SLG47920/21 Errata Note

This document contains the known errata for the SLG47920/21 and the recommended workarounds.

Go Configure software is used to configure the SLG47920 and SLG47921 devices. The configuration is done using Project Settings, setting the properties of the IP blocks in the main Go Configure window and filling out configuration parameters in the IO Planner. In this Errata the configuration settings are indicated by ***BOLD*** labels. Also, refer to the SLG47920/21 data sheet on the Renesas website.

ForgeFPGA Low-density FPGA: Small-scale Programmable Logic Device | Renesas

1. Information

Package(s)	48P-QFN, 48P-WLCSP, and 40P-QFN

2. Errata Summary

Issue #	Issue Title
1	GPIO1 2x Configuration Pull-Up is Enabled during Reset
2	Unexpected SPI GPIO Behavior after Reset
3	Short Pulse On GPIOs Which Are Driven By Status Signals During Transition To Sleep Or Reset Modebelow
4	While Transitioning Into Sleep And Reset Mode Short Pulses Can Be Observed On Dedicated Pins GPIO8 (PLL0_FOUT0) and GPIO9 (PLL1_FOUT)
5	PLL Behavior When BYPASS Mode Is Selected
6	PLL0_FOUT1 Dependent On PLL0_FOUT0 Settings
7	LVDS IO And GPIO Pad Driving Conflict Entering Sleep Or Reset Mode
8	LVDS Interface Is Disabled Before All User Clocks Are Gated Entering Sleep Or Reset Modes

3. Errata Details

3.1 GPIO1 2x Configuration Pull-Up is Enabled during Reset

3.1.1 Effect

During reset the SLG47920/21V GPIO1 defaults to the 2x weak Pull-Up.

3.1.2 Conditions

Occurs at all PVT corners during Reset.

3.1.3 Technical Description

If ***GPIO keep*** in the Go Configure Project Settings is enabled during a Reset operation and the device pin nRESET is LOW, the GPIO settings should not change to the default settings. However, GPIO1 defaults to the 2x pull-up when 1x should have been kept since ***GPIO keep = "Enable".*** Figure 1 shows the logical representation of the SLG47920/21 GPIO structure.

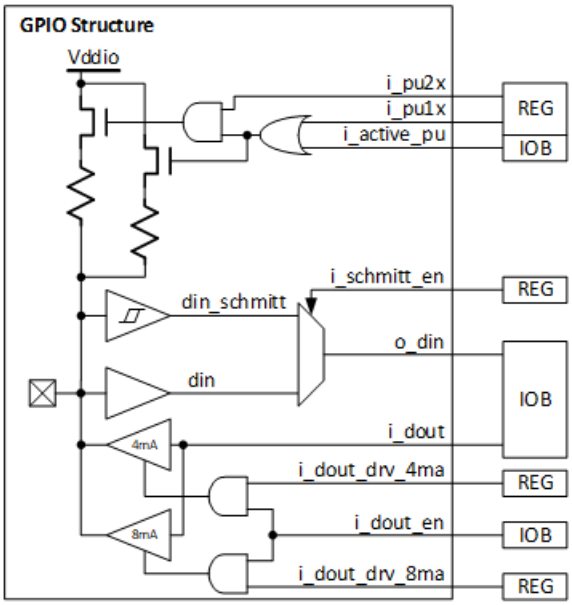


Figure 1. GPIO Structure

3.1.4 Workaround

There is no current workaround but after nRESET goes HIGH, the correct GPIO setting will be loaded.

3.2 Unexpected SPI GPIO Behavior after Reset

3.2.1 Effect

After the chip exits from the Reset state (nRESET = L → H) and tries to load from the external SPI Flash, GPIO0 (SPI_CS) and GPIO1(SPI_SCLK) can be inactive if configuration register bits, ***i_dout_drv_4ma = 0*** and ***i_dout_drv_8ma = 0*** (see Figure 1), were set before the transition to the Reset state. These register bits are set in the Go Configure software using the GPIO property settings.

3.2.2 Conditions

Occurs at all PVT corners exiting reset.

3.2.3 Technical Description

Each GPIO can be configured to meet the design requirements of the customer. If SPI programming mode is selected GPIO0 (SPI_CS) and GPIO1(SPI_SCLK) are controlled by the SLG47920/21 internal circuitry. If the 4ma and 8ma driver configurations are set to zero, the SPI configuration control of GPIO0 and GPIO1 is disrupted.

3.2.4 Workaround

Configure GPIO0 and GPIO1 so that configuration register bits that drives ***i_dout_drv_4ma = 1*** or ***i_dout_drv_8ma = 1*** (see Figure 1). Setting i_dout_drv_4ma and i_dout_drv_8ma is done using the property settings in the Go Configure software by selecting a GPIO from the main Go Configure window and adjusting the property ***Drive strength select.***

3.3 Short Pulse On GPIOs Which Are Driven By Status Signals During Transition To Sleep Or Reset Mode

3.3.1 Effect

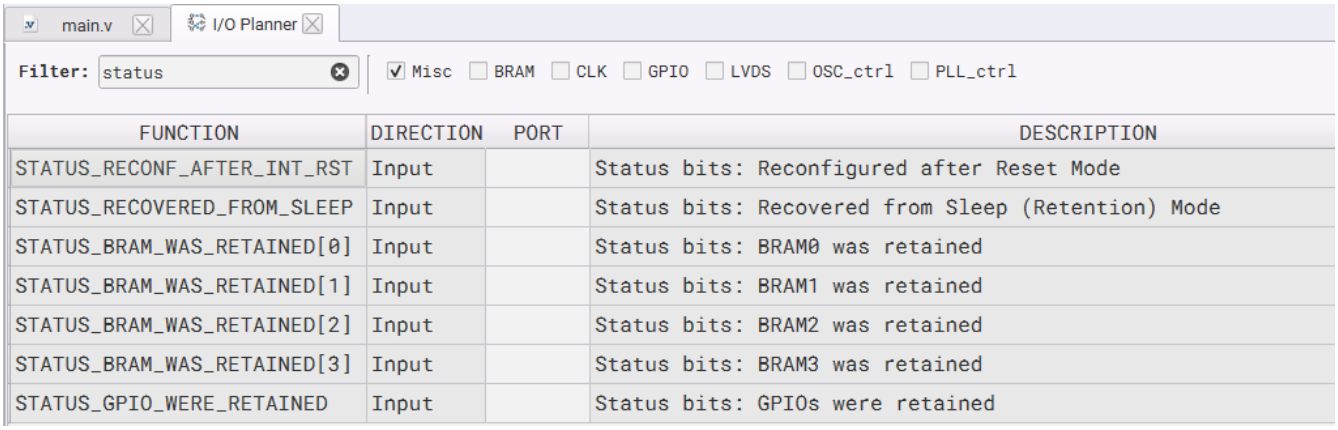
Short pulses are observed on GPIOs during Sleep and Reset modes when it is driven with status signals in a combinatorial way.

3.3.2 Conditions

When status signals drive GPIOs directly in combinatorial way during transition to Sleep and Reset modes, short pulse occurs on corresponding GPIOs.

3.3.3 Technical Description

In SLG47920 and SLG47921 there are status signals that indicate previous states of the device. These signals are enabled in the Go Configure IO planner and can be connected to GPIOs (see Figure 2). These signals are intended to be used inside user logic after returning from Sleep/Reset modes when their states are stable. However, when these signals drive GPIOs directly in combinatorial way, some of them might generate short pulses on GPIOs. This is due to fact that they change their state in process of transition to Sleep and Rest modes.



FUNCTION	DIRECTION	PORT	DESCRIPTION
STATUS_RECONF_AFTER_INT_RST	Input		Status bits: Reconfigured after Reset Mode
STATUS_RECOVERED_FROM_SLEEP	Input		Status bits: Recovered from Sleep (Retention) Mode
STATUS_BRAM_WAS_RETAINED[0]	Input		Status bits: BRAM0 was retained
STATUS_BRAM_WAS_RETAINED[1]	Input		Status bits: BRAM1 was retained
STATUS_BRAM_WAS_RETAINED[2]	Input		Status bits: BRAM2 was retained
STATUS_BRAM_WAS_RETAINED[3]	Input		Status bits: BRAM3 was retained
STATUS_GPIO_WERE_RETAINED	Input		Status bits: GPIOs were retained

Figure 2. Status Bits in IO Planner

3.3.4 Workaround

Avoid driving GPIO with status signals in general and in combinatorial way particularly. If driving GPIO with status signal is necessary – put it through DFF to avoid state change after clocks are stopped during transition to Sleep and Reset modes.

3.4 While Transitioning Into Sleep And Reset Mode Short Pulses Can Be Observed On Dedicated Pins GPIO8 (PLL0_FOUT0) and GPIO9 (PLL1_FOUT)

3.4.1 Effect

Short pulses, extra clocks or unexpected frequency may be observed when transitioning into Sleep and Reset mode on GPIO [9:8] when using ***PLL0_FOUT0*** to GPIO8 and/or ***PLL1_FOUT*** to GPIO9

3.4.2 Conditions

When nSLEEP = L or nRESET = L, ***GPIO keep*** in the project settings is enabled, GPIO [9:8] is enabled and, PLL enabled with options ***PLL_FOUT0 to GPIO8 = "Enable"*** and/or ***PLL_FOUT0 to GPIO9 = "Enable".***

3.4.3 Technical Description

When going into Sleep mode, the PLL enters the standby state, which causes the PLL clock to stop. This transition can result in extra short pulses or an unexpected number of PLL output clocks while the PLL enters Sleep mode.

3.4.4 Workaround

Before the start of transition to Sleep mode put the PLL in standby before bringing nSLEEP low. This can be done by bring to the PLL clock source into an idle state or by using the PLL enable signal to idle the PLL output.

3.5 PLL Behavior When BYPASS Mode Is Selected

3.5.1 Effect

Clock pulses may appear at PLL0_FOUT1 before the LOCK signal when BYPASS = 0 and the PLL clock is gated with the LOCK signal. In BYPASS mode, additional clock pulses are needed to provide a clock signal from the PLL0 and PLL1 external reference input (both synchronous and asynchronous).

3.5.2 Conditions

PLL BYPASS mode is enabled, and an external clock source is being used.

3.5.3 Technical Description

The ForgeFPGA PLL has a BYPASS mode where an external clock source is used to drive the PLL outputs. The clock path has synchronization registers to help maintain clock tree integrity. Additional clock pulses are required for the external clock to propagate to the PLL outputs.

3.5.4 Workaround

None.

3.6 PLL0_FOUT1 Dependent On PLL0_FOUT0 Settings

3.6.1 Effect

The ***PLL0_FOUT1*** output does not work correctly under the default post divider settings of ***PLL0_FOUT0***.

3.6.2 Conditions

The issue occurs when both PLL0 outputs are used but ***PLL0_FOUT0*** has default settings.

3.6.3 Technical Description

The users cannot use ***PLL0_FOUT1*** for clocking the sequential logic in the FPGA Core if ***PLL0_FOUT0*** is not used (meaning post-divider has the default settings of PLL0_POSTDIV0/1_OUT0 = 3'b000 and PLL0_POSTDIV1_OUT0 = 3'b000) or user unmapped PLL0_POSTDIV0/1_OUT0 IOBs. The code below (Figure 3) shows the PLL Verilog code assignments that produce the problem. Each of the assignments is mapped on to the PLL section of the IO planner (Figure 4).

```
//Assigned signals
    assign pll0_en = 1'b1;
    assign pll0_refdiv = 6'b00_001; // Equivalent value in decimal form 6'd1
    assign pll0_fbdiv = 12'b0000_0011_0001; // Equivalent value in decimal form 12'd49
    assign pll0_postdiv0_out0 = 3'b000;
    assign pll0_postdiv1_out0 = 3'b000;
    assign pll0_postdiv0_out1 = 3'b111;  // Equivalent value in decimal form 3'd7
    assign pll0_postdiv1_out1 = 3'b111;  // Equivalent value in decimal form 3'd7
    assign pll0_bypass = 1'b0;
    assign pll0_ref_clk_sel = 2'b10;    //Ext. clk async gate
```

Figure 3. PLL Verilog Code Assignments

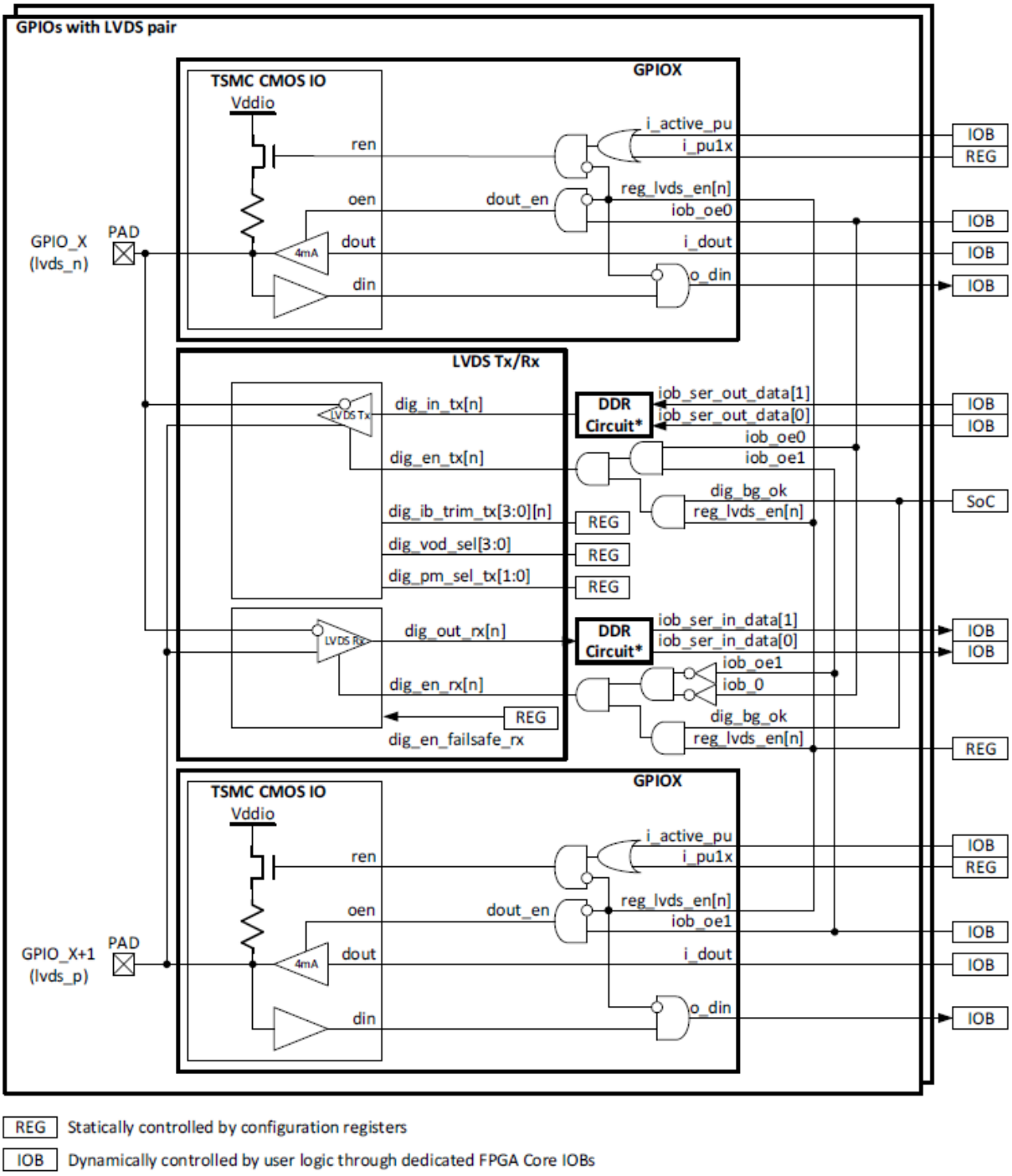
PLL0_FBDIV[6]	Output	pll0_fbdiv[6]	PLL0 Feedback Divider value (bit 6)
PLL0_FBDIV[7]	Output	pll0_fbdiv[7]	PLL0 Feedback Divider value (bit 7)
PLL0_FBDIV[8]	Output	pll0_fbdiv[8]	PLL0 Feedback Divider value (bit 8)
PLL0_FBDIV[9]	Output	pll0_fbdiv[9]	PLL0 Feedback Divider value (bit 9)
PLL0_FBDIV[10]	Output	pll0_fbdiv[10]	PLL0 Feedback Divider value (bit 10)
PLL0_FBDIV[11]	Output	pll0_fbdiv[11]	PLL0 Feedback Divider value (bit 11)
PLL0_POSTDIV0_OUT0[0]	Output		PLL0 Fout0 Postdiv Stage0 Value (bit 0)
PLL0_POSTDIV0_OUT0[1]	Output		PLL0 Fout0 Postdiv Stage0 Value (bit 1)
PLL0_POSTDIV0_OUT0[2]	Output		PLL0 Fout0 Postdiv Stage0 Value (bit 2)
PLL0_POSTDIV1_OUT0[0]	Output		PLL0 Fout0 Postdiv Stage1 Value (bit 0)
PLL0_POSTDIV1_OUT0[1]	Output		PLL0 Fout0 Postdiv Stage1 Value (bit 1)
PLL0_POSTDIV1_OUT0[2]	Output		PLL0 Fout0 Postdiv Stage1 Value (bit 2)
PLL0_REF_CLK_SEL[0]	Output	pll0_ref_clk_sel[0]	PLL0 Reference Clock Source Select (bit 0)
PLL0_REF_CLK_SEL[1]	Output	pll0_ref_clk_sel[1]	PLL0 Reference Clock Source Select (bit 1)
PLL0_BYPASS	Output	pll0_bypass	PLL0 Bypass mode enable
PLL0_EN	Output	pll0_en	PLL0 Enable (active HIGH)

Figure 4. PLL IO Planner Settings

3.6.4 Workaround

Map PLL0_POSTDIV0/1_OUT0 IOBs and use PLL0_POSTDIV0/1_OUT0 (with non-zero settings) before using PLL0_POSTDIV0/1_OUT1.

3.7 LVDS IO And GPIO Pad Driving Conflict Entering Sleep Or Reset Mode



REG Statically controlled by configuration registers

IOB Dynamically controlled by user logic through dedicated FPGA Core IOBs

Figure 5. GPIOs Pair with LVDS Capability

3.7.1 Effect

Temporary LVDS and GPIO driver conflict during transition to Sleep and Reset modes. The conflict is resolved after the Sleep and Reset control signals propagates to steady state.

3.7.2 Conditions

Occurs when entering Sleep or Reset mode and the LVDS Tx drivers are enabled prior to entering Sleep or Reset mode.

3.7.3 Technical Description

Shown in Figure 5 is a GPIO pair with LVDS capability. There is a LVDS IO to GPIO pad driving conflict while entering Sleep or Reset mode. The LVDS IO to GPIO pad driving conflict occurs when the GPIO OE activates the GPIO output while LVDS OE deactivates the LVDS outputs. After about 4 ns, the GPIO starts driving the GPIO pad to LOW, but the LVDS IO continues driving the GPIO pad to HIGH until the Sleep or Reset control signals settle after about 50 ns.

3.7.4 Workaround

To avoid the LVDS IO and GPIO driving conflict, it is proposed to disable the LVDS interface outputs before entering Sleep or Reset mode. The Sleep or Reset mode should be triggered by the FPGA Core dedicated IOB.

3.8 LVDS Interface Is Disabled Before All User Clocks Are Gated Entering Sleep Or Reset Modes

3.8.1 Effect

The LVDS interface is disabling earlier than user clocks being gated.

3.8.2 Conditions

When the LVDS is enabled and prior to entering Sleep or Reset modes.

3.8.3 Technical Description

The LVDS interface is disabling earlier than all user clocks are gated. When the LVDS Enable goes LOW, LVDS outputs are disabled asynchronously to the LVDS clock, but the FPGA core continues transmitting and receiving data. There is no mechanism to prevent data corruption and the LVDS clock output interruption.

3.8.4 Workaround

In the Go Configure software, select the FPGA Core properties (Figure 6). Then use FPGA Core dedicated IOBs to assert Sleep and Reset mode. Use the IOBs to stop LVDS transmit and receive before initiating transition to Sleep or Reset modes.

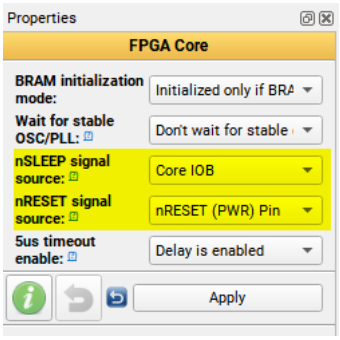


Figure 6. FPGA Core Properties

4. Revision History

Revision	Date	Description
1.01	Feb 09, 2026	Fixed reference link and typos
1.00	Dec 9, 2025	Initial release