, 从器件自动递增寄存器地址, 除非访问的寄存器是0x00, 此时寄存器地址保持不变。突发读操作的步骤如下:

- 1) 主机发送START条件。
- 2) 主机发送7位从地址加一个写标识位(逻辑低)。
- 3) 被寻址从机在数据线上发送ACK。
- 4) 主机发送8位寄存器地址。
- 5) 如果地址有效, 从机在数据线上发送ACK (若地址无效, 则发送NACK)。

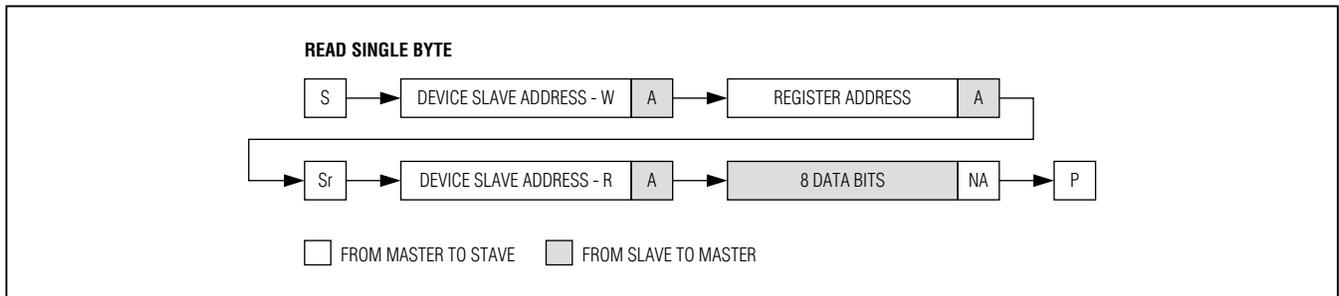


图20. 读字节序列

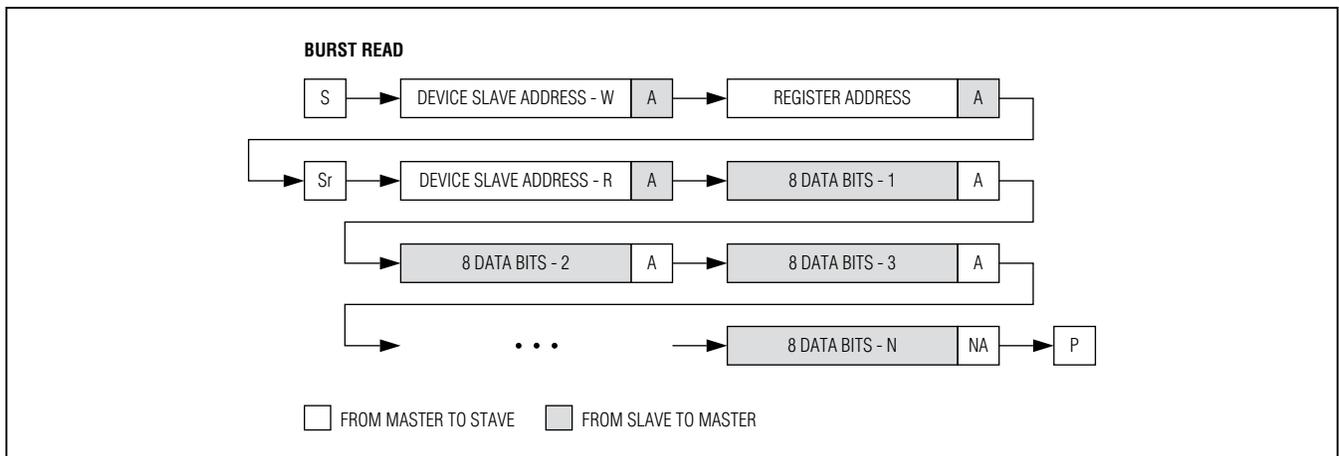


图21. 突发读序列

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

- 6) 主机发送重复START条件。
- 7) 主机发送7位从地址加一个读标识位(逻辑高)。
- 8) 从机在数据线上发送ACK。
- 9) 从机发送8个数据位。
- 10) 主机在数据线上发送ACK。
- 11) 将步骤9和10重复N-2次。
- 12) 从机发送8个数据位。
- 13) 主机在数据线上发送NACK。

- 14) 主机产生STOP条件。

应答位

数据传输由应答位(ACK)或非应答位(NACK)应答。主机和MAX3108A均产生ACK位。为产生ACK, 在第9个时钟脉冲的上升沿之前将SDA拉低, 并在第9个时钟脉冲的高电平期间保持为低电平(图22)。为产生NACK, 在第9个时钟脉冲的上升沿之前使SDA保持为高电平, 并在第9个时钟脉冲期间保持为高电平。监测NACK位可检测失败的数据传输。

应用信息

启动和初始化

上电、硬复位或软复位之后, 器件即可初始化, 如图23所示, 以检查器件在上电或复位后是否做好工作准备。

约200 μ s后, 器件即可工作; 连续读某个已知寄存器, 直到返回预期值。

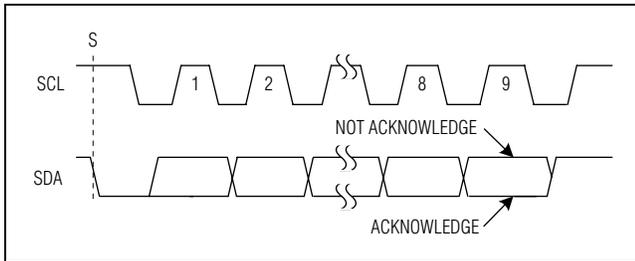


图22. 应答

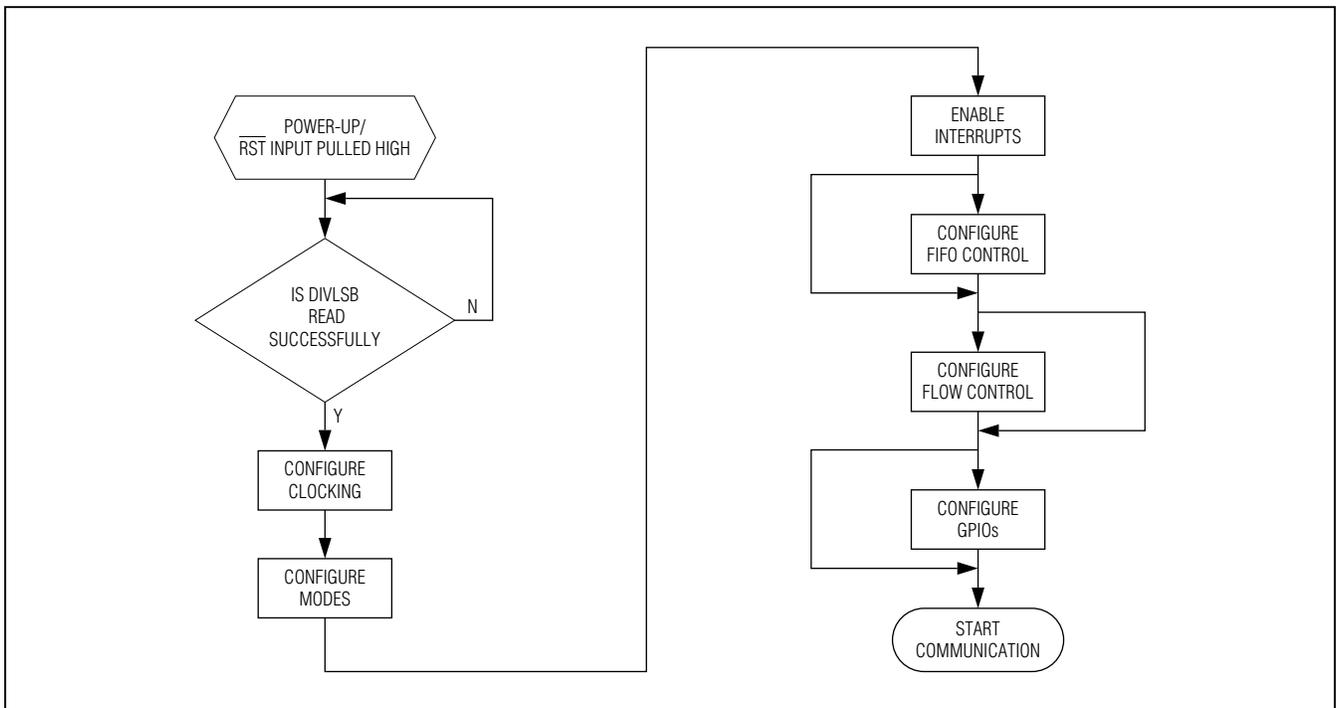


图23. 启动和初始化流程图

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

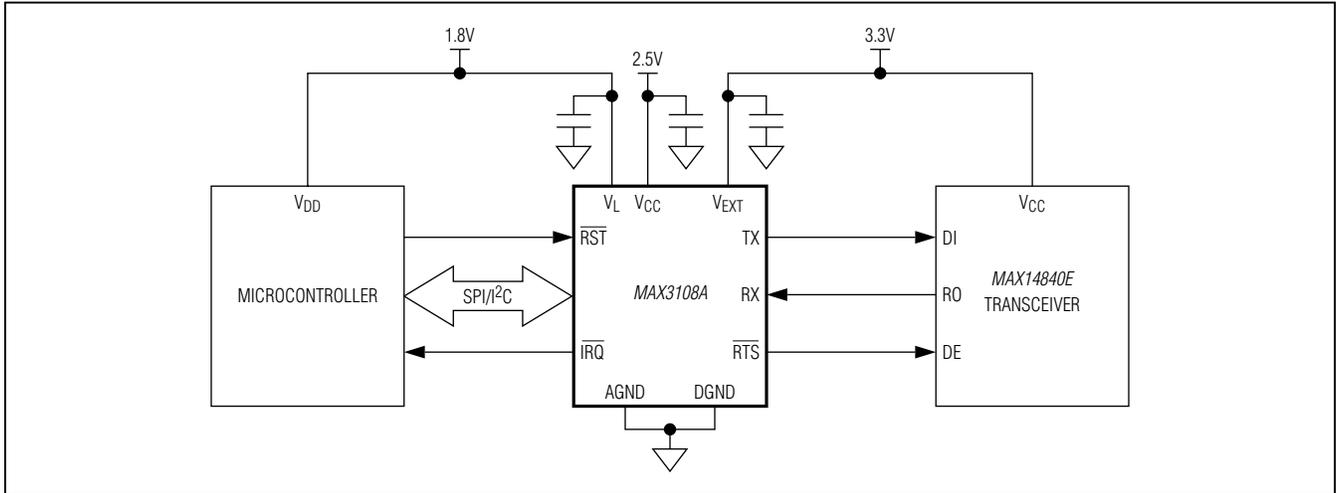


图24. 逻辑电平转换

低功耗工作

为降低常规工作期间的功耗，可采取以下方法：

- 不使用PLL，能节省此处所列选项的功耗。禁用和旁路PLL。使能PLL时，V_{CC}电源的电流为几mA（取决于时钟频率和乘数）；如果将PLL禁止，电流下降至1mA以下。
- 使用外部时钟源。使用外部时钟信号时，时钟模式的功耗最低，功耗下降至约为外部晶体时的一半。
- 使内部时钟速率尽量低。
- V_{CC}电源上使用低电压。
- 使用外部1.8V电源，减少内部1.8V线性稳压器的功耗，该电源为1.8V核电源。将外部1.8V电源连接至V₁₈，通过将LDOEN连接至DGND禁止内部稳压器。

中断和轮询

通过轮询ISR寄存器或通过监测 $\overline{\text{IRQ}}$ 输出，监测器件。轮询模式下，不使用 $\overline{\text{IRQ}}$ 物理中断输出，主控制器定期轮询ISR寄存器，以确定器件的状态。

也可在特定事件之后利用物理中断 $\overline{\text{IRQ}}$ 中断主控制器，非必须使用轮询。 $\overline{\text{IRQ}}$ 输出为开漏输出，需要利用上拉电阻拉至V_L。

逻辑电平转换

器件可直接连接至具有不同供电电压的收发器和控制器。V_L输入定义控制器接口的逻辑电压，V_{EXT}电压定义收发器接口的逻辑，确保选择控制器和收发器时的灵活性。图24所示为控制器、收发器及MAX3108A由不同电源供电的配置示例。

MAX3108A

带有128字FIFO的SPI/I²C UART, WLP封装

电源排序

器件能够以任意顺序上电。无论其它电源是否存在或电平如何，每个电源均可处于整个规定范围之内。向控制器和收发器接口发送输入信号之前，确保接口电源V_L和V_{EXT}处于工作范围之内。

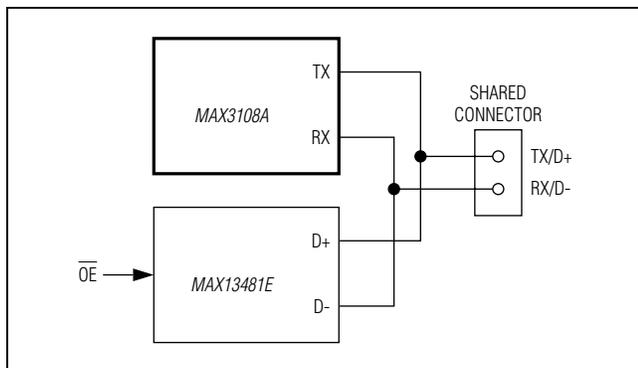


图25. 与USB收发器共用连接器

连接器共用

TX和 $\overline{\text{RTS}}$ 输出可设置为高阻态，该功能用于器件与其它通信器件共用连接器的情况。其它通信器件活动时，将器件的输出置于高阻态。将MODE1[2]: TXHiZ位置高时，TX为高阻态。将MODE1[3]: RTSHiZ位置高时，RTS为高阻态。图25所示为与USB收发器共用连接器的示例。

RS-232 5x3应用

四路GPIO可用于实现ITU V.24定义的其它流控制信号。图26所示为如何利用GPIO实现一些RS-232/V.28接口上的DSR、DTR、DCD和RI信号。

将FlowCtrl[1:0]位置高时，使能自动硬件RTS/CTS流控制。

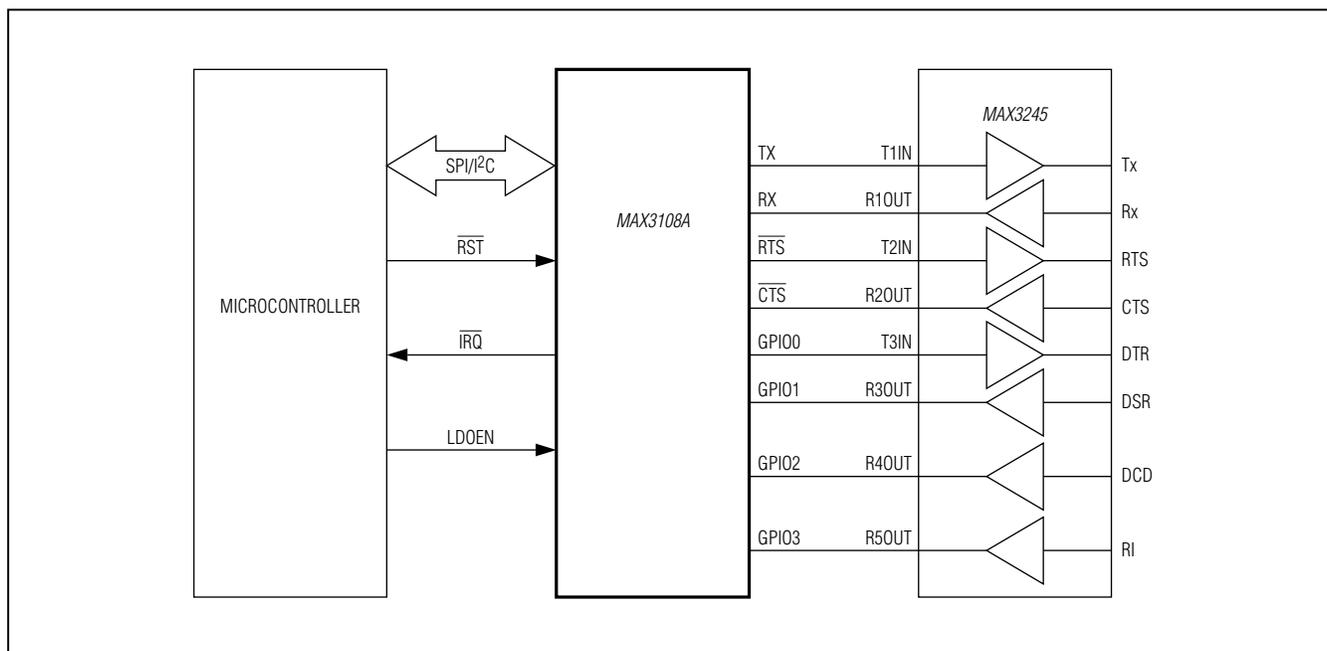


图26. RS-232应用

MAX3108A

带有128字FIFO的SPI/I2C UART, WLP封装

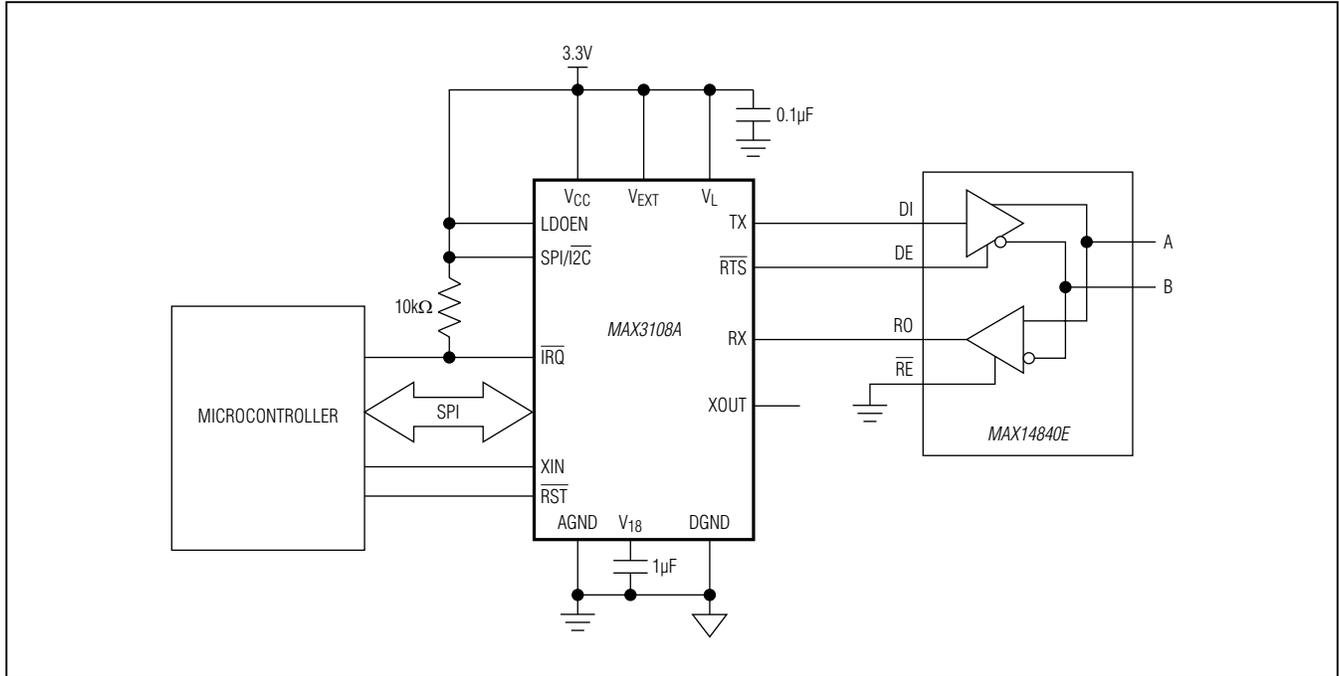


图27. RS-485半双工应用

典型应用电路

图27所示为器件在半双工RS-485中的应用。微控制器、RS-485收发器及器件由3.3V单电源供电。SPI用作控制器的通信接口。微控制器提供外部时钟源，为UART提供时钟。MAX14840E接收器总是使能，所以发生回波。通过将MODE2[7]: EchoSuprs位置高，使能MAX3108A的自动回波抑制。

通过将MODE1[4]: TranscvCtrl位置高使能自动收发器方向控制，以自动控制收发器的DE输入。

芯片信息

PROCESS: BiCMOS

封装信息

如需最近的封装外形信息和焊盘布局(占位面积)，请查询china.maximintegrated.com/packages。请注意，封装编码中的“+”、“#”或“-”仅表示RoHS状态。封装图中可能包含不同的尾缀字符，但封装图只与封装有关，与RoHS状态无关。

封装类型	封装编码	外形编号	焊盘布局编号
25 WLP	W252B2+1	21-0180	参考 应用笔记1891

MAX3108A

带有128字FIFO的SPI/I2C UART，WLP封装

修订历史

修订号	修订日期	说明	修改页
0	9/12	最初版本。	—

Maxim北京办事处

北京8328信箱 邮政编码100083

免费电话：800 810 0310

电话：010-6211 5199

传真：010-6211 5299



Maxim不对Maxim产品以外的任何电路使用负责，也不提供其专利许可。Maxim保留在任何时间、没有任何通报的前提下修改产品资料和规格的权利。电气特性表中列出的参数值(最小值和最大值)均经过设计验证，数据资料其它章节引用的参数值供设计人员参考。