Daisy-Chain SPI Using Quad SLG47011
SLG47011V

This application note describes the design for daisy-chaining SPI signals across four SLG47011 ICs.

Contents

Terms and Definitions

ADC	Analog to Digital Converter
AFE	Analog Front-End
CMD	Command
CNT/DLY	Counter and Delay
LUT	Look Up Table
PGA	Programmable Gain Amplifier
SHR	Shifter Register
SPI	Serial Peripheral interface

References

For related documents and software, please visit:AnalogPAK | Renesas

Download our free Go Configure Software Hub [1] to open the .aap file [2] and view the proposed circuit design. Use the AnalogPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Renesas Electronics provides a complete library of application notes [4] featuring design examples, as well as explanations of features and blocks within the Renesas IC.

[1] GreenPAK Go Configure Software Hub, Software Download and User Guide, Renesas Electronics

[2] AN-CM-414 Daisy-Chain SPI Using Quad SLG47011, AnalogPAK Design File, Renesas Electronics

[3] GreenPAK Development Tools, AnalogPAK Development Tools Webpage, Renesas Electronics

[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage, Renesas Electronics

Author: Paul Yeh, Sr. Application Engineer, Renesas Electronics

1. Introduction

This application note presents a daisy-chained SPI design for four SLG47011 chips. SPI is a high-speed protocol (>1 MHz) that transfers data between an MCU and multiple peripherals using Chip-Select (CS), Serial Clock (SCLK), Master-Out-Slave-In (MOSI), and optionally, Master-In-Slave-Out (MISO) signals.

There are two ways to connect multiple devices:

1. In Figure 1, an MCU provides multiple Chip-Select (CS) signals to corresponding peripheral ADC devices. Each Chip-Select signal is independently connected to a peripheral device. This method of SPI connection requires complicated PCB layout/subnodes and additional GPIO utilization. In addition, the MCU cannot capture ADC data from the ADC devices synchronously.
2. Daisy-Chain SPI, as shown in Figure 2, is developed to simplify subnodes using only one Chip-Select (CS) signal. The MCU provides a single Chip-Select (CS) signal that is sent to all peripheral ADC devices. Meanwhile, MOSI and MISO signals are integrated using a daisy-chain configuration. While the CS signal activates all peripheral ADC devices, each ADC device can transfer its own data in sequence to the MCU. Therefore, the MCU can capture multiple instances of ADC data simultaneously.

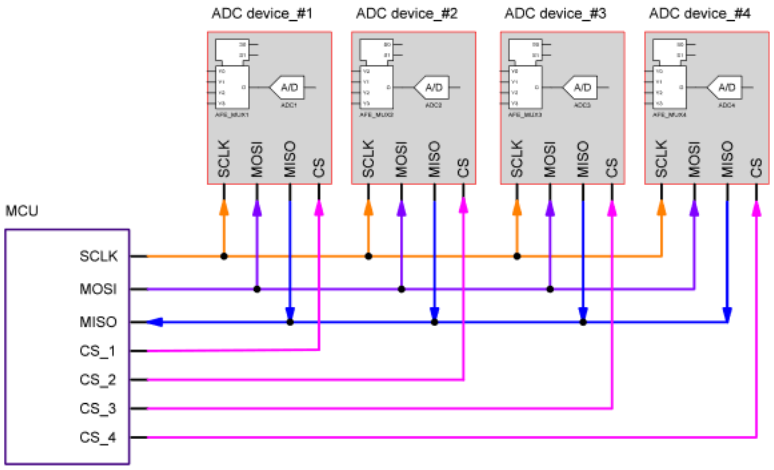

Figure 1: General SPI Connection Diagram

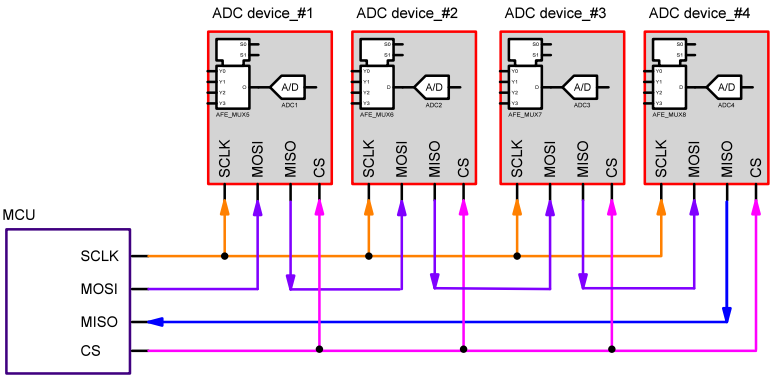

Figure 2: Daisy-Chain SPI Connection Diagram

The SLG47011 is comprised of a diverse collection of digital and analog marcocells for AFE applications. One 14-bit SAR ADC and one 4-channel PGA (shown in the Block Diagram of the SLG47011 in Figure 3 are used to perform Analog Front-End signal capturing. The PGA is capable of operating in different configurations including

single-ended mode and differential mode. Additionally, the reference voltages used by the PGA and ADC can be configured independently as needed. The ADC supports resolutions ranging from 8 bits to 14 bits. Moreover, the SLG47011 offers four Buffer blocks in which averaging and over-sampling functions are used to process ADC data.

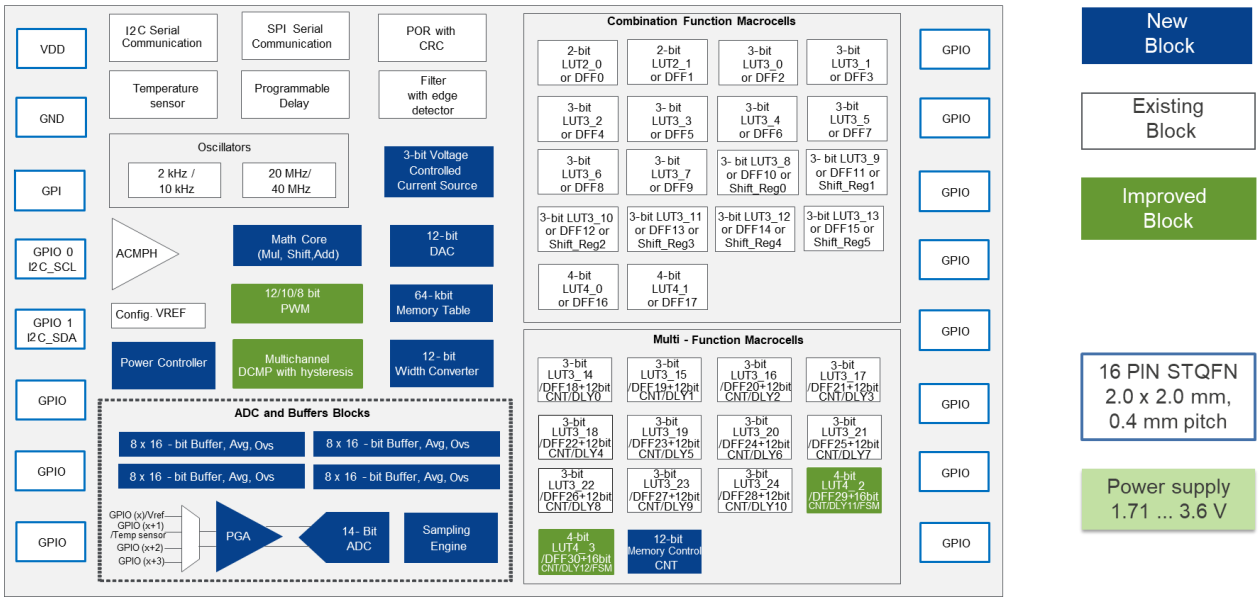

Figure 3: SLG47011 Block Diagram

2. Design Principle

Figure 4 shows the system diagram of a Daisy chain SPI connection. Each SLG47011 provides 3 external AFE inputs (Pin7, Pin8, and Pin9). The SPI interface I/O inherited from the SLG47011 provides the CS(PIN3), SCLK (PIN5), MOSI (PIN4) and MISO (PIN6) connections. To create Daisy-chain SPI based on inherited SPI, we use the internal digital blocks of the SLG47011 and utilize GPIO Pins to synthesize the required function. Pin16, Pin15, and Pin10 are assigned to connect the inherited SPI interface I/O directly. Compared to an Inherited SPI interface I/O, the Daisy Chain SPI interface I/O represented by CS (Pin13), SCLK (PIN5), MOSI (PIN12), and MISO (PIN11). Therefore, MCU would read the data via Daisy Chain SPI interface I/O. Figure 5 shows the Pin connections of a single chip as a reference. Since the inherited SPI macrocell in the SLG47011V cannot support an internal connection using Matirix I/O, GPIO Pins are used to carry out the design. Since the propagation delay of the GPIO Pins is greater than that of the Matrix I/O, it is estimated that the SPI baud rate frequency (SCLK) should be limited to be less than 3.4 MHz.

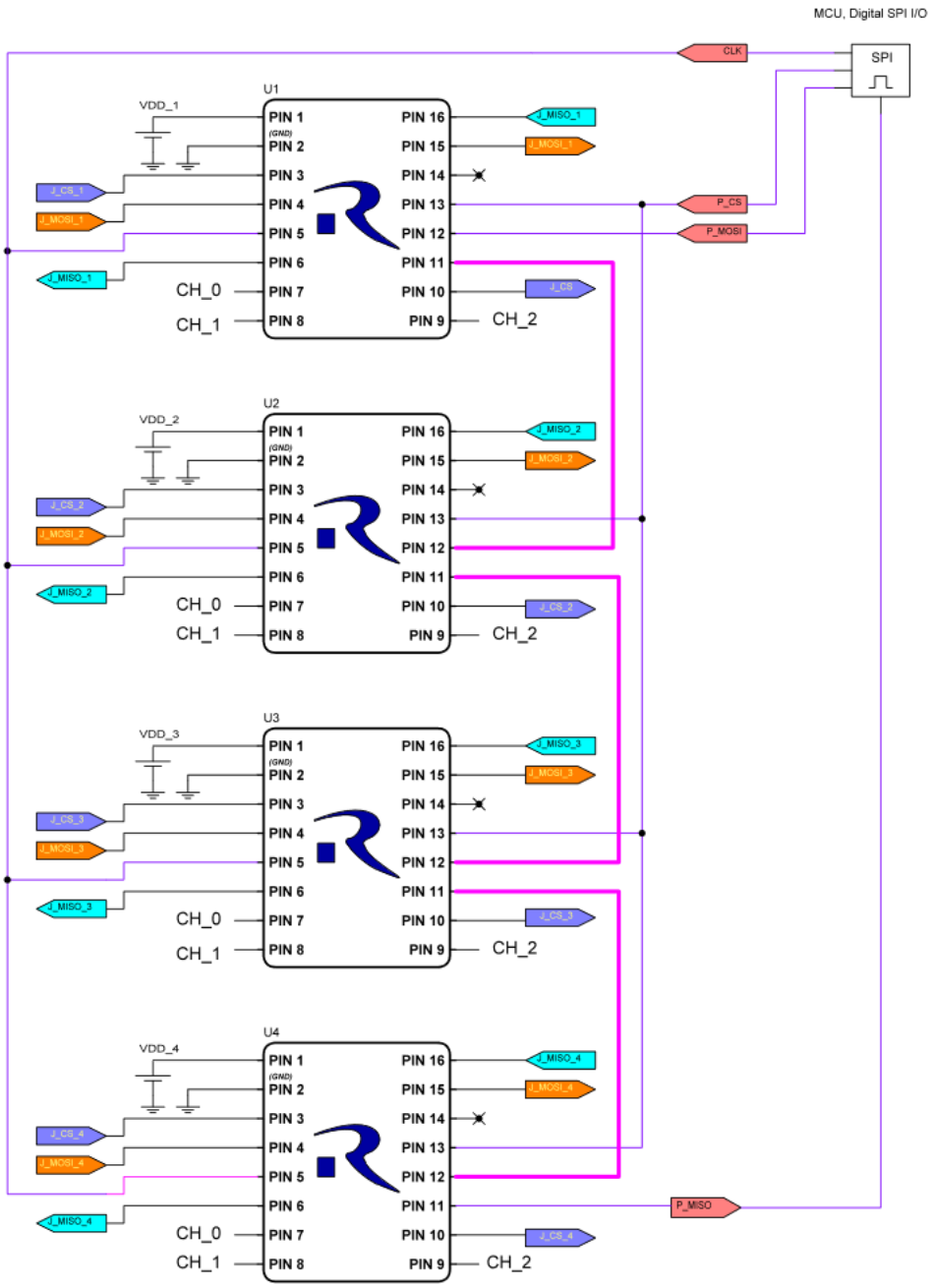

Figure 4: System Diagram of using Quad-Chips in a Daisy-Chain SPI Connection

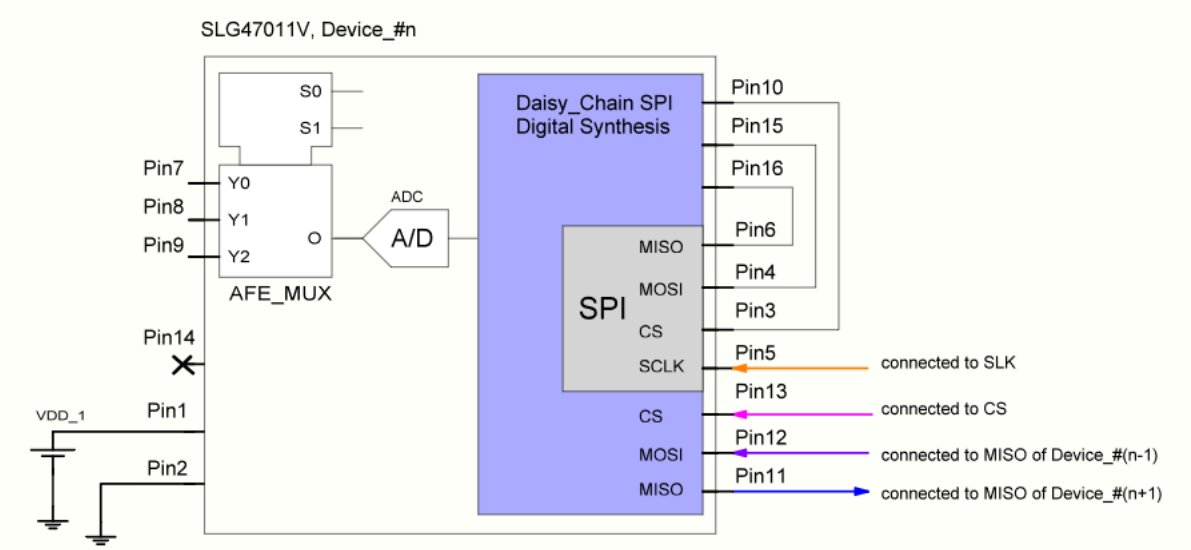

Figure 5: Pin Assignment for a Single-Chip Implemented in a Daisy-Chain SPI Design

A diagram of the SPI signal sequence of Daisy chain SPI is shown in Figure 6. The master MCU device sends one SPI command to collect required data from all peripheral devices (chip_1~4). Once an individual peripheral device receives a command, it will transfer internal data via the inherited SPI macrocell. To manipulate the internal data in sequence, the digital synthesis subcircuit, in Figure 5, needs to initiate the serial-in/serial-out process. Since the CMD signal or Data signals will be propagated through a string of peripheral devices, Pin 10 and Pin 15 are regarded as the main subnodes and must succeed to transfer data and receive the CMD signal, separately. SCLK can be used to allocate the Data sequence transfer and interconnection period.

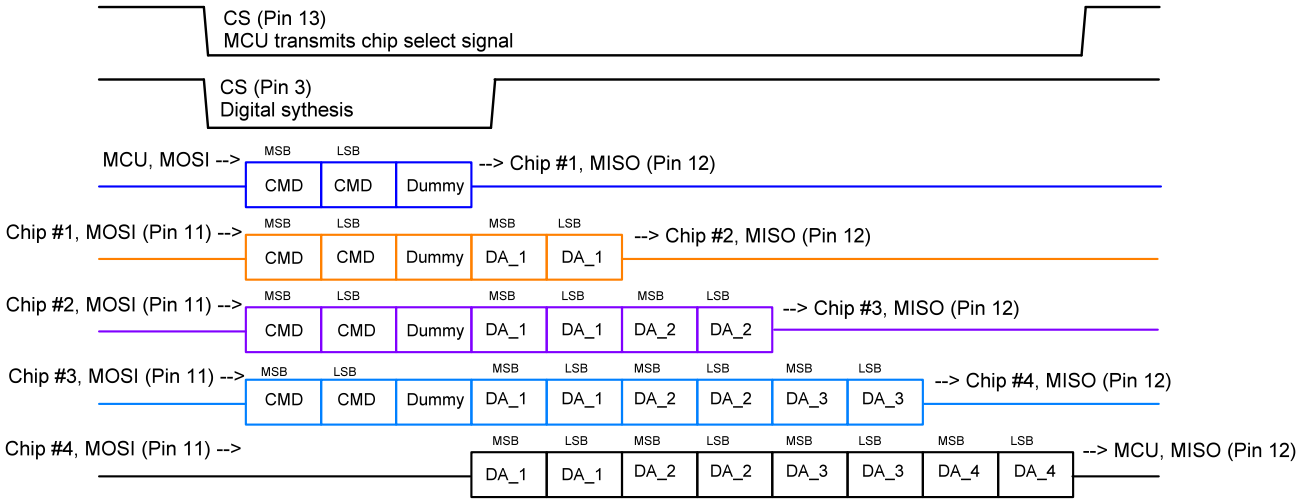

Figure 6: SPI Signal Sequence Diagram of Daisy Chain SPI

In Figure 7, the block diagram illustrates the basic digital synthesis design. SW_A and SW_B are the switches used for sending CMD and Data. The CNT/DLYs + LUTs block provides the CS signal to the inherited SPI block. The Inherited SPI block transfers the corresponding data associated with the unique CMD in all chips. The SHR shifts the data from the inherited SPI block using its specific register lengths setting. Therefore, data from each individual chip is arranged in order. We utilize the macrocells of the SLG47011V to fulfill this required functionality.

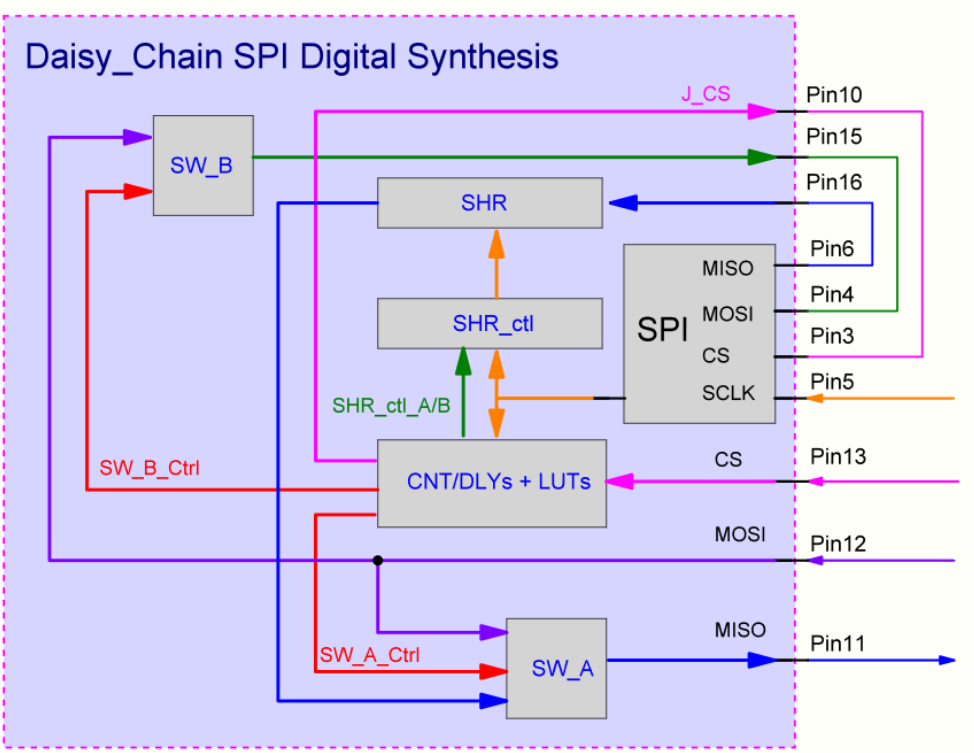

Figure 7: Block Diagram of the Daisy Chain SPI Digital Synthesis

3. GreenPAK Design Internal Block Configuration

All GreenPAK Designs shown in the Go Configure Software are presented in Figure 8 - Figure 11. The configuration descriptions follow.

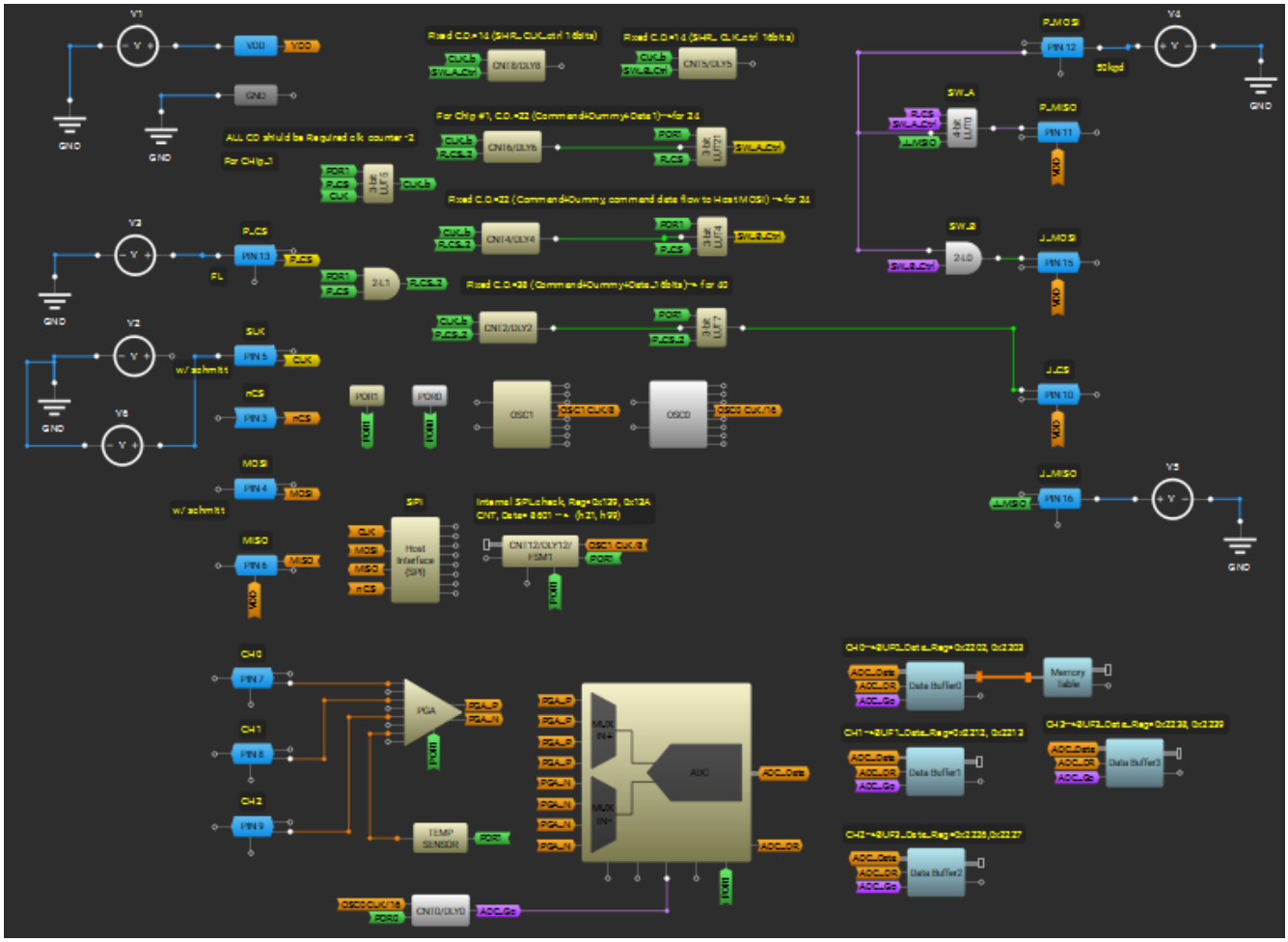

Figure 8: GreenPAK Designer Schematic (Daisychain_SPI_CHIP_1.aap)

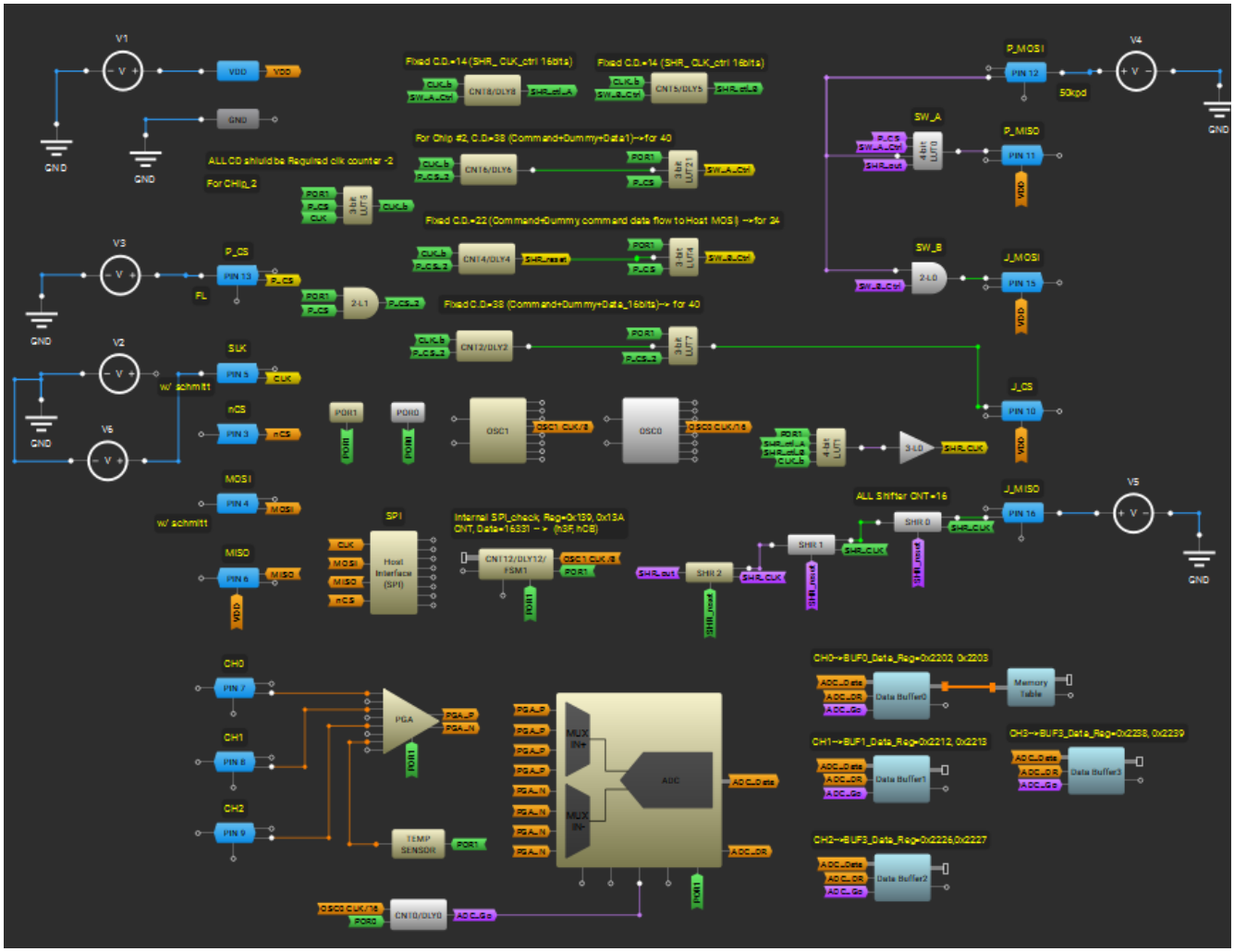

Figure 9: GreenPAK Designer Schematic (Daisychain_SPI_CHIP_2.aap)

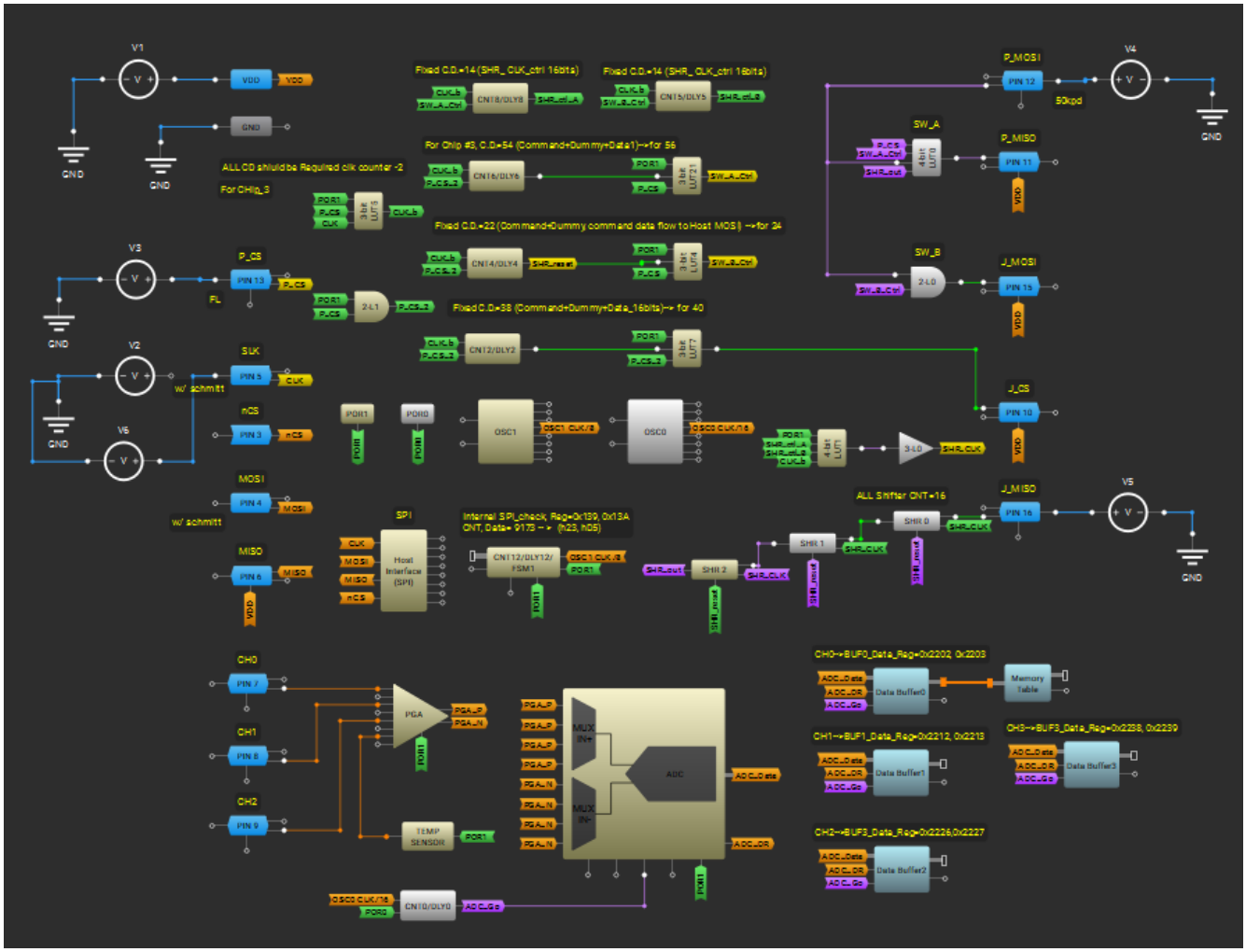

Figure 10: GreenPAK Designer Schematic (Daisychain_SPI_CHIP_3.aap)

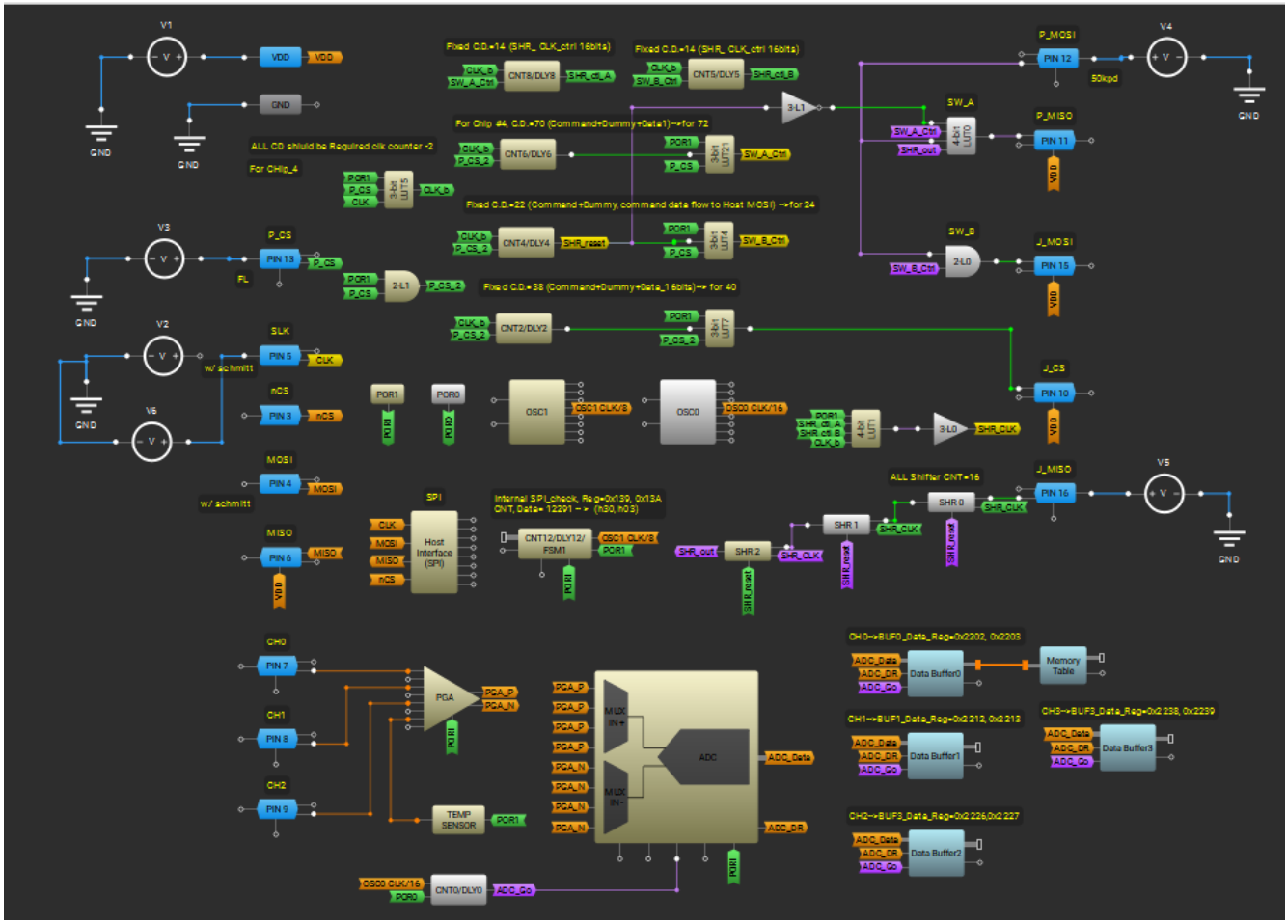

Figure 11: GreenPAK Designer Schematic (Daisychain_SPI_CHIP_4.aap)

As shown in Figure 7, SW_A and SW_B functions are accomplished by using LUT macrocells. The logic configuration of these macrocells is shown in Figure 12.

Figure 12: LUT Configuration for SW_A and SW_B

Also shown in Figure 7, the SHR (Shifter register) processes the data from the inherited SPI block. In order to ensure the data sequence is integrated into the final MOSI bus of chip_4, we have to use three SHR macrocells to output data in sequence. In our design, chip_1 is not equipped with a SHR block due to data first-in-first-out principles, as shown in Figure 8. The total register length settings of the SHR blocks are set to 18. This configuration is shown in Figure 13.

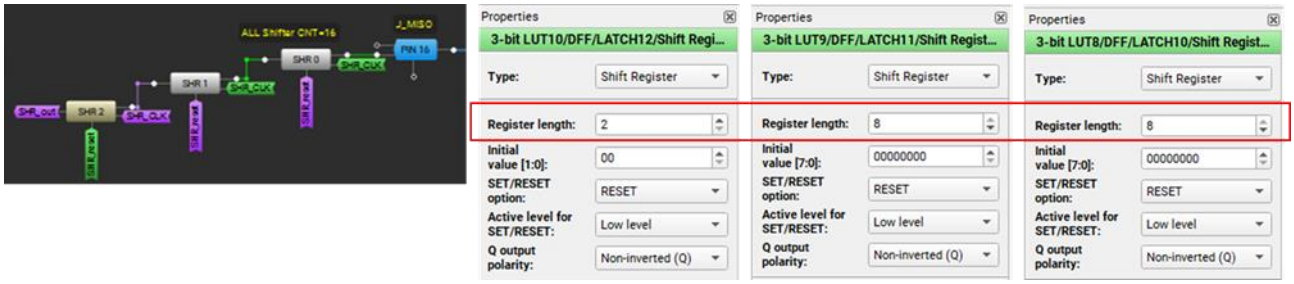

Figure 13: Configuration for the SHR Block

Referring back to Figure 7 and the schematics in the GP file, this design uses multiple combination blocks for **five** control signals: ***SW_A_Ctrl***, ***SW_B_Ctrl***, ***J_CS***, ***SHR_ctl_A,*** and ***SHR_ctl_B***

1. The first combination block (***SW_A_Ctrl***) consists of CNT6 and LUT21. Output is assigned to control SW_A.

   For Chip 1: CNT6 Counter data = **22**. For Chip 2: CNT6 Counter data = **38**.

   For Chip 3: CNT6 Counter data = **54**. For Chip 4: CNT6 Counter data = **70**.

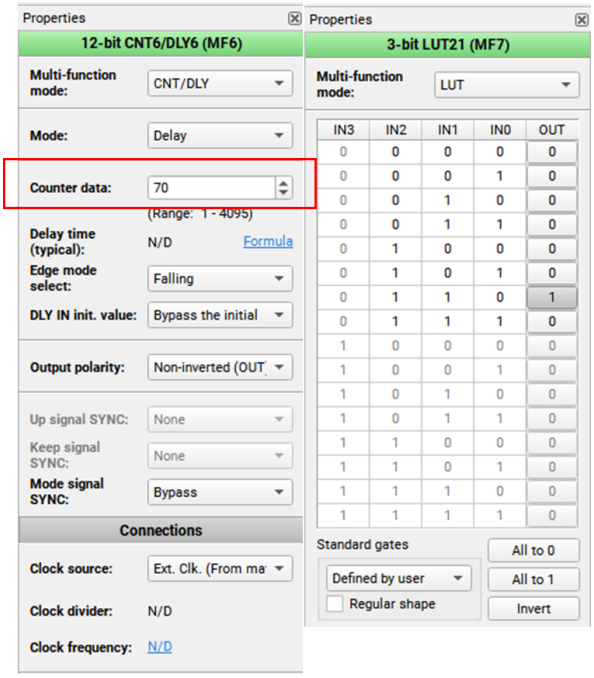

Figure 14: Configuration for the first Combination Block

2. The second combination block (***SW_B_Ctrl***) consists of CNT4 and LUT4. Output is assigned to control SW_B. All macrocells are given the same configuration. For chip_1~4: CNT4 Counter data = **22**.

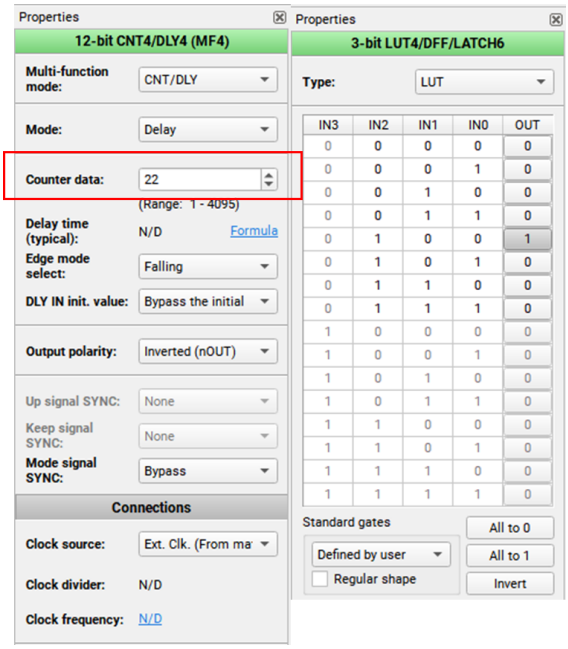

Figure 15: Configuration for second Combination Block

3. The third combination block (***J_CS***) consists of CNT3, CNT4, LUT1, and LUT17. Output is assigned to generate a variant CS signal forward to Pin10. All macrocells are given the same configuration. For chip_ 1~4: CNT2 Counter data = **38**.

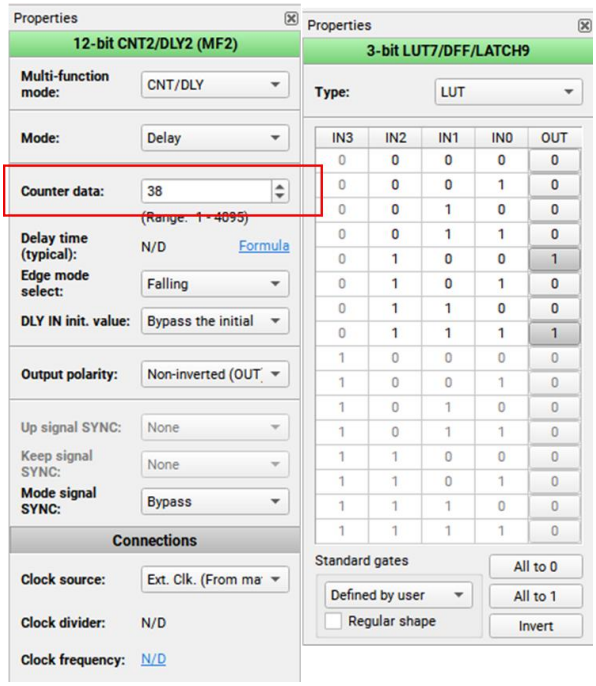

Figure 16: Configuration for the third Combination Block

4. The fourth and fifth combination blocks (***SHR_ctl_A*** and ***SHR_ctl_B***) are used to arrange the data transfer sequences. For chip_1~4: CNT5 and CNT8 Counter data = **14**.

Figure 17: Configuration for the fourth and fifth Combination Blocks

As shown in Figure 6, the MISO bus of Chip 4 transmits data forward to the MCU. In Figure 18 it is shown that SHR_reset is used to blank the CMD transfer in SW_A of Chip 4.

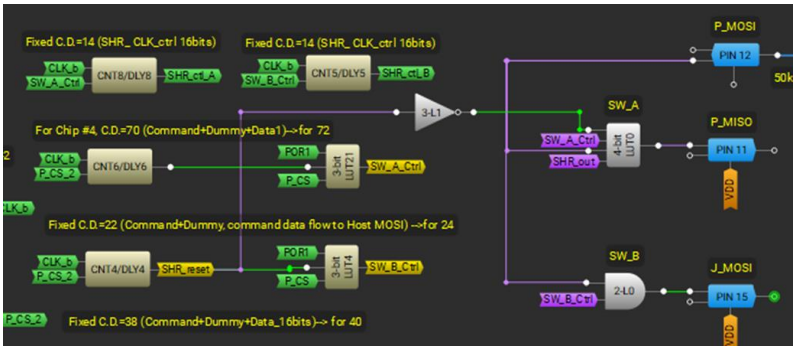

Figure 18: CMD Blank Block

For inspecting the sequence of data transfer in Daisy Chain SPI, we assign the FSM1 macrocell to store a specific counter data. The design and configuration is shown in Figure 19.

For Chip 1: Counter data = 8,601 (h21, h99) For Chip 2: Counter data = 16,331 (h3F, hCB)

For Chip 3: Counter data = 9,173 (h23, hD5) For Chip 4: Counter data = 12,291 (h30, h03)

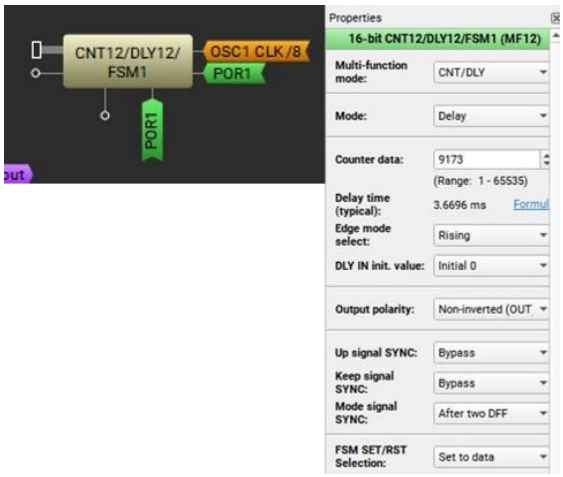

Figure 19: Configuration for Specific Data Inspection

The prototypical Daisy chain SPI is eventually completed. Next, we add the ADC block for multiple AFE applications. As shown in Figure 20, we create four AFE channels associated with the ADC block. Three external AFE channels are assigned to Pin 7, Pin 8, and Pin 9, respectively. The PGA macrocell is configured in single-ended mode for the AFE channels. Additionally, the internal AFE channel is used to monitor the temperature sensor.

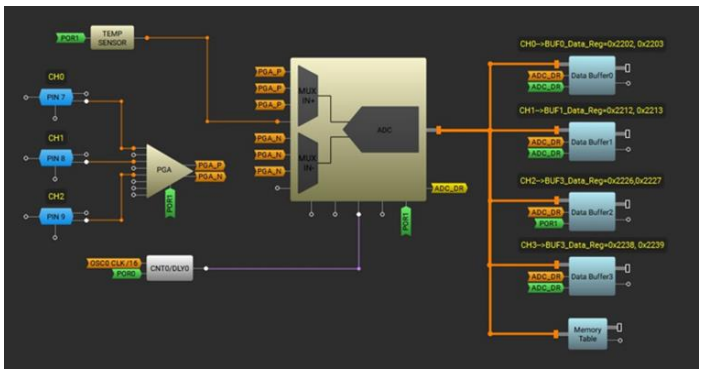

Figure 20: Three External AFE Channels associated with ADC

Figure 21 shows the configuration for the reference design. The ADC macrocell is configured with 14-bit resolution and channel sampling period of 400 µs. ADC data is configured in Buffer Mode to transfer data to the Buffer macrocells, and the Buffer data can be read via SPI protocol directly.

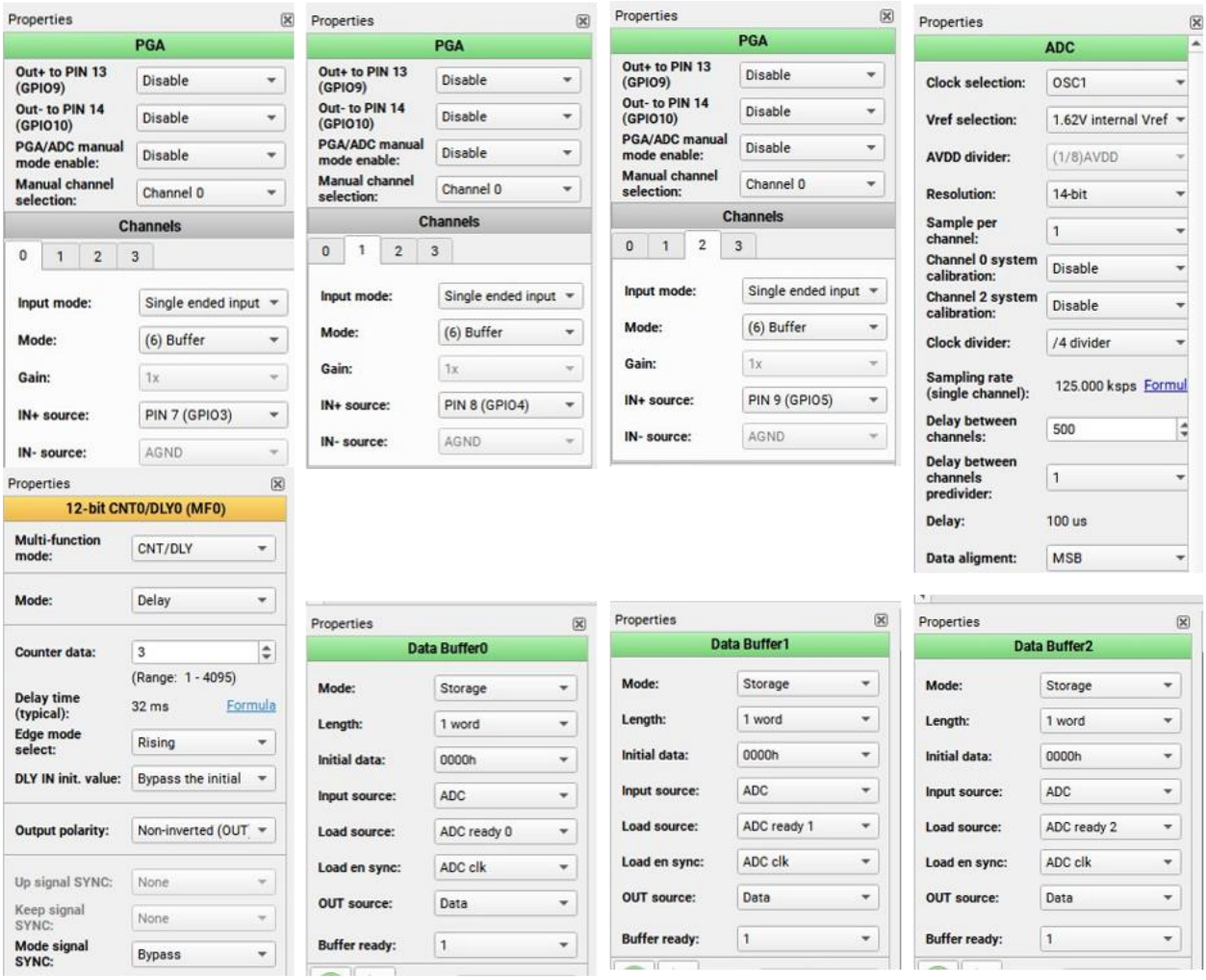

Figure 21: Configuration of Multiple AFE Channel Block

4. Simulation Results

The simulation of Chip 2 is shown in Figure 22.

The CS signal transmitted by the MCU can be reshaped to trigger the inherited SPI to prepare Chip 2 data.

CMD is received and forwarded to MISO.

The SHR block works to propagate the Chip data in turn.

The MISO signal from Chip 2 outputs CMD and the Data of Chip 1 and the Data of Chip 2.

This single chip simulation also applies to the behavior of the other Chips as well.

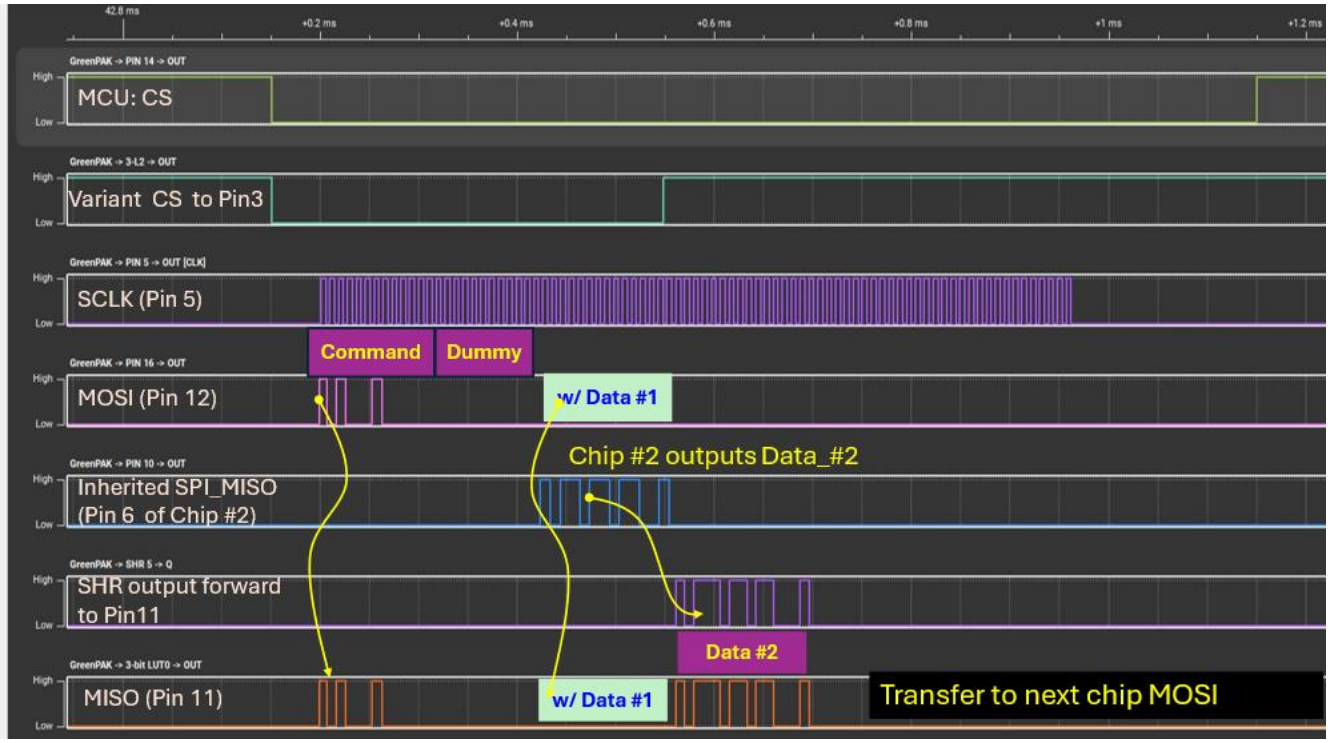

Figure 22: Simulation Results

5. Design Verification Using Hardware Prototype

After programming the NVM on each chip, we can prepare four SLG47011V daughterboards and carry out the wire connection, as shown in Figure 23. A Digilent ® ADP3450 Portable oscilloscope is set up to provide the required SPI send/receive test-pattern and protocol analyzer.

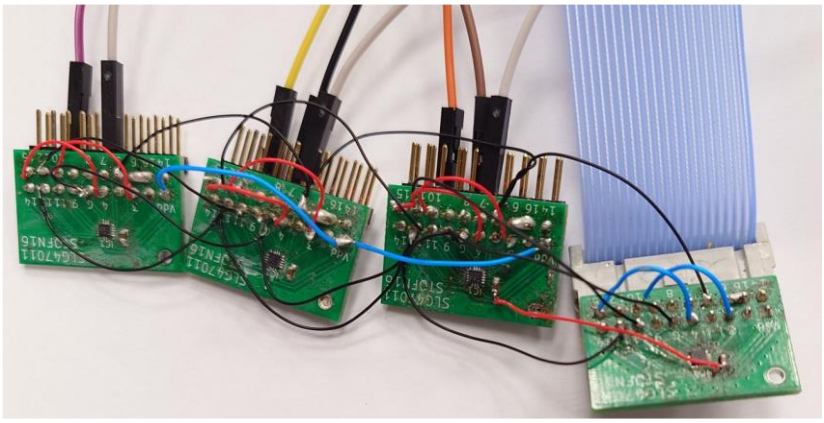

Figure 23: Hardware Prototype and Bench

Basic Testing conditions: V_{DD} = 3.3 V to the four chips; SPI SCLK frequency = 3.4 MHz (provided by the ADP3450)

5.1 Inspection: Reading Data from DLY/CNT, [h0139, h013A]

To inspect the SPI communication and data sequence, the quad SLG47011V prototype is equipped with specific data, independently. The specific data are regarded as check-code allocated to each individual chip's **FSM1** macrocell. From the validation results, the SPI MISO data sequence indicates the data permutations in arrangement. The protocol **analyzer** collects all specific data by sending an SPI command [hA2, h26]. Every chip reflects its individual data including **[h01, h39]** (FSM1_Data_MSB) and **[h01, h3A]** (FSM1_Data_LSB).

Chip 1: specific counter data in FSM1 macrocell is given **h2199**.

Chip 2: specific counter data in FSM1 macrocell is given **h3FCB**.

Chip 3: specific counter data in FSM1 macrocell is given **h23D5**.

Chip 4: specific counter data in FSM1 macrocell is given **h3003**.

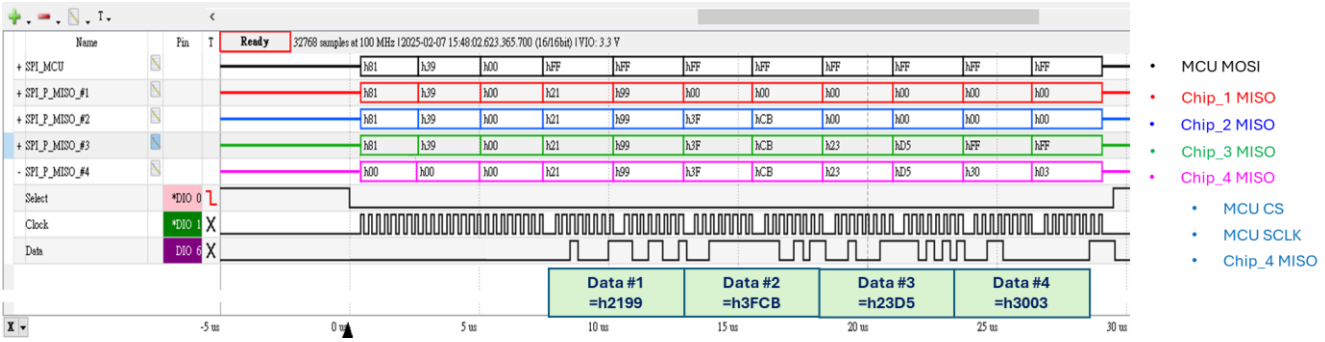

Figure 24: Sequence Inspection for Daisy Chain SPI

5.2 Capturing ADC Data: Reading Channel_0 from Buffer0, [h2202, h2203]

This validation shows that the **protocol analyzer** collects every chip's channel_0 data by sending an SPI command **[h22, h02]**. Every chip reflects its individual data including **[h22, h02]** (BUFFER0_Data_MSB) and **[h22, h03]** (BUFFER0_Data_LSB). BUFFER0_Data collects the data from ADC channel_0.

Chip 1: Channel 0 is given **0.6 V**, Buffer0 return ADC data= **h17F1** for **0.6061 V**

Chip 2: Channel 0 is given **0.2 V**, Buffer0 return ADC data= **h07C6** for **0.1968 V**

Chip 3: Channel 0 is given **1.0 V**, Buffer0 return ADC data= **h27F8** for **1.0118 V**

Chip 4: Channel 0 is given **0.0 V**, Buffer0 return ADC data= **h0002** for **0.0002 V**

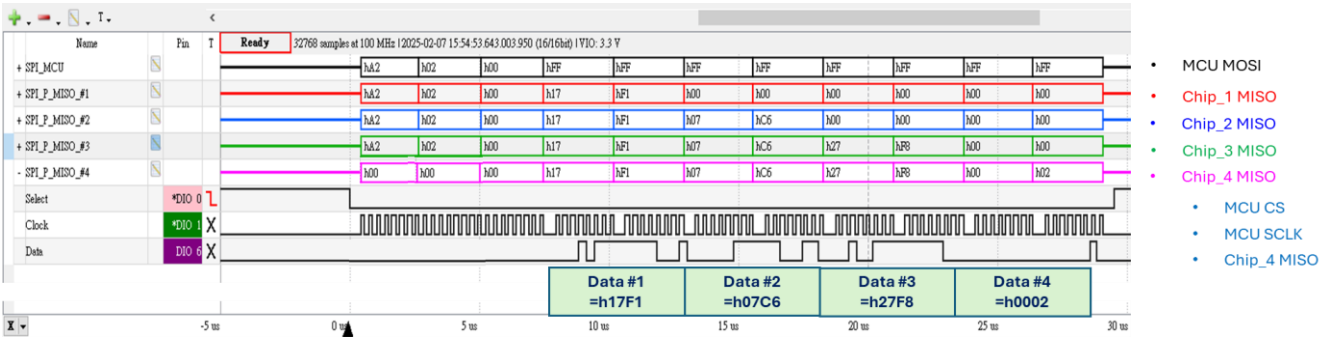

Figure 25: Reading data from ADC channel 0

6. Conclusion

Quad SLG47011V chips with Daisy chain SPI can be utilized to simplify the PCB layout and/or reduce the number of required subnodes. A maximum baud rate frequency of 3.4 MHz for the SCLK is supported. Every SPI command can force all SLG47011V chips to transfer individual ADC data synchronously. The validation result shows the Daisy Chain SPI functionality collects the ADC data corresponding to each individual ADC channel. To ensure ADC accuracy and reduce noise coupling, we recommended that fully PCB layout is necessary.

7. Revision History

Revision	Date	Description
1.00	April 30, 2025	Initial release