Four-Channel Breathing Example
SLG47910

This application shows how to build a four-channel breathing example using the SLG47910 FPGA to control LEDs.

Contents

Terms and Definitions

FPGA	Field Programmable Gate Array
FPGA Editor	Main FPGA design and simulation window
Go Configure Software Hub	Main window for device selection
ForgeFPGA Window	Main FPGA project window for debug and IO programming

References

For related documents and software, please visit:

ForgeFPGA Low-density FPGAs | Renesas

Download our free ForgeFPGA™ Designer software [1] to open the .ffpga design files [1] and view the proposed circuit design.

[1] ForgeFPGA Designer Software, Software Download and User Guide, Renesas Electronics

[2] AN-FG-005 Four-Channel Breathing Example.ffpg ForgeFPGA Design File, Renesas Electronics

[3] SLG47910, Preliminary Datasheet, Renesas Electronics

1. Introduction

Breathing control is a method of slowly fading an LED ON and OFF. This application shows how to build a four-channel breathing example using the SLG47910 ForgeFPGA to control LEDs. This design has four Pulse Width Modulated (PWM) channels that drive LEDs at 50 Hz, 100 Hz, 500 Hz and 1 kHz. The top-level Verilog code combines four instances of the breathing module. Each application of the breathing module has different parameters settings.

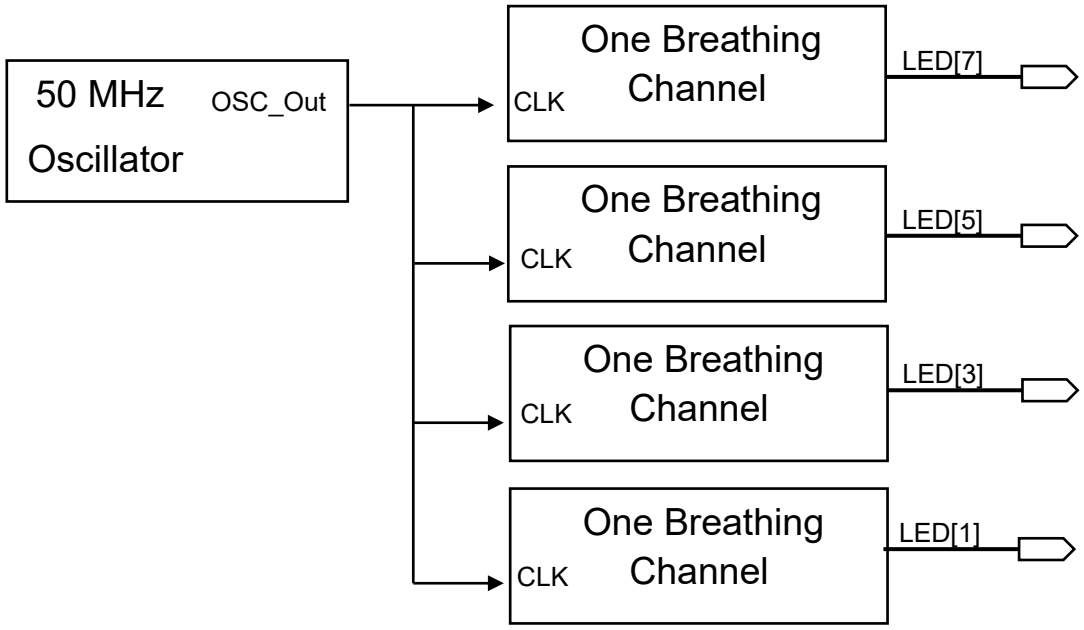


Figure 1: Four-Channel Breathing Application

The following parameters can be set to program the breathing module.

Name	Range	Default Value	Description
IN_CLK_HZ	100MHz – 1MHz	50MHz	Input Frequency
DEPTH	16 - 2	8	It's the depth value of the output PWM counter, defined in bits
PWM_FREQ_HZ	100K - 2	100	It's the output PWM frequency value, defined in Hz
RAMP_MULT	256 - 1	2	It's the value of the ramp multiplier counter, defined in decimal value

For calculating breathing period time in Milli-seconds use the formula below:

PERIOD = (2 * ((2 ^ DEPTH) - 1) * RAMP_MULT) / PWM_FREQ_HZ:

The following signal names are the PINs that are used in the design;

- clk - input clock signal
- nreset - input negative reset signal
- en - input enable signal
- led[7,5,3,1] - output PWM signal

Using the ForgeFPGA Workshop software, the four-channel Verilog code was synthesized, and the bit stream was loaded on to the SLG47910 device. The design uses the internal oscillator as a clocking source for the application. The functional waveforms below (see Figure 2) how four independent channels that have different frequencies and pulse modulation. Each channel drives an LED and turns it ON and OFF with a fading effect.

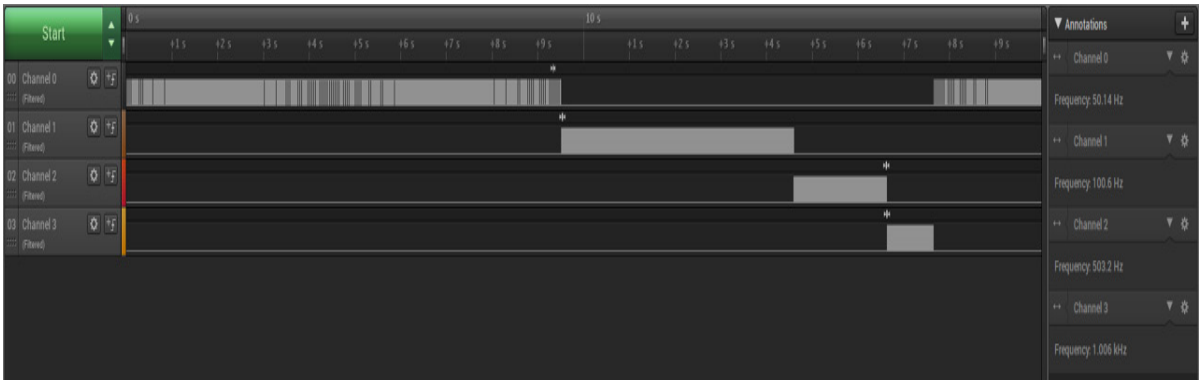


Figure 2: Sequential Breathing Functional Waveforms

2. Ingredients

- ForgeFPGA Device SLG47910V
- ForgeFPGA Development Board with USB cable and power supply
- ForgeFPGA Socket Adaptor Board
- Latest Revision of ForgeFPGA Workshop software
- DIGILENT eight-LED Pmod board (PMOD8LD) or use LEDs on the FPGA development board

3. Four-Channel Breathing Verilog Code

Shown below is the (*top*) module named DemoSequentialBreathing. It consists of four placements of the "breathing" module which is based on the "breathing ctrl " IP block. In each instance of the IP block different parameters are assigned. The Verilog code for the Breathing IP block can be found in the complete design example. It is available for download (AN-FG-005 Four-Channel Breathing Example.ffpga).

```
   // Breathing Demo
(* top *)
module DemoSequentialBreathing #(
parameter IN_CLK_HZ = 44500000,
parameter DEPTH = 8
) (
(* iopad_external_pin *) input nreset,
(* iopad_external_pin, clkbuf_inhibit *) input clk,
(* iopad_external_pin *) output osc_en,
(* iopad_external_pin *) output osc_mode,
(* iopad_external_pin *) output led1,
(* iopad_external_pin *) output led3,
(* iopad_external_pin *) output led5,
(* iopad_external_pin *) output led7
);

// OSC config

assign osc_en = 1'b1;

assign osc_mode = 1'b1;
```

```
wire [3:0] done;

wire next;

assign next = |done;

reg [1:0] seq_counter;

always @(posedge clk) begin

if (!nreset)

seq_counter <= 'h0;

else if (next)

seq_counter <= seq_counter + 1;

end



wire [3:0] en;

assign en[0] = (seq_counter == 0) ? 1 : 0;

assign en[1] = (seq_counter == 1) ? 1 : 0;

assign en[2] = (seq_counter == 2) ? 1 : 0;

assign en[3] = (seq_counter == 3) ? 1 : 0;




breathing #(

IN_CLK_HZ,

DEPTH,

50, // Output PWM = 50 Hz



1 // RAMP_MULT =1

) breathing_led1 (

.clk (clk),

.nreset (nreset),

.en (en[0]),

.out (led1),

.done (done[0])

);



breathing #(

IN_CLK_HZ,

DEPTH,

100, // Output PWM = 100 Hz

1// RAMP_MULT =1
```

```
) breathing_led3 (
.clk (clk),
.nreset (nreset),
.en (en[1]),
.out (led3),
.done (done[1])
);

breathing #(
IN_CLK_HZ,
DEPTH,
500, // Output PWM = 500 Hz
2 // RAMP_MULT = 2
) breathing_led5 (
.clk (clk),
.nreset (nreset),
.en (en[2]),
.out (led5),
.done (done[2])
);

breathing #(
IN_CLK_HZ,
DEPTH,
1000, // Output PWM = 1 MHz
2 // RAMP_MULT = 2
) breathing_led7 (
.clk (clk),
.nreset (nreset),
.en (en[3]),
.out (led7),
.done (done[3])
);
end module
```

4. Floorplan: CLB Utilization

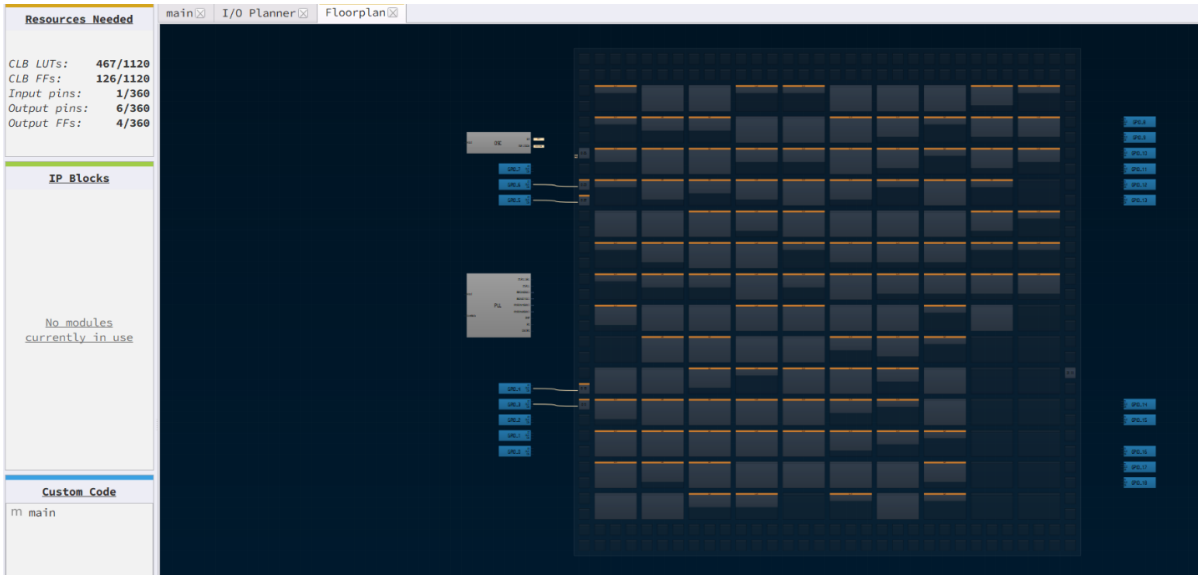


Figure 3: Four Channel Breathing CLB utilization

The Floorplanner tab in the FPGA Editor shows the placement of the CLBs and FFs (Figure 3). The resource utilization is shown in the top left corner.

5. Design Steps

1. Launch the latest version of the Go Configure Software Hub. Select the SLG47910V device and the ForgeFPGA Workshop software will load.
2. Download the design example breathing_demo1.ffpga. If you are not familiar with the ForgeFPGA Workshop software, review the Four-Bit Counter application notes that covers the basic design steps.
3. Open the breathing_demo1.ffpga file after downloading.
4. Open the FPGA editor and review the Verilog code.
5. Open the IO planner tab on the FPGA editor and review the pin assignment.
6. Next select the Synthesize button on the lower left side of the FPGA editor.
7. Select the Generate Bitstream button on the lower left side of the FPGA editor. Check the Logger and Issues tabs to make sure that the bit steam was generated correctly.
8. Now click on the Floorplan tab and see the CLB utilization (Figure 3). Press the Ctrl and the mouse wheel to zoom-in. Confirm that the IOs selected in the IO Planner are shown in the floorplan.
9. Close the FPGA Editor and go to the ForgeFPGA Workshop widow (Figure 4). Selecting the Debug tab will enable the debug controls. Double click on the VDD pin and set VDD= 1.2v. Then double click on VDDIO pin and set VDDIO= 2.0v.
10. Double click on EN and PWR and select Force On. This ensures that all blocks are powered up.
11. The four-channel breathing design is now ready to load onto the FPGA using the development board. Connect the development board to adaptor board. Connect the eight-LED Pmod board to the adapter board. Use PMODA connector on the adaptor board (Figure 5). Connect the USB and power supply to the development board.

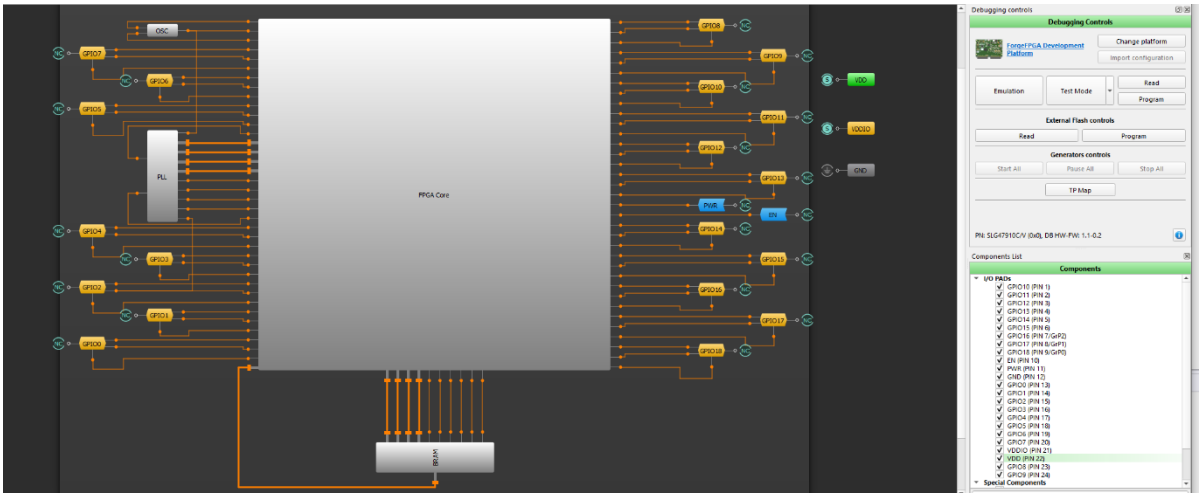
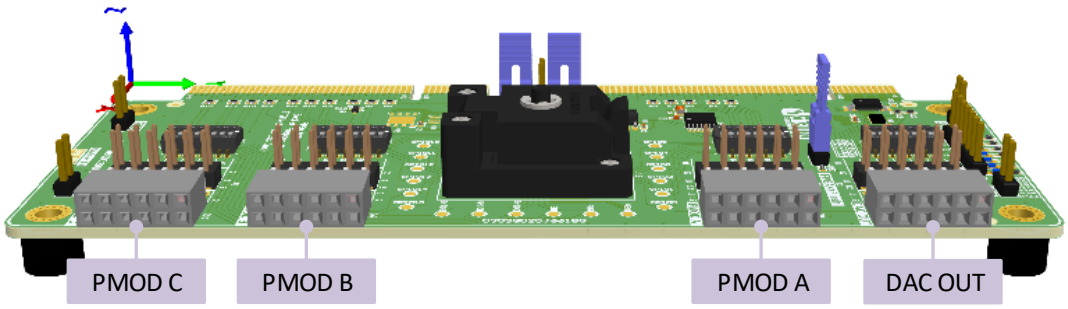


Figure 4: ForgeFPGA Window

12. In the ForgeFPGA Workshop window, select Change platform on the Debugging Controls tab. Choose the ForgeFPGA Development Platform then select Emulation.

Now the design is loaded onto the FPGA device. GPIO[5:2] will have the outputs of the FPGA on them which drive LED[7,5,3,1] on the Pmod board. The LEDs will be fading on and off.



Pin Numbering. View from the connector connection side (PCB, right view)

11	9	7	5	3	1
12	10	8	6	4	2

Figure 5: ForgeFPGA Socket Adaptor

6. Conclusion

This application note shows how the SLG47910 can be used to control a complicated sequencing of LEDs. This testcase is available for download (AN-FG-005 Four-Channel Breathing Example.ffpga,) . If interested, please contact the ForgeFPGA Business Support Team.

7. Revision History

Revision	Date	Description
1.00	Mar 3, 2022	Initial release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use o any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.