AN1215: SINC3 Filter for DSM Applications

A second-order Delta-Sigma Modulator (DSM) ADC is typically followed by a SINC3 filter to reduce signal noise and reduce (decimate) data to a more usable rate. This document discusses the essentials of SINC3 filter implementation for use with Skyworks Si894x and Si8935/36/37 isolated current and voltage sensors with DSM output.

KEY POINTS

- SINC3 filter function
- Implementation requirements
- Implementation example
- VHDL code for SINC3 core

1. SINC3 Filter Function

A second-order Delta-Sigma Modulator (DSM) ADC is typically followed by a SINC3 filter to reduce signal noise and reduce (decimate) the data to a more usable rate. A typical DSM output data clock rate is 20 MHz, which is provided either by the device itself or by an external clock source. The application will dictate an oversampling ratio (OSR), which is a multiple of the signal bandwidth of interest and which obeys the following equation.

$$OSR = \frac{DSM\_clock\_rate}{Filter\_sample\_rate}$$

$$OSR = 2^n$$

Equation 1. Oversampling Ratio

The filter sample rate is two times the signal bandwidth of interest. For example, If the DSM clock rate is 20 MHz and the OSR is 256, then the filter sample rate is 78.125 kHz and the signal bandwidth is 39.0625 kHz.

2. Additional Implementation Requirements

The VHDL code for the basic core of the SINC3 filter is included in this document. Additional functions are required:

- Word clock generator. This takes the DSM clock and OSR as inputs and generates the Filter Sample Rate clock. It is used as the decimation clock. WORD_CLK will be MOD_CLK divided by the OSR.
- Filter output register. This can be a single register with a parallel output, a single register with a serial output, or perhaps a FIFO that accumulates filter output for periodic serial export to a host MCU. If the filter output is not required in real time, a larger RAM can receive the filter data, and the loading and unloading of the RAM can be controlled by an additional finite-state machine.
- Input D-flip-flops to align the data and word clock with the DUT clock (DSM clock).

3. Implementation Example

The following figure shows an implementation example. Also shown is the finite state machine that controls loading and unloading of the output register. Note that the output register function may be something other than a DPRAM, depending on system requirements.

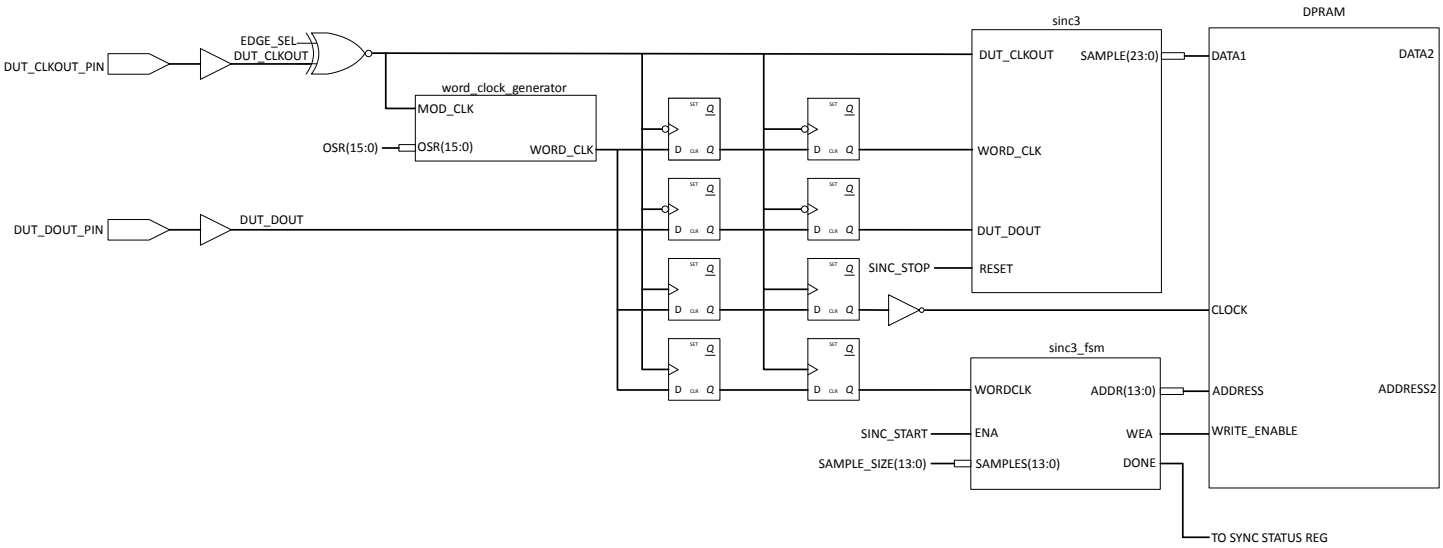

Figure 3.1. Implementation Example

4. VHDL Code

The following source code is for the sinc3 block only. It is provided “AS IS”, with no warranties, and may be copied, compiled into binary form, and distributed in source and binary forms.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.all;
use ieee.numeric_std.all;

entity sinc3 is
port(RESET, DUT_DOUT, DUT_CLKOUT, WORD_CLK : in std_logic;
SAMPLE : out std_logic_vector(23 downto 0));
end sinc3;
architecture RTL of sinc3 is
signal DIF0, DIF1, DIF3, DIF5 : std_logic_vector(24 downto 0);
signal INT1, INT2, INT3, INT4 : std_logic_vector(24 downto 0);
signal UNSHIFTED_SAMPLE : std_logic_vector(24 downto 0);
signal UNSIGNED_SAMPLE : unsigned(24 downto 0);
signal SHIFTED_SAMPLE : std_logic_vector(24 downto 0);
signal DEL1 : std_logic_vector(24 downto 0);

begin

process(DUT_CLKOUT, RESET)
begin
if RESET = '1' then
DEL1 <= (others => '0');
elsif DUT_CLKOUT'event and DUT_CLKOUT = '1' then
if DUT_DOUT = '1' then
DEL1 <= DEL1 + 1;
end if;
end if;
end process;

process(RESET, DUT_CLKOUT)
begin
if RESET = '1' then
INT1 <= (others => '0');
INT2 <= (others => '0');
elsif DUT_CLKOUT'event and DUT_CLKOUT = '1' then
INT1 <= INT1 + DEL1;
INT2 <= INT2 + INT1;
end if;
end process;

process(RESET, WORD_CLK)
begin
if RESET = '1' then
DIF0 <= (others => '0');
DIF1 <= (others => '0');
DIF3 <= (others => '0');
DIF5 <= (others => '0');
elsif WORD_CLK'event and WORD_CLK = '1' then
DIF0 <= INT2;
DIF1 <= DIF0;
DIF3 <= INT3;
DIF5 <= INT4;
end if;
end process;

INT3 <= DIF0 - DIF1;
INT4 <= INT3 - DIF3;
UNSHIFTED_SAMPLE <= INT4 - DIF5;
UNSIGNED_SAMPLE <= unsigned(UNSHIFTED_SAMPLE);
SHIFTED_SAMPLE <= std_logic_vector(UNSIGNED_SAMPLE srl 1);

SAMPLE <= SHIFTED_SAMPLE(23 downto 0);

end RTL;
```

5. References

- https://en.wikipedia.org/wiki/Delta-sigma_modulation

SKYWORKS®

Connecting Everyone and Everything, All the Time

Portfolio
www.skyworksinc.com

Quality
www.skyworksinc.com/quality

Support & Resources
www.skyworksinc.com/support

Copyright © 2021 Skyworks Solutions, Inc. All Rights Reserved.

Information in this document is provided in connection with Skyworks Solutions, Inc. ("Skyworks") products or services. These materials, including the information contained herein, are provided by Skyworks as a service to its customers and may be used for informational purposes only by the customer. Skyworks assumes no responsibility for errors or omissions in these materials or the information contained herein. Skyworks may change its documentation, products, services, specifications or product descriptions at any time, without notice. Skyworks makes no commitment to update the materials or information and shall have no responsibility whatsoever for conflicts, incompatibilities, or other difficulties arising from any future changes.

No license, whether express, implied, by estoppel or otherwise, is granted to any intellectual property rights by this document. Skyworks assumes no liability for any materials, products or information provided hereunder, including the sale, distribution, reproduction or use of Skyworks products, information or materials, except as may be provided in Skyworks' Terms and Conditions of Sale.

THE MATERIALS, PRODUCTS AND INFORMATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, PERFORMANCE, QUALITY OR NON-INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT; ALL SUCH WARRANTIES ARE HEREBY EXPRESSLY DISCLAIMED. SKYWORKS DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. SKYWORKS SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO ANY SPECIAL, INDIRECT, INCIDENTAL, STATUTORY, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS THAT MAY RESULT FROM THE USE OF THE MATERIALS OR INFORMATION, WHETHER OR NOT THE RECIPIENT OF MATERIALS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Skyworks products are not intended for use in medical, lifesaving or life-sustaining applications, or other equipment in which the failure of the Skyworks products could lead to personal injury, death, physical or environmental damage. Skyworks customers using or selling Skyworks products for use in such applications do so at their own risk and agree to fully indemnify Skyworks for any damages resulting from such improper use or sale.

Customers are responsible for their products and applications using Skyworks products, which may deviate from published specifications as a result of design defects, errors, or operation of products outside of published parameters or design specifications. Customers should include design and operating safeguards to minimize these and other risks. Skyworks assumes no liability for applications assistance, customer product design, or damage to any equipment resulting from the use of Skyworks products outside of Skyworks' published specifications or parameters.

Skyworks, the Skyworks symbol, Sky5®, SkyOne®, SkyBlue™, Skyworks Green™, Clockbuilder®, DSPLL®, ISOmodem®, ProSLIC®, and SiPHY® are trademarks or registered trademarks of Skyworks Solutions, Inc. or its subsidiaries in the United States and other countries. Third-party brands and names are for identification purposes only and are the property of their respective owners. Additional information, including relevant terms and conditions, posted at www.skyworksinc.com, are incorporated by reference.

Skyworks Solutions, Inc. | Nasdaq: SWKS | sales@skyworksinc.com | www.skyworksinc.com
USA: 781-376-3000 | Asia: 886-2-2735 0399 | Europe: 33 (0)1 43548540 |