IMPROVING NOISE IMMUNITY FOR SERIAL INTERFACE

A Lattice Semiconductor White Paper

July 2014

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Introduction

Serial interfaces like SPI or I^2C are common methods of communication between ICs or devices. The I^2C interface has two wires (lines), one is clock and the other is data. The data is transferred serially from one device to another at the frequency determined by the clock. The protocol of communication between the two devices should meet all the required specifications of the serial interfaces, in order to send and receive correct data.

FPGAs and CPLDs are programmable devices which often implement serial interface controller using HDL (Hardware Description Language) so that the user can define any general purpose I/O (GPIO) as serial interface ports. The serial interface specification for high speed data transfer (400 kHz and faster) recommends a 50 ns noise filter on the inputs to prevent double-clocking and miss-communication. However, programmable devices generally do not have built-in noise filters on their GPIO pins.

In a complex system there can be noise from a variety of sources such as: switching noise from power supplies, other devices, or the environment itself. Any or all of these may cause an intermittent error in serial communication. This paper discusses some techniques to improve noise immunity for the serial interfaces implemented in devices which do not have built-in noise filter for the serial interface.

Techniques to improve noise immunity

External RC Filter

Adding an external RC low-pass filter on the serial interface pins is one way of implementing a noise filter. However, a simple RC filter with time constant of 50 ns also slows the rising edges and presents a considerable load to the driving pins. A slow edge is more susceptible to noise and the large capacitive load can cause ground-bounce within the device. The circuit shown in Figure 1 is a good compromise; providing noise reduction without slowing the edge too much or loading down the driver pins. This circuit can be used on both the Master and Slave devices on any given bus. The exact component values for C_f, R_{S1}, R_{S2}, and $R_{pull\text{-}up}$ depend upon supply voltage and level of noise reduction desired.

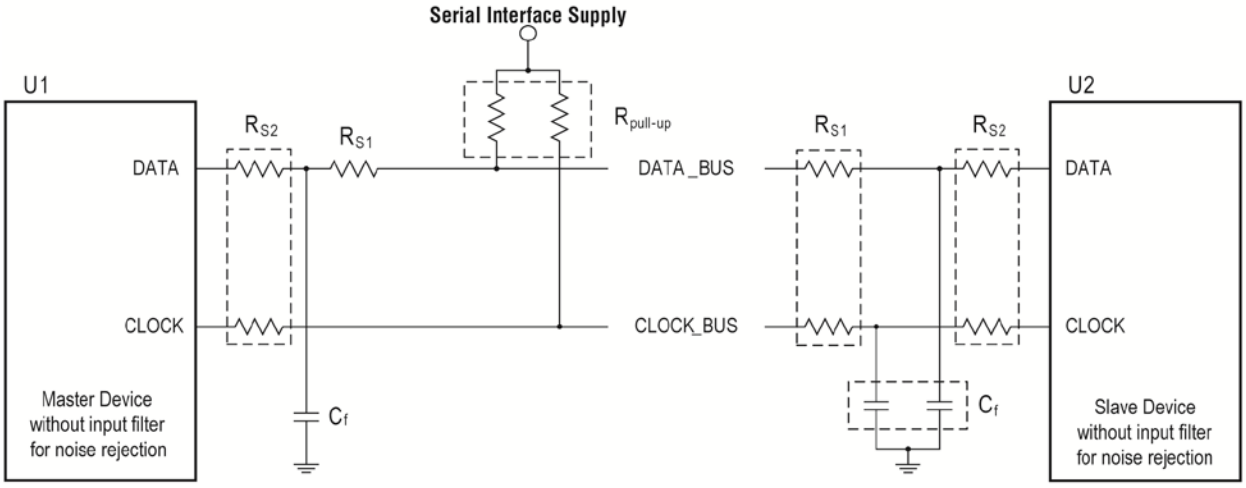

Figure 1: External Low Pass Noise Filter Circuit

The value of the series resistor(s) (R_{S1} and R_{S2}) will effectively raise the logic low output level on to the bus from the device being filtered. If the value of the series resistor(s) is too large, then the voltage swing on the serial bus will be too small for reliable communication. Thus, the value of the series resistor(s) has to balance between enough filtering without reducing the low voltage margin. The tables below show suggested values of components that balance these issues and consider the other serial interface specifications for either a 3.3 V or 2.5 V system. The 3.3 V system has about 100 mV better noise margin over the 2.5 V system, based on the values in the tables.

Time constant = R_{S1} *C_f in seconds.

Margin to V_{IL} = V_{IL} (max) – V_{OL} (calculated) = 0.3 VDD – [V_{OL} (max) + VRs (voltage drop on both R_{S1} and R_{S2})]

$R_{pull-up}$ (Ω)	R_{S1} (Ω)	R_{S2} (Ω)	C_f (Pf)	Time Constant (ns)	Margin to V_{IL} (V)
1800	130	51	180	23.4	0.325
2000	150	51	150	22.5	0.325
2200	160	56	150	24.0	0.331
2400	180	62	130	23.4	0.324

Table 1: Component Values for: V_{DD} =3.3 V V_{IL} (max)=0.99 V V_{OL} (max) =0.4 V

$R_{pull-up}$ (Ω)	R_{S1} (Ω)	R_{S2} (Ω)	C_f (Pf)	Time Constant (ns)	Margin to V_{IL} (V)
1000	30	36	620	19.6	0.220
1500	47	39	390	18.3	0.236
1800	75	39	240	18.0	0.225
2000	91	39	200	18.2	0.222

Table 2: Component Values for: V_{DD} =2.5 V V_{IL} (max)=0.75 V V_{OL} (max) =0.4 V

Decoupling Supply

There are instances where noise on supplies associated with serial interfaces may cause error in communication. Reducing such noise can be accomplished with simple LC filtering circuit. An inductor or Ferrite Bead of about 4-10 uH, with good current carrying characteristics and low resistance should be placed in series with the voltage source to choke off the high frequency noise. Ferrite Bead Inductors such as Murata BLM31B601S, BLM11B601SPB, or equivalent offer good inductive supply isolation. After the inductor multiple capacitors of different values are typically used. The smaller valued capacitor is better at filtering out the highest frequency component of the noise

while the larger valued capacitor is more effective at filtering out the lower frequency component of the noise. Figure 2 shows a typical circuit configuration.

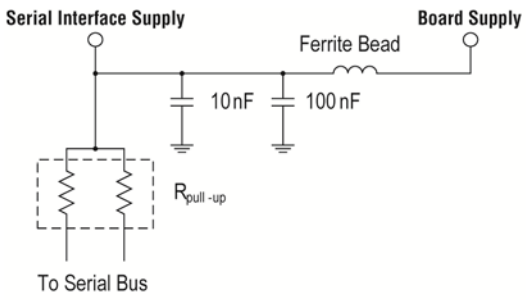

Figure 2: Typical Supply Decoupling Circuit

Digital Filter

In FPGA designs that implement a soft core serial interface, a pair of digital filters can be added; one for the data and one for the clock. Figure 3 is a generic circuit of a scalable single filter that can be realized in VHDL or Verilog. The size of the filter, or number of taps, is based on the oscillator (OSC) frequency and the desired filter time. The filter should have a minimum of three taps. Table 3 lists some recommended frequencies and number of filter taps to provide effective filtering. Generally speaking, both of the serial signals (data and clock) should have identically sized filters. The exception would be if the data and clock have a known relationship that does not meet the specification. Then one of the filters could be extended to properly align the signals at the outputs.

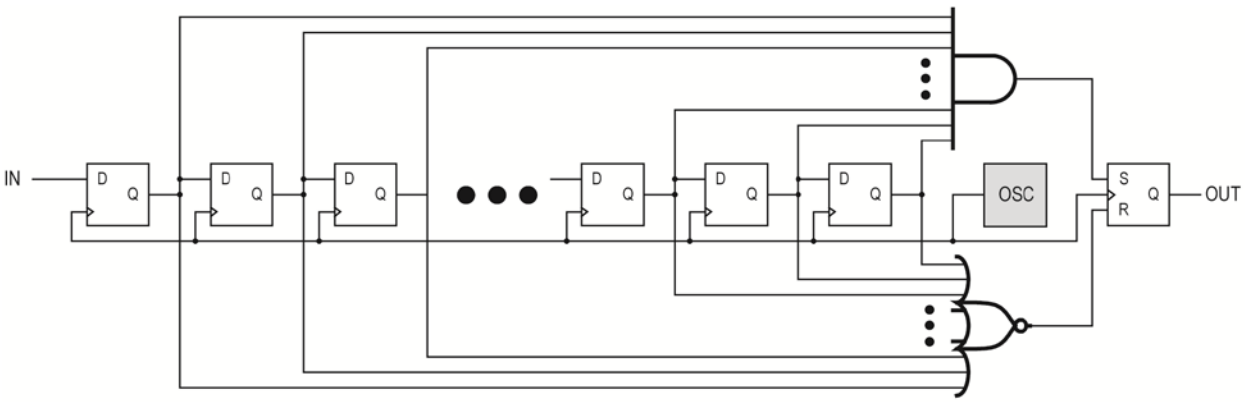

Figure 2: Digital Filter Circuit

OSC Frequency (MHz)	Taps	Filter Time (ns)
60	3	50
80	4	50
100	5	50
150	7	47
200	10	50

Table 3: Digital Filter Taps and Filter Time Based on OSC Frequencies

Summary

For reliable Serial communication, it is essential to minimize the effect of external noise from the data and clock lines. The techniques discussed in this white paper can be used to improve the reliability of serial communication in noisy environments. One or more of the techniques described above may be helpful for designers working with serial interfaces on devices such as the ones mentioned below:

- Any FPGA or CPLD using GPIOs for serial interface ports
- Any device without built-in noise filter on the serial ports
- MachXO2-EFB, POWR1220AT8, POWR1014A, POWR6AT6

Additionally, it is advised that the designer follows the serial interface specifications, when implementing the above techniques.