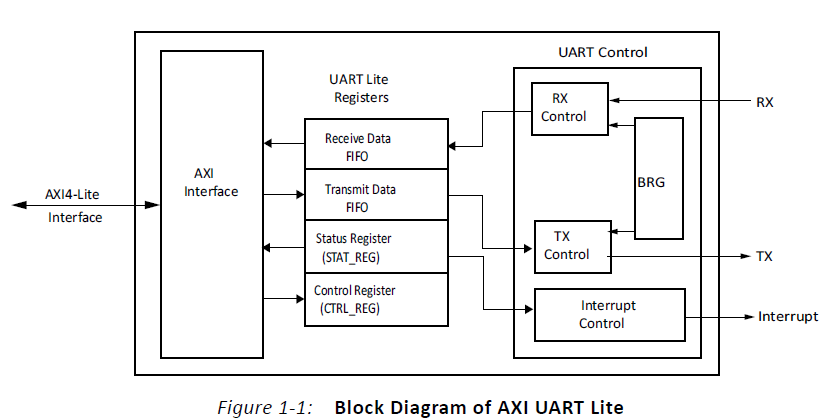
# Using Uart With MicroBlaze

## Overview of Uartlite IP

Uartlite IP uses AXI4-Lite interface for register access and data transfer. It has 16-character transmit and receive FIFOs. The Number of data bits (5-8), parity bit (odd, even or none) and baud rate are configurable on hardware. In software part, there is no access to these configurations.

Receive Data FIFO holds the received data and it is a read only register.

Transmit Data FIFO holds the data to transmit and it is a write only register.

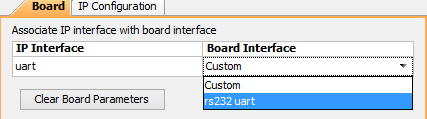
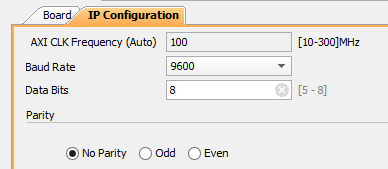
Status register contains the status of the receive and transmit data FIFOs when interrupts are enabled and errors are present. Some of these error and status are Tx FIFO Empty, Rx FIFO Full, Parity Error, Interrupt Enabled etc.

Control register contains the enable interrupt bit and reset pin for the receive and transmit data FIFO.

If the interrupts are enables, a rising edge- sensitive interrupt is generated when the receive FIFO becomes non-empty or when the transmit FIFO becomes empty.

## Hardware Part

In this part adding a Uartlite IP to a MicroBlaze project will be discussed.

* First click on “Add New IP” button and select “AXI Uartlite”.
* Double Click on the IP to configure. 
* Select rs232 Uart on board interface under Board tab.
* Set the configuration under the IP configuration tab
* Click Ok.
* Click “Run Connection Automation”
* Generate bitstream, export hardware and overwrite to existing hardware.
* Open SDK.

## Software Part

There are lots of ways to send data from FPGA to computer via Uart. Two of them will be discussed in this section.

### First and Effortless Way:

|  |
| --- |
| **#include** "stdio.h"  **int** **main**()  {  xil\_printf("Hello World");  xil\_printf("% is an integer.",5);  xil\_printf("%c is a character.",'A');  **return** 0;  } |

Xil\_printf() is the same as the standard printf() function in C. “stdio.h”library must be included to use this function.

### Second Way:

|  |
| --- |
| **#include** "xparameters.h"  **#include** "xuartlite.h"  XUartLite Uart;  **#define** Uart\_Device\_ID XPAR\_AXI\_UARTLITE\_0\_DEVICE\_ID  **#define** BUFFER\_SIZE 20  **int** **main**()  {  u8 data[BUFFER\_SIZE] = "Hello World";  XUartLite\_Initialize(&Uart,Uart\_Device\_ID);  XUartLite\_Send(&Uart,data,BUFFER\_SIZE);  **return** 0;  } |

“xparameters.h ” library contains all parameters and configurations of the IPs including base addresses, device IDs, register addresses and so on. “xuartlite.h” contains APIs, types and definitions for Uartlite IP.

“XUartlite” is a struct type definition for Uart module, and it must be declared before using any function because all functions get it as a parameter. XUartLite\_Initialize() function initializes the Uart module and sets the does the configuration according to the device ID which is XPAR\_AXI\_UARTLITE\_0\_DEVICE\_ID for Uartlite. XUartLite\_Send() function send the specified data via Uart. XUartLite\_Recv()Function is available to receive data to FPGA but it will be discussed after interrupt block.