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

MAX14830

Quad Serial UART with 128-Word FIFOs

General Description

The MAX14830 is an advanced quad universal asynchronous receiver-transmitter (UART), each UART having 128 words of receive and transmit first-in/first-out (FIFO) and a high-speed serial peripheral interface (SPI™) or I²C controller interface. A PLL and fractional baud-rate generators allow a high degree of flexibility in baud-rate programming and reference clock selection.

Each of the four UARTs is selected by in-band SPI/I²C addressing. Logic-level translation on the transceiver and controller interfaces allows ease of interfacing to microcontrollers, FPGAs, and transceivers that are powered by differing supply voltages.

Extensive features simplify transceiver control in half-duplex communication applications. The MAX14830 features the ability to synchronize the start of individual UART's transmission by SPI-based triggering. On-board timers allow programming of delays between transmitters as well as clock generation on GPIOs.

The 128-word FIFOs have advanced FIFO control reducing host processor data flow management.

The MAX14830 is available in a 48-pin TQFN (7mm x 7mm) package and is specified to operate over the extended -40°C to +85°C temperature range.

Applications

Industrial Control Systems

Programmable Logic Controllers (PLC)

IO-Link Master Controllers

Automotive Infotainment Systems

Medical Systems

Point-of-Sales Systems

Airplane Communication Buses

Typical Operating Circuits appear at end of data sheet.

Features

- **SPI Up to 26MHz Clock Rate**
- **Fast-Mode Plus (Fm+) I²C Interface Up to 1MHz**
- **128-Word Transmit and Receive FIFOs Per UART**
- **6Mbaud (max) Data Rate in 16x Sampling Mode**
- **12/24Mbaud (max) Data Rate in 2x/4x Rate Modes**
- **Fractional Baud-Rate Generators, Predivider, and Phase-Locked Loop (PLL)**
- **Transmitter Synchronization Through SPI Commands**
- **Four Timers Routed to GPIOs**
- **Automatic Hardware Flow Control Using $\overline{\text{RTS}}$_ and $\overline{\text{CTS}}$_ Outputs and Inputs**
- **Automatic Software Flow Control (XON/XOFF)**
- **Auto Transceiver Direction Control**
- **Programmable Setup and Hold Times for Transceiver Control**
- **Auto Transmitter Disable**
- **Half-Duplex Echo Suppression**
- **Special Character Detection**
- **9-Bit Multidrop Mode Address Detection and Filtering**
- **SIR- and MIR-Compliant IrDA® Encoder/Decoders**
- **16 Flexible GPIOs with 20mA Drive Capability**
- **+2.35V to +3.6V Supply Range**
- **Logic-Level Translation Down to 1.61V on Controller and Transceiver Interfaces**
- **Small TQFN (7mm x 7mm) Package**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX14830ETM+	-40°C to +85°C	48 TQFN-EP*

+Denotes a lead(Pb)-free/RoHS-compliant package.
**EP = Exposed pad.*

SPI is a trademark of Motorola, Inc.

IrDA is a registered service mark of Infrared Data Association Corporation.

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

19-5547; Rev 3; 1/13

MAX14830

Quad Serial UART with 128-Word FIFOs

TABLE OF CONTENTS

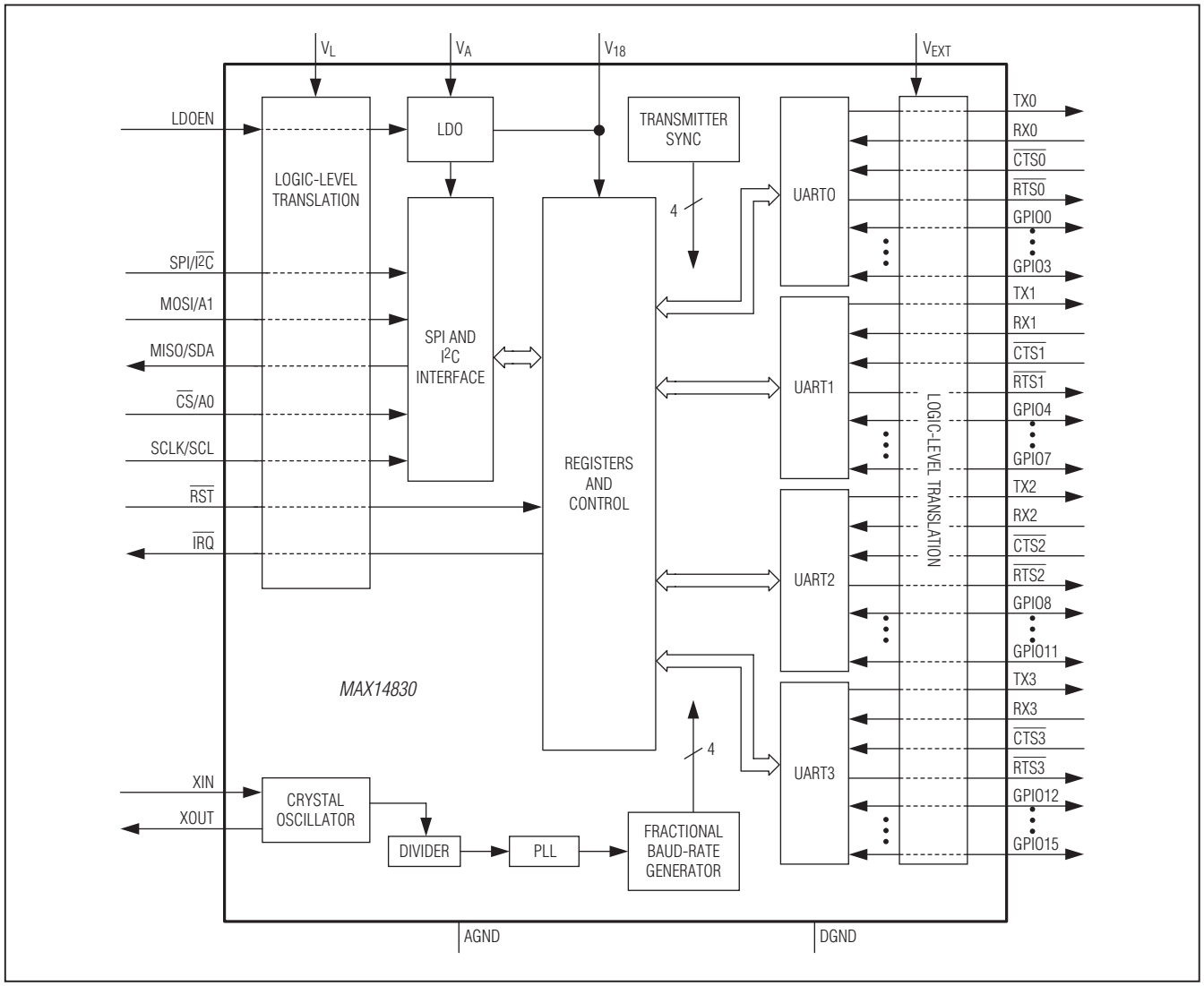
TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF REGISTERS

Functional Diagram



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 the lesser of (V_A + 0.3V) and +2.0V
$\overline{RST}$, $\overline{IRQ}$, MOSI/A1, $\overline{CS}$/A0, SCLK/SCL, MISO/SDA, LDOEN, SPI/$\overline{I2C}$ ..... -0.3V to (V_L + 0.3V)
TX0, RX0, $\overline{CTS0}$, GPIO0, GPIO1, GPIO2, GPIO3..... -0.3V to (V_{EXT} + 0.3V)
TX1, RX1, $\overline{CTS1}$, GPIO4, GPIO5, GPIO6, GPIO7..... -0.3V to (V_{EXT} + 0.3V)
TX2, RX2, $\overline{CTS2}$, GPIO8, GPIO9, GPIO10, GPIO11..... -0.3V to (V_{EXT} + 0.3V)
TX3, RX3, $\overline{CTS3}$, GPIO12, GPIO13, GPIO14, GPIO15..... -0.3V to (V_{EXT} + 0.3V)
DGND ..... -0.3V to +0.3V
Continuous Power Dissipation (T_A = +70°C)
TQFN (derate 38.5mW/°C above +70°C).....3076.9mW
Operating Temperature Range ..... -40°C to +85°C
Maximum Junction Temperature ..... +150°C
Storage Temperature Range ..... -65°C to +150°C
Lead Temperature (soldering, 10s) .....300°C
Soldering Temperature (reflow) .....+260°C

PACKAGE THERMAL CHARACTERISTICS (Note 1)

TQFN
Junction-to-Ambient Thermal Resistance (θ_{JA}).....26°C/W
Junction-to-Case Thermal Resistance (θ_{JC}).....1°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.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_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 = +2.5V, V_L = +1.8V, V_{EXT} = +2.8V, 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_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		1.95	V
CURRENT CONSUMPTION						
V_A Supply Current	I_A	1.8MHz crystal oscillator active, PLL disabled, SPI/I^2C interface idle, UART interfaces idle, V_{LDOEN} = V_L			400	µA
		Baud rate = 1Mbps, 20MHz external clock, SPI/I^2C interface idle, PLL disabled, all UARTs in loopback mode, V_{LDOEN} = 0V			0.5	mA
V_A Shutdown Supply Current	I_{ASHDN}	Shutdown mode, V_{LDOEN} = 0V, $V_{\overline{RST}}$ = 0V, all inputs and outputs are idle			35	µA
V_L Shutdown or Sleep Supply Current	I_L	Shutdown mode, V_{LDOEN} = 0V, $V_{\overline{RST}}$ = 0V, all inputs and outputs are idle			12	µA
V_{EXT} Shutdown Supply Current	I_{EXT}	Shutdown mode, V_{LDOEN} = 0V, $V_{\overline{RST}}$ = 0V, all inputs and outputs are idle			8	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 = +2.5V, V_L = +1.8V, V_{EXT} = +2.8V, T_A = +25°C.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{18} Input Power-Supply Current in Shutdown Mode	I_{18SHDN}	Shutdown mode, V_{LDOEN} = 0V, $V_{\overline{RST}}$ = 0V, all inputs and outputs are idle			200	µA
V_{18} Input Power-Supply Current	I_{18}	Baud rate = 1Mbps, 20MHz external clock, PLL disabled, all UARTs in loopback mode, V_{LDOEN} = 0V (Note 4)			5	mA
SCLK/SCL, MISO/SDA						
MISO/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	
MISO/SDA Output Low Voltage in SPI Mode	$V_{OL,SPI}$	I_{LOAD} = -2mA			0.4	V
MISO/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}	SPI and I²C mode		5		pF
SPI/$\overline{I2C}$, $\overline{CS}$/A0, MOSI/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}	SPI and I²C mode		5		pF
$\overline{IRQ}$ OUTPUT (OPEN DRAIN)						
Output Low Voltage	V_{OL}	I_{LOAD} = -2mA			0.4	V
Output Leakage Current	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
UART INTERFACE						
$\overline{RTS0}$, $\overline{RTS1}$, $\overline{RTS2}$, $\overline{RTS3}$, TX0, TX1, TX2, TX3 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 is three-stated, $V_{\overline{RTS\_}}$ = 0 to V_{EXT}	-1		+1	µA
Input Capacitance	C_{IN}	High-Z mode		5		pF

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 = +2.5V, V_L = +1.8V, V_{EXT} = +2.8V, T_A = +25°C.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
RX0, RX1, RX2, RX3, $\overline{CTS0}$, $\overline{CTS1}$, $\overline{CTS2}$, $\overline{CTS3}$ 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
$\overline{CTS0}$, $\overline{CTS1}$, $\overline{CTS2}$, $\overline{CTS3}$ Input Leakage Current	I_{IN_CTS}	$V_{\overline{CTS\_}}$ = 0 to V_{EXT}	-1		+1	µA
RX0, RX1, RX2, RX3 Pullup Current	$I_{IN\_RX\_}$	$V_{RX\_}$ = 0V, V_{EXT} = 3.6V	-7.5	-5.5	-3.5	µA
Input Capacitance	C_{IN_UART}			5		pF
GPIO0–GPIO15 INPUTS/OUTPUTS						
Output Low Voltage	V_{OL}	I_{LOAD} = -20mA, V_{EXT} > 2.3V, push-pull or open drain			0.45	V
		I_{LOAD} = -20mA, V_{EXT} < 2.3V, push-pull or open drain			0.55	
Output High Voltage	V_{OH}	I_{LOAD} = 5mA, push-pull			V_{EXT} - 0.4	V
Input Low Voltage	V_{IL}	GPIO_ is configured as an input			0.4	V
Input High Voltage	V_{IH}	GPIO_ is configured as an input	2/3 x V_{EXT}			V
Pulldown Current	I_{PD}	GPIO_ = V_{EXT} = 3.6V	3.5	5.5	7.5	µA
XIN						
Input Low Voltage	V_{IL}				0.2	V
Input High Voltage	V_{IH}		1.2			V
Input Capacitance	C_{XIN}			16		pF
XOUT						
Input Capacitance	C_{XOUT}			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.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
INTERNAL OSCILLATOR						
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	f_{REF}	(Note 5)			96	MHz

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.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
I²C BUS: TIMING CHARACTERISTICS (SEE 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	0.6			
		Fast mode plus	0.26			
Rise Time of SDA and SCL Signals Receiving	t_R	Standard mode (0.3 x V_L to 0.7 x V_L) (Note 6)	$20 + 0.1C_b$		1000	ns
		Fast mode (0.3 x V_L to 0.7 x V_L) (Note 6)	$20 + 0.1C_b$		300	
		Fast mode plus			120	
Fall Time of SDA and SCL Signals	t_F	Standard mode (0.7 x V_L to 0.3 x V_L) (Note 6)	$20 + 0.1C_b$		300	ns
		Fast mode (0.7 x V_L to 0.3 x V_L) (Note 6)	$20 + 0.1C_b$		300	
		Fast mode plus			120	
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 (Note 4)	C_b	Standard mode			400	pF
		Fast mode			400	
		Fast mode plus			550	

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.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
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 (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
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
$\overline{CS}$ Hold Time	t_{CSH}		30			ns

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

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

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: Not production tested. Guaranteed by design.

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

Test Circuits/Timing Diagrams

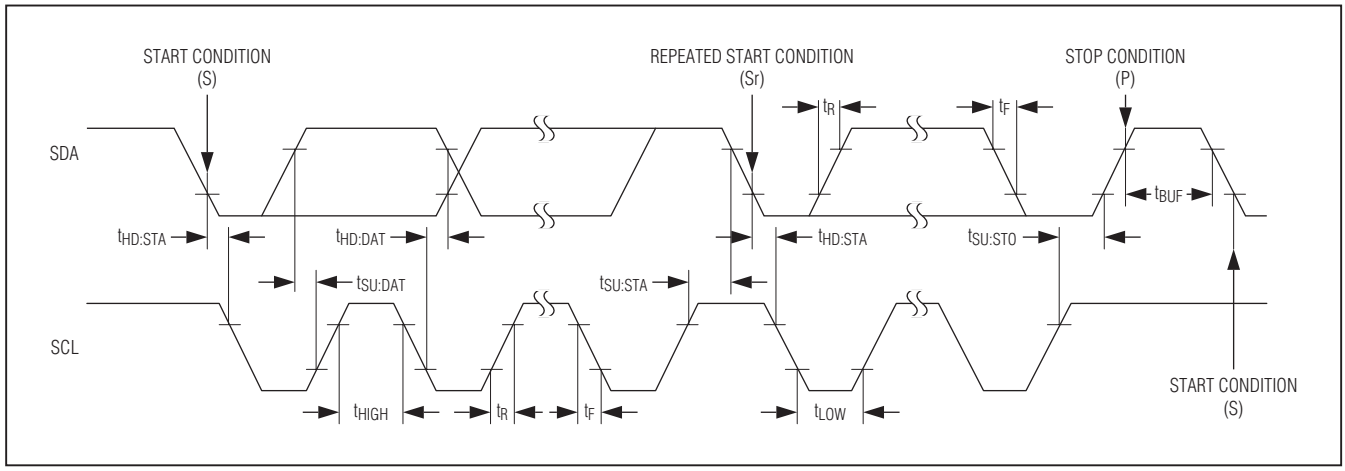


Figure 1. I²C Timing Diagram

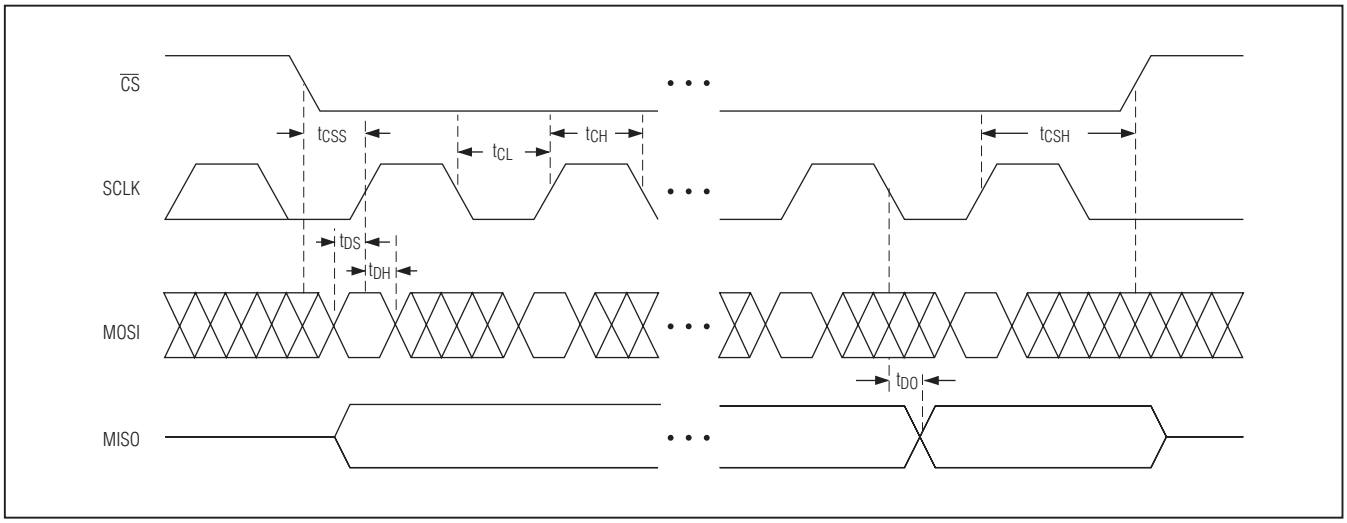
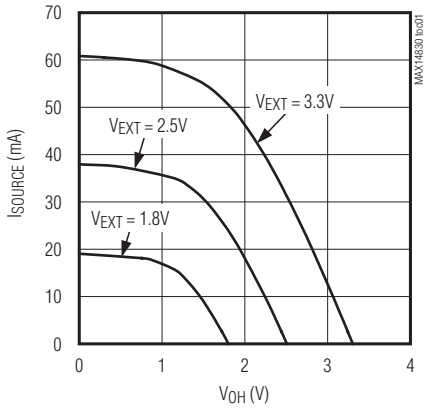


Figure 2. SPI Timing Diagram

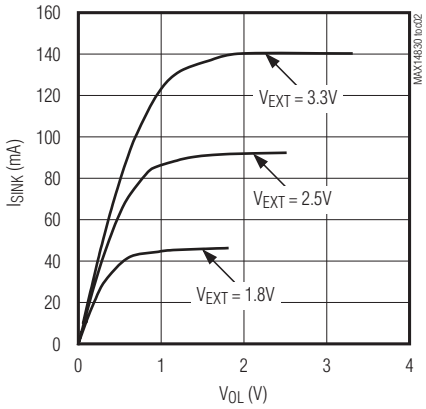
Typical Operating Characteristics

(T_A = +25°C, unless otherwise noted.)

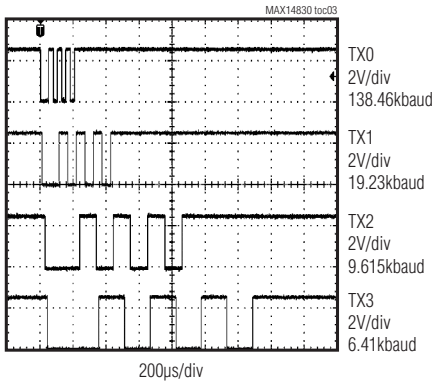
GPIO_ OUTPUT HIGH VOLTAGE vs. SOURCE CURRENT (PUSH-PULL)



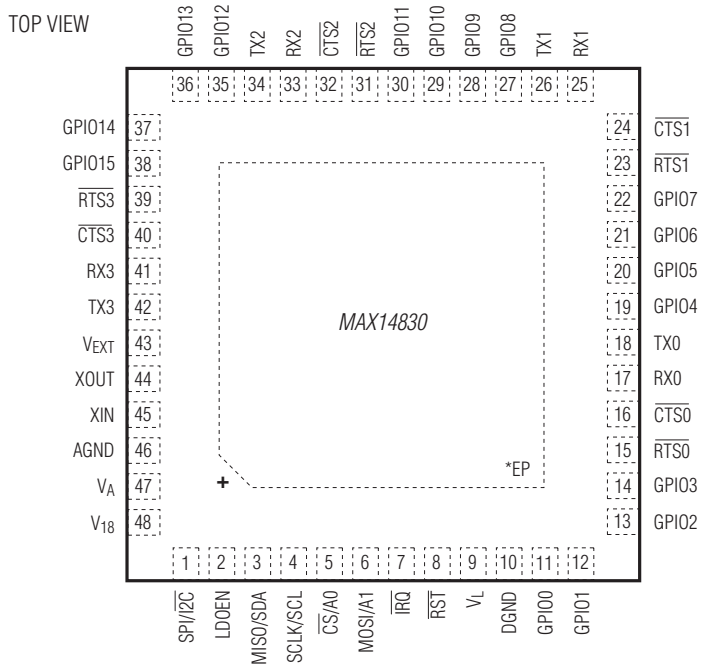
GPIO_ OUTPUT LOW VOLTAGE vs. SINK CURRENT (PUSH-PULL)



TRANSMITTER SYNCHRONIZATION



Pin Configuration



Pin Description

PIN	NAME	FUNCTION
1	SPI/$\overline{I2C}$	SPI or Active-Low I²C Selector Input. Drive SPI/$\overline{I2C}$ high to enable SPI. Drive SPI/$\overline{I2C}$ low to enable I²C.
2	LDOEN	LDO Enable Input. Drive LDOEN high to enable the internal 1.8V LDO. Drive LDOEN low to disable the internal LDO. When LDOEN is low, V_{18} can be supplied by an external voltage source.
3	MISO/SDA	Serial-Data Output. When SPI/$\overline{I2C}$ is high, MISO/SDA functions as the MISO, SPI serial-data output. When SPI/$\overline{I2C}$ is low, MISO/SDA functions as the SDA, I²C serial-data input/output.
4	SCLK/SCL	Serial-Clock Input. When SPI/$\overline{I2C}$ is high, SCLK/SCL functions as the SCLK, SPI serial-clock input (up to 26MHz). When SPI/$\overline{I2C}$ is low, SCLK/SCL functions as the SCL, I²C serial-clock input (up to 1MHz).
5	$\overline{CS}$/A0	Active-Low Chip-Select and Address 0 Input. When SPI/$\overline{I2C}$ is high, $\overline{CS}$/A0 functions as the $\overline{CS}$, SPI active-low chip-select input. When SPI/$\overline{I2C}$ is low, $\overline{CS}$/A0 functions as the A0, I²C device address programming input. Connect $\overline{CS}$/A0 to SDA, SCL, DGND, or V_L when SPI/$\overline{I2C}$ is low.
6	MOSI/A1	Serial-Data and Address 1 Input. When SPI/$\overline{I2C}$ is high, MOSI/A1 functions as the MOSI, SPI serial-data input. When SPI/$\overline{I2C}$ is low, MOSI/A1 functions as the A1, I²C device address programming input. Connect MOSI/A1 to SDA, SCL, DGND, or V_L when SPI/$\overline{I2C}$ is low.
7	$\overline{IRQ}$	Active-Low Interrupt Open-Drain Output. $\overline{IRQ}$ is asserted when an interrupt is pending.
8	$\overline{RST}$	Active-Low Reset Input. Drive $\overline{RST}$ low to force all of the UARTs into hardware reset mode. In hardware reset mode, the oscillator and the internal PLL are shut down and there is no clock activity.

Pin Description (continued)

PIN	NAME	FUNCTION
9	V_L	Digital Interface Logic-Level Supply. V_L powers the internal logic-level translators for $\overline{RST}$, $\overline{IRQ}$, MOSI/A1, $\overline{CS}$/A0, SCLK/SCL, MISO/SDA, LDOEN, and SPI/$\overline{I2C}$. Bypass V_L with a 0.1µF ceramic capacitor to DGND.
10	DGND	Digital Ground
11	GPIO0	General-Purpose Input/Output 0. GPIO0 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO0 has a weak pulldown resistor to ground. GPIO0 is the reference clock output when bit 7 of the TxSynch register is set to 1 (see the *UART Clock to GPIO* section for more information).
12	GPIO1	General-Purpose Input/Output 1. GPIO1 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO1 has a weak pulldown resistor to ground. GPIO1 is the TIMER output when bit 7 of the TIMER2 register is set to 1.
13	GPIO2	General-Purpose Input/Output 2. GPIO2 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO2 has a weak pulldown resistor to ground.
14	GPIO3	General-Purpose Input/Output 3. GPIO3 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO3 has a weak pulldown resistor to ground.
15	$\overline{RTS0}$	Active-Low Request-to-Send Output for UART0. $\overline{RTS0}$ can be set high or low by programming the LCR register. $\overline{RTS0}$ is the UART system clock/fractional divider output when bit 7 of the CLKSource register is set to 1.
16	$\overline{CTS0}$	Active-Low Clear-to-Send Input for UART0. $\overline{CTS0}$ is a flow control status input.
17	RX0	Serial Receiving Data Input for UART0. RX0 has a weak pullup to V_{EXT}.
18	TX0	Serial Transmitting Data Output for UART0
19	GPIO4	General-Purpose Input/Output 4. GPIO4 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO4 has a weak pulldown resistor to ground. GPIO4 is the reference clock output when bit 7 of the TxSynch register is set to 1 (see the *UART Clock to GPIO* section for more information).
20	GPIO5	General-Purpose Input/Output 5. GPIO5 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO5 has a weak pulldown resistor to ground. GPIO5 is the TIMER output when bit 7 of the TIMER2 register is set to 1.
21	GPIO6	General-Purpose Input/Output 6. GPIO6 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO6 has a weak pulldown resistor to ground.
22	GPIO7	General-Purpose Input/Output 7. GPIO7 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO7 has a weak pulldown resistor to ground.
23	$\overline{RTS1}$	Active-Low Request-to-Send Output for UART1. $\overline{RTS1}$ can be set high or low by programming the LCR register. $\overline{RTS1}$ is the UART system clock/fractional divider output when bit 7 of the CLKSource register is set to 1.
24	$\overline{CTS1}$	Active-Low Clear-to-Send Input for UART1. $\overline{CTS1}$ is a flow control status input.
25	RX1	Serial Receiving Data Input for UART1. RX1 has a weak pullup to V_{EXT}.
26	TX1	Serial Transmitting Data Output for UART1
27	GPIO8	General-Purpose Input/Output 8. GPIO8 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO8 has a weak pulldown resistor to ground. GPIO8 is the reference clock output when bit 7 of the TxSynch register is set to 1 (see the *UART Clock to GPIO* section for more information).

Pin Description (continued)

PIN	NAME	FUNCTION
28	GPIO9	General-Purpose Input/Output 9. GPIO9 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO9 has a weak pulldown resistor to ground. GPIO9 is the TIMER output when bit 7 of the TIMER2 register is set to 1.
29	GPIO10	General-Purpose Input/Output 10. GPIO10 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO10 has a weak pulldown resistor to ground.
30	GPIO11	General-Purpose Input/Output 11. GPIO11 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO11 has a weak pulldown resistor to ground.
31	$\overline{RTS2}$	Active-Low Request-to-Send Output for UART2. $\overline{RTS2}$ can be set high or low by programming the LCR register. $\overline{RTS2}$ is the UART system clock/fractional divider output when bit 7 of the CLKSource register is set to 1.
32	$\overline{CTS2}$	Active-Low Clear-to-Send Input for UART2. $\overline{CTS2}$ is a flow control status input.
33	RX2	Serial Receiving Data Input for UART2. RX2 has a weak pullup to V_{EXT}.
34	TX2	Serial Transmitting Data Output for UART2
35	GPIO12	General-Purpose Input/Output 12. GPIO12 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO12 has a weak pulldown resistor to ground. GPIO12 is the reference clock output when bit 7 of the TxSynch register is set to 1 (see the *UART Clock to GPIO* section for more information).
36	GPIO13	General-Purpose Input/Output 13. GPIO13 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO13 has a weak pulldown resistor to ground. GPIO13 is the TIMER output if bit 7 of the TIMER2 register is set to 1.
37	GPIO14	General-Purpose Input/Output 14. GPIO14 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO14 has a weak pulldown resistor to ground.
38	GPIO15	General-Purpose Input/Output 15. GPIO15 is user-programmable as an input or output (push-pull or open drain) or external event interrupt source. GPIO15 has a weak pulldown resistor to ground.
39	$\overline{RTS3}$	Active-Low Request-to-Send Output for UART3. $\overline{RTS3}$ can be set high or low by programming the LCR register. $\overline{RTS3}$ is the UART system clock/fractional divider output when bit 7 of the CLKSource register is set to 1.
40	$\overline{CTS3}$	Active-Low Clear-to-Send Input for UART3. $\overline{CTS3}$ is a flow control status input.
41	RX3	Serial Receiving Data Input for UART3. RX3 has a weak pullup to V_{EXT}.
42	TX3	Serial Transmitting Data Output for UART3
43	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.
44	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.
45	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.
46	AGND	Analog Ground
47	V_A	Analog Supply. V_A powers the PLL, and the internal LDO. Bypass V_A with a 0.1µF ceramic capacitor to AGND.
48	V_{18}	Internal 1.8V LDO Output and 1.8V Logic Supply Input. Bypass V_{18} with a 1µF ceramic capacitor to DGND.
—	EP	Exposed Paddle. Connect EP to AGND. Do not use EP as the main AGND connection.

Detailed Description

The MAX14830 quad UART bridges an SPI/MICROWIRE™ or I²C microprocessor bus to an asynchronous interface like RS-485, RS-232, or IrDA. The MAX14830 contains advanced UARTs and baud-rate generators with a synchronous serial-data interface and an interrupt generator. The MAX14830 is configured by writing an 8-bit word to the configuration registers through either SPI or I²C. These registers are organized by related function as shown in the *Register Map*.

The host controller loads transmit data into the THR register through SPI or I²C. This data is automatically pushed into the Transmit FIFOs, formatted, and sent out at TX_. The MAX14830 adds START and STOP and parity bits to the data and sends the data out at the selected baud rates. The clock configuration registers determine the baud rates, clock source selection, clock frequency prescaling, and fractional baud-rate generators.

The MAX14830 receiver detects a START bit as a high-to-low RX_ transition. An internal clock samples this data at 16 times the data rate. The received data is automatically placed in the Receive FIFOs and can then be read out of the RxFIFOs through the RHRs.

The MAX14830 features four identical UARTS. Text in this data sheet references individual UART operation, unless otherwise noted.

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_ and 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 and 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 FIFOTrigLvl 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, 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 at any time 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 the Transmit Hold Register (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]:TFifoEmptyInt 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).

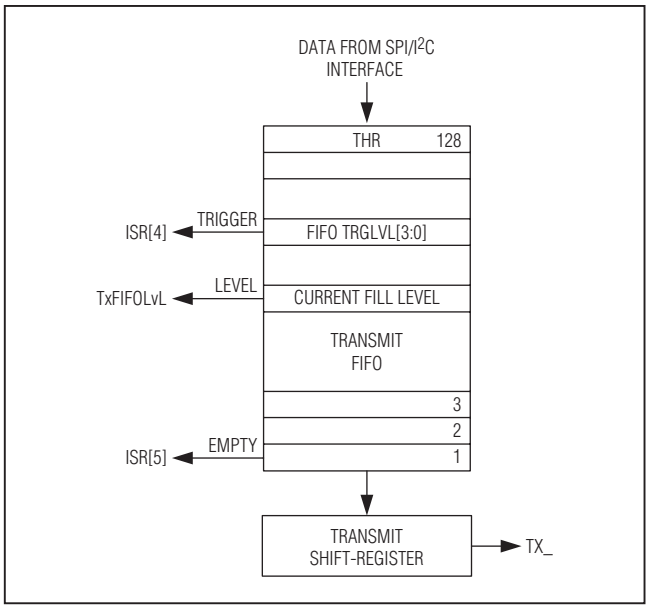


Figure 3. Transmit FIFO Signals

MICROWIRE is a trademark of National Semiconductor Corp.

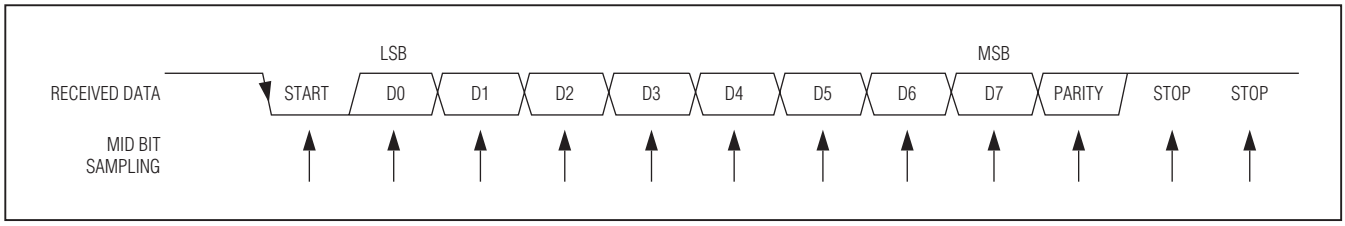


Figure 4. Receive Data Format

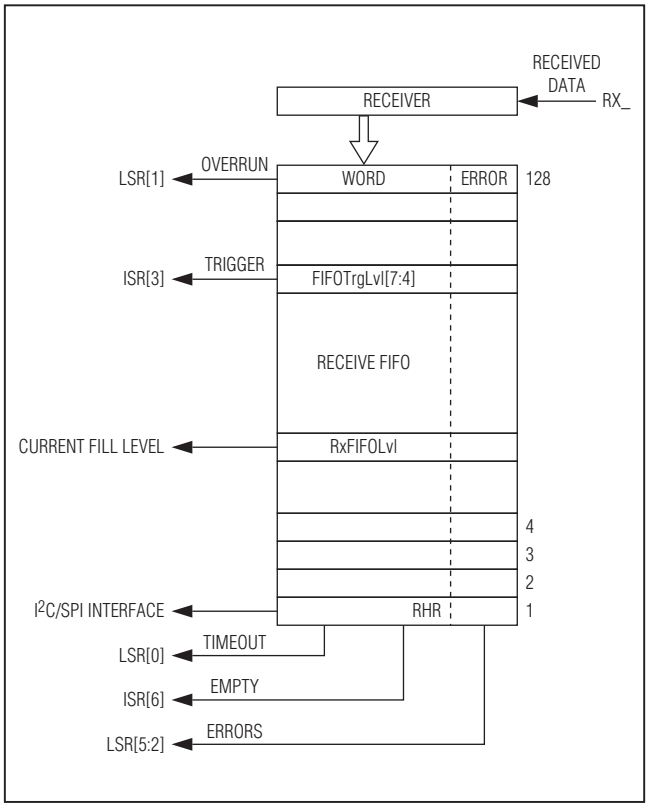


Figure 5. Receive FIFO

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

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 that 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 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 5). The host reads the data out of the Receive FIFO through the Receive Hold Register (RHR), oldest data first. The status information of the most recently read word in 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 6).

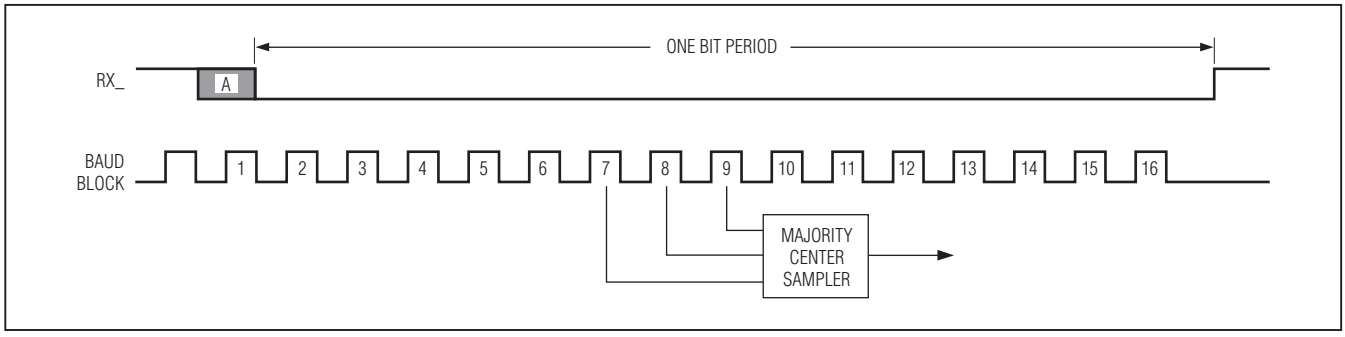


Figure 6. Midbit Sampling

The receiver can be turned off through MODE1[0]: RxDisabl. When this bit is set to 1, the MAX14830 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 or 2x (i.e. not 4x) rate mode, the MAX14830 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 MAX14830 can be clocked by an external crystal, or an external clock source. Figure 7 shows a simplified diagram of the clocking circuitry. When the MAX14830 is clocked by a crystal, the STSInt[5]: ClockReady indicates when the clocks have settled and the baud-rate generator is ready for stable operation.

Each UART baud rate can be individually programmed. To achieve fast baud rate changes, first disable the UART's clock by setting CLKDisabl to 1. Then change the baud rate divisor and subsequently enable the clock via CLKDisabl.

To check that the UART's clocking is programmed as expected, route the baud rate clock to $\overline{RTS}$ using the CLKtoRTS bit. The clock rate of this 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.

Crystal Oscillator

Set BRGConfig[6]: CLKDisabl to 0 and CLKSource[1]: CrystalEn to 1 to enable and select the crystal oscillator. The on-chip crystal oscillator circuit has load capacitances of 16pF (typ) integrated in both XIN and XOUT. Connect an external crystal or ceramic oscillator between XIN and XOUT.

External Clock Source

Connect an external clock source to XIN when not using a crystal oscillator. Leave XOUT unconnected. Set 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 the PLLConfig register. 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.

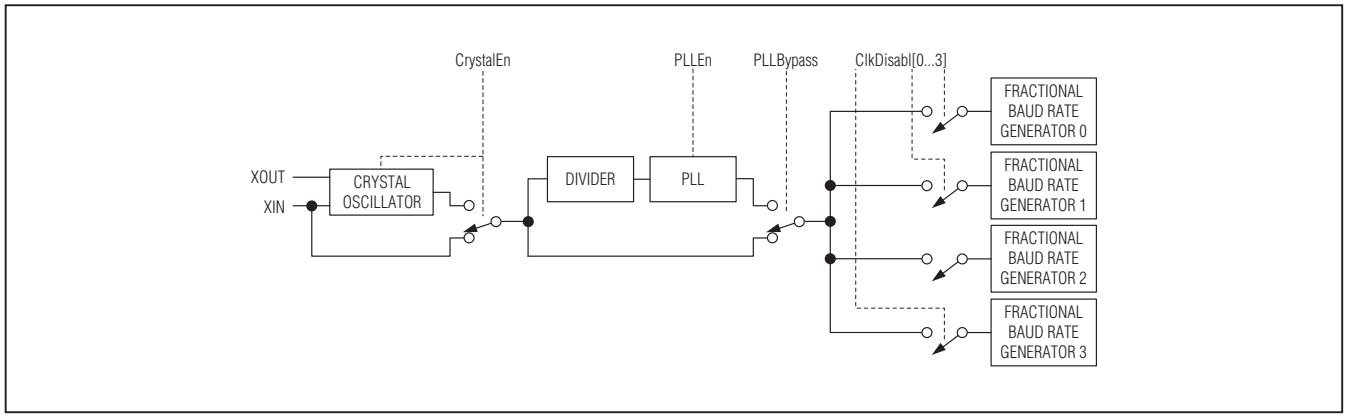


Figure 7. Clock Selection Diagram

Fractional Baud-Rate Generators

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. 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:

$$DIV = TRUNC(D)$$

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

$$FRACT = ROUND(16 \times (D-DIV)).$$

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 default rate mode.

The ideal divisor is calculated as:

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

hence DIV = 9.

$$FRACT = ROUND(4.579) = 0x05$$

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 = 9.313, where D_{ACTUAL} = DIV + (FRACT/16) and

$$BR_{ACTUAL} = 28{,}230{,}000 / (16 \times 9.3125) = 189{,}463.087 \text{ baud.}$$

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 MAX14830 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. In 4x mode only, 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 the programmed baud rate (Figure 8).

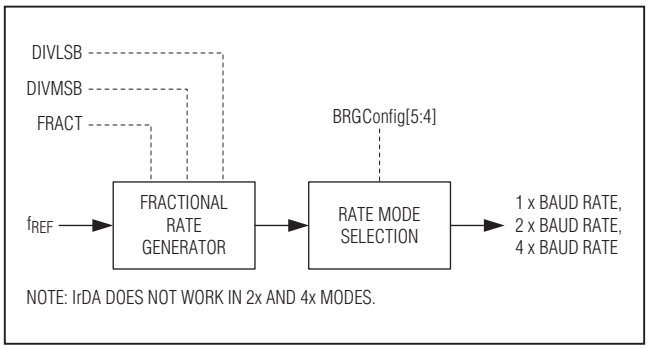


Figure 8. 2x and 4x Baud Rates

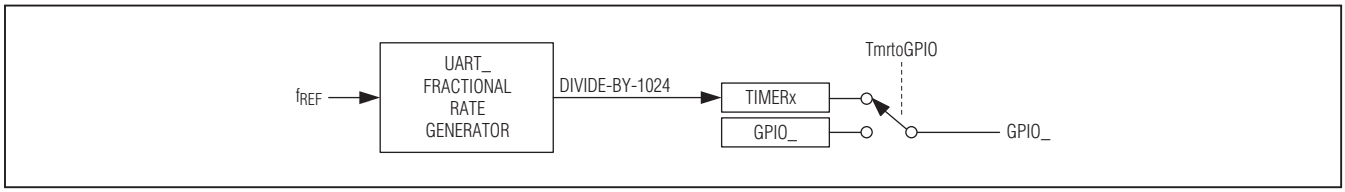


Figure 9. GPIO_ Clock Pulse Generator

Low-Frequency Timer

The general-purpose timer can be used to generate a low-frequency clock at a GPIO output and can, for example, be used to drive external LEDs. The low-frequency clock is a divided replica of a given UART baud-rate clock. The timer is internally routed to the GPIO_ outputs when enabled in the TIMER2 register as follows:

- UART0: GPIO1
- UART1: GPIO5
- UART2: GPIO9
- UART3: GPIO13

The clock pulses at the GPIOs are generated at a rate defined by the baud-rate generator and the timer divider (Figure 9). The baud-rate generator clock is divided by (1024 x TIMERx), where TIMERx is a 15-bit integer programmed into the TIMER1 and TIMER2 registers. The timer output is a 50% duty cycle clock.

UART Clock to GPIO

The MAX14830 reference clock can be routed to the GPIO0, GPIO4, GPIO8, and/or GPIO12 outputs in case a synchronous high-frequency clock is needed by another device. Enable routing a UART clock to GPIO0, GPIO4, GPIO8, and/or GPIO12 in the TxSynch register. This output clock could, for example, be used to clock another UART device (Figure 29).

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 MAX14830 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 value 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_ are 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 MAX14830 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 MAX14830 can automatically enable/disable a transceiver's transmitter and/or receiver. This relieves the host processor of this time-critical task.

The $\overline{\text{RTS\_}}$ output is used to control the transceivers' transmit enable input and is automatically set high when the MAX14830'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 10 shows a typical MAX14830 connection in a RS-485 application.

The $\overline{RTS\_}$ output can be set high in advance of TX_ transmission by a programmable time period called the setup time (Figure 11). The setup time is programmed through HDplxDelay[7:4]. Similarly, the $\overline{RTS\_}$ signal can be held high for a programmable period after the transmitter has completed transmission. The hold time is programmed through HDplxDelay[3:0].

Transmitter Triggering and Synchronization

The MAX14830 allows synchronization of transmitters so that selected UARTs start transmitting data when a trigger command is received. Optional delays can also be programmed, which delay the start of transmission after a trigger command is received. A UART's transmitter can be assigned one of 16 possible SPI/I²C trigger commands. A trigger command is defined as any of 16 special values written into the GloblComnd register (see the *GloblComnd* section for more information). When a byte is written into the GloblComnd register, UART select bits (U0 and U1) are ignored by the MAX14830, and the GloblComnd applies to all four UARTs. Transmission is initiated when the MAX14830 receives the assigned SPI/I²C trigger command if the selected transmitter is initially disabled and data has been loaded into its TxFIFO.

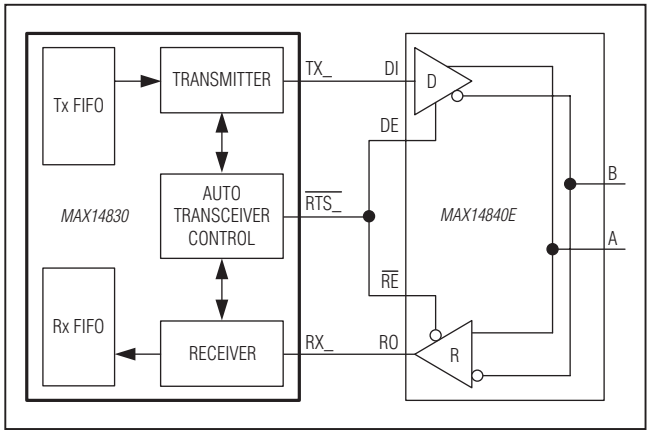


Figure 10. Auto Transceiver Direction Control

Enable and configure transmitter synchronization in the TxSynch register. Triggering and synchronization requires that the TxFIFOs are disabled before the trigger is received. This can be done by setting the MODE1[1] bit to 1 or by utilizing the auto transmitter disable function (TxSynch[4] is 1).

Transmitter Synchronization

Synchronize multiple UARTs so their transmitters start transmission simultaneously by assigning a common trigger command to the UARTs that should be synchronized.

Intrachip and Interchip Synchronization

Intrachip transmitter triggering occurs when any of the four UARTs in a MAX14830 are triggered by one trigger command. This type of synchronization is supported in both SPI and I²C modes, as the trigger commands are global commands that are received by all four UARTs simultaneously.

Interchip transmitter triggering occurs when the UARTs in different MAX14830 devices are synchronized. This type of synchronization is achievable in SPI mode only. Pull the $\overline{CS}$ of all the MAX14830 devices on the bus low during the SPI master's write trigger command so that the commands are received by all UARTs on the shared SPI bus.

I²C protocol does not allow simultaneous addressing of multiple devices.

Delayed Triggering

A delay can be programmed for delaying the start of transmission after the reception of an assigned trigger command. Set the delay by programming the SynchDelay1 and SynchDelay2 registers.

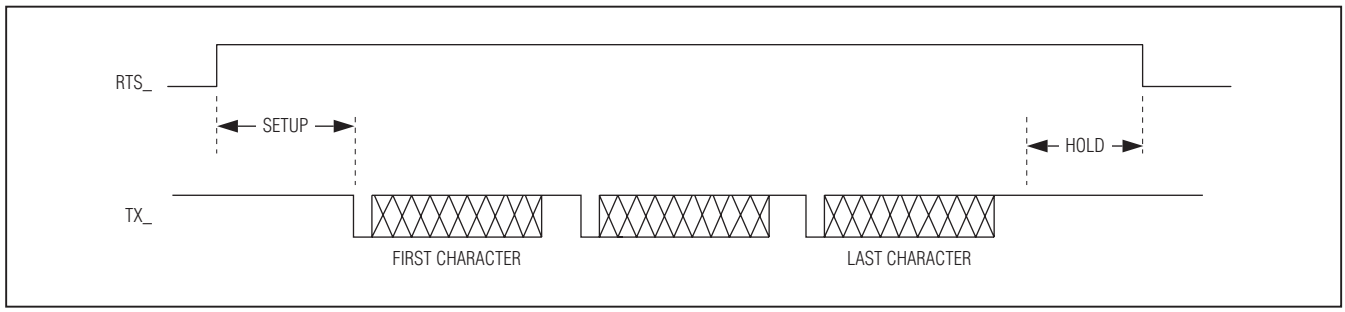


Figure 11. Setup and Hold times in Auto Transceiver Direction Control

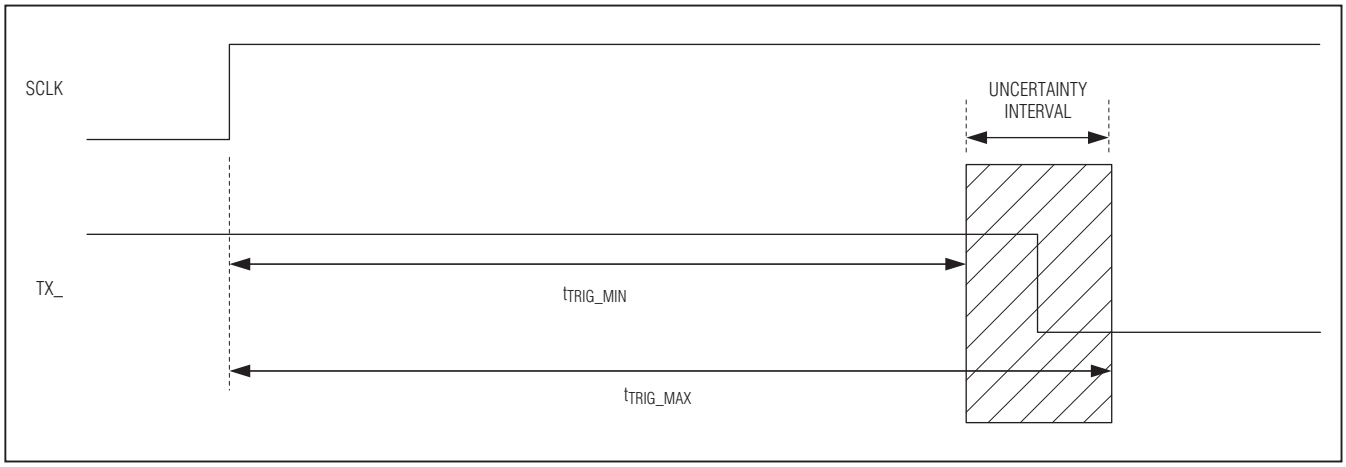


Figure 12. Single Transmitter Trigger Accuracy

Trigger Accuracy

The delay between the time when the MAX14830 receives a trigger command and the time when the associated transmitter starts transmission is made up of a fixed, deterministic portion and a variable, random component. Both portions of the delay are dependent on the UART's clock and baud rates. When the fractional divider is not used, the intrinsic trigger delay, t_{TRIG}, is bounded by the following limits:

$$\frac{5 \times BR}{16} \le t_{TRIG} \le \frac{6 \times BR}{16}$$

where BR is the fractional divider output clock period. This equation is independent on the rate mode. The reference point is the time when the trigger command is received by the MAX14830. This occurs on the final (i.e. the 16th) SPI clock's low-to-high transition (Figure 12).

When the fractional baud-rate generator is used, the random portion is larger than one UART clock period.

Synchronization Accuracy

When synchronizing multiple UART transmitters, the accuracy of the TX_ transmitter outputs is based on the triggering delays of each UART (Figure 13). This skew has a baud-rate dependent component, similar to the trigger accuracy equation for a single transmitter output. Calculate the TX_ transmitter output skew using the following equation:

$$t_{TRIGSKEW}(max) \le \frac{6 \times BR_S - 5 \times BR_F}{16}$$

where BR_S is the fractional divider output clock of the lower/slower baud-rate UART and BR_F is the fractional divider output clock of the higher/faster baud-rate UART.

Auto Transmitter Disable

The MAX14830 allows automatic disabling of the transmitter. Enable auto transmitter disabling functionality by setting TxSynch[4] to 1. When auto transmitter disabling is activated, the MAX14830 disables the specified transmitter after it completes sending all the data in its TxFIFO. New data can then be loaded into the TxFIFO. A disabled transmitter does not send out data on the TX_ output when data is present in its TxFIFO.

To enable transmission, either clear the TxAutoDis bit in the TxSynch register or toggle the TxDisabl bit in the MODE1 register.

Echo Suppression

The MAX14830 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 MAX14830'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 MAX14830 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 (Figure 14). The hold time delay is set by the HDplxDelay[3:0] register. See the *HDplxDelay Register* section for more information.

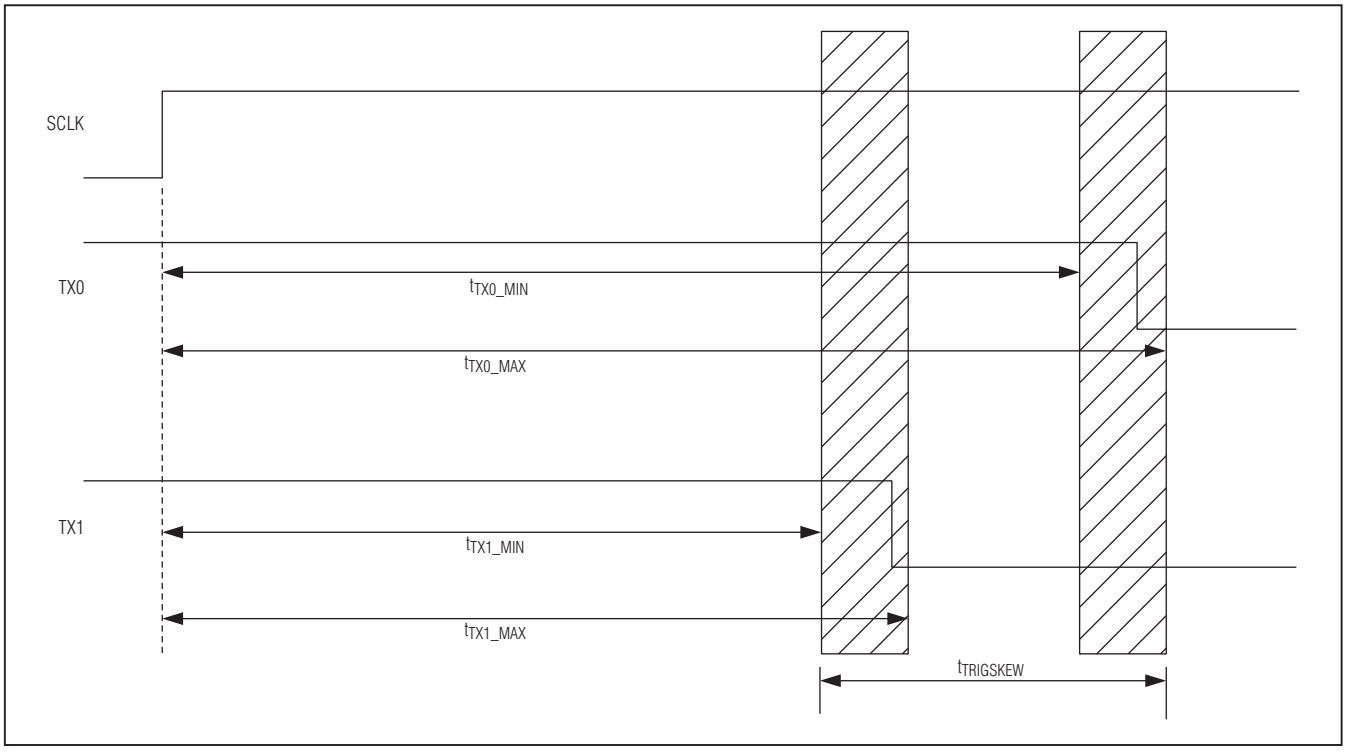


Figure 13. Multiple Transmitter Synchronization Accuracy

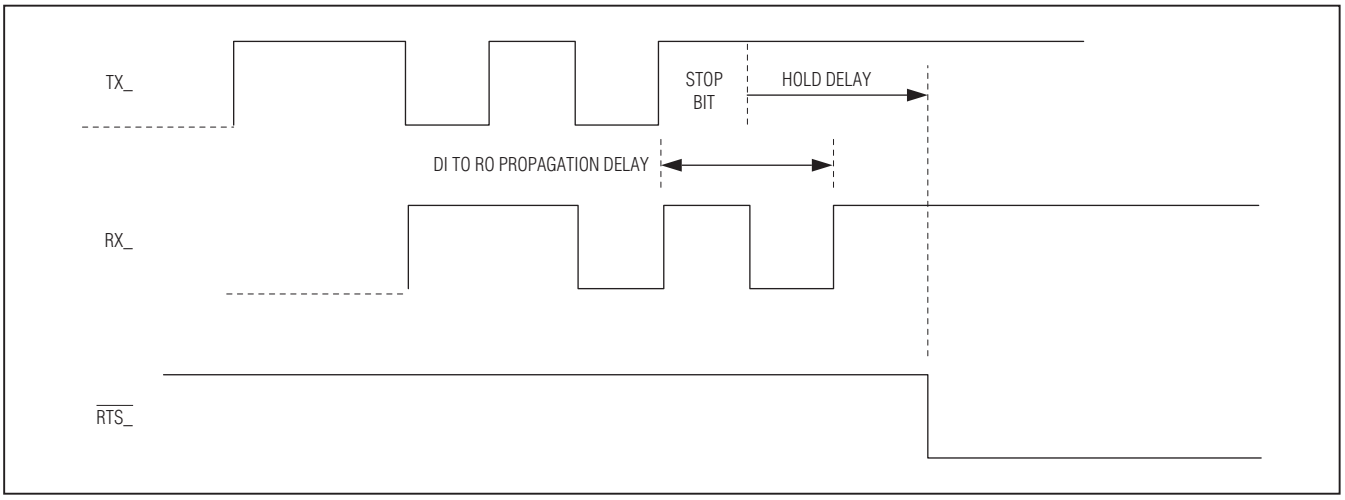


Figure 14. Echo Suppression Timing