Author: Yu-Han Sun
Date: September 19, 2016

Introduction

This app note is a corollary to another app note **AN-1090 Simple I2C IO Controllers with SLG46531V**. AN-1090 explains how to make I2C IO Controllers with separate input and output pins. However, this app note will explain how to setup an 8-bit bus controller which combines input pins with output pins. Refer to Figure 1 for the System Level View.

Digital Input/Output and OE

To combine inputs with outputs, each pin interfacing the bus is set to 'Digital Input/Output'. Only 9 GPIOs in the SLG46531V are 'Digital Input/Output' capable and we will be using 8 of those pins: PINs #19, 18, 16, 14, 13, 7, 5, 3. Each of these pins have an OE signal that toggles the mode. Set the properties as shown in Figure 3 and setup the matrix connections as shown in Figure 2.

Each output signal is controlled by an I2C Virtual Input. OE is controlled by 2-bit LUT0. The inputs to the 2-bit LUT0 are both gnd. Therefore, if the LUT is configured as in Figure 4a, then OE will be logic 1. If the LUT is configured as in Figure 4b, then OE will be logic 0. We will be using I2C to change the LUT configuration on the fly.

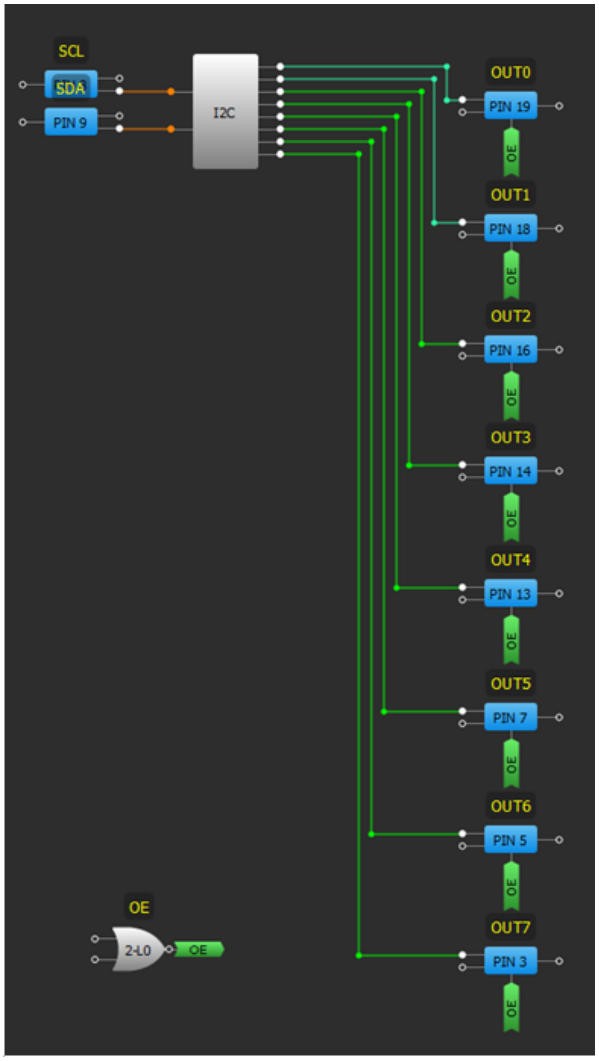

Figure 2. GreenPAK Design

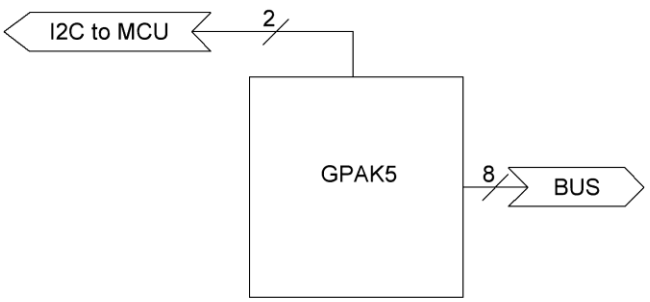

Figure 1. System Level View

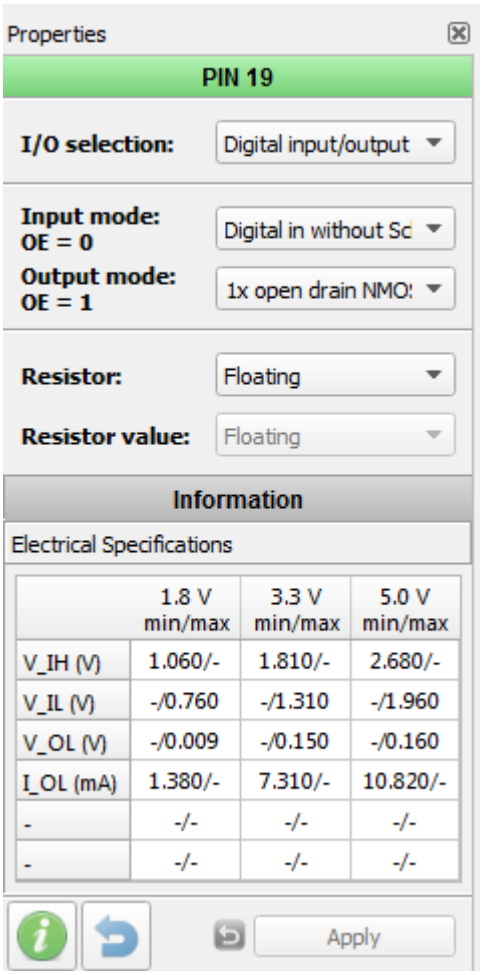

Figure 3. PIN Configuration

IN1	IN0	OUT
0	0	1
0	1	0
1	0	0
1	1	0

Figure 4a. OE = 1

IN1	IN0	OUT
0	0	0
0	1	0
1	0	0
1	1	0

Figure 4b. OE =0

I2C Bus Write

To write the bus, the MCU must send 2 commands: first write to the I2C Virtual Inputs, then set OE = 1 by re-configuring 2-bit LUT0:

i) **W: I2C Virtual Inputs**, write to address 0xF4. Each bit corresponds to an I2C Virtual Input. The order from left to right is PINs #19, 18, 16, 14, 13, 7, 5 and 3.

0xF4	0	0	0	0	0	0	0	0

0xF4	1	1	1	1	1	1	1	1

ii) **W: 2-bit LUT0**, write to the second nibble of address 0x96 with the byte value 0x8. To mask the first nibble, read the byte and change only the latter four bits.

0x96	X	X	X	X	1	0	0	0

I2C Bus Read

To read from the bus, the MCU must send 2 commands: first set OE = 0 by reconfiguring 2-bit LUT0, then read from the GPIO Input Levels.

i) **W: 2-bit LUT0**, write to the second nibble of address 0x96 with the byte value 0x0. To mask the first nibble, read the byte and change only the latter four bits.

0x96	X	X	X	X	0	0	0	0

ii) **R: Input levels**, read from address 0xF0 and 0xF6 the input levels of PIN#3, 5, 7 and PIN#13, 14, 16, 18, 19 respectively. Then parse the data based on bit location.

0XF0	X	X	PIN3	X	PIN5	X	PIN7	X

0XF6	X	PIN13	PIN14	X	PIN16	0	PIN18	PIN19

Warning

The following warning can be ignored because 2-bit LUT0 inputs are intentionally left static and the truth table will be re-configured through I2C.

Time	Event	Rule	Note
14:27:21	Fail	2-bit LUT0/DFF/LATCH0: The truth table is configured incorrectly.	The truth table is configured so that all combinations of the inputs that are connected to the blocks, do not cause changes on the output.
14:27:21	Warning	2-bit LUT0/DFF/LATCH0: No input connected.	2-bit LUT0/DFF/LATCH0's input is not connected.

Figure 5. Warnings and Rules Checker

Examples I2C Commands

The output of each example is shown in Figure 7 through 10, which are screenshots from the I2C Tool.

Emulation

Each Digital IO pin has a pull-up active-low button. Enable I2C Tools after Emulation.

Syntax: **[** is the start bit, **]** is the stop bit, and **SA** is the slave address with a r/w bit.

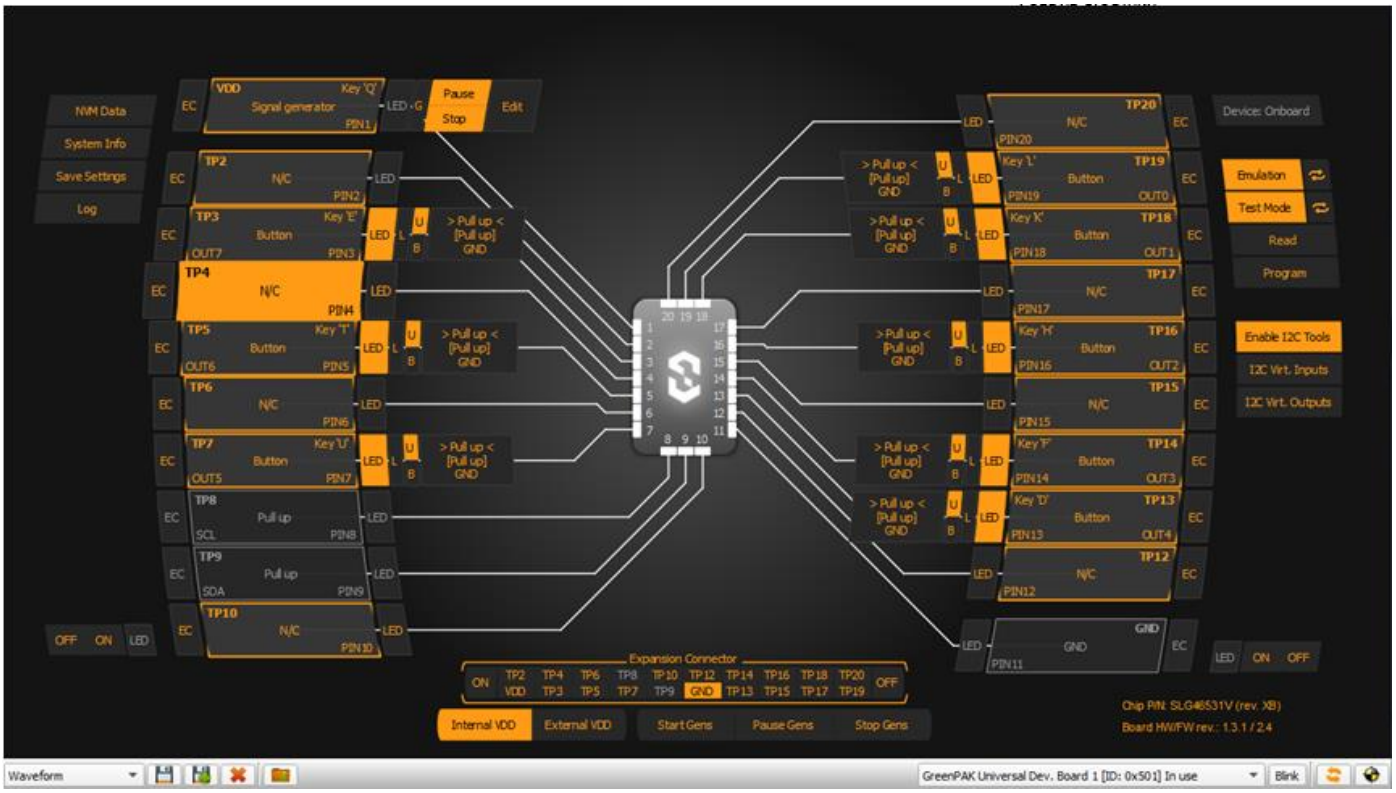

Figure 6. Emulator Configuration

i) I2C write to I2C Virtual Inputs the logic 0, 0, 0, 1, 0, 1, 1, 1.

[SA 0xF4, 0xE8 **]**

ii) I2C write 2-bit LUT0 configuration to 1, 0, 0, 0:

[SA 0x96 0x08 **]**

iii) I2C write 2-bit LUT0 configuration to 0, 0, 0, 0:

[SA 0x96 0x00 **]**

iv) I2C Read the input levels from PINs #7, 5 and 3 and also #19, 18, 16, 14 and 13:

[SA 0xF0 **[** 0xSA read **]**

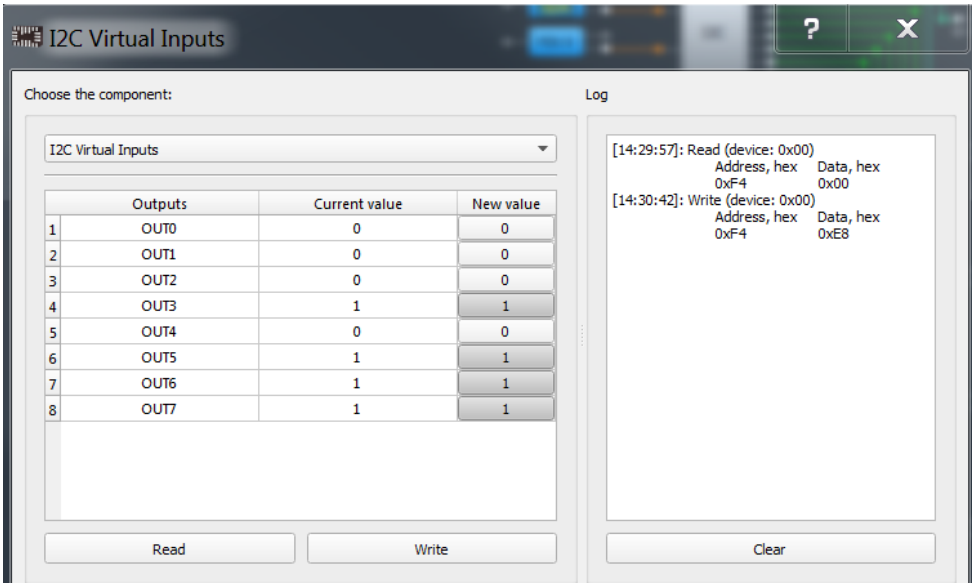

Figure 7. Example i

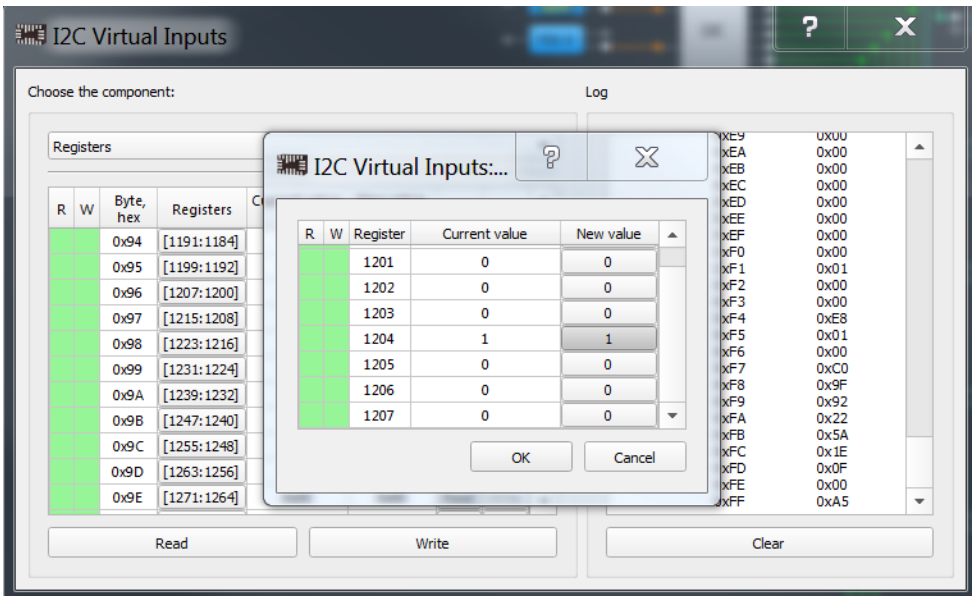

Figure 8. Example ii

v) I2C write 2-bit LUT0 configuration to 0, 0, 0, 0:

[SA 0x96 0x00 **]**

vi) I2C Read the input levels from PINs #7, 5 and 3 and also #19, 18, 16, 14 and 13:

[SA 0xF0 **[** 0xSA read **]**

[SA 0xF6 **[** 0xSA read **]**

Conclusion

By the end of this app note, you should be able to make a GreenPAK Design and the I2C Commands. Unlike AN-1090, this design uses less GPIOs at the expense of more I2C commands.

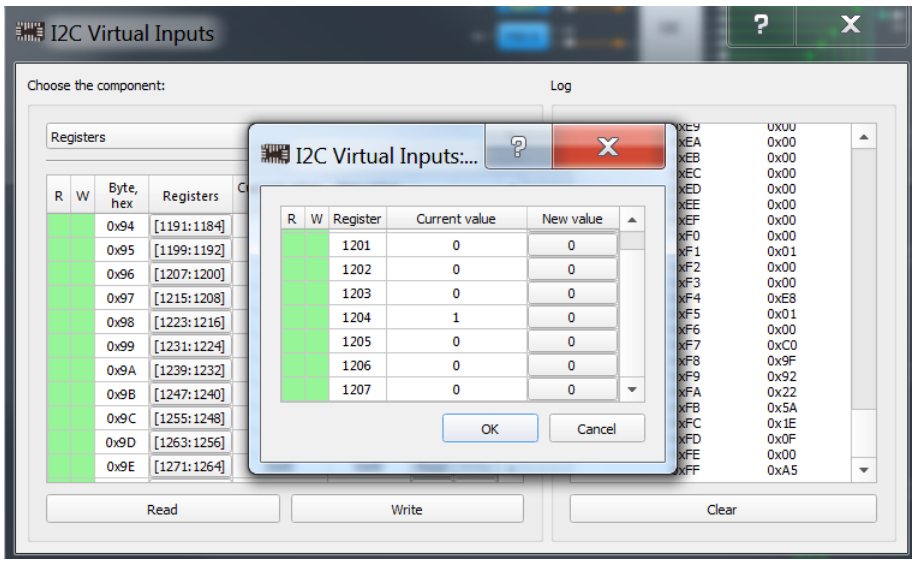

Figure 9. Example iii

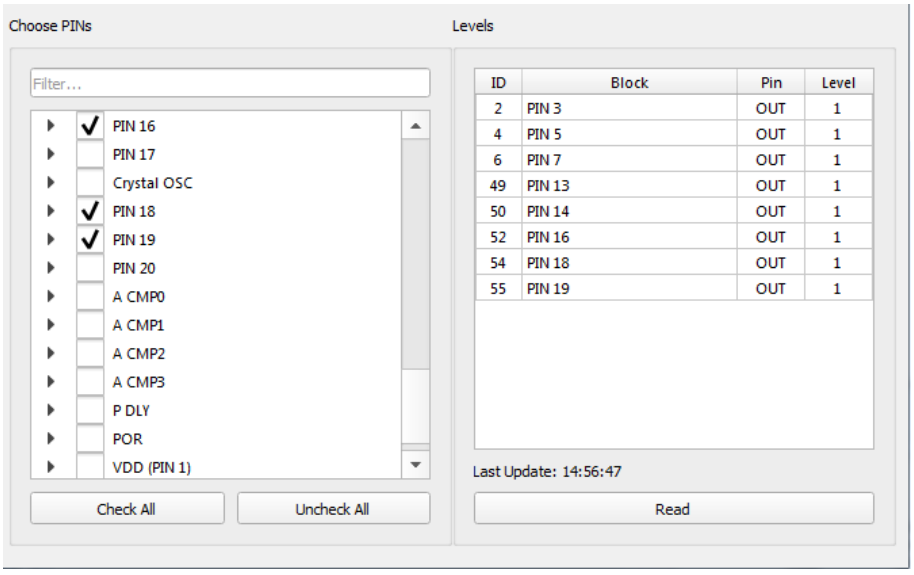

Figure 10. Example iv

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 who are 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 to develop 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 from 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 of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

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-us/.

Trademarks

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