BaudGen IP Core Specification

Author: Andrew Mulcock
amulcock@opencores.org

Rev. 0.1
December 16, 2007

This page has been intentionally left blank

Revision History

Rev.	Date	Author	Description
0.1	26-Nov-07	Andrew Mulcock	First Draft

Contents

1 Introduction

The BaudGen is typically used with a UART (Universal Asynchronous Receiver/Transmitter) core to provide timing for serial communication capabilities.

Further uses are for generating timing 'enables' for such items as SPI, I2C and OneWire peripherals.

Features:

- Stand alone, pure VHDL
- Generates an enable pulse and an n times enable pulse
- 'Programed' by generics at 'compile' time.
- Ease of use, Generics to specify the baud (bit rate) required and the input clock frequency

2

IO ports & generics

2.1 Ports

Port	Width	Direction	Description
clk	1	Input	
rst	1	Input	Active High synchronous reset
baud_x_en	1	Output	Active High one clk wide enable at x times baud rate
baud_en	1	Output	Active High one clk wide enable at baud rate

2.2 Generics

Generic	Type	Default	Description
baudrate	int	115200	Baud rate required (in Hz)
clock_freq_mhz	real	200.0	Frequency of the clock in (in MHz)
over_sample	int	4	Number of over sample pulses per 'Bit'

3 Operation

This BaudGen core is simple, consisting of two programmable counters. Programing is performed by the use of generics, so it's a fixed frequency Baud-rate generator.

The 'clever' bit, and the reason for writing the core, is the user does not have to 'think' about divider ration's or programming the baud rate up using a bus interface, it's just there in the design. Great for debug UART ports, as you know what the UART speed is going to be.

A baud is another way of saying Bit per second, so the BaudGen can also be used to provide any other timing pulses needed in a design. Apart form baud Rate in a UART, uses so far have included a 1 us period pulse used in a OneWire interface, a 64.5 ns period pulse used in a SPI peripheral, and a 2.5 us period pulse used in a I2C interface.

3.1 Initialization

The core has been used in FPGAs and as such does not require a reset, but for the aid of simulation and for customers that require a reset, a synchronous reset has been included. This active high to reset input sets the counters to a known position.

3.2 Generics

Generics are used to define the required baud rate and the frequency of the clock to divide down from.

If no x times output is required, set over_sample to 2, and do not use the baud_x_en output.

3.3 Timing Diagram

Illustration 1 shows the relationship between the two outputs. In this case, over_sample is set at 4, and 4 evenly spaced pulses are generated on baud_x_en for every baud_en.

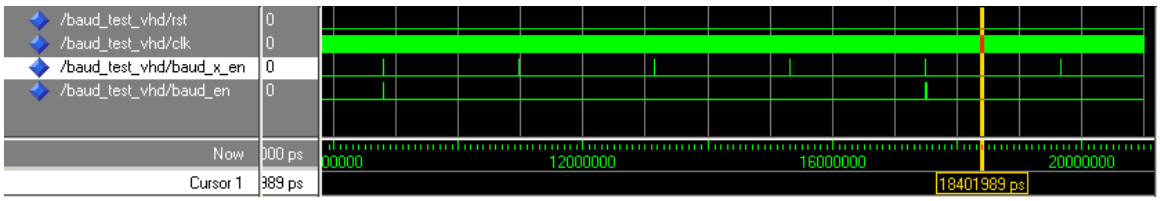

Illustration 1: over_sample set at 4

Illustration 2 shows the one clock width of the enable pulses and the fact that the baud_x_en is one clock ahead of the baud_x output. If both are required to 'line up' then register externally baud_x_en to the clock to align.

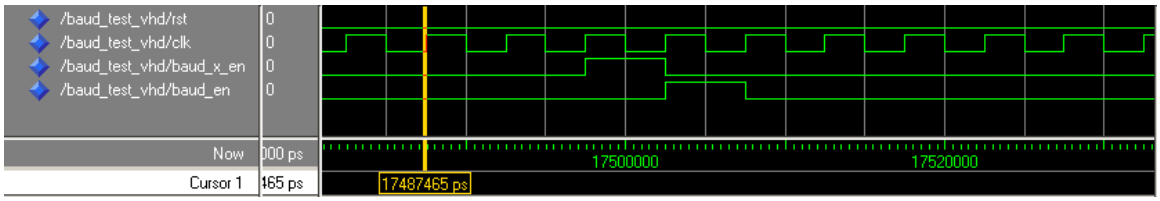

Illustration 2: width of enable pulses and the relative timing