
Using IGLOO and ProASIC3 FIFO for Generating Periodic Waveforms

Table of Contents

Table of Contents	1
FIFO Initialization	1
Creating a FIFO Block	1
Waveform Generator Design	2
Related Documents	4
List of Changes	5

Microsemi® IGLOO® and ProASIC®3 families of FPGAs contain embedded memory blocks that can be used as either RAM or FIFO. These memory blocks also include a dedicated FIFO controller to generate internal addresses and external flag logic (FULL, EMPTY, AFULL, AEMPTY).

In addition to the conventional flags, data, and address ports, IGLOO and ProASIC3 FIFO blocks include two input pins: FSTOP and ESTOP. FSTOP is an active high signal used to stop the FIFO write-address counter when the FIFO is full (FULL flag is high). Similarly, the active high ESTOP signal stops the FIFO read counter from counting once the FIFO is empty (that is, EMPTY flag goes high). When ESTOP is low, the read counter will roll over to the top (that is, address '0') and start counting. If the ESTOP input is low, when the FIFO counter hits the bottom of the FIFO, the EMPTY flag becomes active for only one clock cycle. On the next cycle, when the read counter rolls over to the top, the flag is cleared.

One of the applications of ESTOP is to write into the FIFO once, then read the contents over and over. An example of this application is generating a periodic pattern/waveform out of the FIFO and the FPGA.

FIFO Initialization

As mentioned earlier, the FIFO blocks need be written once (initialized) with the data before the periodic reading starts. The initialization data can be provided to the FIFO by internal logic, embedded FROM (internal nonvolatile Flash memory cells) or external storage devices. Refer to the "SRAM and FIFO Memories in Microsemi Low Power Devices" chapter of the *IGLOO FPGA Fabric User's Guide* for more information.

IGLOO and ProASIC3 memory blocks feature variable-aspect ratio capability. Using this feature, the read and write ports can have different widths (and consequently depths). This may be useful in many applications, including the initialization of the FIFO blocks. For example, if each sample point of the desired waveform consists of eight bits (8-bit read width), the write port can be configured as 32-bit wide. This enables the user to enter four data points into the FIFO in one write clock cycle.

As a reference, the Sine Table on the Microsemi website contains a 1024x8 memory content for half-cycle of sine wave.

Creating a FIFO Block

The FIFO macro, FIFO4K18, can be instantiated directly from the device macro library. After instantiation of the macro, designers must bind all the ports of the FIFO4K18 macro to the rest of the design or terminate them to appropriate logic based on the desired configuration and/or functionality of the FIFO block. This approach can be tedious and prone to errors. Microsemi recommends that designers use SmartGen core generator, integrated in the Microsemi Libero® Integrated Design Environment (IDE) software, to create FIFO blocks. SmartGen instantiates the FIFO macro from the library and connects all

the ports appropriately based on user's entry. Refer to the *IGLOO, Fusion and ProASIC3 Macro Library Guide* for more information.

To create a FIFO with disabled ESTOP, users should check the **Continue counting Read Counter after FIFO is empty** option as shown in [Figure 1](#).

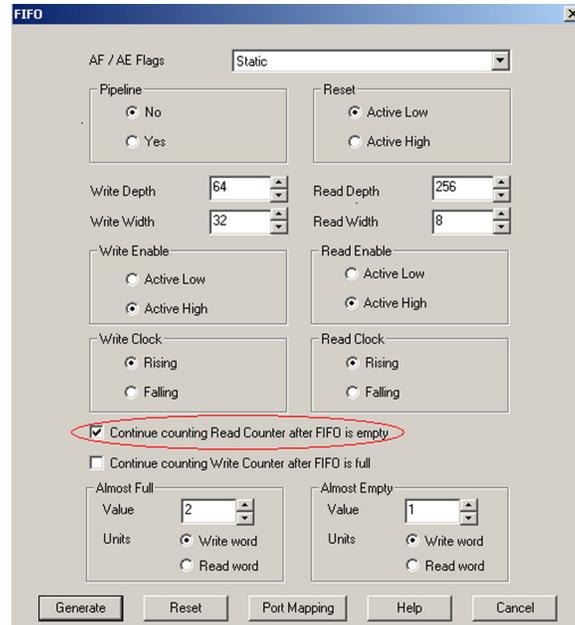


Figure 1 • SmartGen GUI During Device FIFO Generation

Waveform Generator Design

Two essential building blocks of a complete waveform generator design are the initialization block and FIFO. As mentioned in the "FIFO Initialization" section on page 1, there are various solutions to initialize the FIFO block. [Figure 2](#) illustrates the block diagram for a waveform generator design when an external nonvolatile memory is used to initialize device FIFO.

A reference design for initialization block in Figure 2 is presented in the *Embedded SRAM Initialization Using External Serial EEPROM application note*. The "Creating a FIFO Block" section on page 1 discusses the creation and instantiation of the FIFO block.

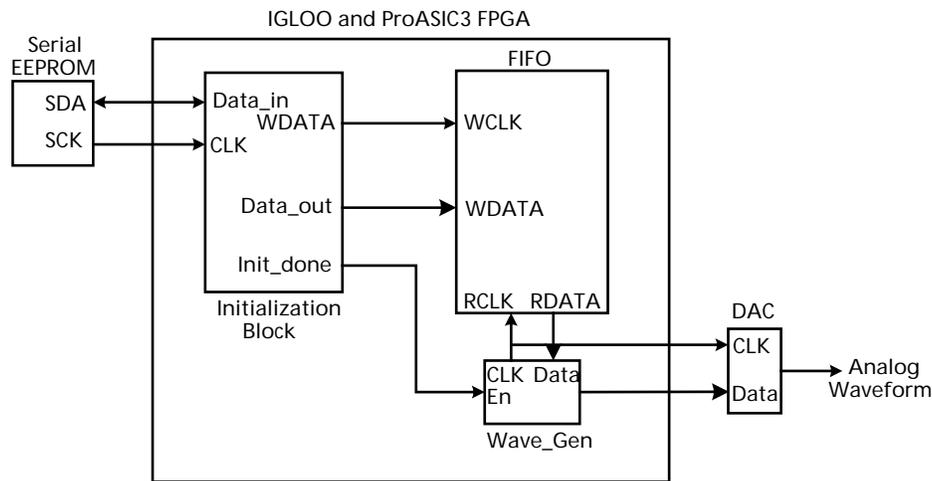


Figure 2 • Block Diagram of the Waveform Generator Using External EEPROM

The Wave_Gen block shown in Figure 2 reads the waveform data points and sends it out of the FPGA to be converted into an analog waveform. The structure and functionality of Wave_Gen block depends entirely to the waveform type (sine wave, square wave, etc.) and the format of the data stored in the FIFO (two data points per row, half or full cycle, etc.). The Sine Table on the Microsemi website contains the initialization data for half of a cycle of a sine wave. If this table is used to initialize FIFO blocks, the contents of the FIFO should be read twice to generate a full cycle of periodic sine wave. The following is an example of the Wave_Gen block in VHDL that uses the half-cycle data, which is stored in the FIFO, and transmits a full sine wave data to the external D/A. The Wave_Gen design in this example, enabled at the end of initialization, reads through the FIFO block (half-cycle sine wave) and changes the sign (+/-) of the read data alternatively at the end of each half-cycle to generate a full-cycle sine wave.

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity wave_gen is
    port( clock, reset_n, enable, empty: in std_logic;
          wave_in : in std_logic_vector(7 downto 0);
          renb_fifo : out std_logic;
          wave_out: out std_logic_vector( 8 downto 0 ));
end;

architecture behave of wave_gen is
    type state_type is ( pos_cycle, neg_cycle);
    signal current_state, next_state: state_type;
    signal current_sign, next_sign: std_logic;
begin

    process( clock, reset_n )
    begin
        if reset_n = '0' then
            current_state <= pos_cycle;
            current_sign <= '0';
        elsif rising_edge( clock ) then
            if enable = '1' then
                if (empty = '1') then
                    current_state <= next_state;
                    current_sign <= next_sign;
                end if;
            end if;
        end if;
    end process;
end behave;

```

```
        end if;
    end if;
end if;
end process;

process(current_state)
begin
    case current_state is
        when pos_cycle =>
            next_state <= neg_cycle;
            next_sign <= '1';
            when neg_cycle =>
                next_state <= pos_cycle;
                next_sign <= '0';
            when others =>
                next_state <= pos_cycle;
                next_sign <= '0';
    end case;
end process;

process(clock, reset_n)
begin
    if (reset_n = '0') then
        wave_out <= (others => '0');
    elsif rising_edge( clock ) then
        wave_out <= current_sign & wave_in;
    end if;
end process;

process (enable, reset_n)
begin
    if (reset_n = '0' or enable = '0') then
        renb_fifo <= '1';
    else
        renb_fifo <= '0';
    end if;
end process;

end;
```

Related Documents

Application Notes

Embedded SRAM Initialization Using External Serial EEPROM

www.microsemi.com/documents/EmbeddedSRAMInit_AN.pdf

User Guides

IGLOO FPGA Fabric User's Guide

http://www.microsemi.com/documents/IGLOO_UG.pdf

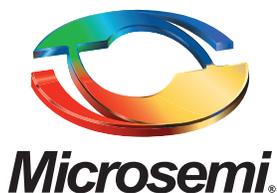
IGLOO, Fusion and ProASIC3 Macro Library Guide

www.microsemi.com/documents/pa3_libguide_ug.pdf

List of Changes

Date	Changes	Page
Revision 2 (July 2015)	Non-technical Updates.	NA
Revision 1 (Nov 2008)	Initial Release.	NA

Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.



Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

© 2015 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi Corporation (MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at www.microsemi.com.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.