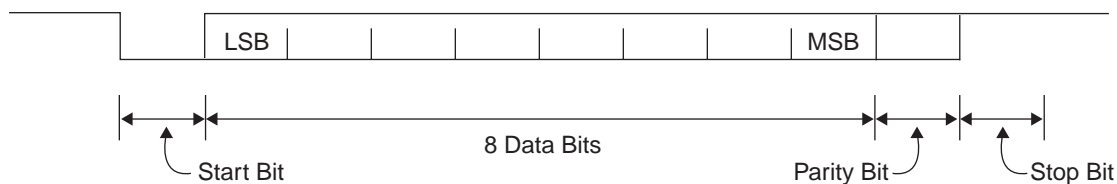


Introduction

The Universal Asynchronous Receiver Transmitter (UART) is a popular and widely-used device for data communication in the field of telecommunication. There are different versions of UARTs in the industry. Some of them contain FIFOs for the receiver/transmitter data buffering and some of them have the 9 Data bits mode (Start bit + 9 Data bits + Parity + Stop bits). This application note describes a fully configurable UART optimized for and implemented in a variety of Lattice devices, which have superior performance and architecture compared to existing semiconductor ASSPs (application-specific standard products).

This UART reference design contains a receiver and a transmitter. The receiver performs serial-to-parallel conversion on the asynchronous data frame received from the serial data input SIN. The transmitter performs parallel-to-serial conversion on the 8-bit data received from the CPU. In order to synchronize the asynchronous serial data and to insure the data integrity, Start, Parity and Stop bits are added to the serial data. An example of the UART frame format is shown in Figure 1 below.

Figure 1. UART Frame Format: (1 Start Bit, 8 Data Bits, 1 Parity Bit, 1 Stop Bit)



This design can also be instantiated many times to get multiple UARTs in the same device. For easily embedding the design into a larger implementation, instead of using tri-state buffers, the bi-directional data bus is separated into two buses, DIN and DOUT. The transmitter and receiver both share a common internal Clk16X clock. This internal clock which needs to be 16 times of the desired baud rate clock frequency is obtained from the on-board clock through the MCLK input directly. However, when implementing the design into ispMACH™ 5000VG devices, the Clk16X clock can be generated flexibly through the ispMACH 5000VG on-chip PLL by using MCLK as the PLL reference clock input.

Features

- Functionally compatible with the NS16450 UART
- Faster performance than industry standard hardwired devices
- Inserts or extracts standard asynchronous communication bits (Start, Stop and Parity) to or from the serial data
- Holding and shifting registers eliminate the need for precise synchronization between the CPU and serial data
- Standard CPU Interface
- Separate interrupt lines for Data Received (RxRdyn) and Data Transmitted (TxRdyn)
- A common interrupt line for all internal UART Data and Error events. Interrupt conditions include: receiver line errors, receiver buffer available, transmit buffer empty and when a modem status flag change is detected
- Fully prioritized interrupt system control
- MODEM interface functions (CTS, RTS, DSR, DTR, RI and DCD)
- Fully programmable serial interface characteristics:
 - 5, 6, 7 or 8-bit characters
 - Even, odd, or no-parity bit generation and detection
 - 1, 1.5 or 2-stop bit generation and detection

- False Start bit detection
- Line break generation and detection
- Interactive control signaling and status reporting capabilities
- Separate input and output data buses for use as an embedded module in a larger design
- Transmitter enabled by new data write to Transmit Holding Register
- Receiver synchronizes off the Start bit
- Receiver samples all incoming bits at the center of each bit

Compatibility

This UART implementation supports a higher operating speed and lower access time compared to the NS16450 function. It is functionally equivalent to the NS16450 with the following variations:

1. The Programmable Clock and Baud Rate Generator is not implemented. The UART internal clock Clk16X runs at 16x of the receiver/transmitter baud clock. Users can create this clock from the on-board system clock (via the on-chip PLL when implementing in ispMACH 5000VG devices or via a separate Divide-by-N clock divisor when implementing in other Lattice CPLD devices) to obtain the desired baud clock frequency with the PLD. The associated control registers (Divisor Latch) are also not implemented.
2. The External Crystal Oscillator connections (XTAL1 and XTAL2) are not supported. Alternatively, an external user-specified MCLK clock drives the transmitter and receiver clock input either directly or through the PLL (ispMACH 5000VG devices only).
3. The auxiliary user-defined output pins are not supported, because the user can modify the reference design, and output any internal signals in the Lattice ISP™ PLD device.
4. The user-accessible Scratchpad register is not supported because all Lattice ISP CPLDs have the built-in User Electronic Signature (UES) register feature.
5. The bidirectional bus D[7:0] is divided into an input bus DIN[7:0] and another output bus DOUT[7:0]. In case the design is used in a larger CPLD design as an embedded module, the DOUT[7:0] should directly drive from the output data multiplexer and not be tri-stated.
6. The internal loop-back capability for on-board diagnostics is not supported because all Lattice CPLD devices have ISP capability, and can perform in-system test with various patterns.

Internal Registers

The UART contains two data buffering registers (RBR and THR), three status registers (IIR, LSR, and MSR), and three control registers (IER, LCR, and MCR). These registers and their addresses are shown below. The further discussions of the data buffering registers are in the Transmitter and Receiver sections. Note that the programmable baud rate control register and the user accessible Scratchpad register are not implemented in this design. For details, refer to the Compatibility section.

Addr[2:0]	Access Mode	Acronym	Register
000	Read only	RBR	Receiver Buffer Register
000	Write only	THR	Transmitter Holding Register
001	Write only	IER	Interrupt Enable Register
010	Read only	IIR	Interrupt Identification Register
011	Write only	LCR	Line Control Register
100	Write only	MCR	MODEM Control Register
101	Read only	LSR	Line Status Register
110	Read only	MSR	MODEM Status Register

Interrupt Enable Register (IER, Addr=001)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	MSI	RLSI	THRI	RBRI

RBRI: Receiver Buffer Register Interrupt (1 = Enable, 0 = Disable)

THRI: Transmitter Hold Register Interrupt (1 = Enable, 0 = Disable)

RLSI: Receiver Line Status Interrupt (1 = Enable, 0 = Disable)

MSI: MODEM Status Interrupt (1 = Enable, 0 = Disable)

Interrupt Identification Register (IIR, Addr=010)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	ID2	ID1	ID0	STAT

Four prioritized interrupt levels and sources:

Level	ISR Bit [3:0]	Source of Interrupt	Interrupt Reset Control
None	x x x 1	None	None
Highest	0 1 1 0	LSR error flags (OE/PE/FE/BI)	Reading LSR
Second	0 1 0 0	LSR receiver data ready flag (DR)	Reading RBR
Third	0 0 1 0	LSR flag THR Empty (THRE)	Reading IIR or Writing THR
Fourth	0 0 0 0	MSR bit 0,1,2,3 indicate MODEM line status changed (DDCD/TERI/DDSR/DCTS)	Reading MSR

Line Control Register (LCR, Addr=011)

The Line Control Register (LCR) is used to specify the asynchronous data communication format:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SB	SP	EPS	PEN	STB	WLS1	WLS0

Lattice Semiconductor

Word Length Select (Bit1-0):

Bit 1	Bit 0	Word Length
0	0	5
0	1	6
1	0	7
1	1	8

Stop Bit Length (Bit 2):

Bit 2	Word Length	Stop Bit Length (Bit Times)
0	5,6,7,8	1
1	5	1.5
1	6,7,8	2

Parity Selection (Bit 5-3):

Bit 5 SP	Bit 4 EPS	Bit 3 PEN	Parity Selection
x	x	0	No Parity
0	0	1	Odd Parity
0	1	1	Even Parity
1	0	1	Stick Parity 1
1	1	1	Stick Parity 0

Set Break (Bit 6):

When enable the Break control bit causes a Break condition to be transmitted (SOUT is forced to a logic 0 state). This condition exists until disabled by resetting this bit to a logic 0.

MODEM Control Register (MCR, Addr=100)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	RTS	DTR

Loopback mode for testing (Bit 4), Auxiliary user-defined output Out2 (Bit 3) and Out1 (Bit 2) are removed, because it can be implemented with in-system programmability (ISP). For details, refer to the Implementation section of this document.

DTR: This bit controls the Data Terminal Ready (DTRn) output.
 DTR=0: force DTRn output to a logic 1 (normal default)
 DTR=1: force DTRn output to a logic 0

RTS: This bit controls the Request to Send (RTSn) output
 RTS=0: force RTSn output to a logic 1 (normal default)
 RTS=1: force RTSn output to a logic 0

Line Status Register (LSR, Addr=101)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	TEMT	THRE	BI	FE	PE	OE	DR

DR: Receiver Data Ready

DR indicates status of RBR. It will be set to logic 1 when RBR data is valid and will be reset to logic 0 when RBR is empty. When line errors (OE/PE/FE/BI) happen, DR will also be set to logic 1 and RBR will be updated to reflect the Data bits portion of the frame. Pin RxRDYn is actually a complement of this bit.

OE: Overrun Error

This bit will be set when the next character is transferred into RBR before the previous RBR data is read by the CPU. Even though DR will still be 1 when OE is set to logic 1, the previous frame data stored in RBR which is not read by the CPU is trashed and can't be recovered.

PE: Parity Error

This bit will be set to logic 1 only when the Parity is enabled and the Parity bit is not at the logic state it should be. For Even Parity, the Parity bit should be 1 if an odd number of 1s in the Data bits is received, otherwise, the Parity bit should be 0. For Odd Parity, the Parity bit should be 1 if an even number of 1s in the Data bits is received, otherwise, the Parity bit should be 0. For Stick Parity '1', the Parity bit should be 1. For Stick Parity '0', the Parity bit should be 0.

FE: Framing Error

FE will be reset to logic 0 whenever SIN is sampled high at the center of the first Stop bit, regardless of how many Stop bits the UART is configured to.

BI: Break Interrupt

BI will be set to logic 1 whenever SIN is low for longer than the whole frame (the time of Start bit + Data bits + Parity bit + Stop bits), not at the SIN rising edge where Break is negated. If SIN is still low after BI is reset to logic 0 by reading LSR, BI will not be set to logic 1 again. Since Break is also a Framing error, FE will also be set to 1 when BI is set.

THRE: THR Empty

THRE will be set to logic 1 whenever THR is empty which indicates that the transmitter is ready to accept new data to transmit. Pin TxRDYn is actually a complement of this bit.

TEMT: Both THR and TSR are Empty

This bit will be set to logic 1 when THRE is set to 1 and the last Data bit in the TSR is shifted out through SOUT.

The four error flags (OE, PE, FE and BI) of LSR will be reset to logic 0 after a LSR read.

MODEM Status Register (MSR, Addr=110)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

Bit 7: DCD (Data Carrier Detect) is the complement of DCDn input

Bit 6: RI (Ring Indicator) is the complement of RIn input

Bit 5: DSR (Data Set Ready) is the complement of DSRn input

Bit 4: CTS (Clear to Sent) is the complement of CTSn input

Bit 3: DDCD (Delta DCD) indicates that the DCDn input has changed state

Bit 2: TERI (Trailing Edge of RI) indicates that the RIn input has changed from a low to high state

Bit 1: DDSR (Delta DSR) indicates that the DSRn has changed state since last time read by CPU

Bit 0: DCTS (Delta CTS) indicates that the CTSn has changed state since last time read by CPU

Whenever bit 0-3 is set to logic 1, a MODEM status interrupt is generated if the interrupt is enabled. These four bits will be reset to logic 0 whenever CPU reads MSR.

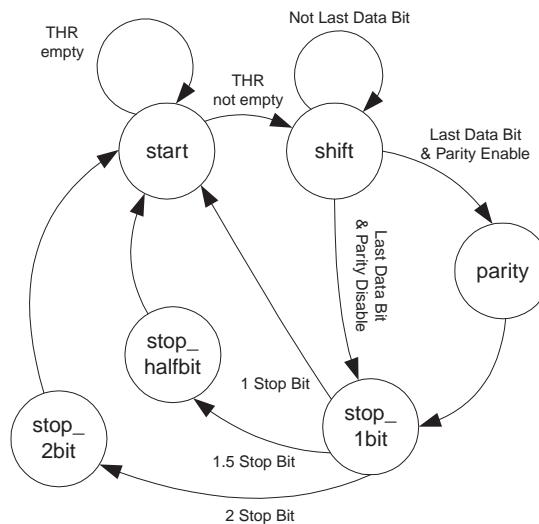
Transmitter

The serial transmitter section consists of an 8-bit Transmitter Hold Register (THR) and Transmitter Shift Register (TSR). There are two ways to indicate the status of THR: an independent TxRDY output pin or the THRE flag in the Line Status Register (LSR). When the THR is empty, pin TxRDYn will be low active, and the THR empty flag in LSR will be set to a logic 1. Only when the THR is empty, can a write operation be performed to transfer the data from CPU to THR without trashing the previous data. After the data is loaded in THR, the THR empty flag in LSR will be reset to logic 0, and pin TxRDYn will go inactive high.

The serial data transmission will be automatically enabled after the data is loaded into THR. First a Start bit (logic 0) is transmitted and the data in THR is parallel loaded to TSR automatically. Then the Data bits will be shifted out of TSR with certain word lengths defined in Line Control Register (LCR) followed by the Parity bit if parity is enabled. Finally, the Stop bit (logic 1) is generated to indicate the end of the frame. This serial data frame (Start bit + Data bits + Parity bit + Stop bit) will be transmitted at the rate of 1/16 of Clk16X frequency. After a frame is fully transmitted, another frame will be transmitted immediately if THR is not empty (due to a THR write occurring during the first frame of transmission). This automatic sequencing causes the frames to be transmitted back-to-back which increases the transmission bandwidth. When no transmission is taking place, the SOUT pin is held in the high state.

The behavior of the transmitter is controlled by the FSM (Finite State Machine) shown in Figure 3:

Figure 3. Transmitter State Machine



<start>: When the UART is reset by MR pin, the transmitter FSM will be reset to this state. When in this state, the transmitter is waiting to assert the Start bit. A Start bit will be asserted as soon as the THR is not empty. Once a low SOUT (Start bit) is asserted, the FSM will switch to <shift> state.

- <shift>:** When the FSM is in this state, it's waiting for the last (most significant) Data bit to be shifted out. After the last Data bit is shifted out, the FSM will switch to <parity> state if parity is enabled. Otherwise, it will switch to <stop_1bit> state.
- <parity>:** When the FSM is in this state, the last Data bit is still in transmission. When the transmission is complete, the FSM will assert the Parity bit. Once the Parity bit is asserted, the FSM switch to the <stop_1bit> state.
- <stop_1bit>:** No matter if the Stop bit is configured to be 1, 1.5 or 2 bits long, the FSM will always switch to this state, wait for a baud clock cycle, and then assert the Stop bit(s). For 1 Stop bit, the FSM switches back to <start> state and waits to assert the Start bit of another frame. For 1.5 Stop bit, it switches to <stop_halfbit> state and stays there for just half baud clock cycle before switching to <start> state. For 2 Stop bits, it switches to <stop_2bit> state then switches back to <start> state. Note that the Stop bit(s) is asserted at the time when the FSM is leaving the <stop_1_bit> state.
- <stop_halfbit>:** This state is for 5-bit Data bits with 1.5 Stop bit. The FSM will stay in this state for only half baud clock cycle and then switch to <start> state.
- <stop_2bit>:** When the FSM is in this state, the first Stop bit is in transmission. It waits for a baud clock cycle, then asserts the second Stop bit and switches to the <start> state.

Receiver

The serial receiver section also contains an 8-bit Receiver Buffer Register (RBR) and Receiver Shift Register (RSR). The status of RBR can be provided by either independent pin RxRDYn or the Receiver Data available flag (DR) in LSR.

Since the serial frame is asynchronous to the receiving clock, a high to low transition of SIN pin will be treated as the Start bit of a frame. However, in order to avoid receiving a incorrect data due to SIN signal noise, the False Start Bit Detection feature is implemented in the design which requires the Start bit to be low at least 50% of the receiving baud rate clock cycle. Since the internal clock Clk16X is 16 times the receiving/transmitting baud rate clock frequency, the Start bit needs to be low at least 8 Clk16X clocks to be considered as a valid Start bit.

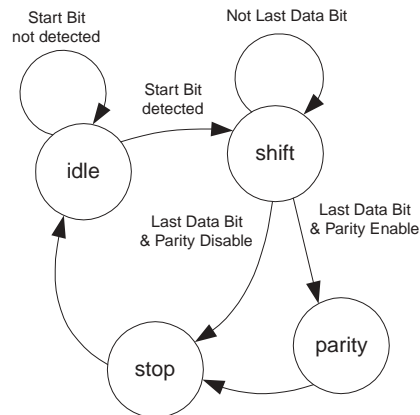
Once a valid 8 Clk16X clocks Start bit is received, the Data bits and Parity bit will be sampled every 16 Clk16X clocks (the receiving baud rate). If the Start bit is exactly 16 Clk16X clocks long, each of the following bits will be sampled at the center of the bit itself. LSR will be updated to show the received frame status when any of the line errors (Overrun error, Parity error, Framing error, Break) is detected.

Whenever the Framing error is detected, the UART assumes that the error was due to the Start bit of the following frame and tries to resynchronize it. To do this, it samples the Start bit twice based on the Clk16X clock. If both samples of the SIN are low, the UART will take in the following frame's Data bits after the 8 Clk16X clocks Start bit is sampled. The resynchronization will not occur for the Framing error caused by the Break.

When the data is available in RBR, the RxRDYn will be low active, and the Receiver Data available flag (DR) in LSR will be set to logic 1 to inform the CPU that the data is ready to be read.

The behavior of the receiver is controlled by the FSM shown in Figure 4:

Figure 4. Receiver State Machine



<idle>: When the MR pin resets the UART, the receiver FSM will be reset to this state. When in this state, it's waiting for SIN to be changed from high to low and stay low for 8 Clk16X clocks to be considered a valid Start bit. Once a valid Start bit is detected, the FSM will switch to <shift> state.

<shift>: When the FSM is in this state, it waits 16 Clk16X clocks for each Data bit to shift into RSR. After the last Data bit is shifted in, the FSM will switch to <parity> state if parity is enabled. Otherwise, it will switch to <stop> state.

<parity>: When the FSM is in this state, it waits for 16 Clk16X clocks and then samples the Parity bit. Once the Parity bit is sampled, the FSM switch to the <stop> state.

<stop>: No matter if the Stop bit length is configured to be 1, 1.5 or 2 bits long, the FSM will always wait for 16 Clk16X clocks and then sample the Stop bit. As long as a logic 1 is sampled at the Stop bit, the Framing error flag (FE) in LSR will not be set. The receiver does not check whether the Stop bit is in the right length as configured. The FSM switches back to <idle> state after the Stop bit sampling.

Interrupt

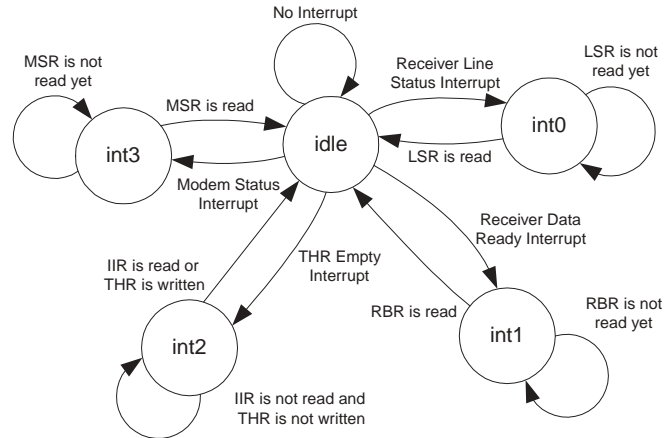
The UART common interrupt request pin INTR will go high active when any of the interrupt conditions is matched and enabled by Interrupt Enable Register (IER).

The UART prioritizes interrupts into four levels to minimize external software interaction, and records these in the Interrupt Identification Register (IIR). The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and MODEM Status.

Performing a read cycle on IIR freezes all interrupts and indicates the highest priority pending interrupt to the CPU. No other interrupts are acknowledged until the pending interrupt is serviced. Whenever the IIR is read, the current pending interrupt is cleared. A lower level interrupt may be seen at next IIR reading.

The behavior of the interrupt is controlled by the FSM shown in Figure 5:

Figure 5. Interrupt State Machine



<idle>: When the MR pin resets the UART, the interrupt FSM will be reset to this state. When in this state, it is waiting for the enabled interrupt conditions to be true and then switches to the interrupt state with highest priority.

<int0>: The FSM switches to this state when the highest priority level interrupt occurs. It stays at this state until the LSR is read.

<int1>: The FSM switches to this state when the second priority level interrupt occurs. It stays at this state until the RBR is read.

<int2>: The FSM switches to this state when the third priority level interrupt occurs. It stays at this state until the IIR is read or after THR is written.

<int3>: The FSM switches to this state when the fourth priority level interrupt occurs. It stays at this state until the MSR is read.

The interrupts continue to be generated as long as the corresponding enable bit in IER is set and the corresponding interrupt condition is matched. Since the interrupt state machine is running at Clk16X, the INTR pin might be low for a Clk16X clock and then return to high again.

Signals (Pin Descriptions)

Pin Name	Type	Pin Description
MR	In	Reset control, active high
MCLK	In	Master clock input. When implementing this design in ispMACH 5000VG devices, MCLK will be fed into the PLL clock input and the PLL clock output will be used as the UART internal Clk16X clock. For other Lattice CPLD devices, MCLK will be connected directly to Clk16X. This internal clock is 16 times the receiving/transmitting baud rate clock frequency.
CS	In	Chip Select
RDn	In	Read data strobe, active low. When it is low and chip is selected, the CPU reads status information or data from the selected UART register
WRn	In	Write data strobe, active low. When it is low and chip is selected, the CPU writes control words or data into the selected UART register
ADSn	In	Address Strobe, active low. When active, it enables A[2:0] and CS to drive the internal logic. A low-to-high transition latches the A[2:0] and CS state for the whole CPU cycle
A[2:0]	In	Address of internal registers.
DIN[7:0]/ DOUT[7:0]	In Out	Data Bus is separated into DIN input and DOUT output so that it can be incorporated to a larger design easily by using MUX data buses
DDIS	Out	Drive Disable, active low. This pin goes to a logic 0, when the CPU is reading data from UART, this signal can be used to disable or control the direction of external data bus transceiver
INTR	Out	Interrupt Request, active high. Used to request service from CPU whenever one of the following four conditions happens: receiver line errors, receiver buffer available, transmit buffer empty and when a modem status flag change is detected.
SIN	In	Receiver Serial Data Input from the communications link
SOUT	Out	Transmitter Serial Data Output to the communications link
TxRDYn	Out	Transmitter Ready. When there is no data in Transmitter Holding Register, it will be low active. Once it is activated, it will go inactive after the character is loaded into the Transmitter Holding Register
RxRDYn	Out	Receiver Ready. Where there is data in Receiver Buffer Register, it will be low active. Once it is activated, it will go inactive when there is no character in the Receiver Buffer Register
DCDn	In	Data Carrier Detect, low active. Indicates that the data carrier has been detected by MODEM
CTSn	In	Clear to Send, low active. Indicates that the MODEM is ready to exchange data
DSRn	In	Data Set Ready, low active. Indicates that the MODEM is ready to establish the communication link with the UART
RIn	In	Ring Indicator, low active. Indicates that a telephone ringing signal has been received by the MODEM
DTRn	Out	Data Terminal Ready. When low, this informs the MODEM that UART is ready to establish a communication link
RTSn	Out	Request to Send. When low, this informs the MODEM that the UART is ready to exchange data

Implementation

This Lattice UART design has been implemented in several Lattice CPLD devices. The details are summarized in the table below. The ispLSI[®] device families are ideally suited to high-speed state machine intensive applications. All Lattice CPLDs feature In-System Programmability (ISP) and can be programmed up to 10,000 times.

The software used for this implementation is the Lattice ispLEVER[™] design tool. After the design was completed, the VHDL code was compiled. The use of VHDL with a CPLD allows functional and hardware changes to be made quickly and easily.

The number of macrocells required varies, depending on the addition of new functions and/or removal of unneeded features. This design uses 37 I/O pins. The maximum operating frequency currently is 55MHz with the M4A5-192/96-6VC. This is compared to the industry-standard UART device with a speed of 24MHz.

The following table lists the implementation data and performance of this UART design when fitting into different Lattice devices:

Device	Macrocells Used	GLB Level	Max. Clk16X Freq
LC51024VG-5F676ES	172	N/A	133.3 MHz
M4A5-192/96-6VC	132	N/A	107.5 MHz
ispLSI2192VE-180LT128	147	3	102 MHz
ispLSI5256VE-165LT128	156	2	104 MHz

Note: The Max CLK16X Freq is obtained by running the Timing Analysis of Lattice Design Software. Please run the timing simulation after you make changes to this design or after you merge it with your design.

7. *The VHDL source code and testbenches for this UART design are also available from Lattice.

Source code:

uart_5kvg_top.vhd : Top level source code when using ispMACH 5000VG devices.
 uart_top.vhd : Top level source code when using other Lattice CPLD devices.
 modem.vhd : MODEM control module
 rxcvr.vhd : Receiver module
 txmitt.vhd : Transmitter module

Test benches:

uart_rx_tb.vhd : Receiver tests
 uart_tx_tb.vhd : Transmitter tests
 uart_rxerr_tb.vhd : Receiver line status error tests
 uart_int_tb.vhd : Interrupt tests

Technical Support Assistance

Hotline: 1-800-LATTICE (Domestic)
 1-408-826-6002 (International)
 e-mail: techsupport@latticesemi.com