Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

EVALUATION KIT AVAILABLE

MAX3107

SPI/I²C UART with 128-Word FIFOs

General Description

The MAX3107 is an advanced universal asynchronous receiver-transmitter (UART) with 128 words each of receive and transmit first-in/first-out (FIFO) that can be controlled through I²C or high-speed SPI™. The 2x and 4x rate modes allow a maximum of 24Mbps data rates. A phase-locked loop (PLL), prescaler, and fractional baud-rate generator allow for high-resolution baud-rate programming and minimize the dependency of baud rate on reference clock frequency.

Autosleep and shutdown modes help reduce power consumption during periods of inactivity. A low 640µA (typ) supply current and tiny 24-pin TQFN (3.5mm x 3.5mm) package make the MAX3107 ideal for low-power portable devices.

Integrated logic-level translation on the controller and transceiver (RX/TX and $\overline{RTS}$/$\overline{CTS}$) interfaces enable use with a wide selection of RS-232/RS-485 transceivers.

Automatic hardware and software flow control with selectable FIFO interrupt triggering offloads low-level activity from the host controller. Automatic half-duplex transceiver control with programmable setup and hold times allow the MAX3107 to be used in high-speed applications, for example Profibus-DP.

The MAX3107 is ideal for use in portable devices, industrial applications, and automotive applications. The MAX3107 is available in a 24-pin SSOP package and a 24-pin TQFN package. It is specified over the -40°C to +85°C extended ambient temperature range.

Applications

Portable Devices

Industrial Control Systems

Fieldbus Networks

Automotive Infotainment Systems

Medical Systems

Point-of-Sale Systems

HVAC or Building Control

Features

- **24-Pin, Lead-Free TQFN (3.5mm x 3.5mm) and 24-Pin, Lead-Free SSOP Packages**
- **24Mbps (max) Data Rate**
- **Integrated PLL and Divider**
- **Fractional Baud-Rate Generator**
- **SPI Up to 26MHz Clock Rate**
- **Auto Transceiver Direction Control**
- **Half-Duplex Echo Suppression**
- **Auto RTS/CTS and XON/XOFF Flow Control**
- **Special Character Detection**
- **GPIO-Based Character Detection**
- **9-Bit Multidrop-Mode Data Filtering**
- **SIR- and MIR-Compliant IrDA Encoder/Decoder**
- **+2.35V to +3.6V Supply Range**
- **Logic-Level Translation on the Controller and Transceiver Interfaces (Down to 1.7V)**
- **Four Flexible GPIOs**
- **Line Noise Indication**
- **Shutdown and Autosleep Modes**
- **Low 640µA (typ) Supply Current at 1Mbaud and 20MHz Clock**
- **Low 20µA (typ) Shutdown Power**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX3107EAG+T	-40°C to +85°C	24 SSOP
MAX3107ETG+T	-40°C to +85°C	24 TQFN-EP*
MAX3107ETG/V+T	-40°C to +85°C	24 TQFN-EP*

+Denotes a lead(Pb)-free/RoHS-compliant package.
**EP = Exposed pad.*
/V denotes an automotive qualified part.
T = Tape and reel.

Functional Diagram appears at end of data sheet.

SPI is a trademark of Motorola, Inc.

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

19-5014; Rev 3; 8/11

SPI/I²C UART with 128-Word FIFOs

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF REGISTERS

ABSOLUTE MAXIMUM RATINGS

(Voltages referenced to AGND.)

V_L, V_A, V_{EXT}, XIN ..... -0.3V to +4.0V
V_{18}, XOUT ..... -0.3V to (V_A + 0.3V)
$\overline{RST}$, $\overline{IRQ}$, DIN/A1, $\overline{CS}$/A0, SCLK/SCL, DOUT/SDA, LDOEN, $\overline{I^2C}$/SPI ..... -0.3V to (V_L + 0.3V)
TX, RX, $\overline{RTS}$/CLKOUT, $\overline{CTS}$, GPIO_..... -0.3V to (V_{EXT} + 0.3V)
DGND ..... -0.3V to +0.3V

Continuous Power Dissipation (T_A = +70°C)
TQFN (derate 15.4mW/°C above +70°C) 1229mW
SSOP (derate 12.3mW/°C above +70°C) 988mW
Operating Temperature Range ..... -40°C to +85°C
Junction Temperature ..... +150°C
Storage Temperature Range..... -65°C to +150°C
Lead Temperature (soldering, 10s) .....+300°C
Soldering Temperature (reflow) .....+260°C

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.

PACKAGE THERMAL CHARACTERISTICS (Note 1)

TQFN
Junction-to-Ambient Thermal Resistance (θ_{JA})..... 65°C/W
Junction-to-Case Thermal Resistance (θ_{JC})..... 15°C/W

SSOP
Junction-to-Ambient Thermal Resistance (θ_{JA}).....81°C/W
Junction-to-Case Thermal Resistance (θ_{JC})..... 32°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 **www.maxim-ic.com/thermal-tutorial**.

DC ELECTRICAL CHARACTERISTICS

(V_A = +2.35V 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_A = +2.8V, V_L = +1.8V, V_{EXT} = +2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Digital Interface Supply Voltage	V_L		1.71		3.6	V
Analog Supply Voltage	V_A		2.35		3.6	V
UART Interface Logic Supply Voltage	V_{EXT}		1.71		3.6	V
Logic Supply Voltage	V_{18}		1.65	1.80	1.95	V
CURRENT CONSUMPTION						
V_A Supply Current	I_A	1.8MHz crystal oscillator active, PLL disabled, V_{LDOEN} = V_L, SPI/I²C interface idle		220	500	µA
		Baud rate = 1Mbps, external clock, SPI frequency is 8MHz, external loopback PLL disabled, V_{LDOEN} = V_L (Note 3)		0.65	1.3	mA
V_A Shutdown Supply Current	$I_{A, SHDN}$	Shutdown mode, V_{LDOEN} = 0V, $V_{\overline{RST}}$ = 0V, all inputs and outputs are idle		20	35	µA
V_A Sleep Supply Current	$I_{A, SLEEP}$	Sleep mode, V_{LDOEN} = V_L, $V_{\overline{RST}}$ = V_L, all inputs and outputs are idle		45	100	µA
V_L Supply Current	I_L	All logic inputs are at V_L or V_{EXT} or 0V		4	15	µA
V_{EXT} Supply Current	I_{EXT}	All logic inputs are at V_L or V_{EXT} or 0V		5	10	µA
V_{18} Input Power-Supply Current in Shutdown Mode	I_{18SHDN}	V_{LDOEN} = 0V (V_{18} is powered by an external 1.85V voltage source), static power consumption		7	50	µA

DC ELECTRICAL CHARACTERISTICS (continued)

(V_A = +2.35V 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_A = +2.8V, V_L = +1.8V, V_{EXT} = +2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCLK/SCL, DOUT/SDA						
DOUT/SDA Output Low Voltage in I²C Mode	$V_{OL,I2C}$	I_{LOAD} = -3mA, V_L > 2V			0.4	V
		I_{LOAD} = -3mA, V_L < 2V			0.2 x V_L	V
DOUT/SDA Output Low Voltage in SPI Mode	$V_{OL,SPI}$	I_{LOAD} = -2mA			0.4	V
DOUT/SDA Output High Voltage in SPI Mode	$V_{OH,SPI}$	I_{LOAD} = 2mA			V_L - 0.4	V
Input Low Voltage	V_{IL}	SPI and I²C mode			0.3 x V_L	V
Input High Voltage	V_{IH}	SPI and I²C mode	0.7 x V_L			V
Input Hysteresis	V_{HYST}	SPI and I²C mode		0.05 x V_L		V
Input Leakage Current	I_{IL}	V_{IN} = 0 to V_L, SPI and I²C mode	-1		+1	µA
Input Capacitance	$C_{IN\_I2C\_SPI}$	SPI and I²C mode		5		pF
$\overline{I2C}$/SPI, $\overline{CS}$/A0, DIN/A1 INPUTS						
Input Low Voltage	V_{IL}	SPI and I²C mode			0.3 x V_L	V
Input High Voltage	V_{IH}	SPI and I²C mode	0.7 x V_L			V
Input Hysteresis	V_{HYST}	SPI and I²C mode		50		mV
Input Leakage Current	I_{IL}	V_{IN} = 0 to V_L, SPI and I²C mode	-1		+1	µA
Input Capacitance	$C_{IN\_I2C\_SPI}$	SPI and I²C mode		5		pF
$\overline{IRQ}$ OUTPUT (OPEN DRAIN)						
Output Low Voltage	V_{OL}	I_{LOAD} = -2mA			0.4	V
Output Leakage	I_{LK}	$V_{\overline{IRQ}}$ = 0 to V_L, $\overline{IRQ}$ is not asserted	-1		+1	µA
LDOEN AND $\overline{RST}$ INPUTS						
Input Low Voltage	V_{IL}				0.3 x V_L	V
Input High Voltage	V_{IH}		0.7 x V_L			V
Input Hysteresis	V_{HYST}			50		mV
Input Leakage Current	I_{IN}	V_{IN} = 0 to V_L	-1		+1	µA
$\overline{RTS}$/CLKOUT AND TX OUTPUTS						
Output Low Voltage	V_{OL}	I_{LOAD} = -2mA			0.4	V
Output High Voltage	V_{OH}	I_{LOAD} = 2mA	V_{EXT} - 0.4			V
Input Leakage Current	I_{IN}	Output three-stated, V_{IN} = 0 to V_{EXT}	-1		+1	µA
Input Capacitance	C_{IN_IRSTB}	High-Z mode		5		pF
RX, $\overline{CTS}$ INPUTS						
Input Low Voltage	V_{IL}				0.3 x V_{EXT}	V
Input High Voltage	V_{IH}		0.7 x V_{EXT}			V
Input Hysteresis	V_{HYST}			50		mV
$\overline{CTS}$ Input Leakage Current	$I_{IN\_\overline{CTS}}$	V_{IN} = 0 to V_{EXT}	-1		+1	µA
RX Pullup Current	I_{IN_RX}	V_{IN} = 0V	0.3	1.5	3	µA
Input Capacitance	C_{IN_IUART}			5		pF
GPIO_ OUTPUTS AND INPUTS						
Output Low Voltage	V_{OL}	I_{LOAD} = -2mA, push-pull or open drain			0.4	V
Output High Voltage	V_{OH}	I_{LOAD} = 2mA, push-pull	V_{EXT} - 0.4			V

DC ELECTRICAL CHARACTERISTICS (continued)

(V_A = +2.35V 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_A = +2.8V, V_L = +1.8V, V_{EXT} = +2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Low Voltage	V_{IL}	Configured as an input			0.4	V
Input High Voltage	V_{IH}	Configured as an input	2/3 x V_{EXT}			V
Pulldown Current	I_{PD}	GPIO_ = V_{EXT}	0.25	1	2.5	µA
Input Capacitance	C_{IN_IUART}	Configured as an input		5		pF
XIN						
Input Low Voltage	V_{IL}				0.3	V
Input High Voltage	V_{IH}		1.2		V_A	V
Input Capacitance	C_{XI}			16		pF
XOUT						
Input Capacitance	C_{XO}			16		pF

AC ELECTRICAL CHARACTERISTICS

(V_A = +2.35V 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_A = +2.8V, V_L = +1.8V, V_{EXT} = +2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
UART CLOCKING						
External Crystal Frequency	f_{XOSC}		1		4	MHz
External Clock Frequency	f_{CLK}		0.5		35	MHz
External Clock Duty Cycle		(Note 3)	45		55	%
Baud-Rate Generator Clock Input	f_{REF}	(Note 3)			96	MHz
I²C BUS: TIMING CHARACTERISTICS (see Figure 1)						
SCL Clock Frequency	f_{SCL}	Standard mode			100	kHz
		Fast mode			400	
Bus Free Time Between a STOP (P) and START (S) Condition	t_{BUF}	Standard mode	4.7			µs
		Fast mode	1.3			
Hold Time for START (S) Condition and Repeated START (Sr) Condition (Note 3)	$t_{HD:STA}$	Standard mode	4.0			µs
		Fast mode	0.6			
Low Period of the SCL Clock	t_{LOW}	Standard mode	4.7			µs
		Fast mode	1.3			
High Period of the SCL Clock	t_{HIGH}	Standard mode	4.0			µs
		Fast mode	0.6			
Data Hold Time	$t_{HD:DAT}$	Standard mode	0		0.9	µs
		Fast mode	0		0.9	

AC ELECTRICAL CHARACTERISTICS (continued)

(V_A = +2.35V 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_A = +2.8V, V_L = +1.8V, V_{EXT} = +2.5V, T_A = +25°C.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Data Setup Time	$t_{SU:DAT}$	Standard mode	250			ns
		Fast mode	100			
Setup Time for Repeated START (Sr) Condition	$t_{SU:STA}$	Standard mode	4.7			µs
		Fast mode	0.6			
Rise Time of SDA and SCL Signals Receiving	t_R	Standard mode (0.3 x V_L to 0.7 x V_L) (Note 5)	20 + 0.1C_B		1000	ns
		Fast mode (0.3 x V_L to 0.7 x V_L) (Note 5)	20 + 0.1C_B		300	
Fall Time of SDA and SCL Signals	t_F	Standard mode (0.7 x V_L to 0.3 x V_L) (Note 5)	20 + 0.1C_B		300	ns
		Fast mode (0.7 x V_L to 0.3 x V_L) (Note 5)	20 + 0.1C_B		300	
Setup Time for STOP (P) Condition	$t_{SU:STO}$	Standard mode	4.7			µs
		Fast mode	0.6			
Capacitive Load for SDA and SCL (Note 3)	C_B	Standard mode			400	pF
		Fast mode			400	
I/O Capacitance (SCL, SDA)	$C_{I/O}$				10	pF
Pulse Width of Spike Suppressed	t_{SP}				50	ns
SPI BUS: TIMING CHARACTERISTICS (see Figure 2)						
SCLK Clock Period	t_{CH+CL}		38.4			ns
SCLK Pulse-Width High	t_{CH}		16			ns
SCLK Pulse-Width Low	t_{CL}		16			ns
$\overline{CS}$ Fall to SCLK Rise Time	t_{CSS}		0			ns
DIN Hold Time	t_{DH}		3			ns
DIN Setup Time	t_{DS}		5			ns
Output Data Propagation Delay	t_{DO}				20	ns
DOUT Rise and Fall Times	t_{FT}				10	ns
$\overline{CS}$ Hold Time	t_{CSH}		32			ns

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

Note 3: Not production tested. Guaranteed by design.

Note 4: When V_{18} is powered by an external voltage regulator, the external power supply must have current capability above or equal to I_{18}.

Note 5: C_B is the total capacitance of either the clock or data line of the synchronous bus in pF.

SPI/I²C UART with 128-Word FIFOs

Test Circuits/Timing Diagrams

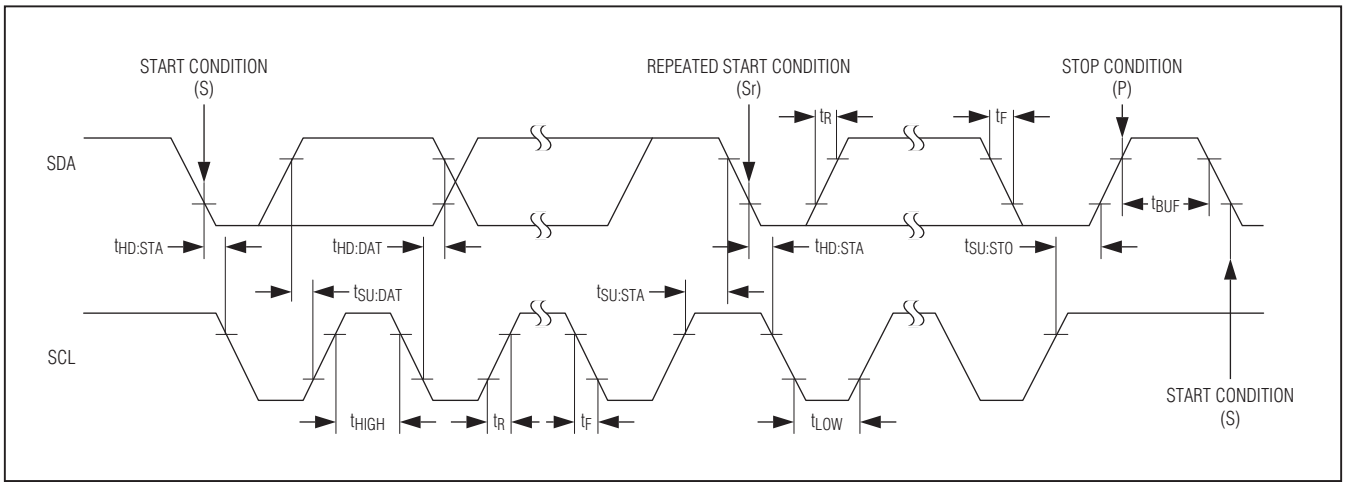


Figure 1. I²C Timing Diagram

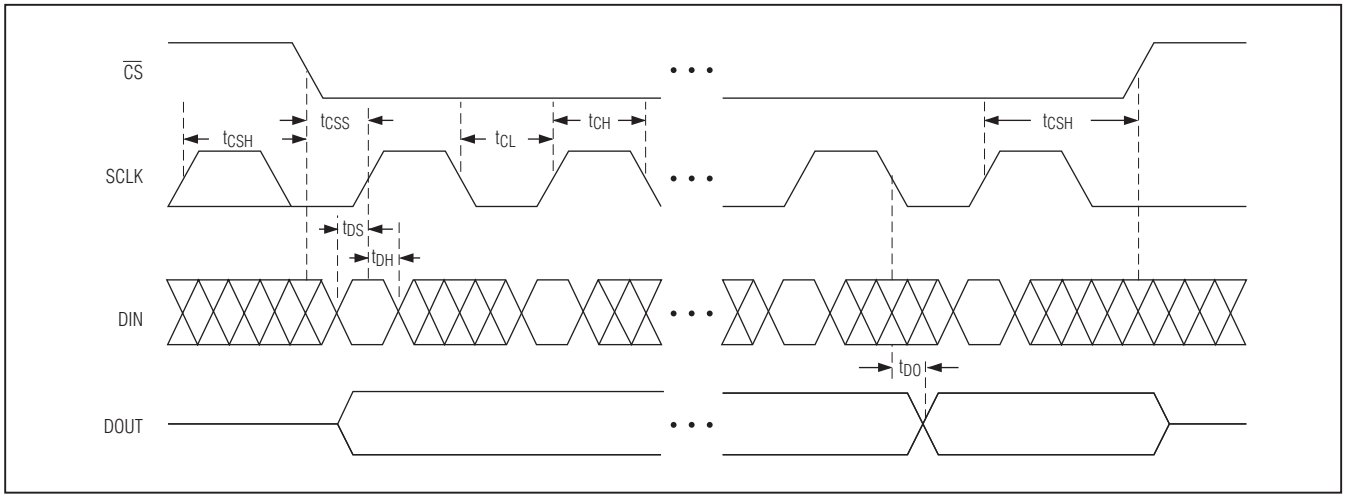
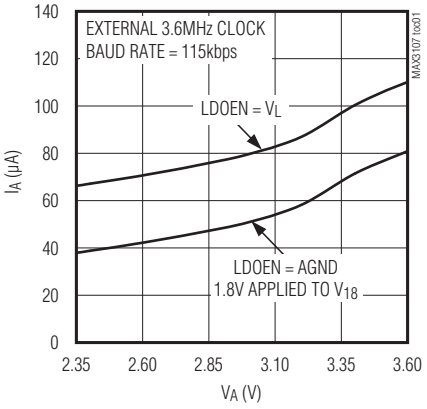


Figure 2. SPI Timing Diagram

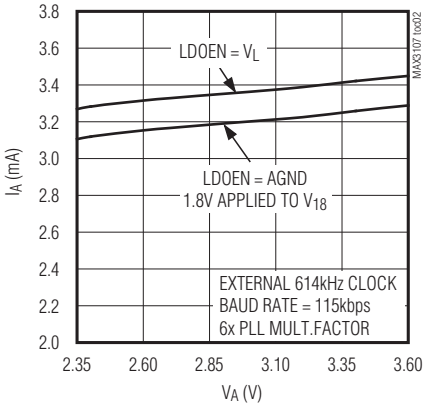
Typical Operating Characteristics

(V_A = 2.5V, V_L = 2.5V, V_{EXT} = 2.5V, LDOEN = VL, T_A = +25°C, unless otherwise noted.)

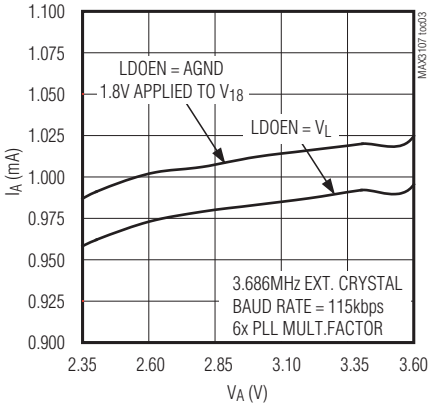
I_A SUPPLY CURRENT vs. V_A VOLTAGE (EXTERNAL CLOCK, PLL DISABLED)



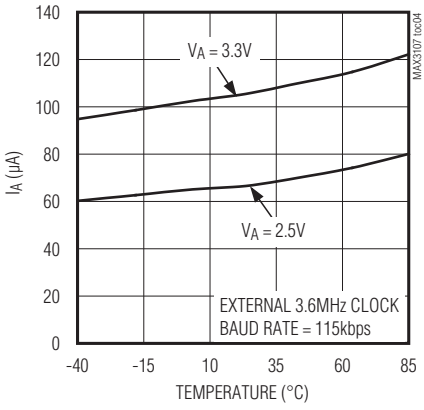
I_A SUPPLY CURRENT vs. V_A VOLTAGE (EXTERNAL CLOCK, PLL ENABLED)



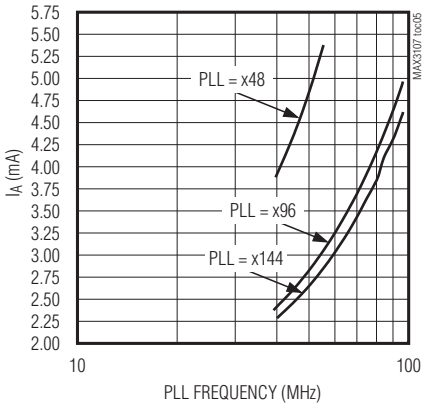
I_A SUPPLY CURRENT vs. V_A VOLTAGE (EXTERNAL CRYSTAL, PLL ENABLED)



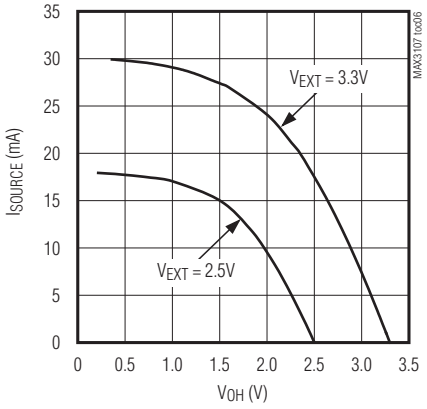
I_A SUPPLY CURRENT vs. TEMPERATURE



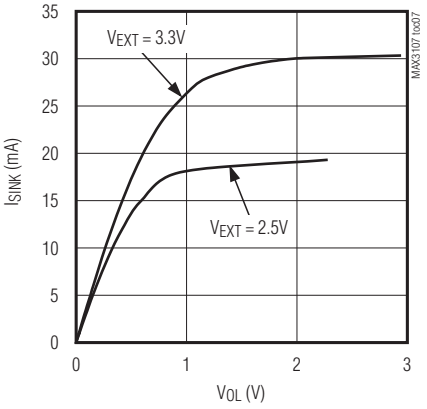
I_A SUPPLY CURRENT vs. PLL FREQUENCY



GPIO_ OUTPUT HIGH VOLTAGE vs. SOURCE CURRENT (PUSH-PULL)

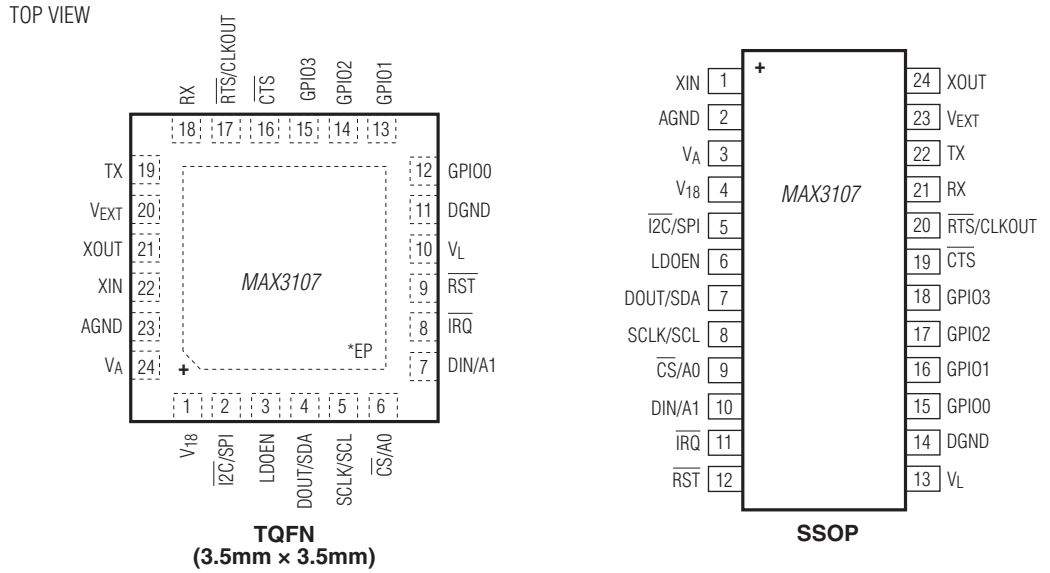


GPIO_ OUTPUT LOW VOLTAGE vs. SINK CURRENT (OPEN DRAIN)



SPI/I²C UART with 128-Word FIFOs

Pin Configurations



Pin Descriptions

PIN		NAME	FUNCTION
TQFN-EP	SSOP		
1	4	V_{18}	Internal 1.8V LDO Output and 1.8V Logic Supply Input. Bypass V_{18} with a 1µF ceramic capacitor to DGND. Keep V18 powered in shutdown mode.
2	5	$\overline{I2C}$/SPI	SPI or Active-Low I²C Selector Input. Drive $\overline{I2C}$/SPI high to enable SPI. Drive $\overline{I2C}$/SPI low to enable I²C.
3	6	LDOEN	LDO Enable Input. Drive LDOEN high to enable the internal 1.8V LDO. Drive LDOEN low to disable the internal LDO. Power V18 with an external 1.8V supply when LDOEN is low.
4	7	DOUT/SDA	Serial-Data Output. When $\overline{I2C}$/SPI is high, DOUT/SDA functions as the DOUT SPI serial-data output. When $\overline{I2C}$/SPI is low, DOUT/SDA functions as the SDA I²C serial-data input/output.
5	8	SCLK/SCL	Serial-Clock Input. When $\overline{I2C}$/SPI is high, SCLK/SCL functions as the SCLK SPI serial-clock input (up to 26MHz). When $\overline{I2C}$/SPI is low, SCLK/SCL functions as the SCL I²C serial-clock input (up to 400kHz).
6	9	$\overline{CS}$/A0	Active-Low Chip-Select and Address 0 Input. When $\overline{I2C}$/SPI is high, $\overline{CS}$/A0 functions as the $\overline{CS}$ SPI active-low chip select. When $\overline{I2C}$/SPI is low, $\overline{CS}$/A0 functions as the A0 I²C device address programming input. Connect $\overline{CS}$/A0 to DGND or V_L.

Pin Descriptions (continued)

PIN		NAME	FUNCTION
TQFN-EP	SSOP		
7	10	DIN/A1	Serial-Data and Address 1 Input. When $\overline{I2C}$/SPI is high, DIN/A1 functions as the DIN SPI serial-data input. When $\overline{I2C}$/SPI is low, DIN/A1 functions as the A1 I²C device address programming input and connects to DIN/A1 DGND or V_L.
8	11	$\overline{IRQ}$	Active-Low Interrupt Open-Drain Output. $\overline{IRQ}$ is asserted when an interrupt is pending.
9	12	$\overline{RST}$	Active-Low Reset Input. Drive $\overline{RST}$ low to force the UART into hardware reset mode. In hardware reset mode, the oscillator and the internal PLL are shut down; there is no clock activity.
10	13	V_L	Digital Interface Logic-Level Supply. V_L powers the internal logic-level translators for $\overline{RST}$, $\overline{IRQ}$, DIN/A1, $\overline{CS}$/A0, SCLK/SCL, DOUT/SDA, LDOEN, and $\overline{I2C}$/SPI. Bypass V_L with a 0.1µF ceramic capacitor to DGND. V_L must be powered in all modes.
11	14	DGND	Digital Ground
12	15	GPIO0	General-Purpose Input/Output 0. GPIO0 is user programmable as an input or output (push-pull or open drain). GPIO0 has a weak pulldown resistor to ground.
13	16	GPIO1	General-Purpose Input/Output 1. GPIO1 is user programmable as an input or output (push-pull or open drain). GPIO1 has a weak pulldown resistor to ground.
14	17	GPIO2	General-Purpose Input/Output 2. GPIO2 is user programmable as an input or output (push-pull or open drain). GPIO2 has a weak pulldown resistor to ground.
15	18	GPIO3	General-Purpose Input/Output 3. GPIO3 is user programmable as an input or output (push-pull or open drain). GPIO3 has a weak pulldown resistor to ground.
16	19	$\overline{CTS}$	Active-Low Clear-to-Send Input. $\overline{CTS}$ is a flow-control input.
17	20	$\overline{RTS}$/CLKOUT	Active-Low Request-to-Send Output. $\overline{RTS}$/CLKOUT can be set high or low by programming bit 7 ($\overline{RTS}$) of the LCR register.
18	21	RX	Receive Input. Serial UART data input. RX has an internal weak pullup resistor to V_{EXT}.
19	22	TX	Transmit Output. Serial UART data output.
20	23	V_{EXT}	Transceiver Interface Level Supply. V_{EXT} powers the internal logic-level translators for RX, TX, $\overline{RTS}$, $\overline{CTS}$, and GPIO_. Bypass V_{EXT} with a 0.1µF ceramic capacitor to DGND.
21	24	XOUT	Crystal Output. When using an external crystal, connect one end of the crystal to XOUT and the other to XIN. When using an external clock source, leave XOUT unconnected.
22	1	XIN	Crystal/Clock Input. When using an external crystal, connect one end of the crystal to XIN and the other one to XOUT. When using an external clock source, drive XIN with the external clock.
23	2	AGND	Analog Ground
24	3	V_A	Analog Supply. V_A powers the PLL and internal LDO. Bypass V_A with a 0.1µF ceramic capacitor to AGND.
—	—	EP	Exposed Paddle. Connect EP to AGND. EP is not intended as an electrical connection point. Only for TQFN-EP package.

SPI/I²C UART with 128-Word FIFOs

Register Map

(All default reset values are 0x00, unless otherwise noted. All registers are R/W, unless otherwise noted.)

REGISTER	ADDR	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
FIFO DATA									
RHR†*	0x00	RData7	RData6	RData5	RData4	RData3	RData2	RData1	RData0
THR†	0x00	TData7	TData6	TData5	TData4	TData3	TData2	TData1	TData0
INTERRUPTS									
IRQEn	0x01	CTSIEn	RxEmtyIEn	TxEmtyIEn	TxTrgIEn	RxTrgIEn	STSIEn	SpclChrIEn	LSRErrIEn
ISR*†	0x02	CTSInt	RxEmptyInt	TxEmptyInt	TFifoTrigInt	RFifoTrigInt	STSInt	SpCharInt	LSRErrInt
LSRIntEn	0x03	—	—	NoiseIntEn	RBreakIEn	FrameErrIEn	ParityIEn	ROverrIEn	RTimoutIEn
LSR*†	0x04	$\overline{\text{CTS}}$bit	—	RxNoise	RxBreak	FrameErr	RxParityErr	RxOverrun	RTimeout
SpclChrIntEn	0x05	—	—	MltDrpIntEn	BREAKIntEn	XOFF2IntEn	XOFF1IntEn	XON2IntEn	XON1IntEn
SpclCharInt †	0x06	—	—	MultiDropInt	BREAKInt	XOFF2Int	XOFF1Int	XON2Int	XON1Int
STSIntEn	0x07	—	SleepIntEn	ClkRdyIntEn	—	GPI3IntEn	GPI2IntEn	GPI1IntEn	GPI0IntEn
STSInt*†	0x08	—	SleepInt	ClockReady	—	GPI3Int	GPI2Int	GPI1Int	GPI0Int
UART MODES									
MODE1	0x09	IRQSel	AutoSleep	ForcedSleep	TrnscvCtrl	RTSHiZ	TXHiZ	TxDisabl	RxDisabl
MODE2	0x0A	EchoSuprs	MultiDrop	Loopback	SpecialChr	RxEmtyInv	RxTrigInv	FIFORst	RST
LCR*	0x0B	$\overline{\text{RTS}}$	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
FIFO CONTROL									
FlowLvl	0x0F	Resume3	Resume2	Resume1	Resume0	Halt3	Halt2	Halt1	Halt0
FIFOTrgLvl*	0x10	RxTrig3	RxTrig2	RxTrig1	RxTrig0	TxTrig3	TxTrig2	TxTrig1	TxTrig0
TxFIFOLvl†	0x11	TxFL7	TxFL6	TxFL5	TxFL4	TxFL3	TxFL2	TxFL1	TxFL0
RxFIFOLvl†	0x12	RxFL7	RxFL6	RxFL5	RxFL4	RxFL3	RxFL2	RxFL1	RxFL0
FLOW CONTROL									
FlowCtrl	0x13	SwFlow3	SwFlow2	SwFlow1	SwFlow0	SwFlowEn	GPIAddr	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									
GPIOConfg	0x18	GP3OD	GP2OD	GP1OD	GP0OD	GP3Out	GP2Out	GP1Out	GP0Out
GPIOData	0x19	GPI3Dat	GPI2Dat	GPI1Dat	GPI0Dat	GPO3Dat	GPO2Dat	GPO1Dat	GPO0Dat
CLOCK CONFIGURATION									
PLLConfig*	0x1A	PLLFactor1	PLLFactor0	PreDiv5	PreDiv4	PreDiv3	PreDiv2	PreDiv1	PreDiv0
BRGConfig	0x1B	—	—	4xMode	2xMode	FRACT3	FRACT2	FRACT1	FRACT0
DIVLSB	0x1C	Div7	Div6	Div5	Div4	Div3	Div2	Div1	Div0
DIVMSB	0x1D	Div15	Div14	Div13	Div12	Div11	Div10	Div9	Div8
CLKSource*	0x1E	CLKtoRTS	—	—-	ClockEn	PLLBypass	PLLEn	CrystalEn	—
REVISION									
RevID*†	0x1F	1	0	1	0	0	0	0	1

**Denotes nonzero default reset value: ISR = 0x60, LCR = 0x05, FIFOTrgLvl = 0xFF, PLLConfig = 0x01, DIVLSB = 0x01, CLKSource = 0x08, RevID = 0xA1.*

†Denotes nonread/write value: RHR = R, THR = W, ISR = COR, SpclCharInt = COR, STSInt = R/COR, LSR = R, TxFIFOLvl = R, RxFIFOLvl = R, RevID = R.

Detailed Description

The MAX3107 UART is a bridge between an SPI/MICROWIRE™ or I²C microprocessor bus and an asynchronous serial-data communication link, such as RS-485, RS-232, or IrDA. The MAX3107 contains an advanced UART, a fractional baud-rate generator, and four GPIOs. The MAX3107 is configured and monitored, and data is written and read from 8-bit registers through SPI or I²C. These registers are organized by related function as shown in the *Register Map*.

The host controller loads data into the Transmit Holding register (THR) through SPI or I²C. This data is automatically pushed into the transmit FIFO and sent out at TX. The MAX3107 adds START, STOP, and parity bits to the data and sends the data out at the selected baud rate. The clock configuration registers determine the baud rate, clock source selection, and clock frequency prescaling.

The receiver in the MAX3107 detects a START bit as a high-to-low RX transition. An internal clock samples this data. The received data is automatically placed in the receive FIFO and can then be read out of the RxFIFO through the RHR.

Register Set

The MAX3107 has a flat register structure without shadow registers. The registers are 8 bits wide. The MAX3107 registers have some similarities to the 16C550 registers.

Receive and Transmit FIFOs

The UART's receiver and the transmitter each have a 128-word deep FIFO, reducing the intervals that the host processor needs to dedicate for high-speed, high-volume data transfer. As the data rates of the asynchronous RX, TX interfaces increase and get closer to those of the host controller's SPI/I²C data rates, UART management and flow control can make up a significant portion of the host's activity. By increasing FIFO size, the host is interrupted less often and can utilize SPI/I²C burst data block transfers to/from the FIFOs.

FIFO trigger levels can generate interrupts to the host controller, signaling that programmed FIFO fill levels have been reached. The transmitter and receiver trigger levels are programmed through FIFOTrgLvl with a resolution of eight FIFO locations. When a receive FIFO trigger is generated, the host knows that the receive FIFO has a defined number of words waiting to be read out or that a known number of vacant FIFO locations are available and ready to be filled. The transmit FIFO trigger generates an interrupt when the transmit FIFO level is above the programmed trigger level. The host then knows to throttle data writing to the transmit FIFO.

The host can read out the number of words present in each of the FIFOs through the TxFIFOLvl and RxFIFOLvl registers.

Transmitter Operation

Figure 3 shows the structure of the transmitter with the TxFIFO. The transmit FIFO can hold up to 128 words that are written to it through THR.

The current number of words in the TxFIFO can be read out through the TxFIFOLvl register. The transmit FIFO can be programmed to generate an interrupt when a programmed number of words are present in the TxFIFO through the FIFOTrgLvl register. The TxFIFO interrupt trigger level is selectable through FIFOTrgLvl[3:0]. When the transmit FIFO fill level reaches the programmed trigger level, the ISR[4] interrupt is set.

The transmit FIFO is empty when ISR[5]: TxEmtyInt is set. ISR[5] turns high when the transmitter starts transmitting the last word in the TxFIFO. Hence, the transmitter is completely empty after ISR[5] is set with an additional delay equal to the length of a complete character (including START, parity, and STOP bits).

The contents of the TxFIFO and RxFIFOs are both cleared through MODE2[1]: FIFORst.

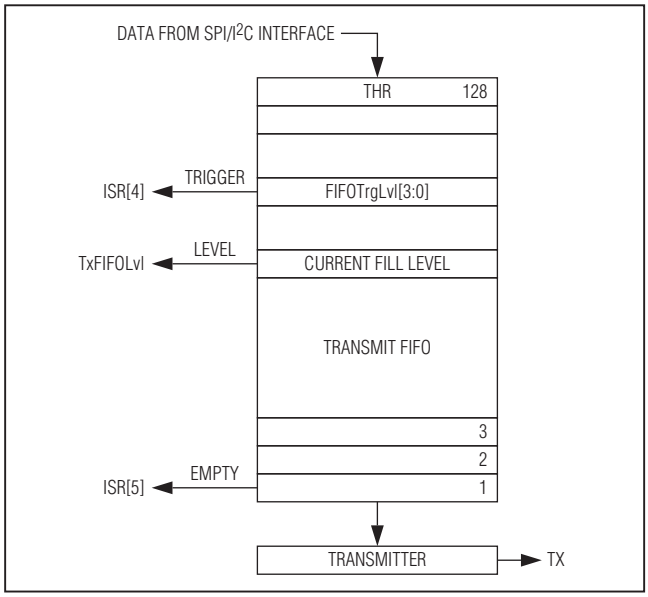


Figure 3. Transmit FIFO Signals

MICROWIRE is a trademark of National Semiconductor Corp.

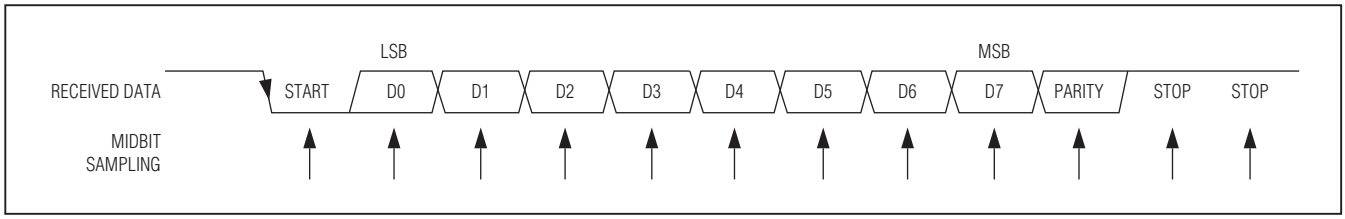


Figure 4. Receive Data Format

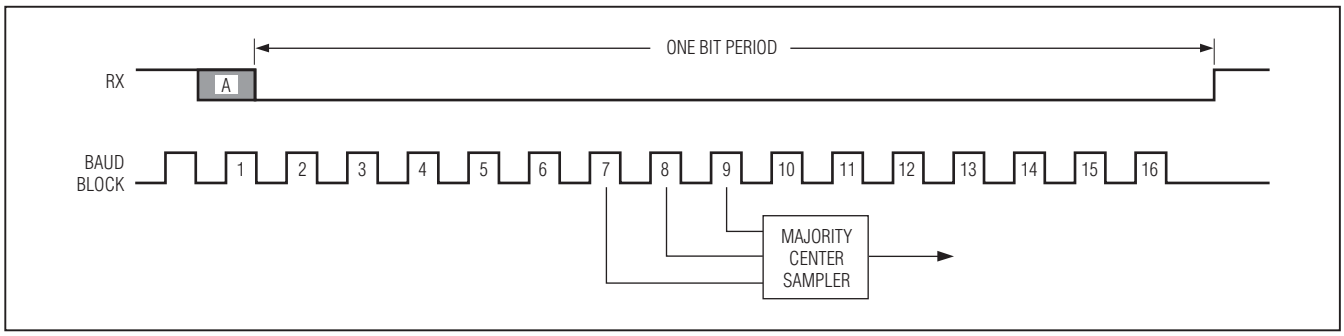


Figure 5. Midbit Sampling

To halt transmission, set MODE1[1]: TxDisabl to 1. After MODE1[1] is set, the transmitter completes transmission of the current character and then ceases transmission.

The TX output logic can be inverted through IrDA[5]: TxInv. If not stated otherwise, all transmitter logic described in this data sheet assumes IrDA[5] is 0.

Receiver Operation

The receiver expects the format of the data at RX to be as shown in Figure 4. The quiescent logic state is a high and the first bit (the START bit) is logic-low. The receiver samples the data near the midbit instant (Figure 4). The received words and their associated errors are deposited into the receive FIFO. Errors and status information are stored for every received word (Figure 6). The host reads data out of the receive FIFO through the Receive Holding register (RHR), oldest data first. The status information of the word previously read out of the RHR is located in the Line Status register (LSR). After a word is read out of the RHR, the LSR contains the status information for that word.

The following three error conditions are determined for each received word: parity error, framing error, and noise on the line. Line noise is detected by checking the consistency of the logic of the three samples (Figure 5).

The receiver can be turned off through MODE1[0]: RxDisabl. When this bit is set to 1, the MAX3107 turns the receiver off immediately following the current word and does not receive any further data. The RX input logic can be inverted through IrDA[4]: RxInv.

Line Noise Indication

When operating in standard (i.e., not 2x or 4x rate) mode, the MAX3107 checks that the binary logic level of the three samples per received bit are identical. If any of the three samples have differing logic levels, then noise on the transmission line has affected the received data and is considered to be noisy. This noise indication is reflected in the LSR[5]: RxNoise bit for each received byte. Parity errors are another indication of noise, but are not as sensitive.

Clocking and Baud-Rate Generation

The MAX3107 can be clocked by an external crystal or an external clock source. Figure 7 shows a simplified diagram of the clocking circuitry. When the MAX3107 is clocked by the crystal, the STSInt[5]: ClockReady indicates when the clocks have settled and the baud-rate generator is ready for stable operation.

The baud-rate clock can be routed to the $\overline{RTS}$/CLKOUT output. The clock rate is 16x the baud rate in standard operating mode, and 8x the baud rate in 2x rate mode. In 4x rate mode, the CLKOUT frequency is 4x the programmed baud rate. If the fractional portion of the baud-rate generator is used, the clock is not regular and exhibits jitter.

SPI/I²C UART with 128-Word FIFOs

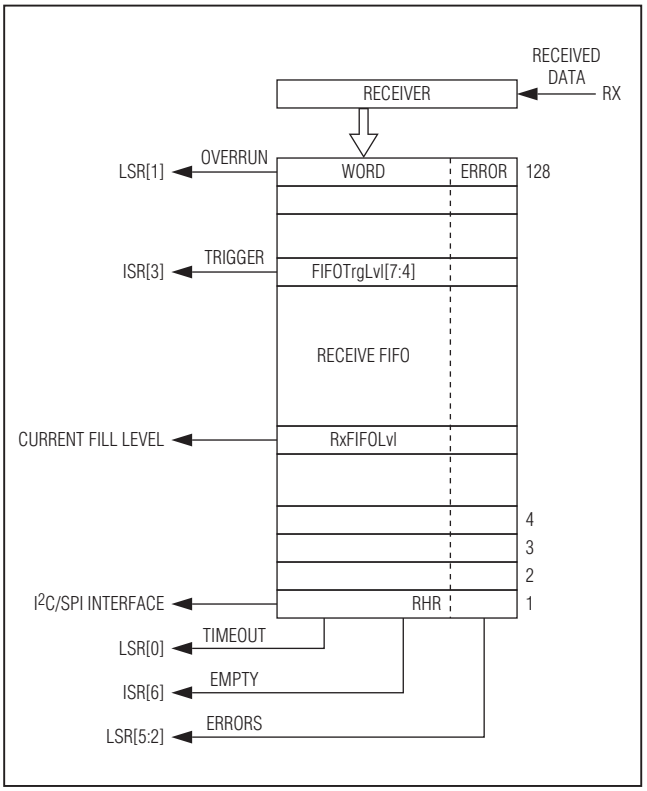


Figure 6. Receive FIFO

Crystal Oscillator

Set CLKSource[4]: ClockEn to 1 and CLKSource[1]: CrystalEn to 1 to enable and select the crystal oscillator. The on-chip crystal oscillator has load capacitances of 20pF integrated in both XIN and XOUT. Connect an external crystal or ceramic oscillator between XIN and XOUT.

External Clock Source

When an external clock signal is used, this should be connected to XIN. Leave XOUT unconnected. Set CLKSource[4]: ClockEn to 1 and CLKSource[1]: CrystalEn to 0 to select external clocking.

PLL and Predivider

The internal predivider and PLL allow for a wide range of external clock frequencies and baud rates. The PLL can be configured to multiply the input clock rate by a factor of 6, 48, 96, or 144 through PLLConfig[7:6]. The predivider, located between the input clock and the PLL, allows division of the input clock by a factor between 1 and 63 by writing to PLLConfig[5:0]. See the PLLConfig register description for more information.

Fractional Baud-Rate Generator

The internal fractional baud-rate generator provides a high degree of flexibility and high resolution in baud-rate programming. The baud-rate generator has a 16-bit integer divisor and a 4-bit word for the fractional divisor. The fractional baud-rate generator can be used with the external crystal or clock source.

The integer and fractional divisors are calculated through the divisor, D:

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

where f_{REF} is the reference frequency input to the baud-rate generator and D is the ideal divisor. f_{REF} must be less than 96MHz. In 2x and 4x rate modes, replace the divisor 16 by 8 or 4, respectively.

The integer divisor portion, DIV, of the divisor, D, is obtained by truncating D:

$$DIV = TRUNC(D)$$

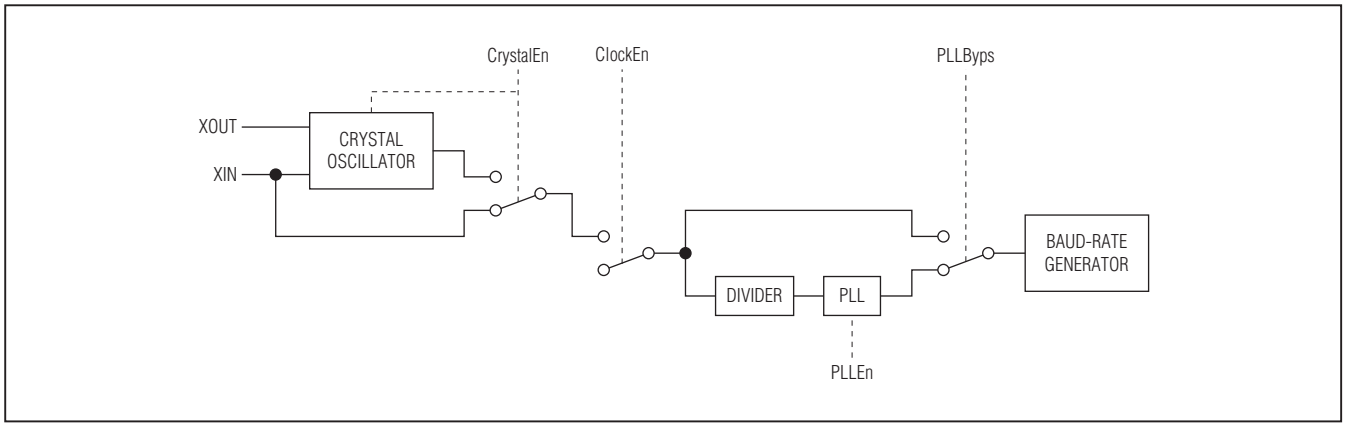


Figure 7. Clock Selection Diagram

DIV can be a maximum of 16 bits wide and is programmed into the 2-byte-wide registers DIVMSB and DIVLSB. The minimum allowed for DIVLSB is 1.

The fractional portion of the divisor, FRACT, is a 4-bit nibble, which is programmed into BRGConfig[3:0]. The maximum value is 15, allowing the divisor to be programmed with a resolution of 0.0625. FRACT is calculated as:

FRACT = ROUND(16 x (D-DIV))

The following is an example of calculating the divisor. It is based on a required baud rate of 190kbaud and a reference input frequency of 28.23MHz and 1x (default) rate mode.

The ideal divisor is calculated as:

D = 28,230,000/(16 x 190,000)

= 9.286184210526315789473684210526 3

hence DIV = 9.

FRACT =

ROUND(4.57894736842105263157894736842 11) = 5

so that DIVMSB = 0x00, DIVLSB = 0x09, and BRGConfig[3:0] = 0x05.

The resulting (actual) baud rate can be calculated as:

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

For this example: D_{ACTUAL} = 9 + 5/16 = 9.3125

where

$$D_{ACTUAL} = DIV + FRACT/16$$

and

$$BR_{ACTUAL} = 28,230,000/(16 \times 9.3125)$$

= 189463.08724832214765100671140 94 baud

Thus, the baud rate is within 0.28% of the ideal rate.

2x and 4x Rate Modes

To support higher baud rates than possible with standard (16x sampling) operation, the MAX3107 offers 2x and 4x rate modes. In this case, the reference clock rate only needs to be either 8x or 4x of the baud rate, respectively. The bits are only sampled once at the midbit instant instead of the usual three samples to determine the logic value of the bits. This reduces the tolerance to line noise on the received data. The 2x and 4x modes are selectable through BRGConfig[5:4]. Note that IrDA encoding and decoding does not operate in 2x and 4x modes.

When 2x rate mode is selected, the actual baud rate is twice the rate programmed into the baud-rate generator. If 4x rate mode is enabled, the actual baud rate on the line is quadruple that of programmed baud rate (Figure 8).

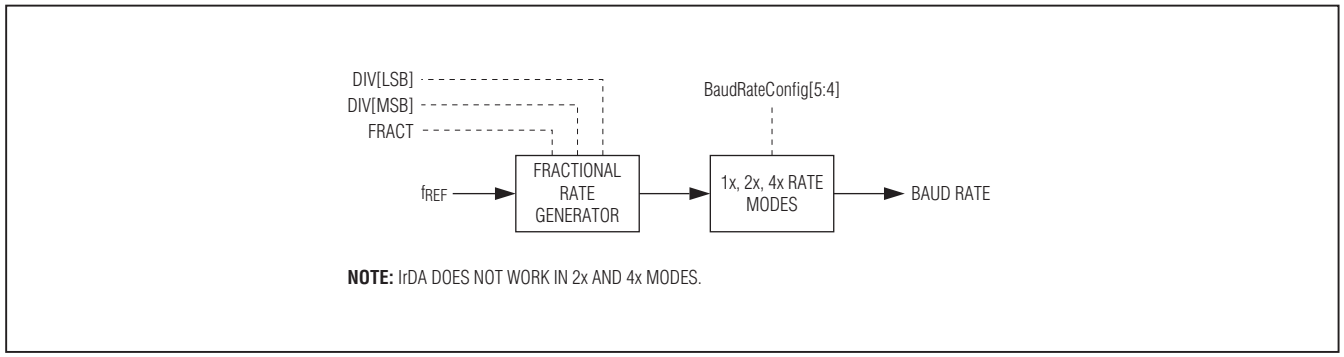


Figure 8. 2x and 4x Baud Rates

Multidrop Mode

In multidrop mode, also known as 9-bit mode, the word length is 8 bits and a 9th bit is used for distinguishing between an address and a data word. Multidrop mode is enabled through MODE2[6]: MultiDrop. Parity checking is disabled and an SpclCharInt[5]: MultiDropInt interrupt is generated when an address (9th bit set) is received.

It is up to the host processor to filter out the data intended for its address. Alternatively, the auto data-filtering mode can be used to automatically filter out the data intended for the station's specific 9-bit mode address.

Auto Data Filtering in Multidrop Mode

In multidrop mode, the MAX3107 can be configured to automatically filter out data that is not meant for its address. The address is user-definable either by programming a register value or a combination of a register values and GPIO hardware inputs. Use either XOFF2 or XOFF2[7:4] in combination with GPIO_ to define the address.

Enable multidrop mode by setting MODE2[6]: MultiDrop to 1 and enable auto data filtering by setting MODE2[4]: SpecialChr to 1.

When using register bits in combination with GPIO_ to define the address, the MSB of the address is written to XOFF2[7:4] register bits, while the LSBs of the address are defined through the GPIOs. To enable this mode, set FlowCtrl[2]: GPIAddr, MODE2[4]: SpecialChr, and MODE2[6]: MultiDrop to 1. GPIO_ is automatically read when FlowCtrl[2]: GPIAddr is set to 1, and the address is updated on logic changes at GPIO_.

In the auto data-filtering mode, the MAX3107 automatically accepts data that is meant for its address and places this into the receive FIFO, while it discards data that is not meant for its address. The received address word is not put into the FIFO.

Auto Transceiver Direction Control

In some half-duplex communication systems, the transceiver's transmitter must be turned off when data is being received so as not to load the bus. This is the case in half-duplex RS-485 communication. Similarly in full-duplex multidrop communication, like RS-485 or RS-422/V.11, only one transmitter can be enabled at any one time and the others must be disabled. The MAX3107 can automatically enable/disable a transceiver's transmitter and/or receiver. This relieves the host processor of this time-critical task.

The $\overline{RTS}$/CLKOUT output is used to control the transceivers' transmit enable input and is automatically set high when the MAX3107's transmitter starts transmission. This occurs as soon as data is present in the transmit FIFO. Auto transceiver direction control is enabled through MODE1[4]: TrnscvCtrl. Figure 9 shows a typical MAX3107 connection in a RS-485 application.

The $\overline{RTS}$/CLKOUT output can be set high in advance of TX transmission by a programmable time period called the setup time (Figure 10). The setup time is programmed through HDplxDelay[7:4]. Similarly, the $\overline{RTS}$/CLKOUT signal can be held high for a programmable period after the transmitter has completed transmission. The hold time is programmed through HDplxDelay[3:0].

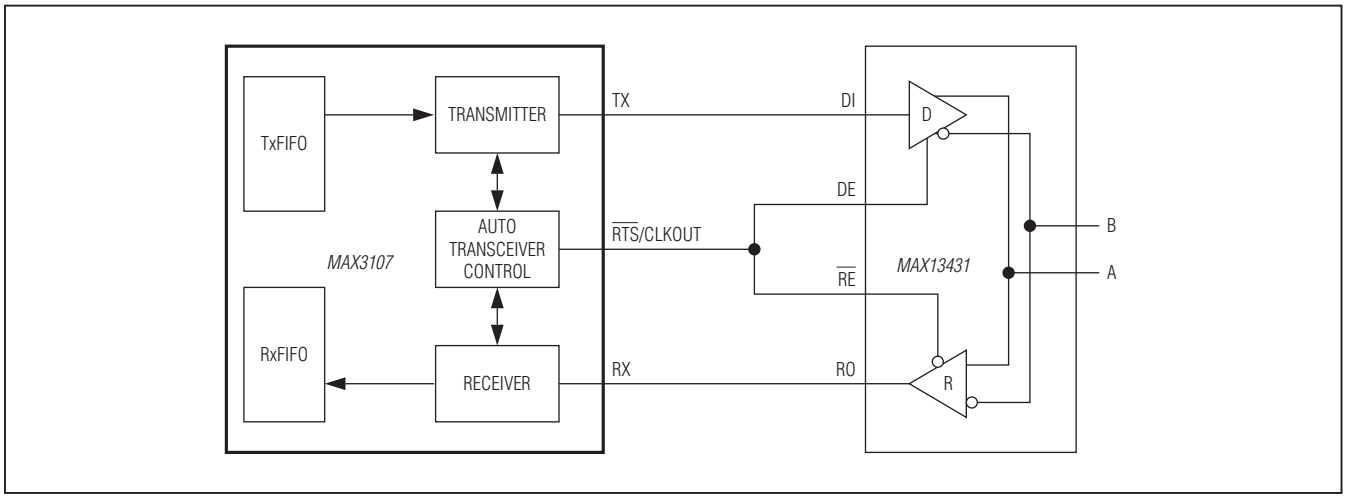


Figure 9. Auto Transceiver Direction Control

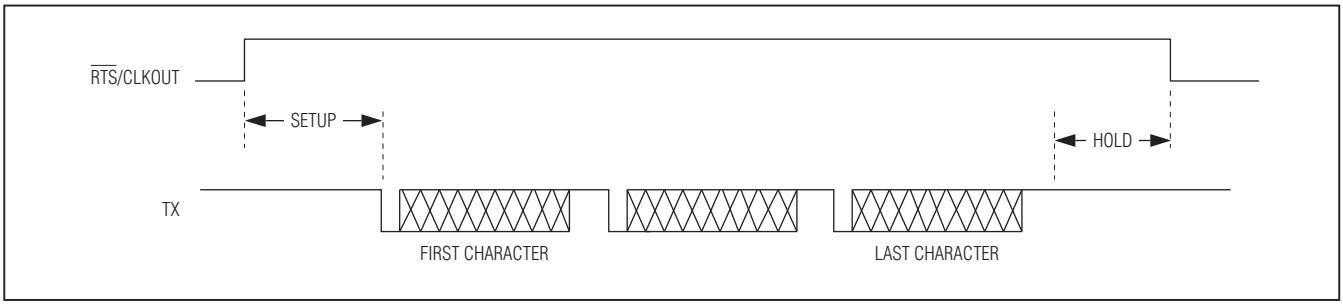


Figure 10. Setup and Hold Times in Auto Transceiver Direction Control

Echo Suppression

The MAX3107 can suppress echoed data, sometimes found in half-duplex communication (e.g., RS-485 and IrDA). If the transceiver's receiver is not turned off while the transceiver is transmitting, copies (echoes) are received by the UART. The MAX3107's receiver can block the reception of this echoed data by enabling echo suppression. Set MODE2[7]: EchoSuprs to 1 to enable echo suppression.

The MAX3107 receiver can block echoes with a long round trip delay. The transmitter can be configured to remain enabled after the end of transmission for a programmable period of time: the hold time delay. The hold time delay is set by the HDplxDelay[3:0] register. See the HDplxDelay description in the *Detailed Register Descriptions* section for more information.

Auto transceiver direction control and echo suppression can operate simultaneously.

Auto Hardware Flow Control

The MAX3107 is capable of auto hardware ($\overline{RTS}$ and $\overline{CTS}$) flow control without the need for host processor intervention. When AutoRTS control is enabled, the MAX3107 automatically controls the $\overline{RTS}$ handshake without the need for host processor intervention. AutoCTS flow control separately turns the MAX3107's transmitter on and off based on the $\overline{CTS}$ input. AutoRTS and AutoCTS flow control are independently enabled through FlowCtrl[1:0].

AutoRTS Control

AutoRTS flow control ensures that the receive FIFO does not overflow by signaling to the far-end UART to stop data transmission. The MAX3107 does this automatically by controlling $\overline{RTS}$/CLKOUT. AutoRTS flow control is enabled through FlowCtrl[0]: AutoRTS. The HALT and RESUME levels determine the threshold levels at which $\overline{RTS}$/CLKOUT is asserted and deasserted. HALT and

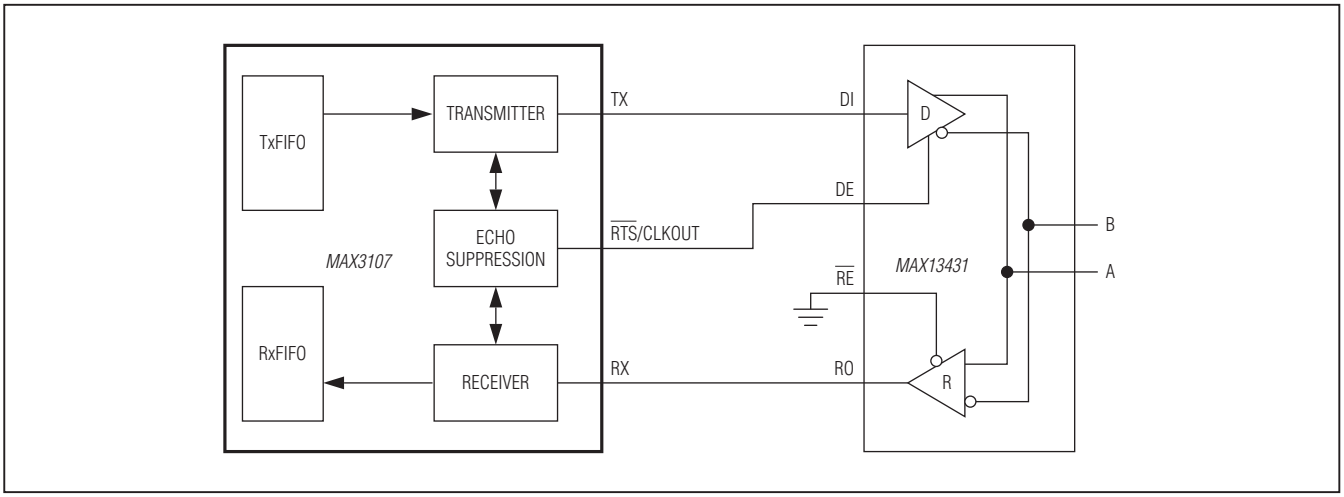


Figure 11. Half-Duplex with Echo Suppression

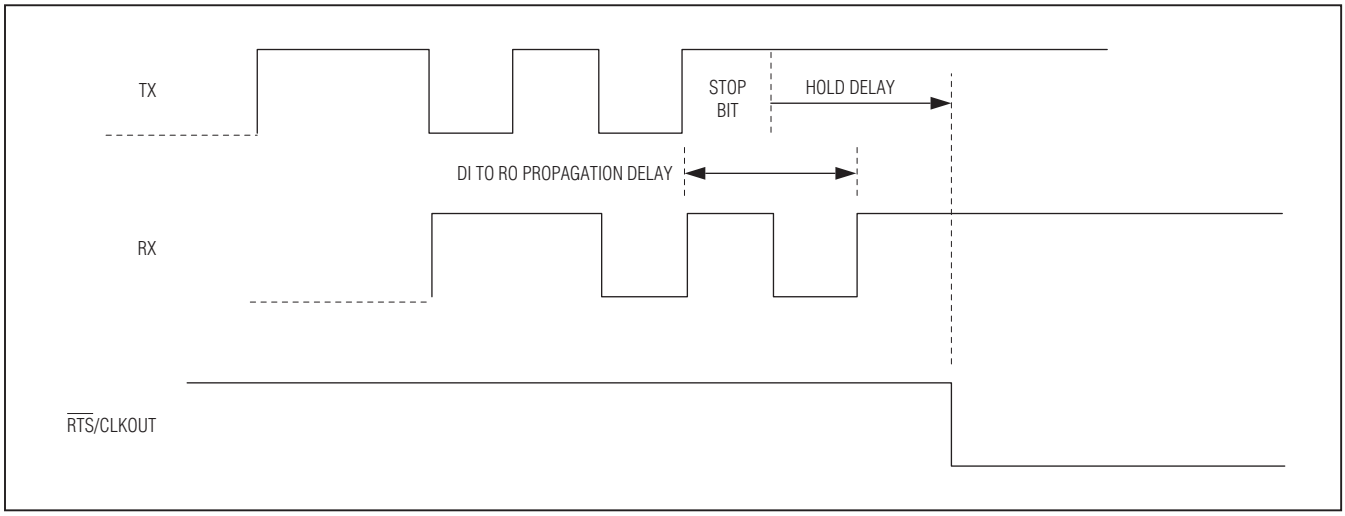


Figure 12. Echo Suppression Timing

RESUME are programmed in FlowLvl. With differing HALT and RESUME levels, hysteresis can be defined for the RTS/CLKOUT transitions.

When the RxFIFO fill level reaches the HALT level (FlowLvl[3:0]), the MAX3107 deasserts RTS/CLKOUT. RTS/CLKOUT remains deasserted until the RxFIFO is emptied and the number of words falls to the RESUME level.

Interrupts are not generated when the HALT and RESUME levels are reached. This allows the host controller to be completely disengaged from RTS flow control management.

AutoCTS Control

When AutoCTS flow control is enabled, the UART automatically starts transmitting data when the CTS input is logic-level low and stops transmitting when CTS is logic-high. This frees the host processor from managing this timing-critical flow-control task. AutoCTS flow control is enabled through FlowCtrl[1]: AutoCTS. During AutoCTS flow control the CTS interrupt works normally. Set the IRQEn[7]: CTSIntEn to 0 to disable CTS interrupts; then ISR[7]: CTSInt is fixed to logic 0 and the host does not receive interrupts from CTS. If CTS is set high during transmission, the MAX3107 completes transmission of the current word and halts transmission afterwards.

Turn the transmitter off by setting MODE1[1] to 1 before enabling AutoCTS control.

Auto Software (XON/XOFF) Flow Control

When auto software flow control is enabled, the MAX3107 recognizes and/or sends predefined XON/XOFF characters to control the flow of data across the asynchronous serial link. Auto flow works autonomously and does not involve host intervention, similar to auto hardware flow control. To reduce the chance of receiving corrupted data that equals a single-byte XON or XOFF character, the MAX3107 allows for double-wide (16-bit) XON/XOFF characters. XON and XOFF are programmed into the XON1, XON2 and XOFF1, XOFF2 registers.

FlowCtrl[7:3] are used for enabling and configuring auto software flow control. An ISR[1] interrupt is generated when XON or XOFF are received and details are found in SpclCharInt. The IRQ can be masked by setting IRQEn[1]: SpclChrIEn to 0.

Software flow control consists of transmitter control and receiver overflow control, which can operate independently of each other.

Transmitter Flow Control

If auto transmitter control (FlowCtrl[5:4]) is enabled, the receiver compares all received words with the XOFF and XON characters. If a XOFF is received, the MAX3107 halts its transmitter from sending further data. The receiver is not affected and continues reception. Upon receiving an XON, the transmitter restarts sending data. The received XON and XOFF characters are filtered out and are not put into the receive FIFO, as they do not have significance to the higher layer protocol. An interrupt is not generated.

Turn the transmitter off (MODE1[1]) before enabling transmitter control.

Receiver Flow Control

If auto receiver overflow control (FlowCtrl[7:6]) is enabled, the MAX3107 automatically sends XOFF and XON control characters to the far-end UART to avoid receiver overflow. XOFF1/XOFF2 are sent when the receive FIFO fill level reaches the HALT value set in the FlowLvl register. When the host controller reads data from the Receive FIFO to a level equal to the RESUME level programmed into the FlowLvl register, XON1/XON2 are automatically sent to the far-end station to signal it to resume data transmission.

If dual-character (XON1 and XON2/XOFF1 and XOFF2) flow control is selected, XON1/XOFF1 are transmitted before XON2/XOFF2.

FIFO Interrupt Triggering

Receive and transmit FIFO fill-dependent interrupts are generated if FIFO trigger levels are defined. When the number of words in the FIFOs reach or exceed a trigger level, as programmed in FIFOTrgLvl, an ISR[3] or ISR[4] interrupt is generated. There is no relationship between the trigger levels and the HALT or RESUME levels.

The FIFO trigger level can, for example, be used for a block data transfer, since it gives the host an indication when a given block size of data is available for readout in the teceive FIFO or available for transfer to the transmit FIFO.

Low-Power Standby Modes

The sleep and shutdown modes reduce power consumption during periods of inactivity. In both sleep and shutdown modes, the UART disables specific functional blocks to reduce power consumption.

Forced Sleep Mode

In forced sleep mode, all UART-related on-chip clocking is stopped. The following are inactive: the crystal oscillator, the PLL, the predivider, the receiver, and the transmitter. The SPI/I²C interface and the registers remain active. Thus, the host controller can access the resisters. To enter sleep mode, set MODE1[5] to 1. To wake up, set MODE1[5] to 0.

Autosleep Mode

The MAX3107 can be configured to operate in autosleep mode by setting MODE1[6] to 1. In autosleep mode, the MAX3107 automatically enters sleep mode when all the following conditions are met:

- Both FIFOs are empty.
- There are no pending $\overline{IRQ}$ interrupts.
- There is no activity on any input pins for a period equal to 65,536 UART characters lengths.

The MAX3107 exits autosleep mode as soon as activity is detected on any of the GPIO_, RX, or $\overline{CTS}$ inputs.

To manually wake up the MAX3107, set MODE1[6] to 0. After wake-up is initiated, the internal clock starts up and a period of time is needed for clock stabilization. The STSInt[5]: ClockReady bit indicates when the clocks are stable. If an external clock source is used, the STSInt[5] bit does not indicate clock stability.

Shutdown Mode

Shutdown mode is the lowest power consumption mode. In shutdown mode, all the MAX3107 circuitry is off. This includes the I²C/SPI interface, the registers, the FIFOs, and clocking circuitry. The LDO is kept on. To enter shutdown mode, connect $\overline{RST}$ to DGND.

When the $\overline{RST}$ input is toggled high, the MAX3107 exits shutdown mode. When the MAX3107 sets $\overline{IRQ}$ to logic-high, the chip initialization is completed. The MAX3107 needs to be reprogrammed following a shutdown. Keep V18 powered by the internal LDO or an external 1.8V supply during shutdown.

Power-Up and $\overline{IRQ}$

$\overline{IRQ}$ has two functions. During normal operation (MODE1[7] is 1), $\overline{IRQ}$ operates as a hardware interrupt output, whereby the $\overline{IRQ}$ is active when an interrupt is pending. An $\overline{IRQ}$ interrupt is only produced during normal operation, if at least one of the IRQEn interrupt enable bits are enabled.

During power-up or following a reset, $\overline{IRQ}$ has a different function. It is held low until the MAX3107 is ready for programming following an initialization delay. Once $\overline{IRQ}$ goes high, the MAX3107 is ready to be programmed. The MODE1[7]: IRQSel bit should then be set in order to enable normal $\overline{IRQ}$ interrupt operation.

In polled mode, the RevID register can be polled to check whether the MAX3107 is ready for operation. If the controller gets a valid response from RevID, then the MAX3107 is ready for operation.

Interrupt Structure

The structure of the interrupt is shown in Figure 13. There are four interrupt source registers: ISR, LSR, STSInt, and SpclCharInt. The interrupt sources are divided into top-level and low-level interrupts. The top-level interrupts typically occur more often and can be read out directly through the ISR. The low-level interrupts typically occur less often and their specific source can be read out through the LSR, STSInt, or SpclChar registers. The three LSBs of the ISR point to the low-level interrupt registers that contain the source detail of the interrupt source.

Interrupt Enabling

Every interrupt bit of the four interrupt registers can be enabled or masked through an associated interrupt enable register bit. These are the IRQEn, LSRIntEn, SpclChrIntEn and STSIntEn registers.

Interrupt Clearing

When an ISR interrupt is pending (i.e., any bit in ISR is set) and the ISR is subsequently read, the ISR bits and $\overline{IRQ}$ are cleared. Both the SpclCharInt and the STSInt registers also are clear on read (COR). The LSR bits are only cleared when the source of the interrupt is removed, not when LSR is read.

Detailed Register Descriptions

The MAX3107 has a flat register structure, without shadow registers, that makes programming and code simple and efficient. All registers are 8 bits wide.

RHR—Receiver Hold Register

ADDRESS:	0x00							
MODE:	R							
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NAME	RData7	RData6	RData5	RData4	RData3	RData2	RData1	RData0
RESET	X	X	X	X	X	X	X	X

Bits 7–0: RData[7:0]

The RHR is the bottom of the receive FIFO and is the register used for reading data out of the receive FIFO. It contains the oldest (first received) character in the receive FIFO. RHR[0] is the LSB of the character received at the RX input. It is the first data bit of the serial-data word received by the receiver.

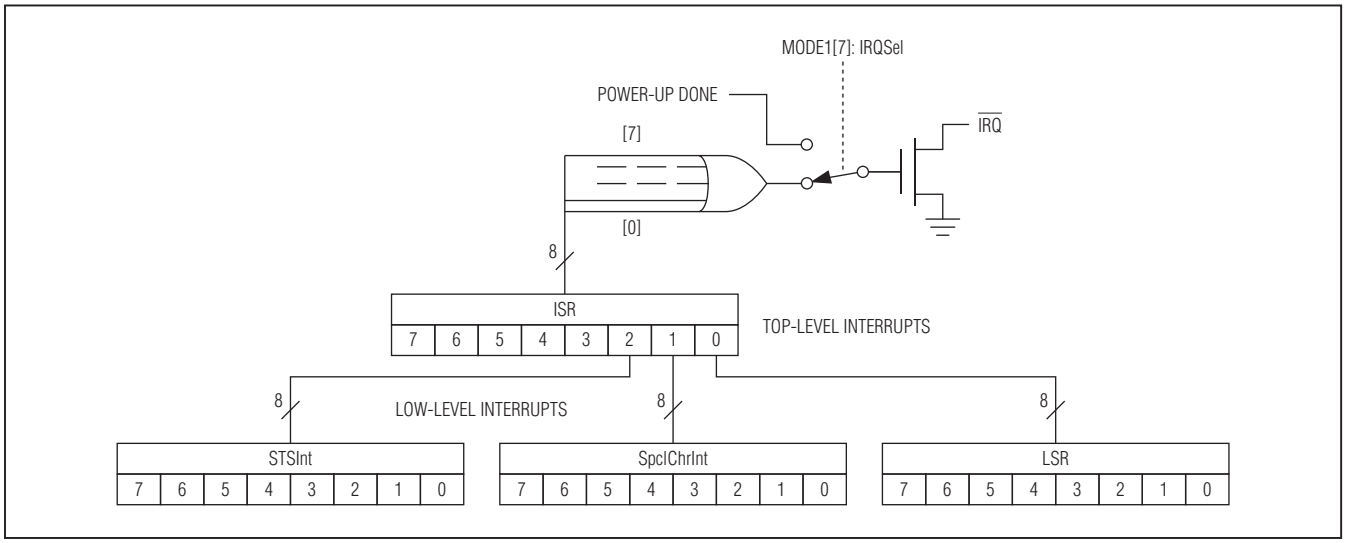


Figure 13. Simplified Interrupt Structure

THR—Transmit Hold Register

ADDRESS:		0x00						
MODE:		W						
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NAME	TData7	TData6	TData5	TData4	TData3	TData2	TData1	TData0

Bits 7–0: TData[7:0]

The THR is the register that the host controller writes data to for subsequent UART transmission. This data is deposited in the transmit FIFO. THR[0] is the LSB. It is the first data bit of the serial-data word that the transmitter sends out, right after the START bit.

IRQEn—IRQ Enable Register

ADDRESS:		0x01						
MODE:		R/W						
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NAME	CTSIEn	RxEmtyIEn	TxEmtyIEn	TxTrgIEn	RxTrgIEn	STSIEn	SpclChrIEn	LSRErrIEn
RESET	0	0	0	0	0	0	0	0

The IRQEn is used to enable the $\overline{IRQ}$ physical interrupt. Any of the eight ISR interrupt sources can be enabled to generate an $\overline{IRQ}$. The IRQEn bits only influence the $\overline{IRQ}$ output and do not have any effect on the ISR contents or behavior. Every one of the IRQEn bits operates on an ISR bit.

Bit 7: CTSIEn

The CTSIEn bit enables $\overline{IRQ}$ interrupt generation when the CTSInt interrupt bit is set in the ISR. Set CTSIEn bit low to disable $\overline{IRQ}$ generation from CTSInt.

Bit 6: RxEmtyIEn

The RxEmtyIEn bit enables $\overline{IRQ}$ interrupt generation when the RxEmtyInt interrupt bit is set in the ISR. Set RxEmtyIEn bit low to disable $\overline{IRQ}$ generation from RxEmtyInt.

Bit 5: TxEmtyIEn

The TxEmtyIEn bit enables $\overline{IRQ}$ interrupt generation when the TxEmptyInt interrupt bit is set in the ISR. Set TxEmtyIEn bit low to disable $\overline{IRQ}$ generation from TxEmptyInt.

Bit 4: TxTrgIEn

The TxTrgIEn bit enables $\overline{IRQ}$ interrupt generation when the TFifoTrigInt interrupt bit is set in the ISR. Set TxTrgIEn bit low to disable $\overline{IRQ}$ generation from TFifoTrigInt.

Bit 3: RxTrgIEn

The RxTrgIEn bit enables $\overline{IRQ}$ interrupt generation when the RFifoTrigInt interrupt bit is set in the ISR. Set RxTrgIEn bit low to disable $\overline{IRQ}$ generation from RFifoTrigInt.

Bit 2: STSIEn

The STSIEn bit enables $\overline{IRQ}$ interrupt generation when the STSInt interrupt bit is set in the ISR. Set STSIEn bit low to disable $\overline{IRQ}$ generation from STSInt.

Bit 1: SpclChrIEn

The SpclChrIEn bit enables $\overline{IRQ}$ interrupt generation when the SpCharInt interrupt bit is set in the ISR. Set SpclChrIEn bit low to disable $\overline{IRQ}$ generation from SpCharInt.

Bit 0: LSRErrIEn

The LSRErrIEn bit enables $\overline{IRQ}$ interrupt generation when the LSRErrInt interrupt bit is set in the ISR[0]. Set LSRErrIEn low to disable $\overline{IRQ}$ generation from LSRErrInt.

ISR—Interrupt Status Register

ADDRESS:	0x02							
MODE:		COR						
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NAME	CTSInt	RxEmptyInt	TxEmptyInt	TFifoTrigInt	RFifoTrigInt	STSInt	SpCharInt	LSRErrInt
RESET	0	1	1	0	0	0	0	0

The ISR provides an overview of all interrupts generated in the MAX3107. These interrupts are cleared on reading the ISR. When the MAX3107 is operated in polled mode, the ISR can be polled to establish the UART's status. In interrupt-driven mode, $\overline{IRQ}$ interrupts are enabled through the appropriate IRQEn bits. The ISR contents give direct information on the cause for the interrupt or point to other registers that contain more detailed information.

Bit 7: CTSInt

The CTSInt is set when a logic state transition occurs at the $\overline{CTS}$ input. This bit is cleared after ISR is read. The current logic state of the $\overline{CTS}$ input can be read out through the LSR[7]: $\overline{CTS}$bit.

Bit 6: RxEmptyInt

The RxEmptyInt is set when the receive FIFO is empty. This bit is cleared after ISR is read. Its meaning can be inverted by setting the MODE2[3]: RxEmtyInv bit.

Bit 5: TxEmptyInt

The TxEmptyInt bit is set when the transmit FIFO is empty. This bit is cleared once ISR is read.

Bit 4: TFifoTrigInt

The TFifoTrigInt bit is set when the number of characters in the transmit FIFO is equal to or greater than the transmit FIFO trigger level defined in FIFOTrgLvl[3:0]. TFifoTrigInt is cleared when the transmit FIFO level falls below the trigger level or after the ISR is read. It can be used as a warning that the transmit FIFO is nearing overflow.

Bit 3: RFifoTrigInt

The RFifoTrigInt bit is set when the receive FIFO fill level reaches the receive FIFO trigger level, as defined in the FIFOTrgLvl[7:4]. This can be used as an indication that the receive FIFO is nearing overrun. It can also be used to report that a known number of words are available which can be read out in one block. The meaning of RFifoTrigInt can be inverted through MODE2[2]. RFifoTrigInt is cleared when ISR is read.

Bit 2: STSInt

The STSInt bit is set high when any bit in the STSInt register that is enabled through a STSIntEn bit is high. The STSInt bit is cleared on reading ISR.

Bit 1: SpCharInt

The SpCharInt bit is set high when a special character is received, a line BREAK is detected, or an address character is received in multidrop mode. The cause for the SpCharInt interrupt can be read from the SpclCharInt register, if enabled through the SpclChrIntEn bits. The SpCharInt interrupt is cleared when the ISR is read.

Bit 0: LSRErrInt

The LSRErrInt bit is set high when any LSR bits, which are enabled through the LSRIntEn, are set. This bit is cleared after the ISR is read.