Application Note

SLG46880/1 Asynchronous State Machine

AN-CM-238

Abstract

This application note focuses on the Asynchronous State Machine in a broad approach, with discrete sections devoted to the ASM blocks that have been added or significantly changed in SLG46880 and SLG46881 when compared to previous GreenPAK devices.

Contents

Figures

Tables

1   Terms and Definitions

ASM	Asynchronous state machine
IC	Integrated circuit
GUI	Graphical user interface

2   References

For related documents and software, please visit:

https://www.dialog-semiconductor.com/products/greenpak.

[1]   GreenPAK Designer Software, Software Download and User Guide, Renesas Electronics
[2]   GreenPAK Development Tools, GreenPAK Development Tools Webpage, Renesas Electronics
[3]   GreenPAK Application Notes, GreenPAK Application Notes Webpage, Renesas Electronics
[4]   AN-CM-237 SLG46880/1 F(1) Block, Application note, Renesas Electronics
[5]   AN-CM-236 SLG46880/1 Dynamic Memory Block, Application note, Renesas Electronics

Author: Alexander Richardson

3   Introduction

Renesas Electronics has released the SLG46880 and SLG46881 GreenPAK™ ICs. This application note focuses on the Asynchronous State Machine (ASM) in a broad approach, with discrete sections devoted to the ASM blocks that have been added or significantly changed when compared to previous GreenPAK devices.

Some of the most notable features that differentiate the SLG46880 and SLG46881 GreenPAK ICs from previously-released GreenPAK ICs are within the ASM. To begin, the SLG46880 and SLG46881 contain 12 states; a greater number of states than previous GreenPAK ICs. Also, the ASM now contain several new state-dependent blocks; Dynamic Memory, a F(1) Computational Macrocell, and an extra, redesigned ASM Output block. These extra states and additional blocks enable the user to create more modular designs that reduce both complexity and power consumption.

4   The Asynchronous State Machine

The SLG46880 and SLG46881 both contain 12 asynchronous states, and more flexibility in tying states together than earlier GreenPAK devices. As an example of the ASM, a solar panel state machine using the SLG46880 is shown in Figure 1.
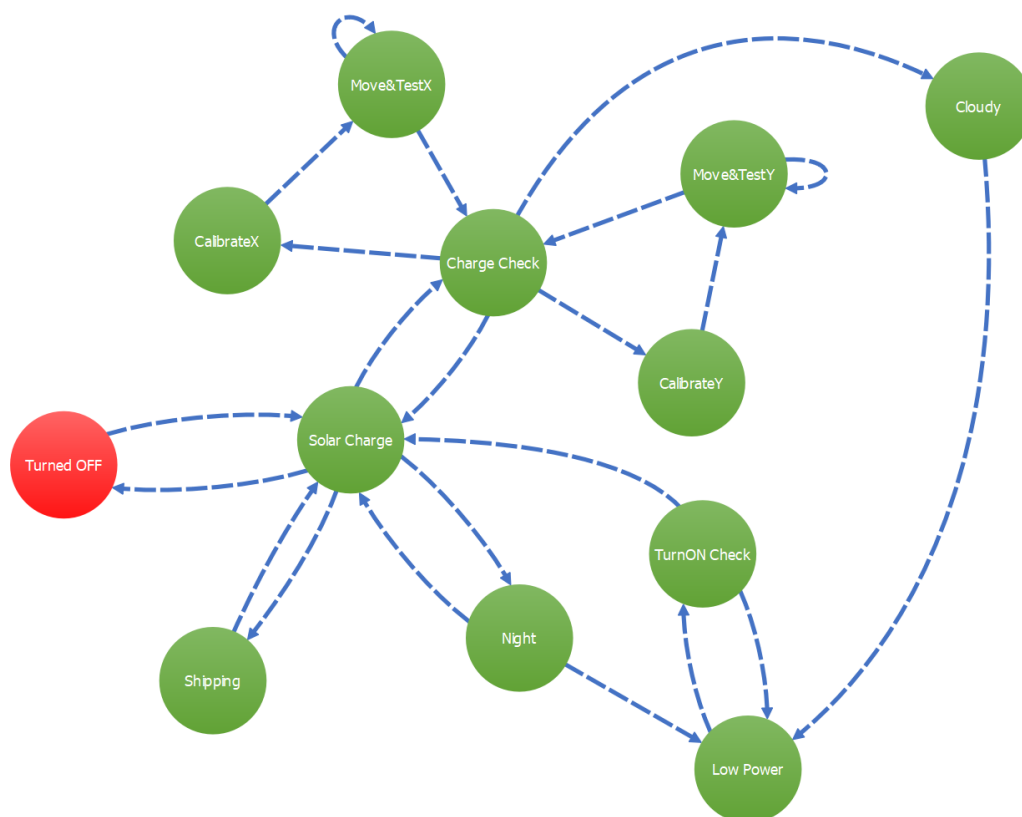
Figure 1: Solar Panel 12-State Machine

The solar panel state machine shown above uses all 12 of the available states, which lets the user maximize options for power savings, delays between transitions, and different modes of operation. Figure 1 also showcases a new capability within the "Move&TestX" and "Move&TestY" blocks; a state is capable of transitioning to itself! Test and check operations, as well as other iterative actions, are easy to accomplish and require fewer blocks.

SLG46880/1 Asynchronous State Machine

Configuring States

States can be configured in the same manner as previous ASM-compatible GreenPAK devices. First, the ASM window is opened using the ASM Editor button in the top toolpanel (Figure 2). If states have not yet been defined the ASM Editor will show two unconnected states. To connect two states, select the outside ring of one state, then click on the ring of the second state.

Figure 2: ASM Button

Figure 3: ASM Connection Flow

When connecting a state to another, the first state chosen will always be the initial state, and the second the transitioned state. A state may be looped to itself by selecting the state's outside ring on both the first and second click. There is no limit to the number of states to which a state may transition; a state may transition to all 12 states, or it may transition to none.

To rename a state, simply double click on the inside of the state circle, or select the state to access the "State name" setting from the left-hand Properties toolbar. Renaming states from the default is a good practice for ensuring that state transitions are correctly made; a 12-state process can become confusing when the states are named "State1" through "State12".

State Transitions

ASM state changes are triggered by a HIGH signal at the ASM Subsystem block, found within the Components list. Prior to selecting this box, ensure that your ASM states have all been properly connected In the ASM Editor. Once the ASM Subsystem component has been check-marked in the Components list, a box will appear in the State Dependent Items section of your GreenPAK Designer workspace. This new block will show the other available states to which the current state may transition. Figure 4 shows the ASM Subsystem block in the "Night" state from the Figure 1 example; the only transition available for the "Night" state is to "Low Power", which will occur on the rising edge of the 6 Hour timer implemented in the DM0_0 (NightTimer) DM Block. Use of the dynamic memory blocks are outlined in a following section.
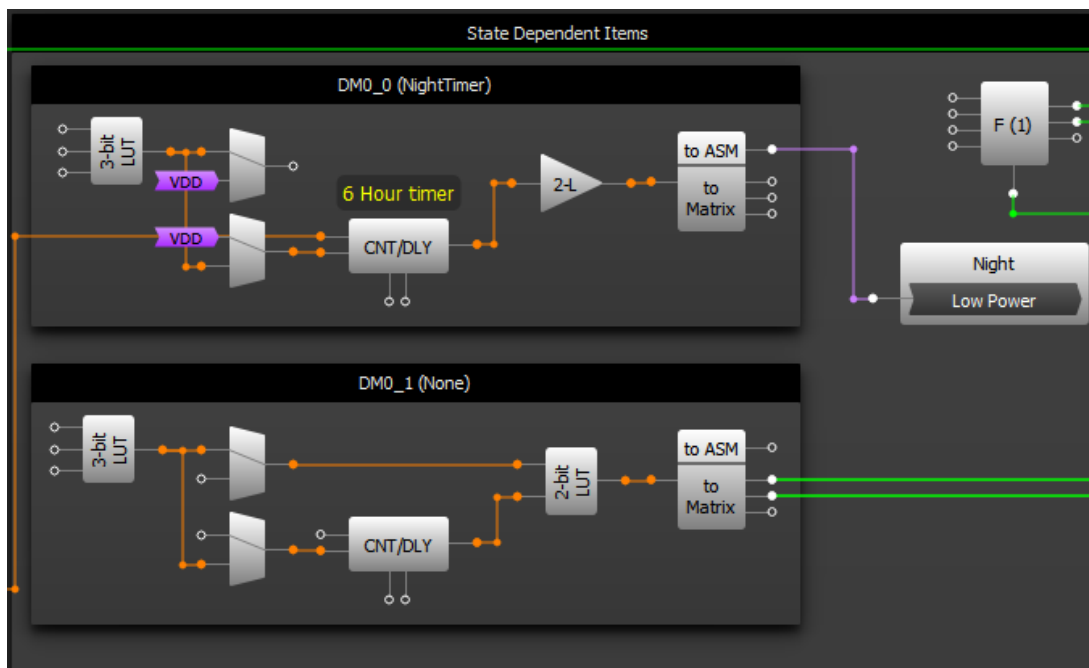
SLG46880/1 Asynchronous State Machine

Figure 4: ASM Subsystem Block

ASM Transition Timing

Several key factors should be kept in mind when considering the asynchronous behavior of the state machine:

1. No clock source is needed, it reacts only to input signals.
   a. Input signals must be greater than the minimum active-HIGH pulse width. The minimum time is 12 to 28 nanoseconds, depending upon VDD.
2. The input signals do not have to be synchronized to each other, the macrocell will react to the earliest valid signal for state transition.
   a. It will be indeterminate which state takes over if the signals occur at precisely the same time. However, a change from the original state will absolutely occur.
3. The macrocell only consumes power while in state transition.
4. If a state transitions to another state, in which the condition for transition is already met, the total time within the second state will be Tst_sequential_delay, nominally rated at 0.2 to 0.5 microseconds, before transitioning to the third state.
   a. This may create a closed loop, in which two or more states cycle to each other quickly and continuously. Much like when coding, this behavior may be undesirable and should be considered when deciding conditions for state transitions.

Further information about asynchronous timing behavior can be viewed in section 16.7 of the SLG46870/71/80/81 datasheet, available online.

5 New Blocks Overview

Several new blocks have been introduced in the SLG46880 and SLG46881. Regarding state-machine operations, several Dynamic Memory [DM] components, a F(1) macrocell computation block, and two ASM Output blocks have been added. These blocks enable greater differences in the operation of states, which has the benefit of saving components, power, and complexity.

For a more in-depth look of the DM or F(1) blocks than what is presented within this application note the associated application notes should be viewed. The following overview sections are a condensed view of the information from these application notes.

DM Blocks

AN-CM-236 SLG46880/1 Dynamic Memory Block describes the Dynamic Memory (DM) blocks in SLG46880/1 and how to use them.

The main advantage of DM blocks is that they can be reconfigured to perform different functions in different states of the SLG46880/1's 12-state ASM. This makes them a very flexible component, since they can be used one way in State 0 and another way in State 1.

For example, in the solar panel use-case from Figure 1, the DM CNT/DLY block within the "Night" state and the "Cloudy" state might be different. Both states eventually transition to "Low Power", which checks for sunlight with a periodic transition to "TurnON Check", but "Cloudy" might wish to switch within a couple of minutes or seconds, whereas the "Night" state might want to idle for hours before switching to the "Low Power" state.

There are 4 DM blocks in the GreenPAK SLG46880/1. An unconfigured DM block is shown in Figure 5. All the DM blocks in the SLG46880/1 have the following resources:

- 2 look-up tables: a 3-bit LUT and a 2-bit LUT
- 2 multiplexers
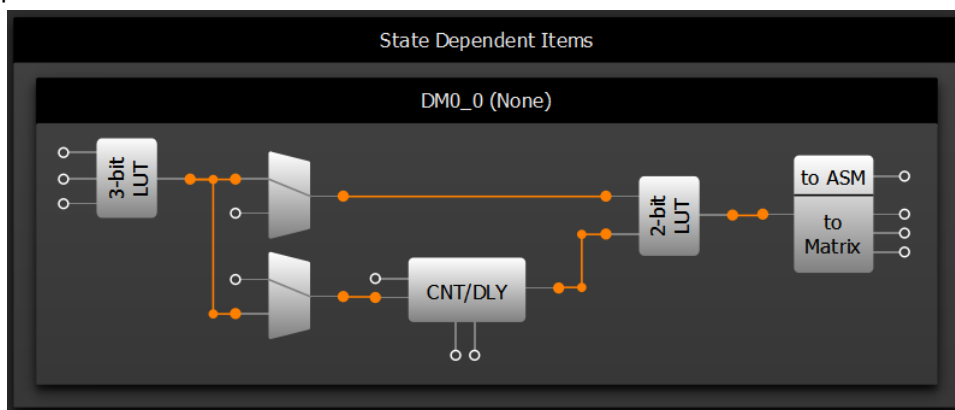- 1 CNT/DLY
- 1 Output block

Figure 5: An Unconfigured DM Block

Figure 6 shows the same DM block with colored-in connectors. (These colors do not appear inside GreenPAK Designer, they are merely for illustrative purposes.) The Green connectors are inputs to the DM block from the Matrix. The orange connections are dedicated connections within the DM block, which cannot be changed or moved. The blue connectors are clock connections for the counter block. The purple connector can be used to trigger a state transition, but is not a general matrix connection. The yellow connectors are matrix outputs from the DM block.
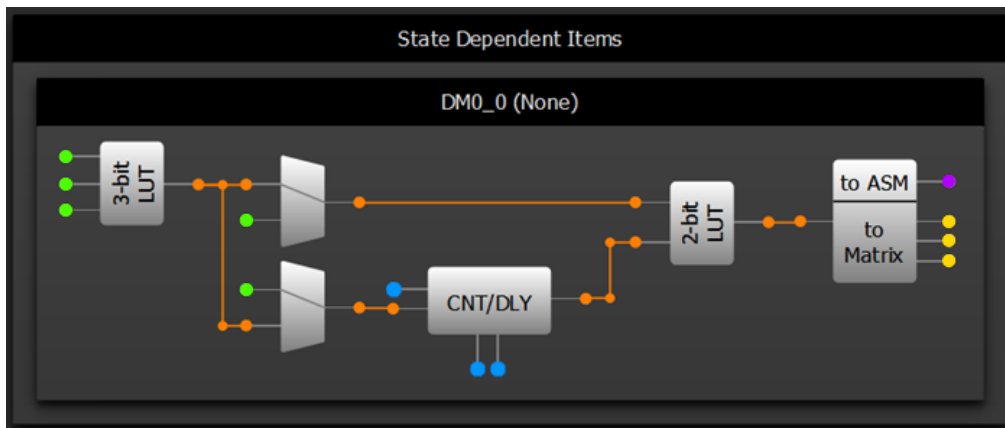
Figure 6: An Unconfigured DM Block with Colored Connections

Each DM block can have up to 6 different configurations. Any DM block configuration can be used in any of the ASM's 12 states, but only one configuration per DM block per state is permitted.

F(1) Computational Macrocell

AN-CM-237 SLG46880/1 F(1) Block describes how to use the F(1) computation macrocell in the SLG46880/1.

The computational macrocell (Figure 7) is a new component used to execute a predetermined sequence of actions during a state transition. This macrocell provides up to 4 different F(1) computations, each capable of executing up to 12 commands and holding 16 bits in its register. The list of available commands is shown in Table 1. Commands that specify a LOAD or OUT correspond to the configurable outputs of the F(1) block, shown in Figure 7.
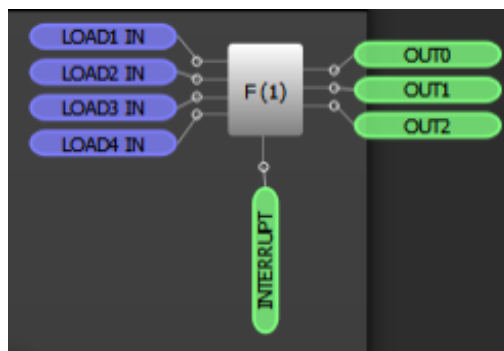

Figure 7: F(1) Block

The F(1) block has 4 analog inputs coming from various pins that can be muxed into the positive input for the F(1) block's dedicated Analog Comparator.

Table 1: F(1) Commands

Command	Command Name	Command Description
0000	LOAD1	Loads a one bit value to the top of the memory stack (location 0). During the execution of this command, all other values are shifted down 1 location, and the value in location 15 is lost. The data loaded by Load 1 command is defined in the Load 1 register, which defines where the value is to be loaded from such as a pin, the f(1) comparator, etc.

Command	Command Name	Command Description
0001	LOAD2	Loads a one bit value to the top of the memory stack (location 0). During the execution of this command, all other values are shifted down 1 location, and the value in location 15 is lost. The data loaded by Load 2 command is defined in the Load 2 register, which defines where the value is to be loaded from such as a pin, the f(1) comparator, etc.
0010	LOAD3	Loads a one bit value to the top of the memory stack (location 0). During the execution of this command, all other values are shifted down 1 location, and the value in location 15 is lost. The data loaded by Load 3 command is defined in the Load 3 register, which defines where the value is to be loaded from such as a pin, the f(1) comparator, etc.
0011	LOAD4	Loads a one bit value to the top of the memory stack (location 0). During the execution of this command, all other values are shifted down 1 location, and the value in location 15 is lost. The data loaded by Load 4 command is defined in the Load 4 register, which defines where the value is to be loaded from such as a pin, the f(1) comparator, etc
0100	AND	Performs a logical AND to the top two locations in the memory stack (location 0 and location 1). During execution of this command, the two values in the top two stack locations are deleted, and the logical AND result is pushed on the top of stack (location 0). In the process, all other values in the stack are shifted up 1 location, and a 0 is loaded in location 15.
0101	OR	Performs a logical OR to the top two locations in the memory stack (location 0 and location 1). During execution of this command, the two values in the top two stack locations are deleted, and the logical OR result is pushed on the top of stack (location 0). In the process, all other values in the stack are shifted up 1 location, and a 0 is loaded in location 15.
0110	XOR	Performs a logical XOR to the top two locations in the memory stack (location 0 and location 1). During execution of this command, the two values in the top two stack locations are deleted, and the logical XOR result is pushed on the top of stack (location 0). In the process, all other values in the stack are shifted up 1 location, and a 0 is loaded in location 15.
0111	INV	Performs a logical Invert (INV) to the top location in the memory stack (location 0). During execution of this command, the value in the top stack location is deleted, and the logical INV result is pushed on the top of stack. There is no effect on all other values in the stack.
1000	PUSH0	Pushes a 0 into the top location in the memory stack (location 0). During the execution of this command, all other values are shifted down 1 location, and the value in the location 15 is lost.
1001	POP	During execution of this command, values in the stack are shifted up 1 location, and a 0 is loaded in location 15.
1010	DELAY	The execution of commands by the f(1) Computation microcell is delayed by a period of time defined in the configuration of the delay function. Once this time period is completed, the next f(1) instruction will execute.
1011	LOOP with DELAY	If the top location in the memory stack (location 0) is equal to 0, the command execution in the f(1) Macrocell is delayed by a period of time defined in the configuration of the delay function and then executes the f(1) sequence at the specified jump location. If the top location in the memory stack (location 0) is equal to 1, the f(1) Macrocell proceeds with execution of the next command in sequence.
1100	OUT1	Outputs the top location in the memory stack (location 0) on matrix input OUT1
1101	OUT2	Outputs the top location in the memory stack (location 0) on matrix input OUT2

Command	Command Name	Command Description
1110	OUT3	Outputs the top location in the memory stack (location 0) on matrix input OUT3
1111	END	Immediately ends execution of commands by f(1) Macrocell, and control is returned to ASM Macrocell

The F(1) block's stack can serve as a data source or data destination for the commands running in the F(1) block. LOADx commands will push data down into the stack. OUTx commands will retrieve data from the stack, and send to the contents to one of three outputs to the Connection Matrix. The contents of this memory are not changed during state transitions, and are only changed by the commands running inside the F(1) Computation Macrocell itself.

The INTERRUPT signal is used to halt all macrocell actions and return control to the ASM. The INTERRUPT may additionally be used to reset the macrocell's memory.

To create a new F(1) configuration, display the F(1) block on your workspace by clicking on its checkbox in the Components List. Then double-click the F(1) block to bring up its Properties panel.

Create a new F(1) configuration by clicking on the green "+" button at the top right of the Properties panel. You can rename configurations to whatever you want, but for this app note I'll leave the default configuration names in place.

For each of your F(1) configurations, you can choose whether or not you want an interrupt to reset the stack's memory. You can also select the initial state for each of the F(1) block's 3 outputs. The options for the OUTx initial states are: keep, 0, 1, and none.

ASM Output Blocks

The SLG46880 and SLG46881 ASM OUTPUT0 and ASM OUTPUT1 blocks can be used to create state-specific signals which are sent to the rest of the GreenPAK IC design. ASM OUTPUT0 (Figure 8) contains 4 output signals, and ASM OUTPUT1 contains 8. These signals are ideal for changing logic outside of the DM blocks, or for regulating power to larger devices, such as comparators.

The ASM OUTPUT0 block, shown in Figure 8, has 4 outputs which can be used to connect state-dependent outputs to the rest of the matrix. However, the ASM OUTPUT1 is only able to directly send the state-dependent information to the GPO's of the IC.
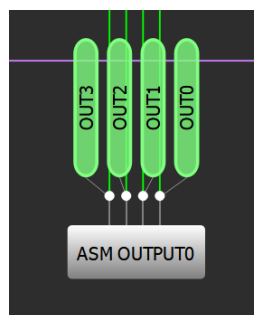

Figure 8: ASM OUTPUT0

6 Changes to the Graphical User Interface

Due to the increase in state-dependent items, the graphical user interface (GUI) has been changed for ease of use. Several new GUI features are outlined in Figure 9 below:

- State Window Selection
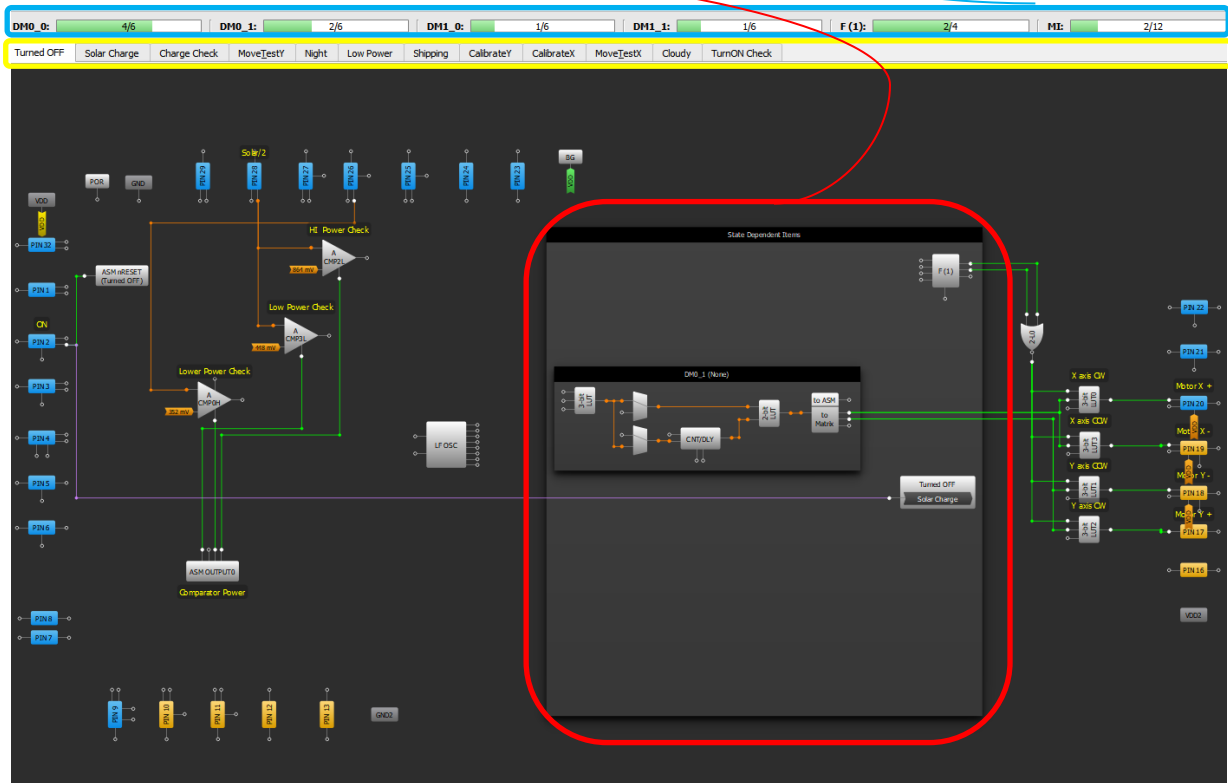- ASM Resource Manager
- State Dependent Items

Figure 9: System Diagram

State Window Selection

The highlighted state button, such as "Turned Off" in Figure 9, indicates the current state shown in the GreenPAK Designer workspace. Selecting another state from the tab will switch the GreenPAK Designer workspace to the selected state.

ASM Resource Manager

The ASM Resource Manager shows the current number of available configurations for the state-dependent items, such as the DM blocks and the F(1) macrocell. The available number of components for the state-dependent blocks are:

- DM blocks: Each of the 4 DM blocks contains 6 configurations
- F(1) Macrocell: 4 configurations
- Matrix Interface cells: 4 MIx configuration registers, each with 3 unique configurations, allowing 12 possible combinations

Though the total resource configurations are limited, any number of states may utilize the same resource, meaning that an F(1) macrocell may use one function within several states, or one DM delay block within two or more similar states.

State Dependent Items

State Dependent Items consist of all items that may change because of a state-change, such as the DM and F(1) blocks. Any ASM block shown within the ASM Resource Manager will appear and remain within the state-dependent items window.

State Dependent Items may be configured by selecting the visible block, then navigating to the Properties window of the block. The "Configuration" option, circled red in Figure 10, shows the current configuration of the F(1) block within the selected state. To create a new configuration simply press the green plus button, and to select a different configuration (or to choose no configuration) select the state from the dropdown menu. Selecting the red minus button will remove the configuration entirely, which may affect other states, but will also free the ASM resource.
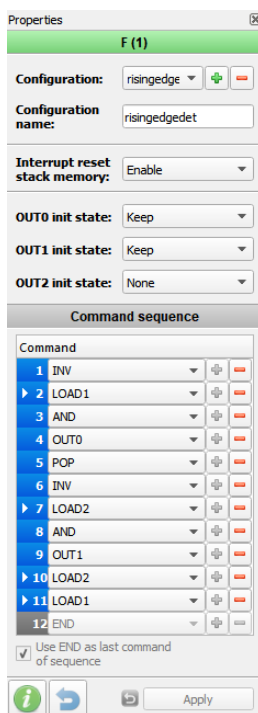

Figure 10: F(1) Properties

7   Conclusion

The SLG46880 and SLG46881 have widely configurable asynchronous state machines, allowing users to develop more complex designs while using fewer components. Many of the changes from previous GreenPAK ICs that make the new SLG46880 and SLG46881 so appealing can be attributed to the new implementation of the asynchronous state machine, which now accommodate further states and greater configuration per state than previous GreenPAK devices. Dynamic memory, a F(1) computational macrocell, and other new features add key elements to a user's toolkit to help save power and complexity.

Revision History

Revision	Date	Description
1.0	19-Mar-2018	Initial Version

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)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

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

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/.