Intro to UARTs

April 2010

Why use a UART?

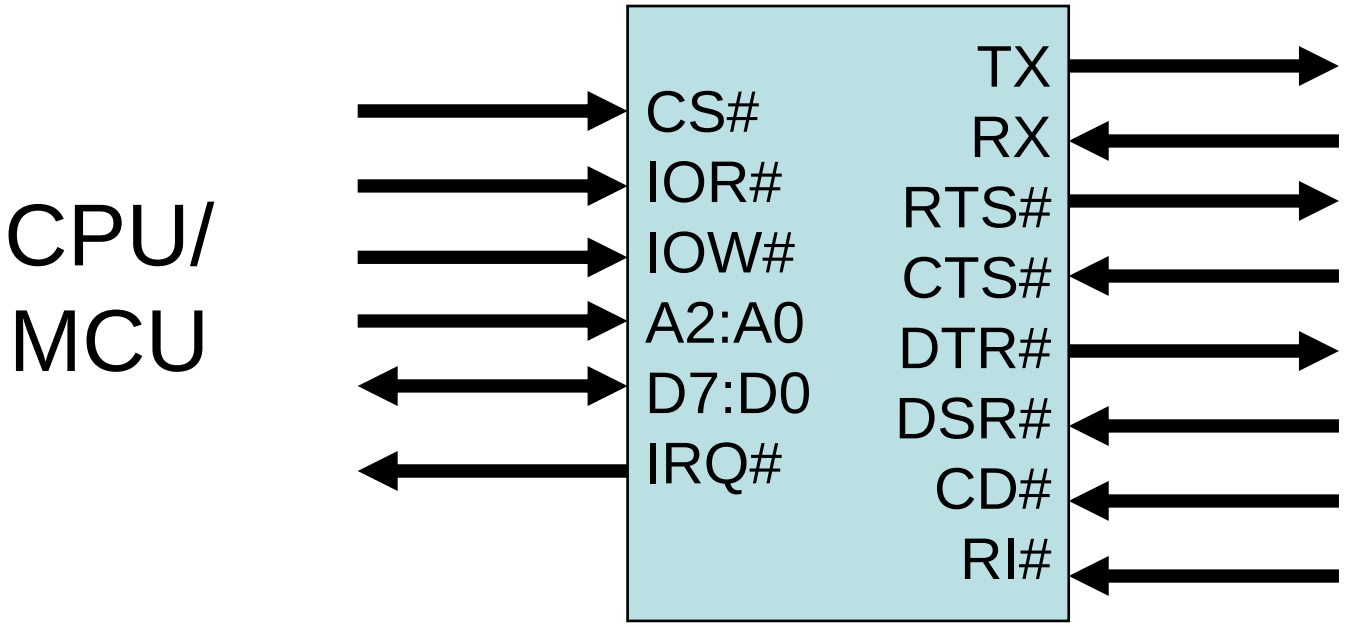
- **UARTs are everywhere!**
- **Simple way to send data from one system to another system**
- **Add additional functionality to an application**

Why use a UART from Exar?

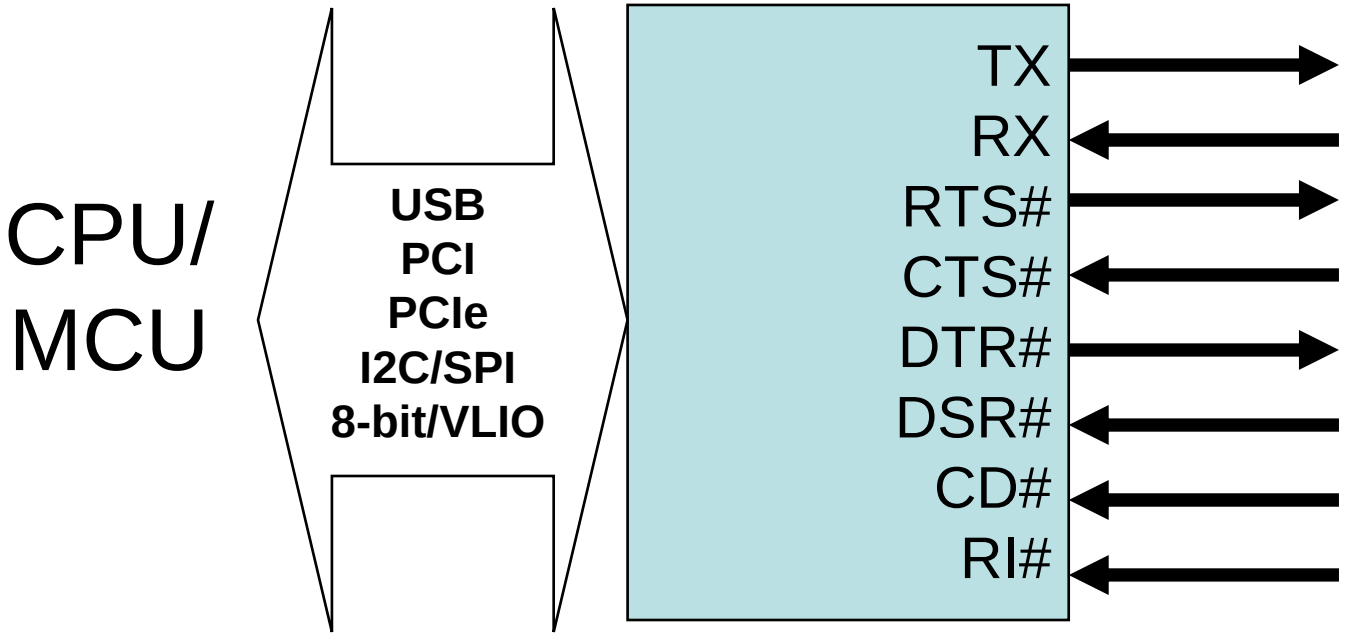
- **Largest and broadest UART portfolio**
- **Highest performance UARTs**
- **UARTs with the most enhanced features**
- **Excellent technical support**
- **Exar also has serial transceivers!**

What is a UART?

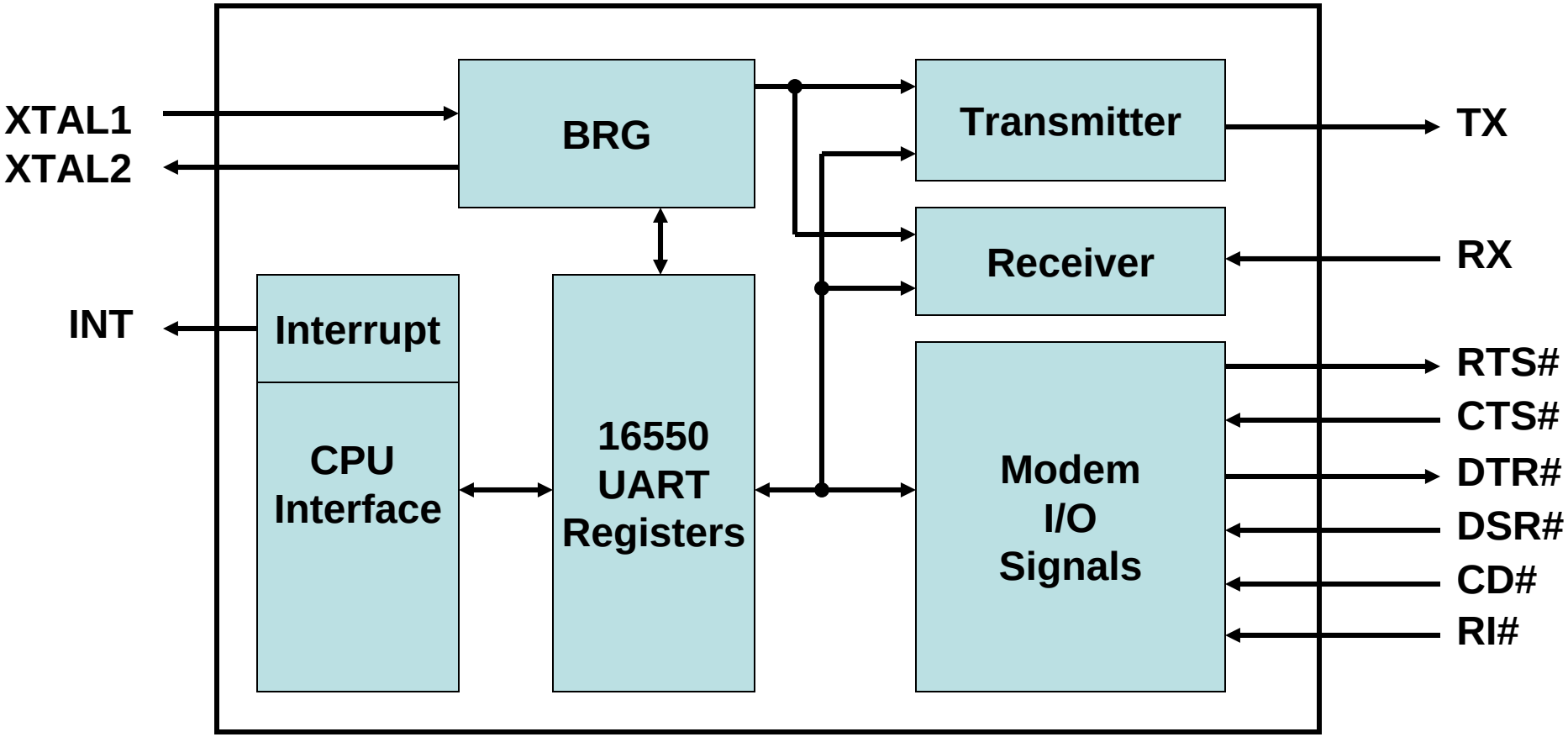
- **Universal Asynchronous Receiver/Transmitter**
- **Traditional Definition: Converts parallel (8-bit) data to serial data and vice versa**



What is a UART?



UART Block Diagram



16550 UART Registers

Address A2-A0	Register Name	Read/Write	Register Function	Comment
0 0 0	DLL – Divisor LSB	Write-Only	Divisor (LSB) for BRG	LCR bit-7 = 1
0 0 1	DLM – Divisor MSB	Read-Only	Divisor (MSB) for BRG	LCR bit-7 = 1
0 0 0	THR – Transmit Holding Register	Write-Only	Loading data into TX FIFO	LCR bit-7 = 0
0 0 0	RHR – Receive Holding Register	Read-Only	Unloading data from RX FIFO	LCR bit-7 = 0
0 0 1	IER – Interrupt Enable Register	Read/Write	Enable interrupts	
0 1 0	FCR – FIFO Control Register	Write-Only	FIFO enable and reset	
0 1 0	ISR – Interrupt Status Register	Read-Only	Status of highest priority interrupt	
0 1 1	LCR – Line Control Register	Read/Write	Word length, stop bits, parity select, send break, select divisor registers	
1 0 0	MCR – Modem Control Register	Read/Write	RTS# and DTR# output control Interrupt output enable Internal Loopback enable	
1 0 1	LSR – Line Status Register	Read-Only	RX Errors/Status TX Status	
1 1 0	MSR – Modem Status Register	Read-Only	Modem Input Status	
1 1 1	SPR – Scratch Pad Register	Read/Write	General Purpose Register	

16550 UART Registers

Address A2-A0	Register Name	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0 0 0	DLL	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0 0 1	DLM	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0 0 0	THR	W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0 0 0	RHR	R	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0 0 1	IER	R/W	0	0	0	0	MSR	LSR	TX (THR)	RX (RHR)
0 1 0	FCR	W	RX Trig Level	RX Trig Level	0	0	DMA Mode	TX FIFO Reset	RX FIFO Reset	FIFO Enable
0 1 0	ISR	R	FIFOs Enabled	FIFOs Enabled	0	0	INT Source	INT Source	INT Source	INT Source
0 1 1	LCR	R/W	Divisor Enable	Set TX Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length	Word Length
1 0 0	MCR	R/W	0	0	0	Internal Loopback	INT / OP2#	(OP1#)	RTS# Control	DTR# Control
1 0 1	LSR	R	RX FIFO Error	THR/TSR Empty	THR Empty	RX Break	RX Framing	RX Parity	RX Overrun	RX Data Ready
1 1 0	MSR	R	CD#	RI#	DSR#	CTS#	Delta CD#	Delta RI#	Delta DSR#	Delta CTS#
1 1 1	SPR	R/W	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0

Baud Rate Generator (BRG)

- **Used to generate the baud rates for both the transmitter and receiver**
- **Not required for any other function including reads and writes**
- **Crystal or External Clock**
- **16-bit divisor programmed in DLM/DLL registers**

Baud Rate Generator (BRG)

$$\text{Baud Rate} = \frac{\text{Clock Frequency}}{(\text{Sampling Rate}) \text{ X } (\text{Divisor})}$$

- **Standard clock frequencies are multiples of 1.8432 MHz**
  - 3.6864 MHz, 7.3728 MHz, 14.7456 MHz, 18.432 MHz, 22.1184 MHz
- **Standard baud rates are multiples of 9600 bps**
  - 19200 bps, 38400 bps, 57600 bps, 115200 bps, 230400 bps, 460800 bps, 921600 bps
- **Sampling rate is 16**
- **Divisor values are written into the DLM and DLL registers**
  - Divisor values are 1 to ($2^{16} - 1$) in increments of 1

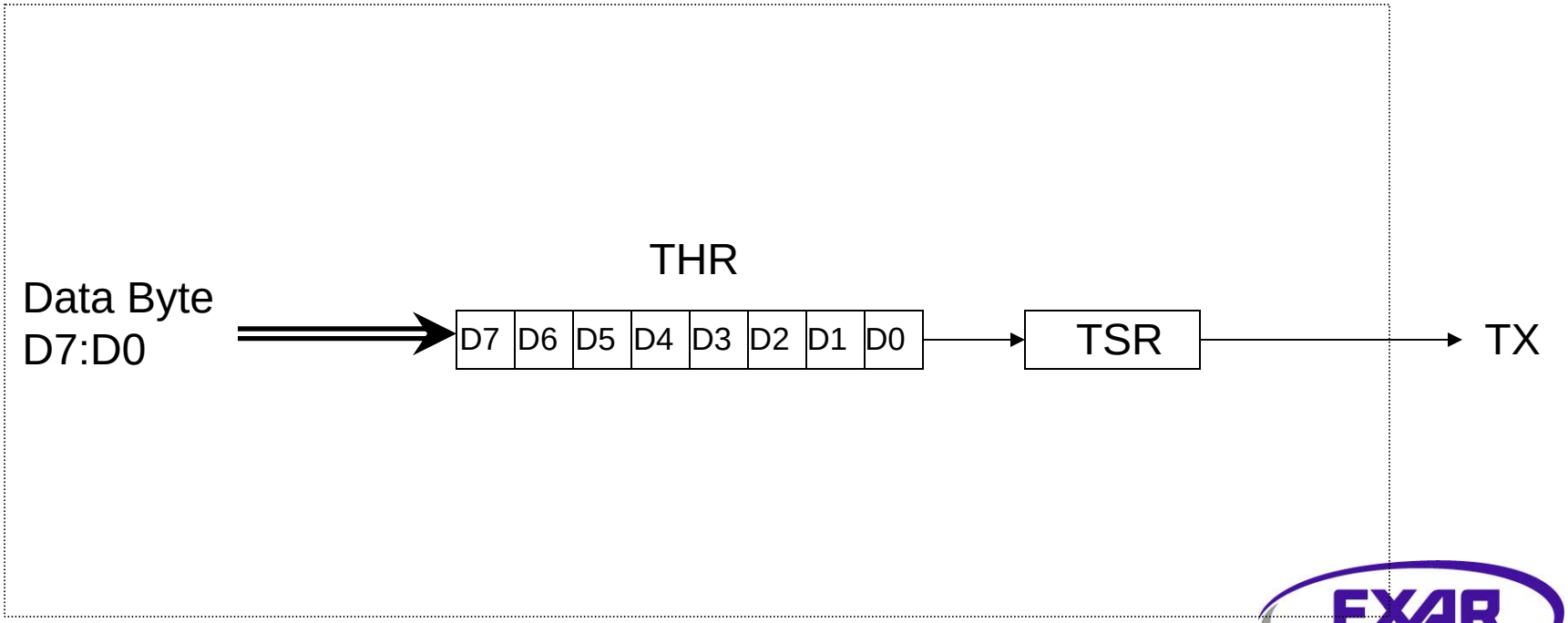
$$\text{Baud Rate} = \frac{14.7456 \text{ MHz}}{(16) \text{ X } (1)} = 921600 \text{ bps}$$

Transmitter

- **Parallel-to-serial conversion**
- **Non-FIFO Mode**
  - Transmit Holding Register (THR) and Transmit Shift Register (TSR)
- **FIFO Mode**
  - Transmit (TX) FIFO and Transmit Shift Register (TSR)
- **16X timing for bit shifting**
- **Character Framing**
- **Parity Insertion**
- **TX FIFO interrupt and status**

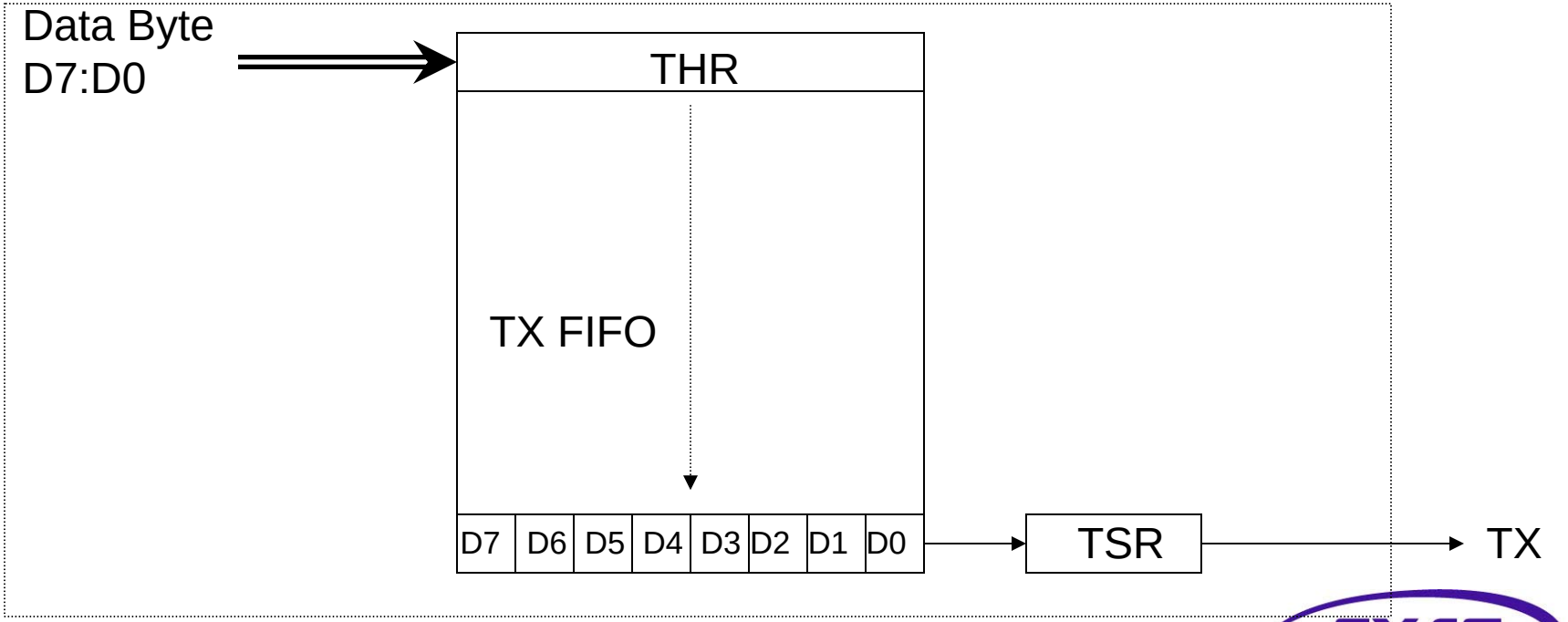
Transmitter – Non-FIFO mode

- **Write Data to Transmit Holding Register (THR)**
- **Data in THR is transferred to Transmit Shift Register (TSR) when TSR is empty**
- **TSR shifts the data out on the TX output pin**



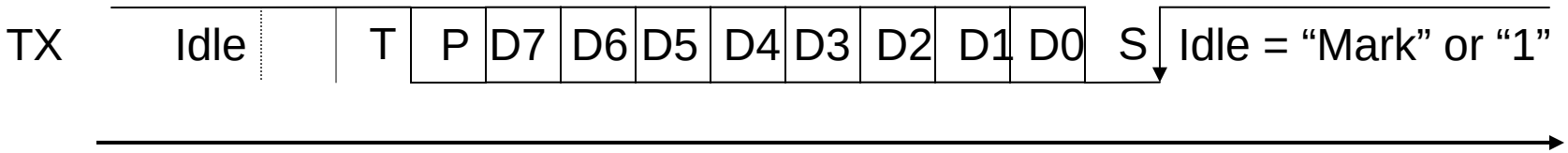
Transmitter – FIFO Mode

- **Write Data to Transmit Holding Register (THR)**
- **Transmit data is queued in TX FIFO**
- **Data in TX FIFO is transferred to Transmit Shift Register (TSR) when TSR is empty**
- **TSR shifts data out on TX output pin**



TX Character Framing

- **Start Bit**
- **Data Bits of 5, 6, 7 or 8**
- **Parity Bit**
- **Stop Bit of 1, 1.5 or 2**
- **Example:**
  - Start, 8 data, parity, with 1 stop bit

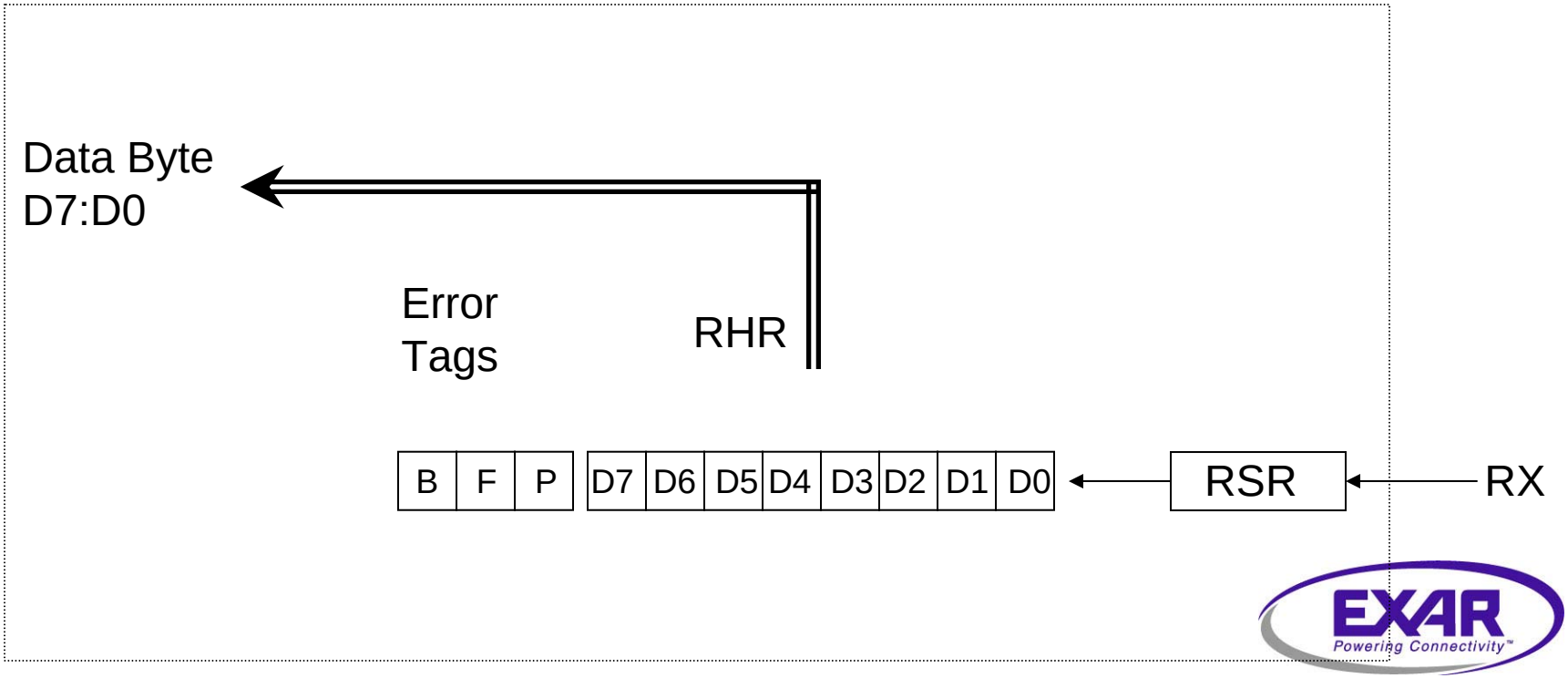


Receiver

- **Serial-to-Parallel Conversion**
- **Non-FIFO Mode**
  - Receive Holding Register (RHR) and Receive Shift Register (RSR)
- **FIFO Mode**
  - RX FIFO and RSR
- **16X timing clock for mid bit sampling**
- **Start bit detection and verification**
- **RX FIFO is 11 bits wide**
  - 8 data bits
  - 3 error bits or error tags

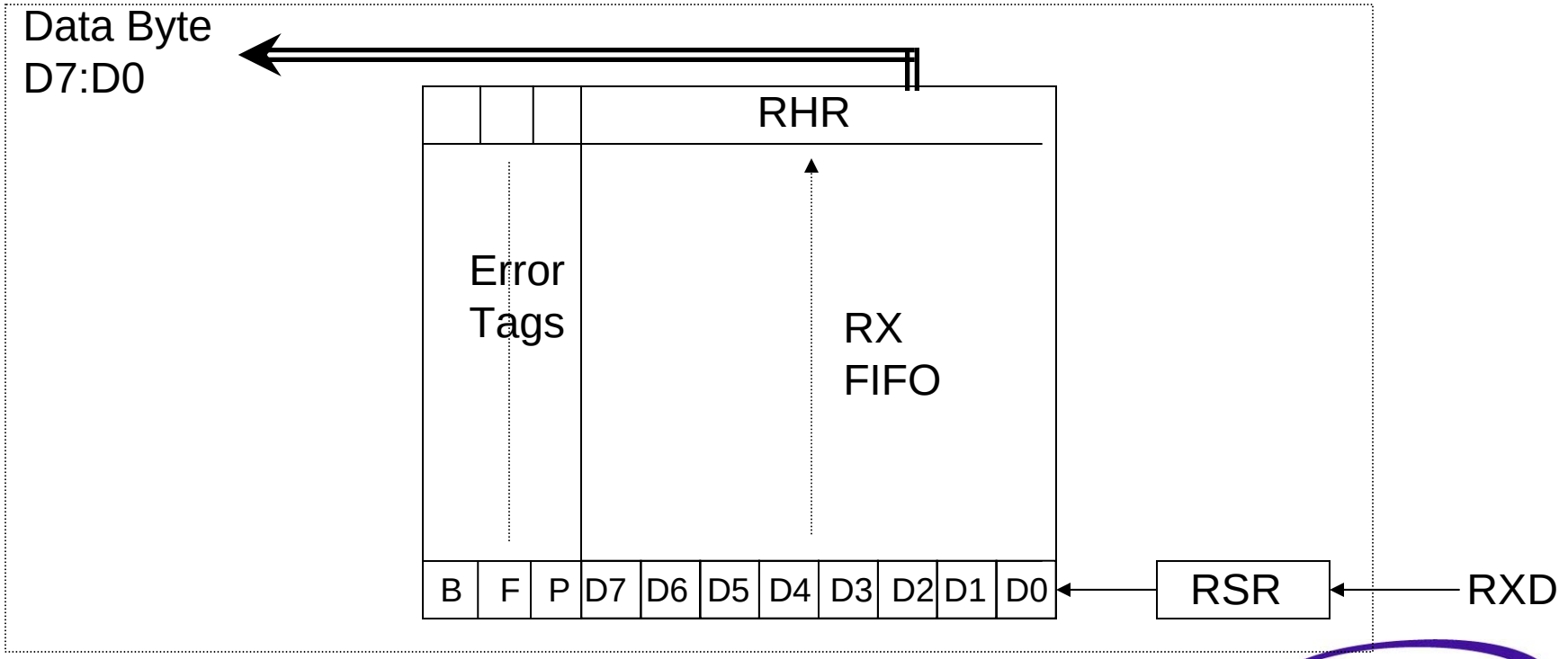
Receiver – Non-FIFO Mode

- **Incoming data is received in the Receive Shift Register (RSR)**
- **Received data is transferred to the RHR**
- **Error tags associated with data in RHR can be read via LSR**
- **Read RHR to read the data out**



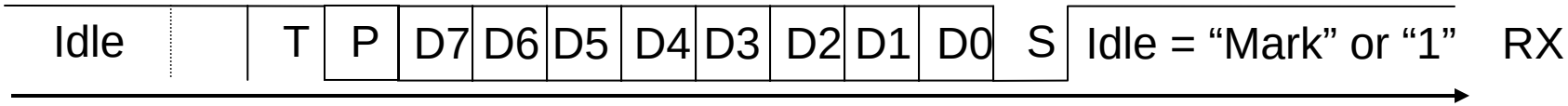
Receiver – FIFO Mode

- **Incoming data is received in the Receive Shift Register (RSR)**
- **Received data is queued in the RX FIFO**
- **Error tags associated with data in RHR can be read via LSR**
- **Read RHR to read the data out**



RX Character Validation

- **Start bit detection and validation**
  - HIGH to LOW transition indicates a start bit
  - Start bit validated if RX input is still LOW during mid bit sampling
- **Data, parity and stop bits are sampled at mid bit**
- **A valid stop bit is HIGH when the stop bit is sampled**



RX Error Reporting

- **Line Status errors**
  - Error tags are associated with each byte
    - Framing error if stop bit is not detected
    - Parity error if parity bit is incorrect
    - Break detected if RX input is LOW for duration of one character time and stop bit is not detected
  - Overrun error if character is received in RSR when RX FIFO is full
    - Non-FIFO mode
      - RHR has a data byte and data received in RSR
      - RSR data overwrites RHR data
    - FIFO mode
      - RX FIFO is full and data is received in RSR
      - Data in RX FIFO is not overwritten by data in RSR

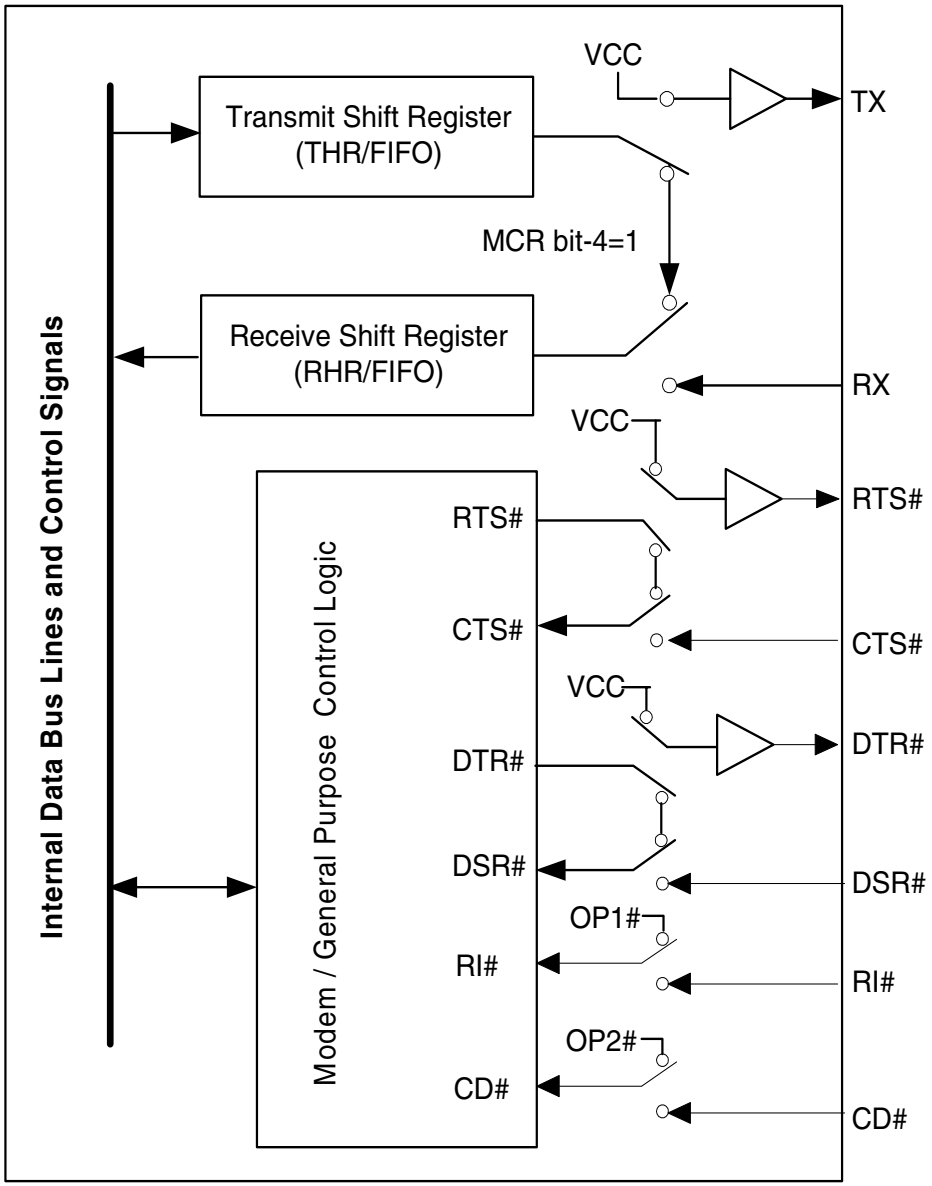
Modem I/Os

- **Legacy Modem Signals**

Signal Name	Description	Input/Output
RTS#	Request-to-Send	Output
CTS#	Clear-to-Send	Input
DTR#	Data-Terminal-Ready	Output
DSR#	Data-Set-Ready	Input
CD#	Carrier-Detect	Input
RI#	Ring-Indicator	Input

- **Used for hardware flow control or as general purpose inputs or outputs**

Internal Loopback Mode



Interrupts

Priority Level	ISR bit-3	ISR bit-2	ISR bit-1	ISR bit-0	Source of Interrupt
1	0	1	1	0	LSR (RX Data Error)
2	1	1	0	0	RXRDY (RX Data Time-out)
3	0	1	0	0	RXRDY (RX Data Ready)
4	0	0	1	0	TXRDY (TX Empty)
5	0	0	0	0	MSR (Modem Status)
-	0	0	0	1	None

- **Interrupt Source Register (ISR)**
  - If there are multiple interrupts, ISR reports only the highest pending interrupt
  - Lower priority interrupts will be reported when higher priority interrupts are cleared

E-mail hotline: uarttechsupport@exar.com