Noise Immune and Area Optimized Serial Interface for FPGA based Industrial Interfaces

D. Ravi[1], K. S. R Murthy[2]

[1]M.E Student, Department of ECE, M.V.S.R. Engineering College, Nadergul, Andhra Pradesh, India
[2]Professor, Department of ECE, M.V.S.R. Engineering College, Nadergul, Andhra Pradesh, India

Abstract: ***This project describes a novel architecture based on Recursive Running Sum (RRS) filter implementation for wire and wireless data processing. UARTs are used for asynchronous serial data communication between remote embedded systems. If physical channel is noisy then, serial data bits get corrupted during transmission. The UART core described here, utilizes recursive running sum filter to remove noisy samples. Input data signal is directly sampled with system clock and samples are accumulated over a window size. The window size is user programmable and it should be set to one tenth of required bit period. The intermediate data bit is decoded using magnitude comparator. The advantage of this architecture is that baud rate is decided by the window size so there is no need of any external "timer module" which is normally required for standard UARTs. The Recursive Running Sum (RRS) filter architecture with programmable window size of M is designed and modules are implemented with VHDL language. This project implementation includes many applications in wireless data communication Systems like RF, Blue tooth, WIFI, ZigBee wireless sensor applications. Total coding written in VHDL language. Simulation in Modelsim Simulator, Synthesis done by XILINX ISE 9.2i. Synthesis result is verified by the Chipscope. Input signal given from the keyboard and output is seen by the help of HyperTerminal***.

Keywords: Universal, data processing, signals, communication, noise

1. Introduction

Universal Asynchronous Receiver Transmitter (UART) is used for asynchronous serial data communication between remote embedded systems. Standard UART cores [1],[2],[3],[4]. The window size is user programmable and it should be set to one tenth of required bit period. The intermediate data bit is decoded using magnitude comparator. The advantage of this architecture is that baud rate is decided by the window size so there is no need of any external "timer module" which is normally required for standard UARTs. But if the physical channel is noisy then data bits get corrupted during transmission and it leads to wrong data decoding at receiver. To overcome the noise problem a digital low pass filter based architecture is proposed in this paper. Recursive Running Sum (RRS) is simple low pass filter, it can be used to remove noise samples from data samples at receiver [5]. Serial receive data signal is directly sampled with system clock and samples are fed to RRS filter. The window size of the filter is user programmable and it decides baud rate. RS filter hardware implementation is described in section-2. Window size selection criteria is described in section-3. The UART Architecture is described in section-4 while section-5 gives simulation results and comparison with standard UART core. The robust UART core described here is designed using VHDL and implemented on Xilinx Virtex FPGA .

2. RRS Filter Implementation

The Recursive Running Sum (RRS) filter with window size of M is described by following equations.

$$H(z) = \frac{1-Z^{-M}}{1-Z^{-1}}$$

$$y(n) = x(n) + y(n-1) – x(n-M)$$

The hardware realization of the above equation is as shown in the Figure-1. It requires an Adder, subtracter, a unit delay and an M samples delay element. The window size (M) is related to baud rate which is user programmable. So M is variable, if a 16 bit register is used to hold value of M, it can have values from 0 to 65535. The hardware implementation of variable delay with above range would require 65535 D flip-flops and large number of combinatorial logic for MUX and selection logic implementation. So this implementation is not feasible for FPGA or ASIC platform.

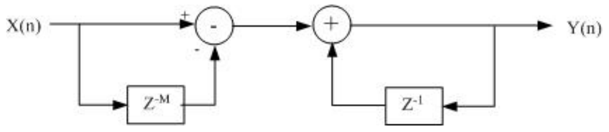


Figure 1: Hardware realization of RRS filter

The Other approach for hardware realization of RRS filter using a factor of M decimator is shown in Figure-2. It requires a Adder, subtracter, two unit delays and a down sampler of factor M. The implementation of down sampler requires a 16 bit counter and a magnitude comparator which is much simpler than previous approach. So this approach is selected for robust UART implementation.

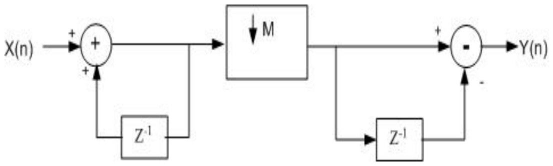


Figure 2: Implementation of RRS using down sampler

3. Window Size Determination

The serial data signal is directly sampled with system clock and samples are fed to the input of the RRS filter. The window size (M) of the filter is user programmable, it is the main thing to decode the correct data bits shown in Figure-1.6, and it also decides BAUD RATE. Baud rate is number of bits transmitted per second, where external timer module is not required.

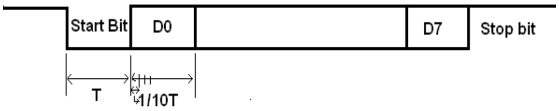


Figure 3: window size determination

The window size (M) is used to drive the corrupted signal of each bit period into ten parts, by a mathematical sampling process in each part we getting one output (‘0’s or ‘1’s) at the output of RRS filter. Window Size = Bit Period /10

4. UART Architecture

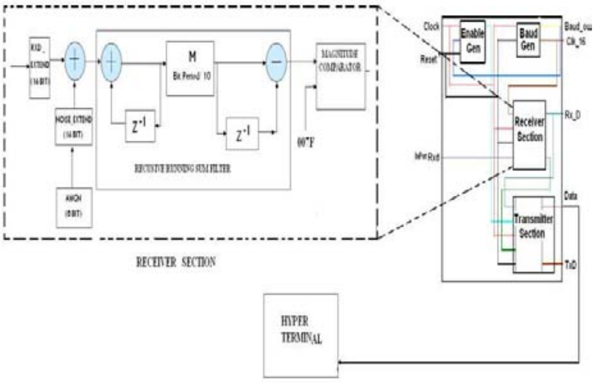


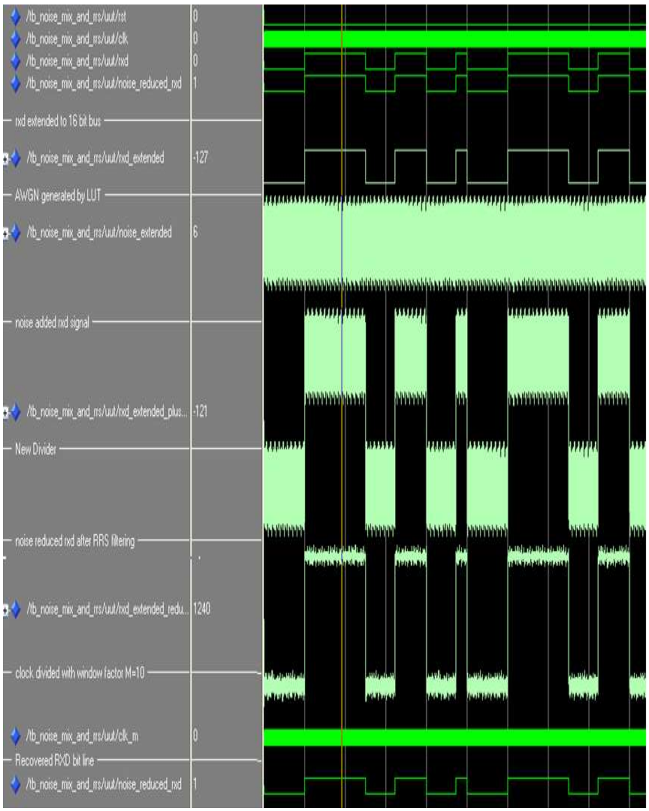
Figure 4: The UART block diagram

The 1 bit input given to rxd_extended, based on the input if it is ‘1’ then the value of output is converted to +127 in 16 bit signed magnitude format else -127 16 bit signed magnitude format. Next stage is RSS filter; the two basic building blocks of a RSS filter are an integrator and a comb. An integrator is simply a single-pole IIR filter with a unity feedback coefficient Y [n] =Y [n-1] + X [n]. This system is also known as an accumulator. The transfer function for an integrator on the z- plane is $H_I(Z)$ =1 / 1-Z $^{-1}$. The power response is basically a low-pass filter with a 20 dB per decade (- 6 dB per octave) roll off, but with infinite gain at DC. This is due to the single pole at z = 1; the output can grow without bound for a bounded input. In other words, a single integrator by itself is unstable. After that passed through a decimator block i.e The digital clock manager (Decimator) module implemented in our project divides input clk signal by 10 times. The input signal x is the input data of 16 bit and y is the output of the given data input. Then given to a comb filter which is running at the low sampling rate, f $_s$ / R, for a rate change of R is an odd-symmetric FIR filter described by Y [n] = X [n] -- X [n-RM] In this equation, M is a design parameter and is called the differential delay. M can be any positive integer, but it is usually limited to 1 or 2. The corresponding transfer at f $_s$ / R=> $H_C(Z)$ =1 / 1-Z $^{-RM}$

5. Results

The UART core described here has been designed using VHDL. Noise insertion is done by additive White Gaussian Noise. The core has been simulated using Model Sim Simulator while hardware implementation is done on Xilinx Spartan 3e FPGA and has been checked hardware with On Chip Debugging tool ChipScope pro with the help of icon and ila component

5.1 Simulation Results



```
/tb_noise_mix_and_rrs/uut/rst                      0
/tb_noise_mix_and_rrs/uut/clk                      0
/tb_noise_mix_and_rrs/uut/rxd                      0
/tb_noise_mix_and_rrs/uut/noise_reduced_rxd        1
rxd extended to 16 bit bus
/tb_noise_mix_and_rrs/uut/rxd_extended             -127
AWGN generated by LUT
/tb_noise_mix_and_rrs/uut/noise_extended           6
noise added rxd signal
/tb_noise_mix_and_rrs/uut/rxd_extended_plus...     -121
New Divider
noise reduced rxd after RRS filtering
/tb_noise_mix_and_rrs/uut/rxd_extended_redu...     1240
clock divided with window factor M=10
/tb_noise_mix_and_rrs/uut/clk_m                    0
Recovered RXD bit line
/tb_noise_mix_and_rrs/uut/noise_reduced_rxd        1
```

Figure 5: Top Module Simulation Wave Form

5.2 RTL Schematic

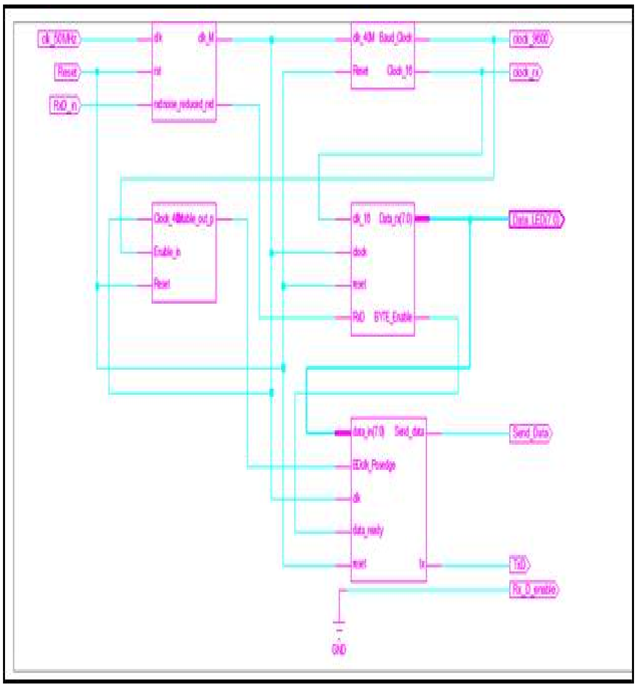
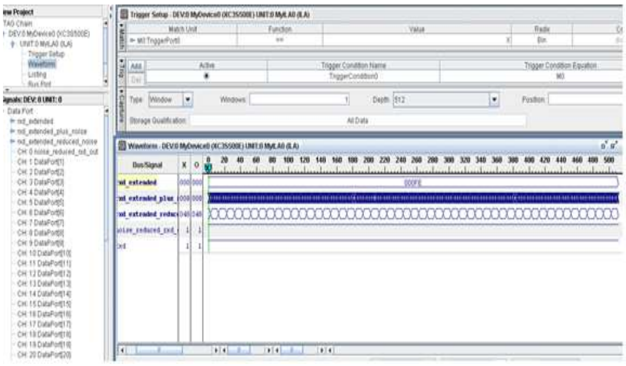
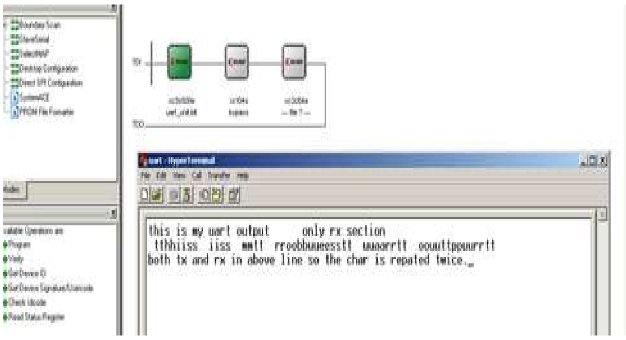


Figure 6: Top Module RTL schematic diagram

5.3 Chipscope Result


Trigger Setup - DEV:0 MyDevice0 (XC3S500E) UNIT:0 MyILA0 (ILA)
Waveform - DEV:0 MyDevice0 (XC3S500E) UNIT:0 MyILA0 (ILA)
Bus/Signal
xd_extended
xd_extended_plus_
xd_extended_reduc
oise_reduced_rxd_
rxd

5.4 Out Put for UART



```
this is my uart output     only rx section
 tthhiiss  iiss  mm!!  rroobbuueesstt  uuaarrtt  oouuttppuurrtt
both tx and rx in above line so the char is repated twice._
```

6. FPGA Implementation

The robust UART described in this paper has been implemented on xilinx® Spartan XC3S500E-5FG320. The synthesis report is as shown below, it consumes only 655 Spartan slices, thus UART core occupies very small area

Selected Device:	3S500E-5	
Number of Slices:	394 of 4656	8%
Number of Slice Flip Flops:	219 out of 9312	2%
Number of 4 input LUTs:	671 out of 9312	7%
Number of bonded IOBs:	16 out of 232	6%
Number of GCLKs:	2 out of 24	8%

7. Conclusion

This project describes a robust UART architecture based on recursive running sum filter for removal of channel induced noise. The baud rate of serial communication is decided by the window size of the filter, which is user programmable and should be set to one tenth of required bit period. Thus it does not require any external module for baud rate generation. From the above results we can say that the robust UART has better performance than standard UART at higher noise levels and also from synthesis results the area occupied on FPGA board also very low

8. Future Scope

In existing system we have one baud rate (i.e. 9600) and one data width. Variable width data and variable baud rate robust UART can be implemented in future implementation models.

References

[1] Dr. Garima Bandhawarkar Wakhle, Iti Aggarwal and Shweta Gaba “Synthesis and Implementation of UART using VHDL Codes” 2012 International Symposium on Computer, Consumer and Control

[2] Himanshu Patel, Sanjay Trivedi, R. Neelkanthan, V. R. Gujraty “ a robust uart architecture based on recursive running sum filter for better noise performance”

[3] Intel® MCS-51 microcontroller family user’s manual

[4] Synopsys DesignWare® DW8051 MacroCell databook

[5] Liakot Ali, Roslina Sidek, Ishak Aris, Alauddin Mohd. Ali, and Bambang Sunaryo Suparjo “Design of a micro-UART for SoC application” Computers & ElectricalEngineering Journal, Elsevier Publ. Volume 30, Issue 4 , June 2004, Pages 257-268

[6] Norhuzaimin, J. Maimun, H.H. “The design of high speed UART” Asia-Pacific Conference on Applied Electromagnetics (APACE 2005). Dec. 2005

[7] Yong Lian Poh Choo Ho “ECG noise reduction using multiplier-free FIR digital filters” 7th International Conference on Signal Processing (ICSP ‘04.) 2004. Volume: pages: 2198 - 2201

[8] M. Delvai, U. Eisenmann, W. Elmenreich, “Intelligent UART Module for Real-Time Applications” First Workshop on Intelligent Solutions in Embedded Systems (WISES),pages 177-185, Vienna, Austria, June 2002

[9] P.J. Ashenden, “The Designer’s guide to VHDL”, Morgan Kaufmann Publ., 2000

[10] Mitra, S. K., “Digital Signal Processing - A computer based approach,” Tata McGraw-Hill, 2001

[11] Xilinx® Virtex FPGA datasheets

[12] Spartan-3 FPGA Family:Complete Data Sheet