Advanced iniUART Information

Universal Asynchronous Receive and Transmit (UART) functions are one of the most widely used interfaces. By having a technology independent synchronous design approach, the iniUART can be synthesized to any target technology. This advanced iniUART information shows you how to use this function and how it compares with standard off the shelf ICs.

System Integration

iniUART incorporates the basic functionality of an universal asynchronous receiver/ transmitter (UART). It is not an one-to-one replacement of existing off the shelf ICs, but it's complainant with their asynchronous data transfer protocol. By focusing on the basic functionality gives designers several advantages:

- Improved Reusability
  A state-of-the-art UART function is used in a lot of different systems. Project specific requirements (e.g., data FIFO) are add-ons and don't influence the main function of the iniUART core.

- Smaller Area
  Only required functions are implemented.

- Efficient System Integration
  The interface is designed for system integration. In this case having a built-in microprocessor interface makes integration more difficult. iniUART has a flexible interface towards system logic which simplifies system embedding.

- Stand Alone System
  By having a HDL representation of the code configuration signals can be hard wired. This means that the configuration is active right after system reset and the core is ready-to-run. There is no need to configure anything.

- Reduced System Cost
  iniUART has an advanced built-in baud rate generator. The required baud rate can directly be derived out of the system clock. No additional clock generator is required.

iniUART Design Advantages

- Ready to Run

  - Configuration is active right after reset.

- Improved Data Accuracy

  - 3 Point Input Sampling
  - Glitch Rejection on Serial Input Line
  - Data Format Check

- Efficient Core Implementation

  - Designed for In-System Use
  - Only utilized mode functions are implemented
  - Easy integration into custom system with predefined interface

- Synchronous HDL Design

  - Only one clock domain required

- Technology Independent

  - Can be synthesized to any FPGA and ASIC target technology

iniUART versus Standard Asynchronous Communication ICs

The table below shows how the iniUART functionality can be compared with standard off the shelf products. This comparison takes only the most often used operating mode into account: The asynchronous mode.

	iniUART	**8251 Asynchronous Mode**	**16450**	**16550**	**6402**
Dataformat	7 / 8 bit	5 / 6 / 7 / 8 bit	5 / 6 / 7 / 8 bit		5 / 6 / 7 / 8 bit
Idle line	high	high	high		high
Startbit	low bit after idle line	low bit after idle line	low bit after idle line		low bit after idle line
Parity	even / odd on/off	even / odd on/off	even / odd on/off		even / odd on/off
Parity Error	Error Flag Data is discarded	Error Flag	Error Flag		Error Flag
Stopbit	1 / 2	1 / 1,5 / 2	1 / 1,5 / 2		1 / 1,5 / 2
Baudrate	$= f_{clk} * n / 2^{22}$ n = 16 bit parameter	$= f_{clk} / n$ n = 1/16/64	$= f_{clk} / (16*n)$ n = 16 bit parameter		$= f_{clk} / 16$
FIFO	external add-on	n.a.	n.a.	2*16 byte	n.a.

Differences:

- 5 and 6 bit data format
  This format was used a long time ago. Today, you can hardly find this data format. If there is a customer who needs this option then don't walk away because with a minor modification we could support this mode.

- 1,5 stop bits
  They were used together with 5/6-bit data format.

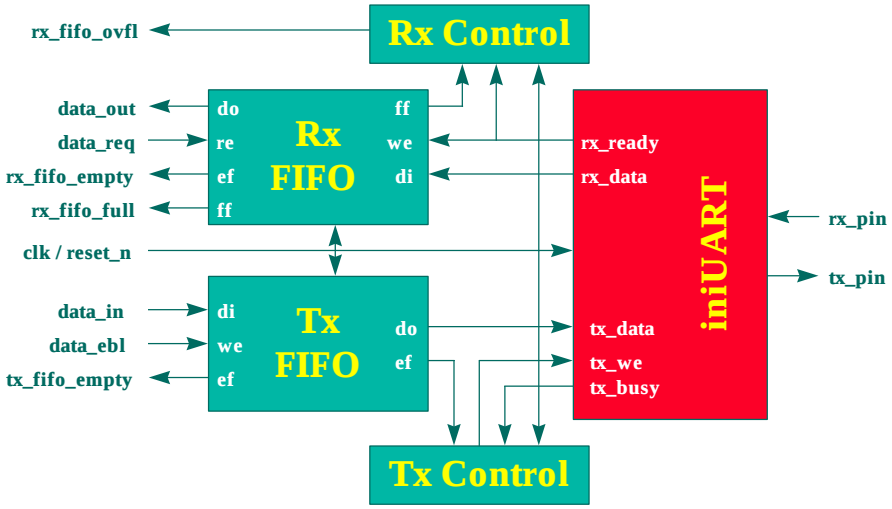
- Baud rate
  There are two main differences in generating the baud rate:
  a) by dividing the clock by a constant or a fixed value
    - 6402: one baud rate
    - 8251: three different baud rates
    - 16x50: Derived baud rate is not linear over the whole range which results in a different resolution.

  b) by using a sophisticated solution:
  iniUART uses a baud rate generator which is based on an accumulator structure. The synthesized baud rate is linear to the configuration value and the resolution is the same over the whole range.
  By having this integrated baud rate generator no external clock generator is needed because the baud rate can be derived from the already available system clock. Thus minimizes system cost and number of components on the print circuit board.
  The maximum selectable baud rate with a 8 MHz clock is 125kbps. By having a higher clock speed, higher baud rates can be achieved.

- FIFO
  iniUART has not a built in data FIFO because it is not a requirement for all applications. By designing the FIFO depending on the actual system requirements its size can be chosen accordingly. Please see the paragraph below for detailed discussion how the iniUART can be connected to a FIFO.

Connecting a FIFO to iniUART

For certain application it is required to have a on chip data buffer. Most often this buffer is organized as a FIFO. The block diagram below shows, how a FIFO can be integrated.



ef: empty flag ff: full flag re: read enable we: write enable do: data out di: data in

Corporate Headquarters:

INICORE AG
Mattenstrasse 6a
2555 Brügg, Switzerland
Tel. + +41 32 374 32 00
Fax. + +41 32 374 32 01

Email: ask_us inicore.sme.ch

Web: http://www.inicore.com

US Office:

INICORE INC.
44350 Grimmer Blvd.
Fremont, CA 94538
Tel. 510.445.1529
Fax. 510.656.0995

Email: ask_us inicore.com