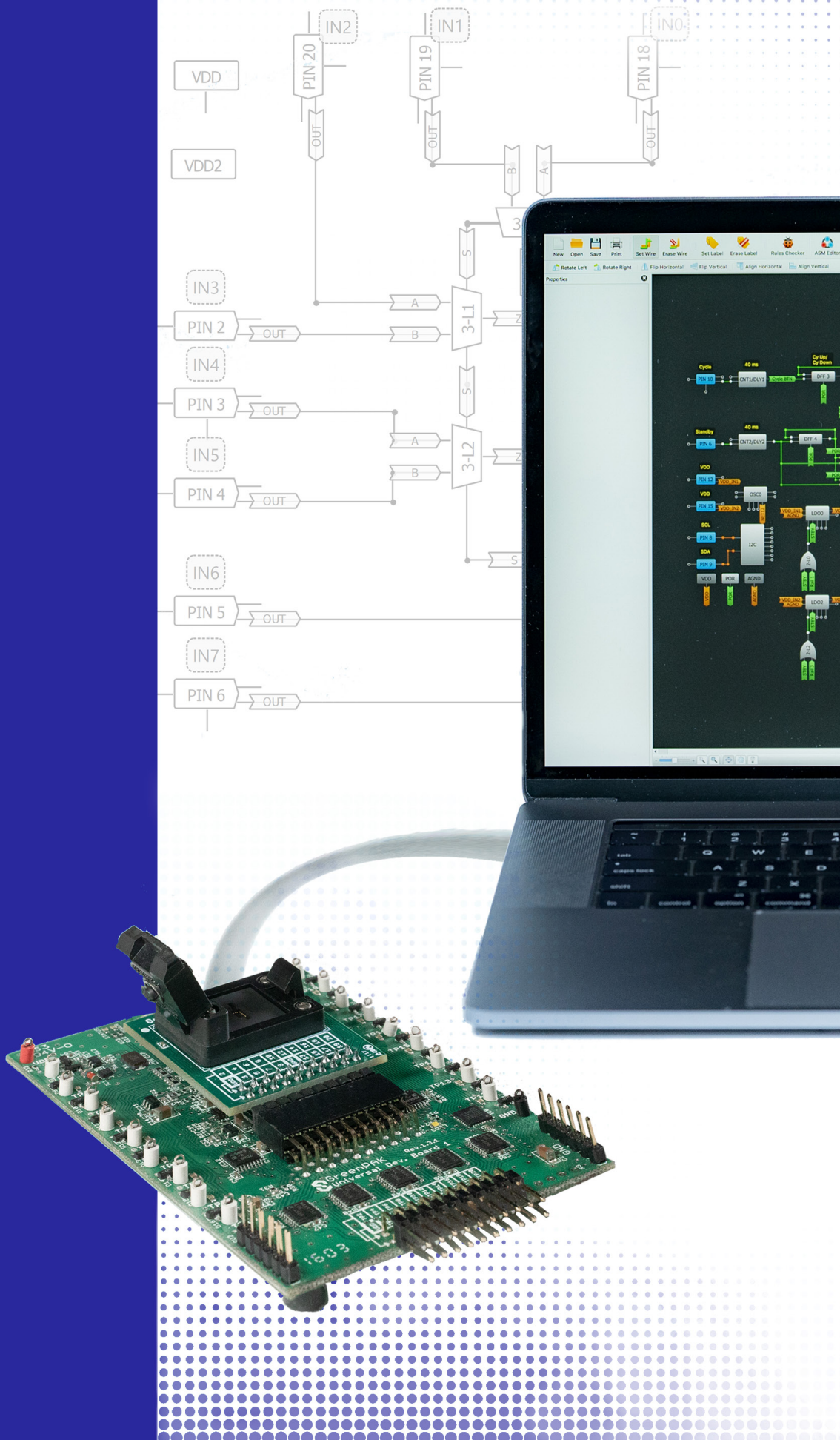
RENESAS

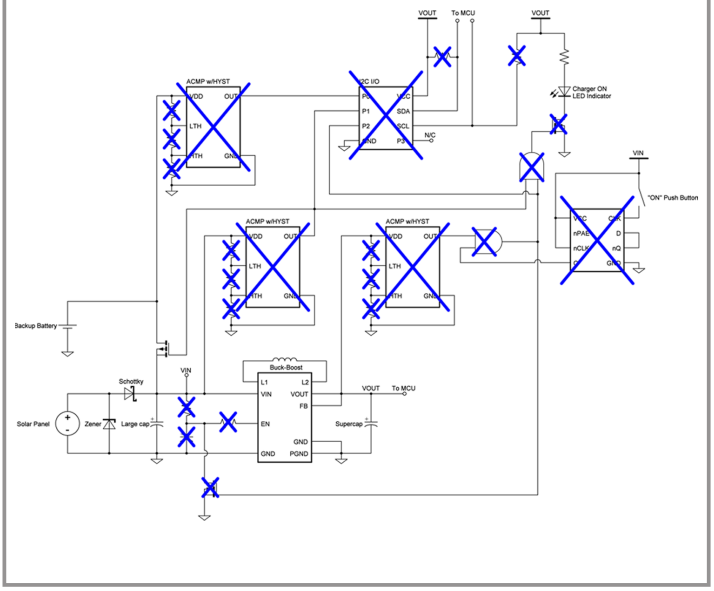
The GreenPAK™ Cookbook



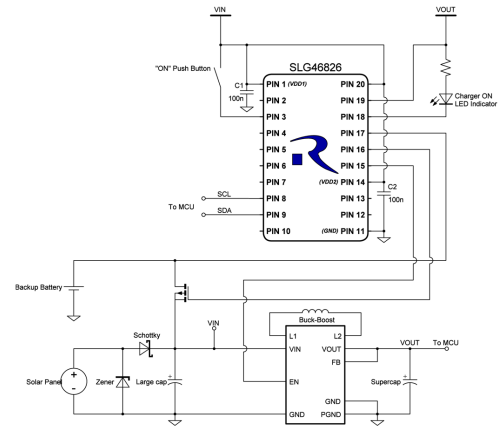
Introduction to GreenPAK

Renesas Electronics GreenPAK ICs are a family of Programmable Mixed-signal ICs that provide a small, costfriendly, and personalized solution to common problems that system-level circuit designers face. GreenPAK provides a means of considerably reducing PCB size, BOM cost, and design time.

Original Design



GreenPAK Benefits



Design reduced by

- 5 ICs
- 2 NMOS transistors
- 14 passive components

Example of Size Reduction w/ GreenPAK

Due to the features and configurability of GreenPAK narrowing the scope of possible applications can be difficult.
With the right motivation a designer can use a GreenPAK in almost any application within most industries. This document is designed to bolster this motivation and know-how: we provide a "cookbook" to designers to highlight where a GreenPAK can be used within their projects. We outline different techniques and provide completed applications to help designers use GreenPAK on their own.

Cookbook Structure

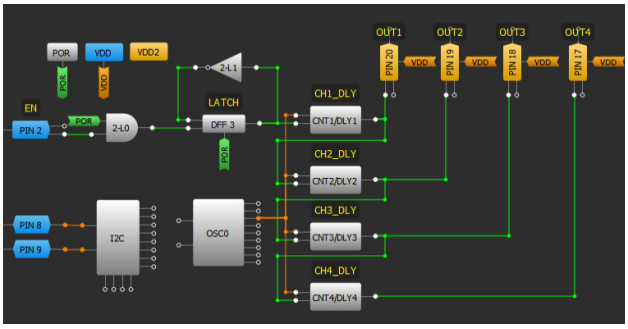
The majority of subsections within this document are organized into two categories: Techniques and Applications. Techniques focus on a task accomplished using only one or a few macrocells. Application sections describe how techniques can be meshed together to create real, valuable applications. Generally, the easiest techniques and applications will be at the beginning of a chapter.

Each application has an associated GreenPAK Designer file that can be viewed and edited.

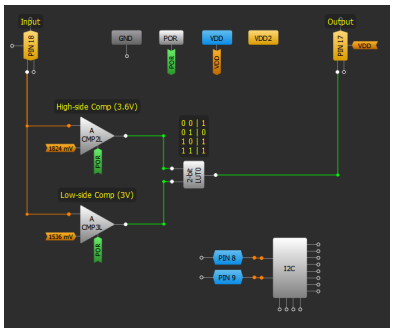
Making Your Own Design With the Cookbook

The applications outlined in the Cookbook are simple realizations of real-world applications. However, GreenPAK ICs have the macrocells and functionality to add far more value than the designs in this cookbook. Renesas Electronics has helped designers create thousands of unique designs, where simple applications both similar and different to the cookbook applications were expanded, combined, and personalized.

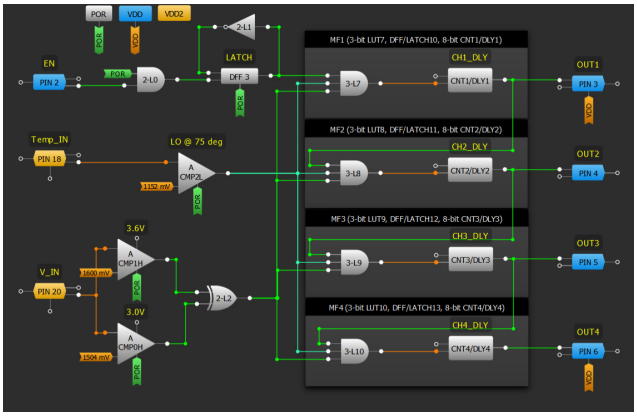
For example, the Application: Basic Sequencer can be combined with many of the applications within Chapter 4: Safety Features to create a self-regulating, customized sequencing application.



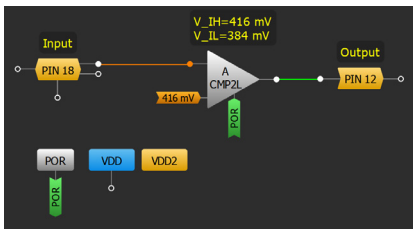
Power Sequencing



Window Comparator



Unique Solution



Overtemperature Detection

The resulting integrated solution is more complex yet still doesn't incorporate all available macrocells. With the full GreenPAK family of ICs at your disposal, the number of permutations and modifications available for the designs in this cookbook are endless. Whether you wish to completely reuse a design shown in this cookbook, or you'd rather incorporate some of the techniques in this paper into your own design, feel free. After all, it's your recipe.

Table of Contents

Chapter 1
Basic Blocks & Functions

This chapter introduces many of the basic building blocks found in the GreenPAK that will be used throughout the Cookbook. It will also present a few simple combinational logic designs that utilize look-up tables (LUTs).

Technique: Learning More About a Macrocell

This technique works with any version of GreenPAK Designer.

While using GreenPAK Designer you may wish to learn more about a specific macrocell. This can be done by selecting the macrocell in the GreenPAK Designer, then clicking the Information button at the bottom-left of the Properties window.

Info Button

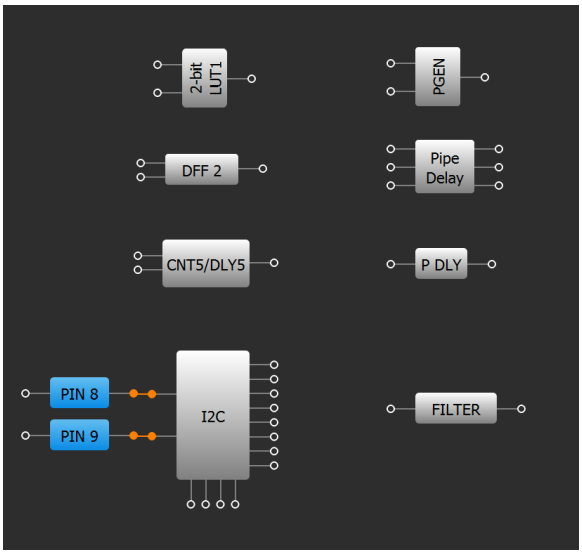
Overview: Digital Macrocells

Digital Macrocells are the basic functional components of any GreenPAK. They include:

Common Digital Macrocells:
- Look-Up Table (LUT)
- D Flip-Flop (DFF) / Latch
- Counter / Delay (CNT/DLY)

Communication:
- I2C (many devices)
- SPI (select devices)

Less Common:
- Pattern Generator (PGEN)
- Pipe Delay
- Programmable delay (PDLY)
- Filter / Edge Detector



Digital Macrocells

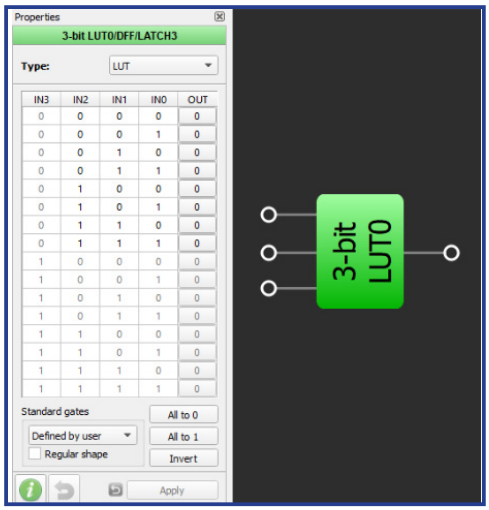
Many of the components in GreenPAK Designer can be configured to be one of multiple types of macrocells. This is indicated by the name of the digital macrocell: for example, 2-bit LUT0/DFF/LATCH0 can be, as the name implies, a LUT, DFF, or Latch. The selection of macrocell type is configured using the Type option in the Properties window.

Technique: Configuring Standard Logic w/ LUT Macrocells

This technique works with any GreenPAK.

Look-up tables are used in GreenPAK Designer to configure any digital logic for a two, three or four input, single output logic macrocell. The logic configuration is edited in the Properties window.

Most logic implemented in GreenPAK designs is standardized logic, such as MUX, AND, OR, etc… To expedite these common configurations, the Properties window has a Standard gates option that automatically convert the logic table into a standard gate configuration. If the Regular shape option is left unchecked the LUT shape will change to the standardized gate symbol.



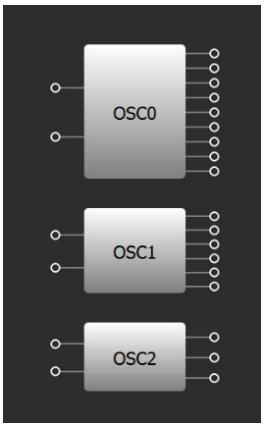
Config for 3-bit LUT0

Overview: Oscillators

GreenPAK ICs contain at least two oscillators. Many GreenPAKs, such as the SLG46826, have three oscillators. The most common, non-divided frequencies of the oscillators within GreenPAK are:

- 2KHz low speed, low power oscillator
- 2MHz medium speed
- 25MHz high speed

Each oscillator has several outputs, each with several pre-dividers to allow flexibility in clocking. To save power Auto-power on allows you to turn off the oscillator when the clock is not needed.
More information about oscillators can be found by using the Information button when the component is selected.

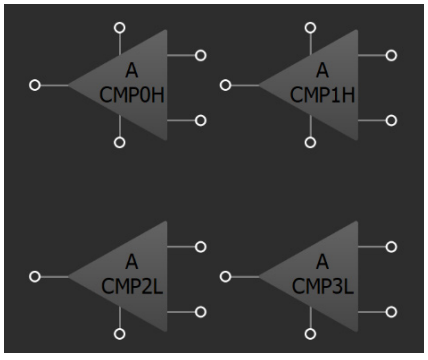


Oscillators

Overview: Analog Comparators

Almost every GreenPAK is equipped with two or more analog comparators [ACMPs], each with two input sources; IN+ and IN-. The input source to each is configured in the Properties window.

More information about analog comparators can be found by using the Information button when the component is selected.



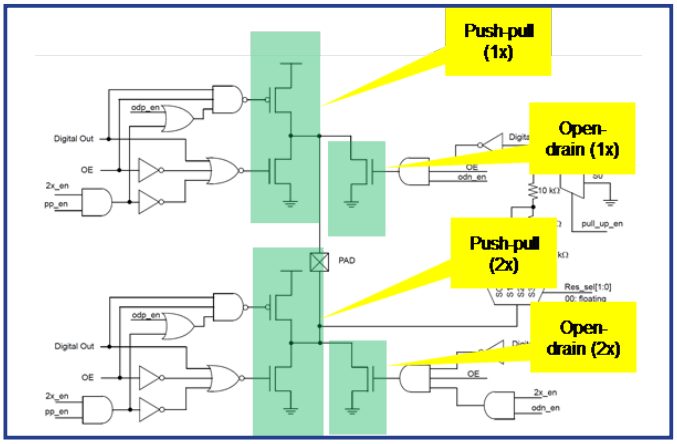
ACMPs

Overview: I/Os

I/Os within GreenPAK are very flexible. The I/O capabilities vary from pin to pin and part to part, so a design should be mapped to the necessary pin configuration before choosing a specific GreenPAK.

Outputs can configure to be Push-pull or open-drain in either a NMOS or PMOS configuration. A scaling factor, such as 2x, indicates that the output strength is doubled.

Additionally, pull-up and pull-down resistor options of 10kΩ, 100kΩ, and 1MΩ are available on output pins.
Multiple input options are available as well, such as: Digital-In, Digital-In with Schmitt trigger, Low Voltage Digital-In and Analog-In. Analog in is used as an input to an ACMP.

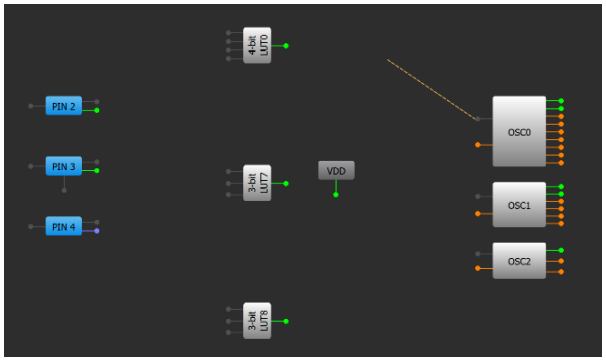


Typical I/O Structure

Overview: Interconnections

Interconnection with GreenPAK Designer is easy. The system will guide you on which connections you can make. When you click on any connection point, the system:

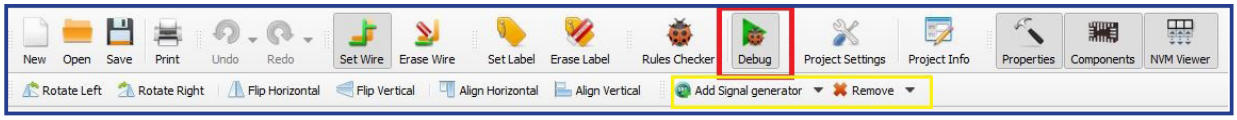
- Highlights all available connections in green
- Gives you a "rubber band" connection that you can stretch to any of these green connection
- This results in a green wire to show you interconnections you have made



Interconnections

Technique: Simulation and Emulation Using GreenPAK Designer

Emulation is available for all GreenPAK parts and Simulation is available on many GreenPAK ICs.



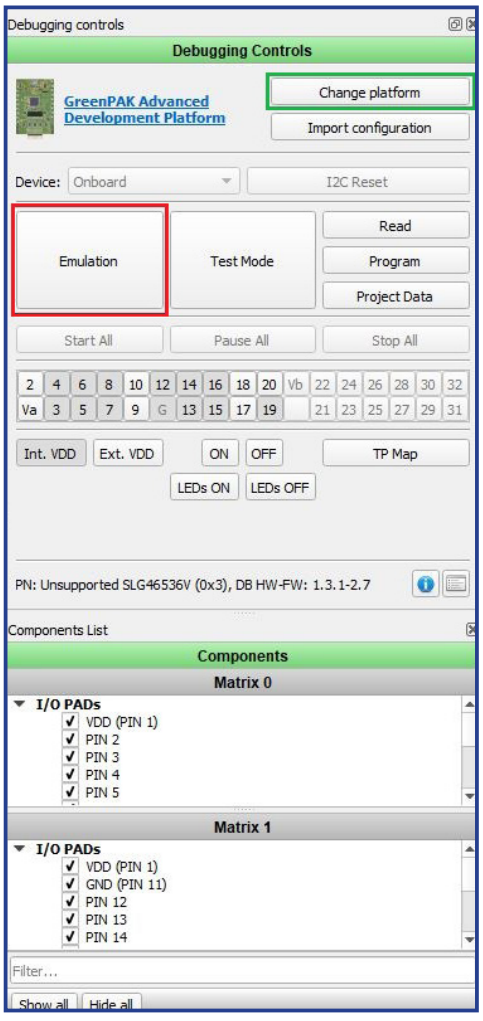
Toolbar

When developing a design, it is important to be able to quickly test the functionality. GreenPAK Designer makes debugging effective and easy.

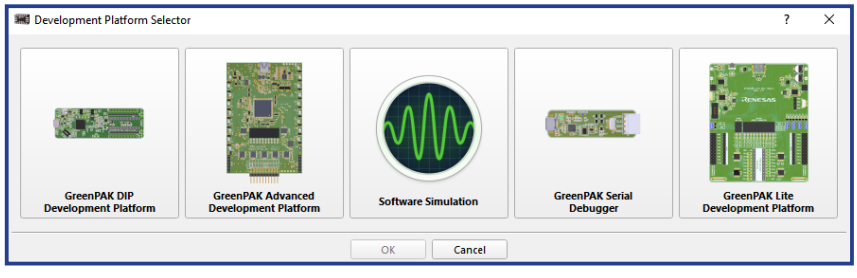
There are two ways to quickly check your design:
1. Simulation
2. Emulation

Simulation simulates the operation of the circuit in conditions depictive of reality without the need of a physical IC. It should be kept in mind that simulation can't provide for all the nuances of a real-world system.

Emulation allows, with the presence of a demo board and the GreenPAK chip, to check the operation of your design directly in the hardware without permanently programming a part. This allows you to quickly make changes to the project and use your emulation to check your guesswork.
1. If the design is ready for debug select the Debug button (boxed in red in the figure above) to go to the emulation/simulation selection
menu.
2. Next, select the platform with which you want to check your design.
3. After selecting a platform, go to the debug menu, where further actions will be suggested depending upon the platform you choose.
4. If you need to change the platform, you can do this at any time by selecting Change platform.



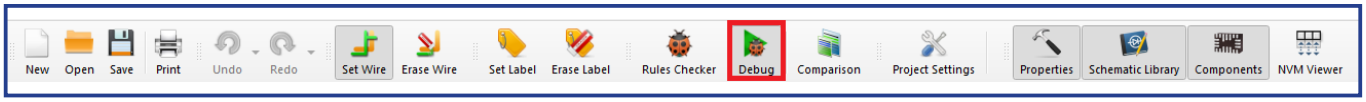
Debugging Menu



Platform Selection Menu

Technique: GreenPAK Programming

Debugging control is available for all GreenPAKs.



Toolbar in GreenPAK Designer

When developing a design, it is important to be able to quickly test the functionality. GreenPAK Designer makes debugging effective and easy.

Change platform

Select the type of hardware platform with the supported features.

Import configuration

Import user configuration of test points from other platforms.

Device

Allows a user to work with an external chip on a specified device address.

I2C Reset

Suppose I2C serial communication is established with the device. In that case, it is possible to reset the device to initial power-up conditions, including the configuration of all macrocells and all connections provided by the Connection Matrix. This is implemented by setting the register I2C reset bit to "1", which causes the device to re-enable the Power-On Reset (POR) sequence, including the reload of all register data from NVM.

Emulation

- Emulation of the current project will be loaded to the chip (but not programmed) and will be ready for a test on the hardware board.
- Emulation (sync).In addition to Emulation, each change made in the project will be immediately loaded onto the chip.

Test mode

Test mode is used for connecting or disconnecting the chip's I/O pads to TP controls configured by a user. Also, a user can check a programmed chip using the test mode without emulation. To do this: turn on the test mode and internal VDD button. The test mode can work without power on the chip. The user will control the power manually.

Read

Read chip data using the hardware board.

Program

Program chip with the current project. For some chip models, a user can configure the programming process by clicking the Programming options at the Program button. Available programming options:

- Program NVM (Program the chip's NVM)
- Program EEPROM (Program the chip's EEPROM)

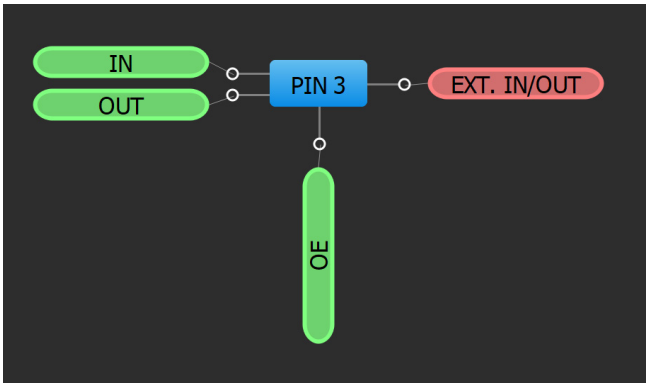
Project Data

The table of NVM and EEPROM (available for specific chip revisions) bits.

Technique: OE Pin

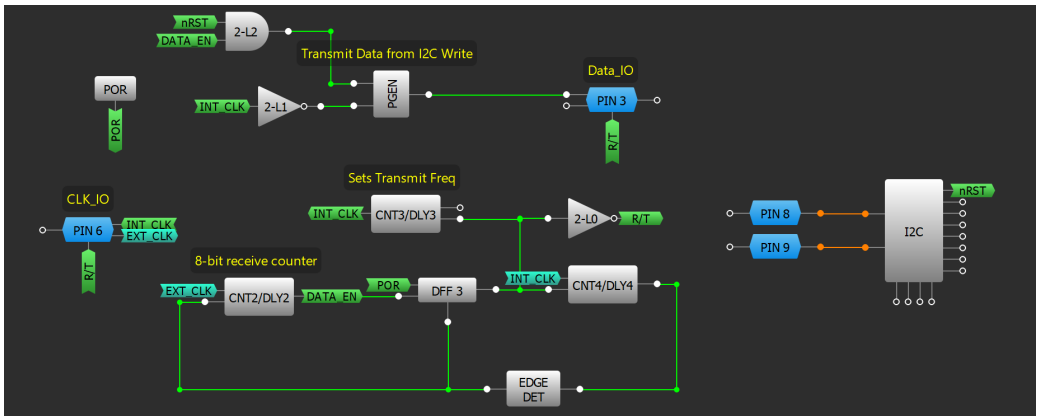
This technique can be used within any GreenPAK with OE pins.

Typically, GreenPAK I/O are configured as an input or output. Output enable (OE) pins are select pins within most GreenPAK that allow the pin to dynamically change between a Digital input and Digital output. When the OE GPIO is set as a permanent input the OE pin is set to ground and if the GPIO is set as a permanent output the OE pin is set to VDD. Setting the GPIO as a "Digital input/output" allows for this selection to be made in the matrix.

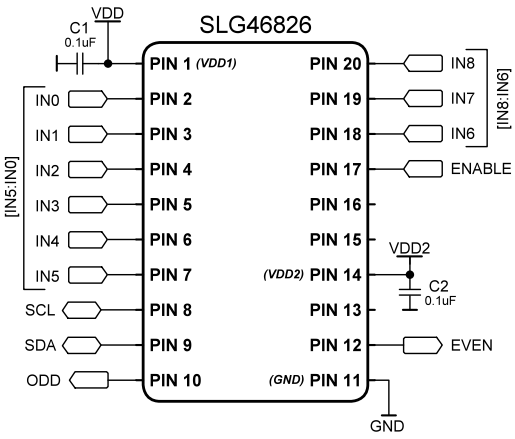


Setting a GPIO as a digital input/output allows for two-way communication. It also allows for the GPIO to be set to Hi-Z in addition to a logical high and low.

If the GPIO is used for two-way communication, it's important to implement a timing circuit for OE selection. In the example circuit below the OE pins of CLK_IO and Data_IO are switched from low to high after CNT2 sees 8 clocks, consequently setting the OE pins as outputs to transmit the internal signals. After another 8 clocks the OE pins are reset low, setting them as inputs again to receive an external signal.



Application: Parity Bit Generator

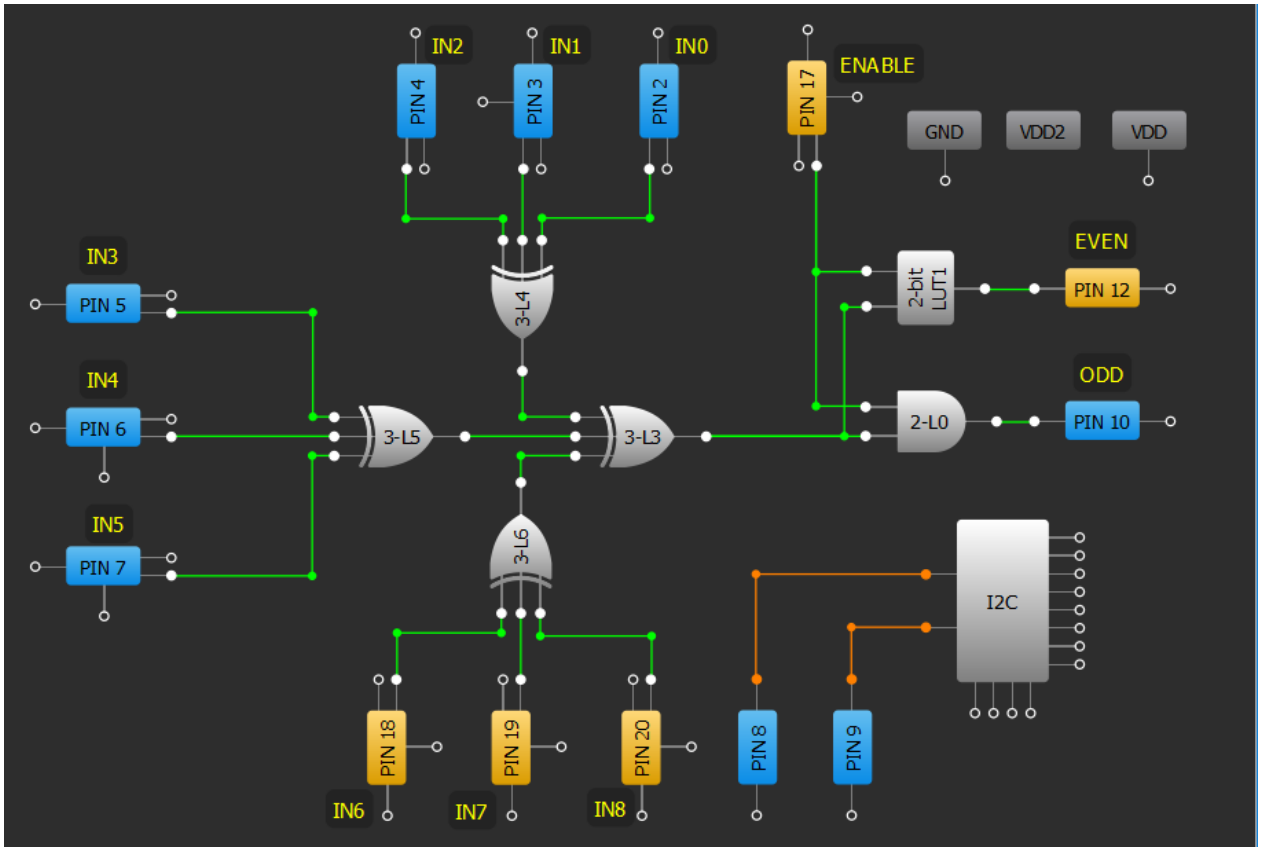


Parity Bit Generators are used to check the integrity of a signal; it is the simplest implementation of a Cyclic Redundancy Check (CRC). Parity Bits are used prior to committing data to an MCU or other control unit to ensure the incoming data hasn't been corrupted.

Ingredients

- Any GreenPAK
- No other components are needed

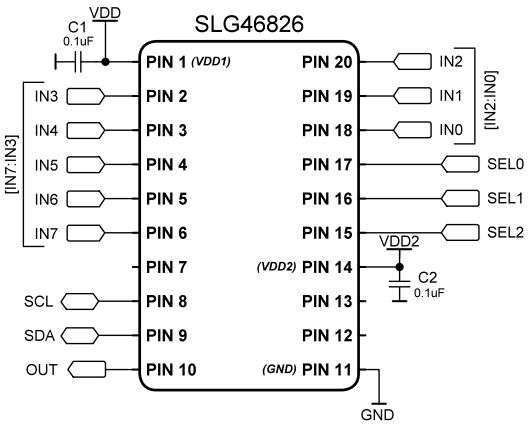
GreenPAK Diagram



Design Steps

1. Connect input pins using XOR gates using Technique: Configuring Standard Logic w/ LUT Macrocells. XOR gates are used to calculate the running sum of 1's.
2. Add logic for the ENABLE signal.

Application: 8-bit Multiplexer

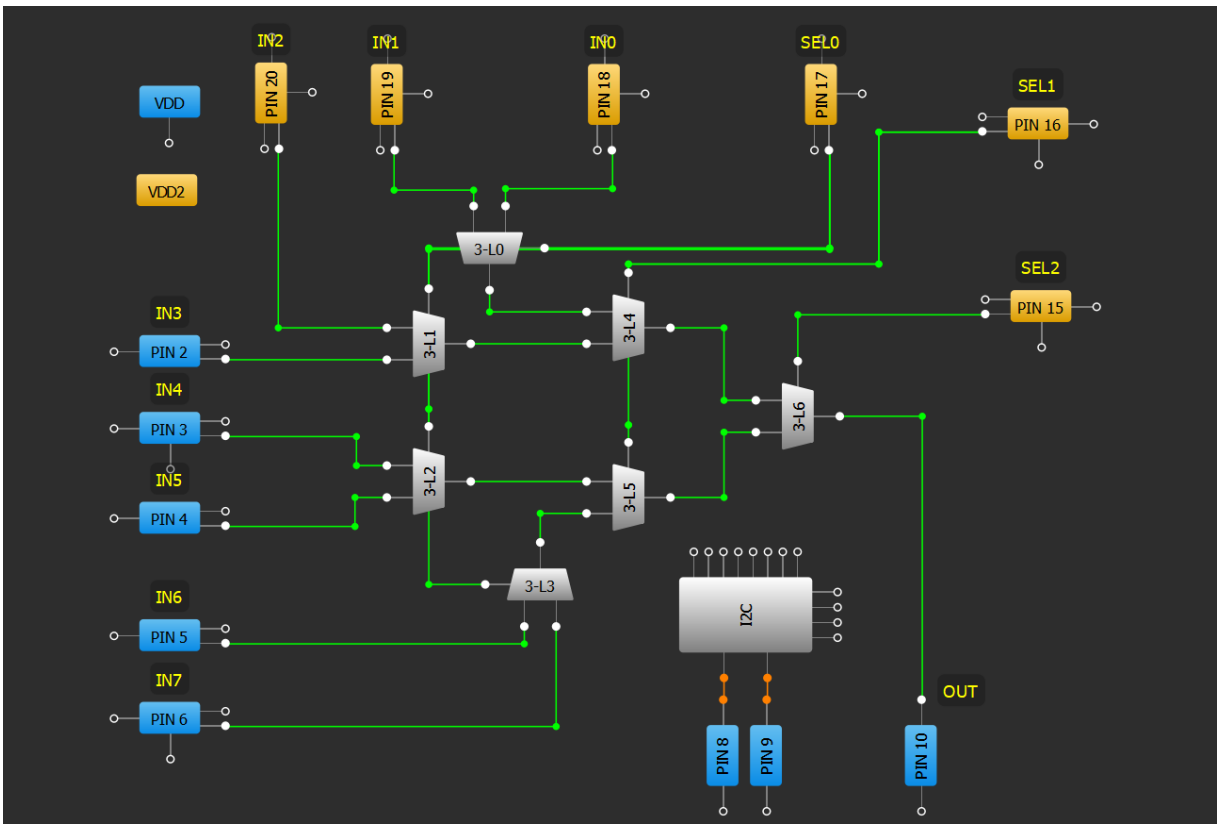


A one-hot is a group of bits that contain a single HIGH (1) with all other bits set LOW (0), i.e. 0001. The reverse implementation with a collection of 1's and a single 0 is called a one-cold. This design is an encoder that will output a specified one-hot output to its outputs based on a two-bit code on its inputs.

Ingredients

- Any GreenPAK
- No other components are needed

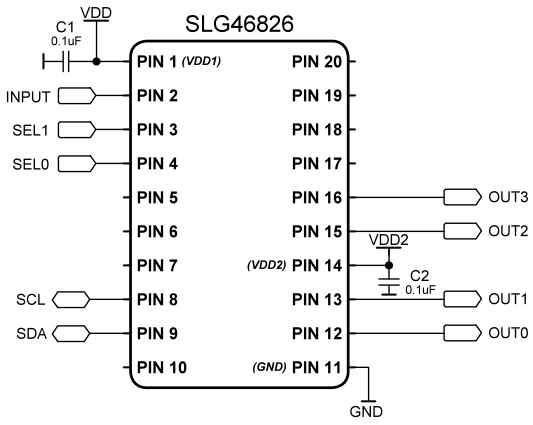
GreenPAK Diagram



Design Steps

1. Connect input pins to 4 LUT's configured as multiplexers using Technique: Configuring Standard Logic w/ LUT Macrocells. INx should connect to A or B, SEL0 should connect to S on all 4 mux's.
2. Add second and third stage cascading multiplexer blocks to create the more significant SEL bits.
3. Add an output pin connected to the last-stage multiplexer.

Application: Demultiplexer

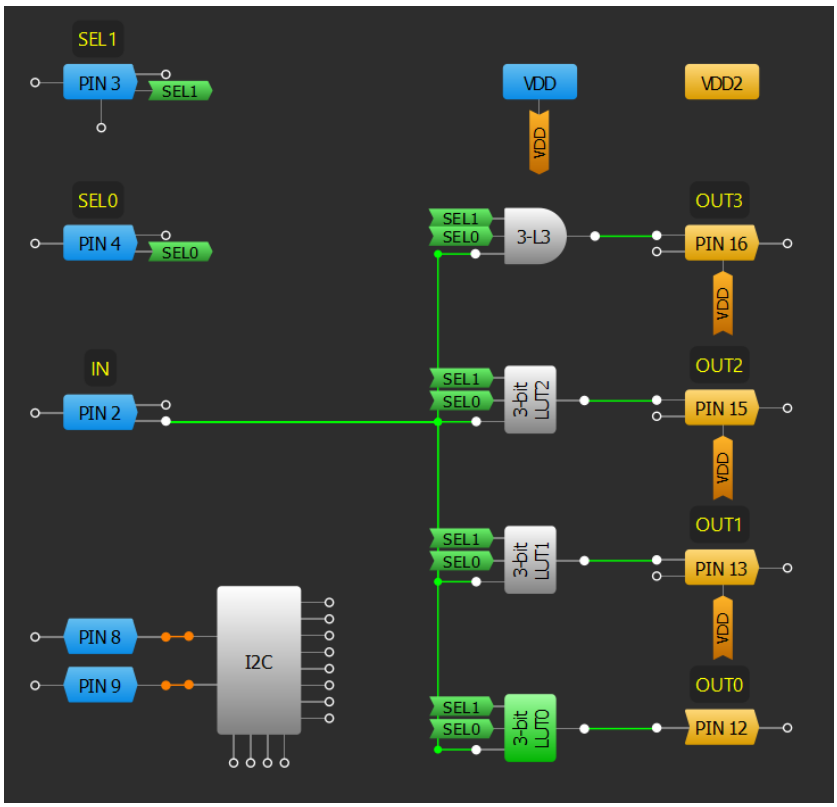


A one-hot is a group of bits that contain a single HIGH (1) with all other bits set LOW (0), i.e. 0001. The reverse implementation with a collection of 1's and a single 0 is called a one-cold. This design is an encoder that will output a specified one-hot output to its outputs based on a two-bit code on its inputs.

Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram



Design Steps

1. Configure input pins for a signal input (IN), two select lines (SELx) and four output pins (OUTx).
2. Configure the LUTs to each pass the signal from IN upon a specific logic input on the select lines. For example, 3-L3 will be HIGH when SEL0, SEL1 and IN are HIGH.

Chapter 2
Sequential Logic

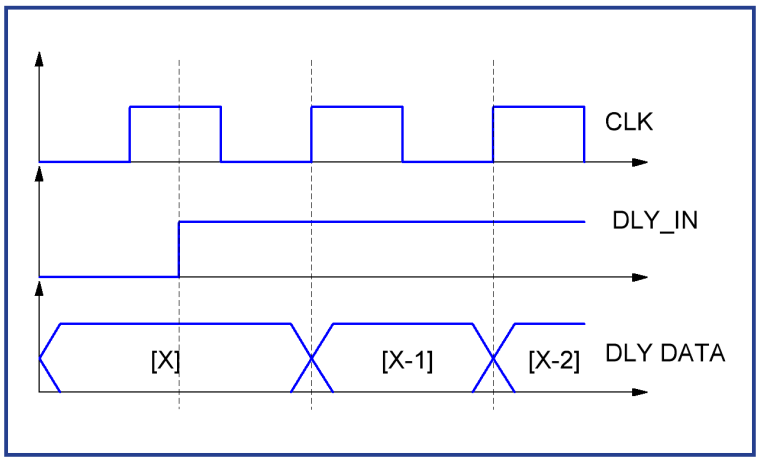
This chapter presents applications that involve sequential logic. Some sequential logic applications are counters, system reset circuits, power sequencers, and state machines

Technique: Optimizing CNT/DLY Accuracy

This technique works with any GreenPAK. The accuracy of the oscillator and CNT/DLY blocks vary from part-to-part.

GreenPAK ICs, like all chips with internal oscillators, have inherent variation in timing. This is attributed to factors like manufacturing, temperature and, in the case of GreenPAK, user design practices. By using simple design principles the accuracy of counters and delays within a GreenPAK design can be improved.

The relationship between the oscillators and CNT/DLY blocks should be considered. The oscillators are global oscillators; they are used for any number of CNT/DLY blocks and aren't initially synchronized to the start/stop of a delay or counter. Consequently, when a counter or delay is enabled it will only begin to increment on the next clock edge. This is depicted in figure below, where an enable signal for a delay is activated mid-clock-cycle and doesn't begin to decrement until the next rising edge.



Behavior of Rising Enable for Delay

This is factored into the typical delay time calculation for the CNT/DLY blocks:

$$Delay_{time}\ (typical) = ((Counter\_Data+1)+t)/clock,$$

where t is between 0 and 1

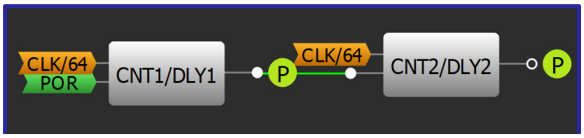
Thus, as the value of "Counter_Data" increases, the influence of "t" on the delay time will be proportionately less. Additionally, the absolute value of the delay time is kept the same, despite using a larger "Counter_Data" value, if a faster "clock" value is used. In the Properties window of the selected CNT/DLY block both the Counter Data value and the Clock Source can be modified.

Additionally, the timing characteristics within the datasheet of the respective GreenPAK should be referenced to ensure factors such as Power-ON time, frequency settling time and percent deviation across temperature are considered.

Technique: Sequencing CNT/DLY Blocks

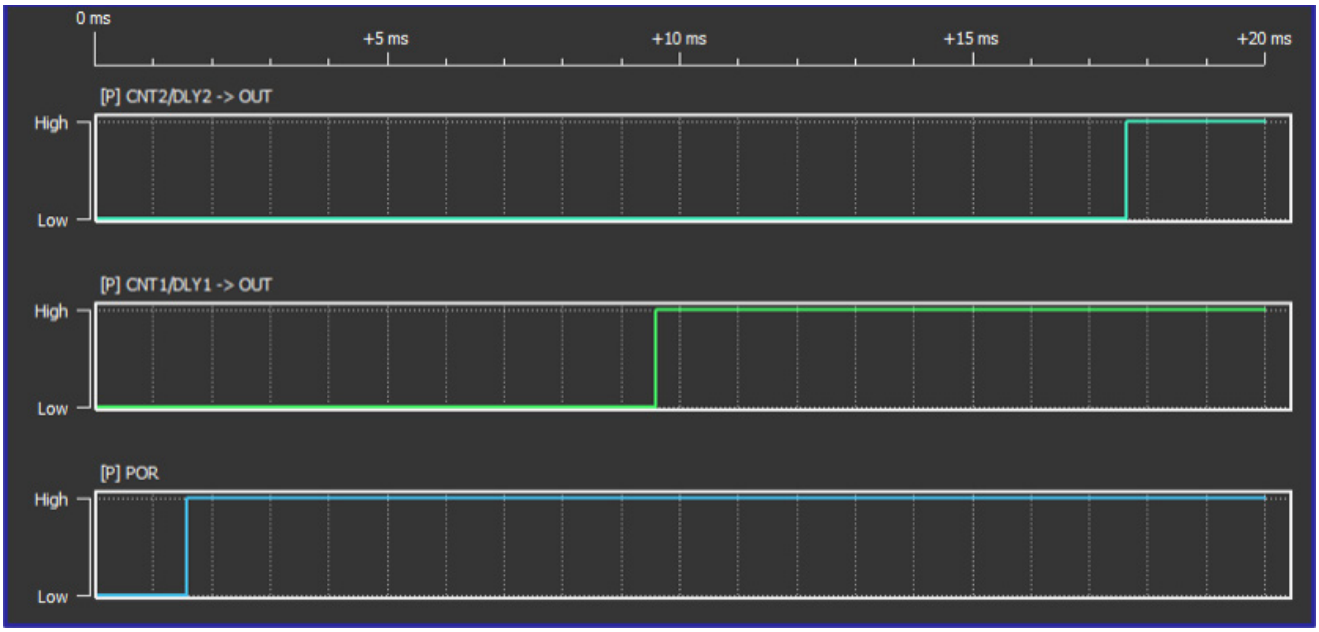
This technique will work with any GreenPAK.

Delay blocks can be chained together to sequence signals. By chaining the output of one delay block to the input of another a sequential set of delays is made.



Sequential Rising Edge Delays

The CNT/DLY blocks in the sequential set should be set to "Delay" in the Mode setting within the Properties window. Typically, the Edge select setting should be the same for all sequenced components. The figure below shows the effect of the two sequential CNT/DLY blocks set to rising edge, 8ms delays of the Power-On-Reset (POR) signal.



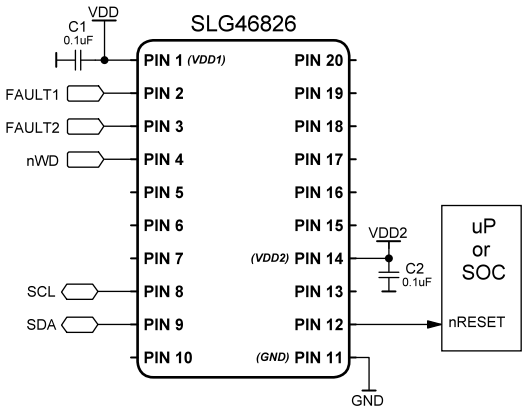
Sequential Delay Simulation

One can also chain together CNT blocks for a longer counted time. When Chaining CNT blocks together, the CLK of the CNT should be driven by the output of the previous counter. This is done in the properties by selecting a CNT block and in the Properties window, choosing the Clock connection to be sourced from the previous CNTx/DLYx.

Sequential Counters

Application: System Reset

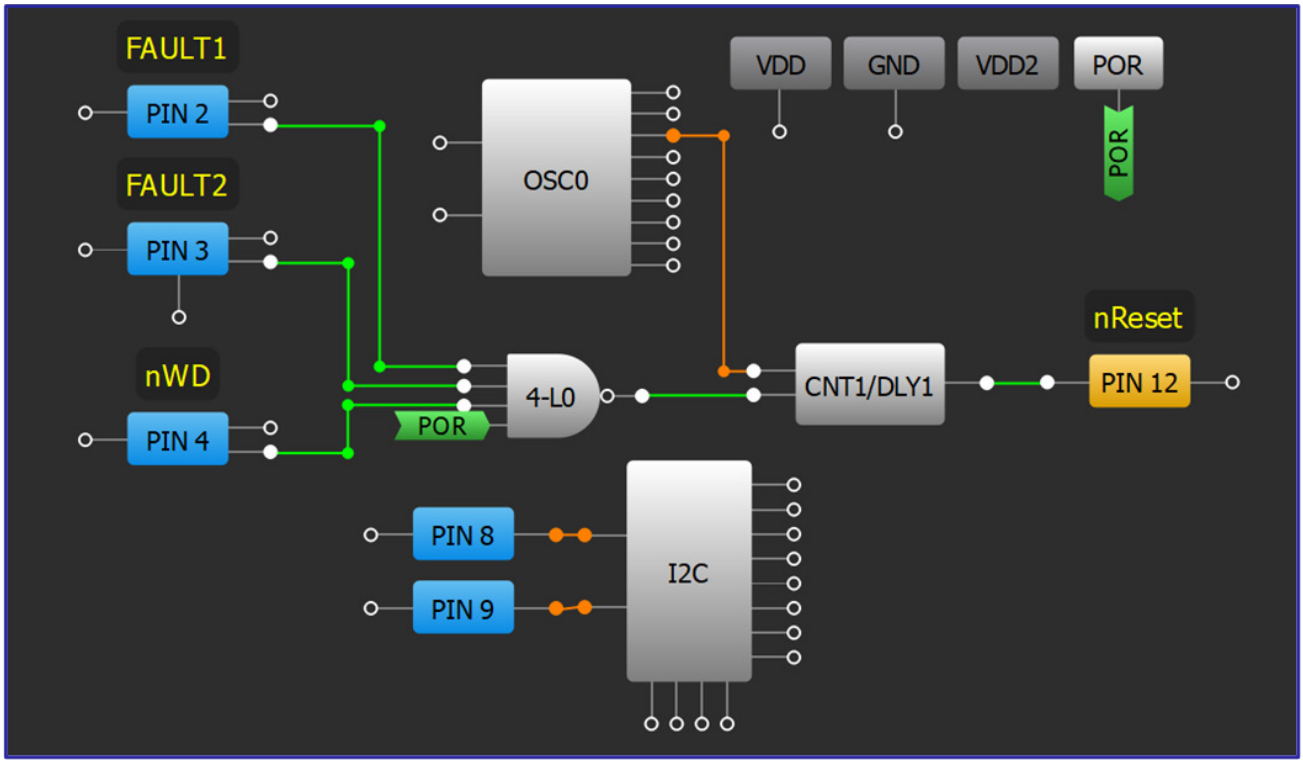
System Reset ICs are used to provide a reset to a microprocessor during faults, manual resets, brown-outs and more.



Ingredients

- Any GreenPAK
- No other components are needed

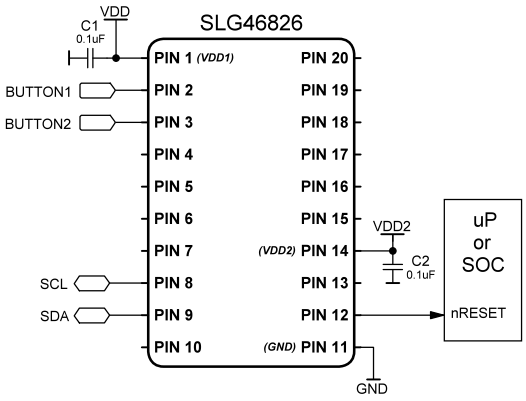
GreenPAK Diagram



Design Steps

1. Configure an I/O as an input for each input signal.
2. Add LUT logic to create a HIGH signal when any of the lines are active. The logic is dependent on whether each signal is active-high or active-low.
3. Configure a CNT/DLY block to "One shot" mode, with Edge select configured to "Rising." Set the Counter data to create the desired length of pulse. For an active-low pulse change the Output polarity to "Inverted (nOUT)."
4. Connect the CNT/DLY block's output to an output pin.

Application: Several Button Reset

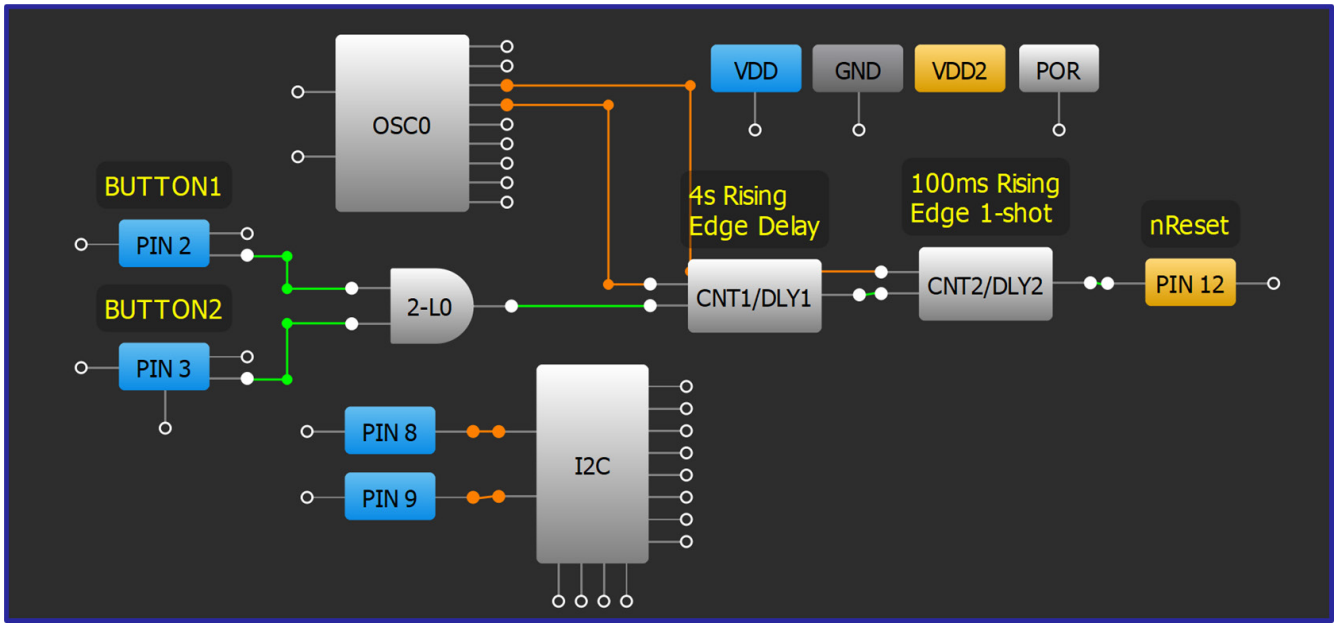


Pressing and holding several buttons to initiate a hard reset is a common interface in many devices. Implementing this application in a separate IC ensures the reset will be acknowledged and acted upon, even if the rest of the system is experiencing one or more software, firmware, or hardware issues.

Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram

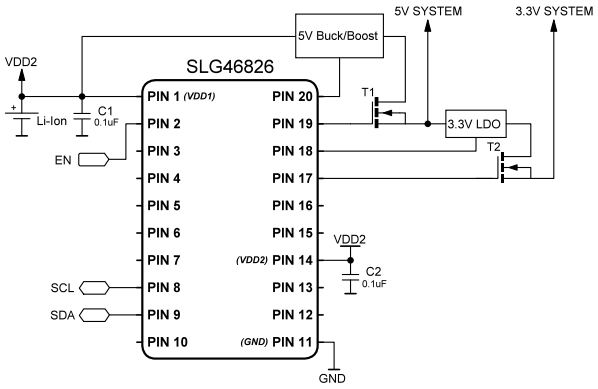


Design Steps

1. Configure an I/O as an input for each button.
2. Add LUT logic to create a HIGH signal when both buttons are active. The logic is dependent on whether each signal is active-high or active-low.
3. Configure a CNT/DLY block to “Delay” mode, with Edge select configured to “Rising”. Set the Counter data to create the desired length of button hold time. For an active-low pulse change the Output polarity to “Non-inverted (OUT).”
4. Configure a second CNT/DLY block to “One shot” mode, with Edge select configured to “Rising.” Set the Counter data to create the desired length of pulse. For an active-low pulse change the Output polarity to “Inverted (nOUT).”
5. Connect the CNT/DLY block’s output to an output pin

Application: Basic Sequencer

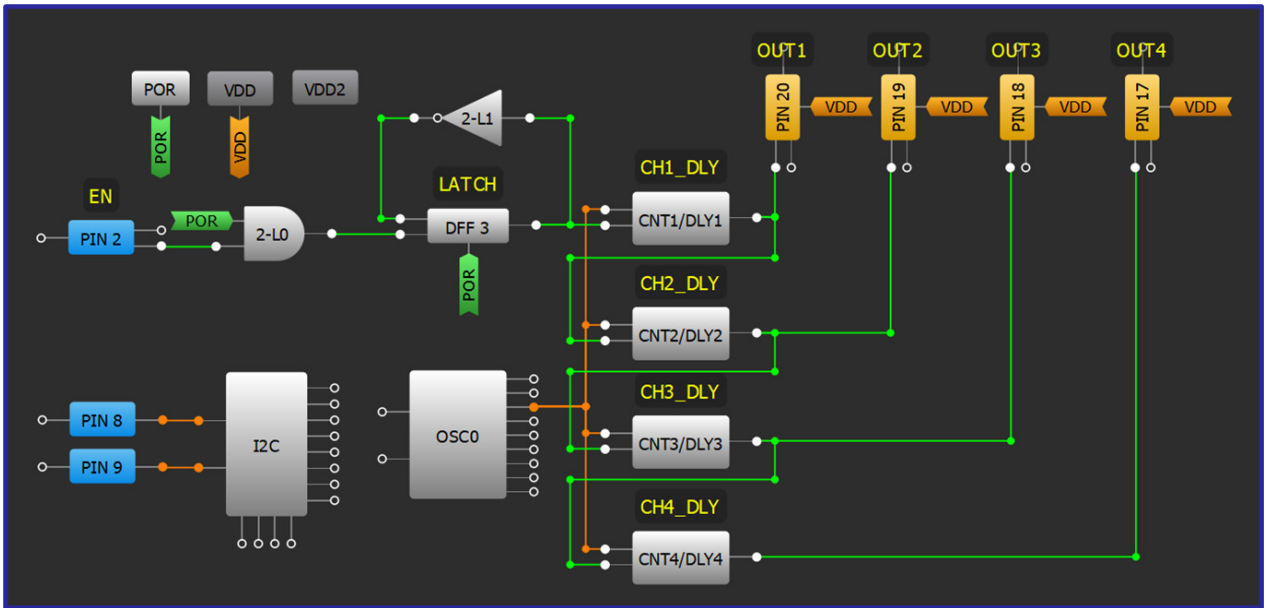
Sequencers are used when a designer needs to sequentially activate different portions of a system. This is critical for applications that require several power rails.



Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram

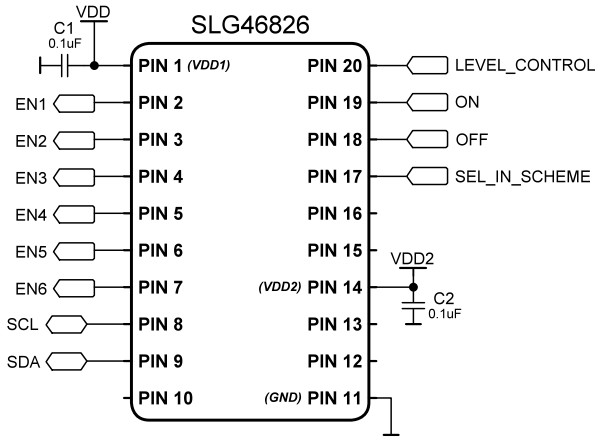


Design Steps

1. Use LUTs to configure the desired start-up condition.
2. Use a latch or DFF to maintain the start-up signal so it can be read by DLY blocks.
3. Chain the Delays using Technique: Sequencing CNT/DLY Blocks.
4. Connect each delay channel to the desired output pins.

Application: Cascaded Sequencer

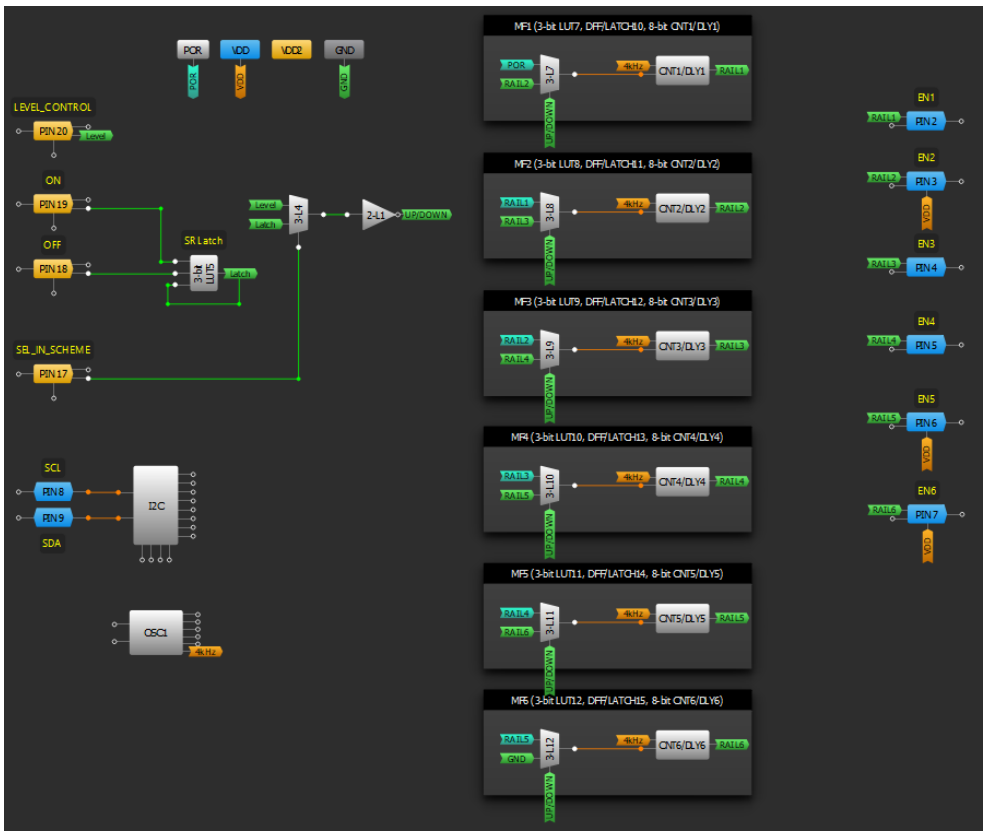
Sequencers are used when a designer needs to sequentially activate different portions of a system. It is typical to have a cascaded sequence, such that a rail does not turn off until all the rails below it have turned off.



Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram

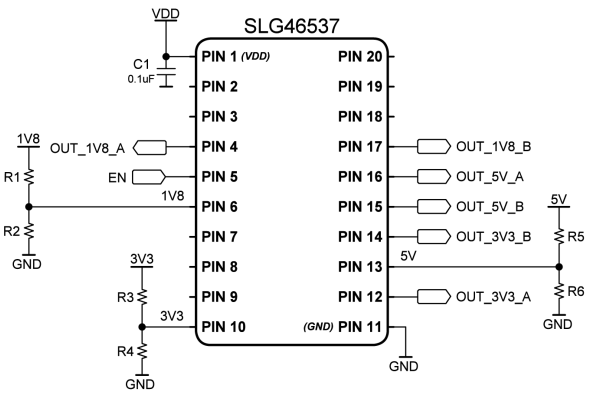


Design Steps

1. Configure input structure. Here you can select between level and latching control with PIN17.
2. Multiplex inputs to DLYs within a Multi-function block to achieve a cascaded effect.
3. Connect the DLY outputs to push pull output pins.

Application: Voltage Monitoring Power Sequencer

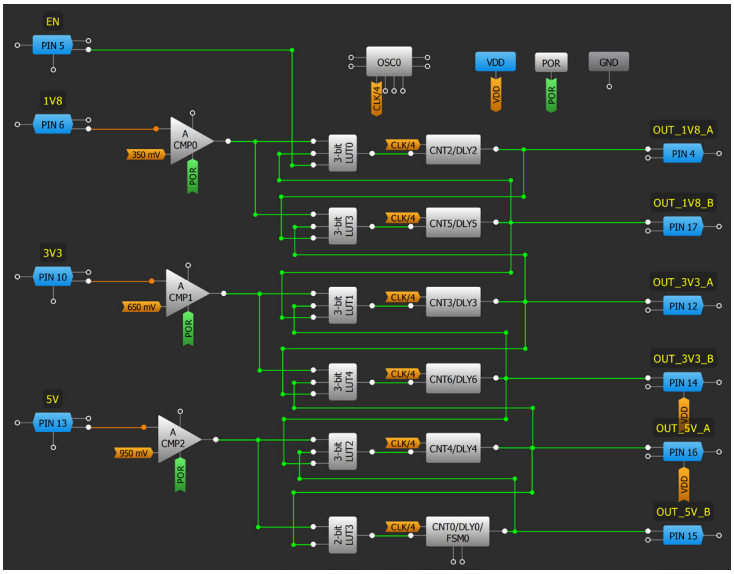
Sequencers are used when a designer needs to sequentially activate different portions of a system. It is typical to have a cascaded sequence, such that a rail does not turn off until all the rails below it have turned off.

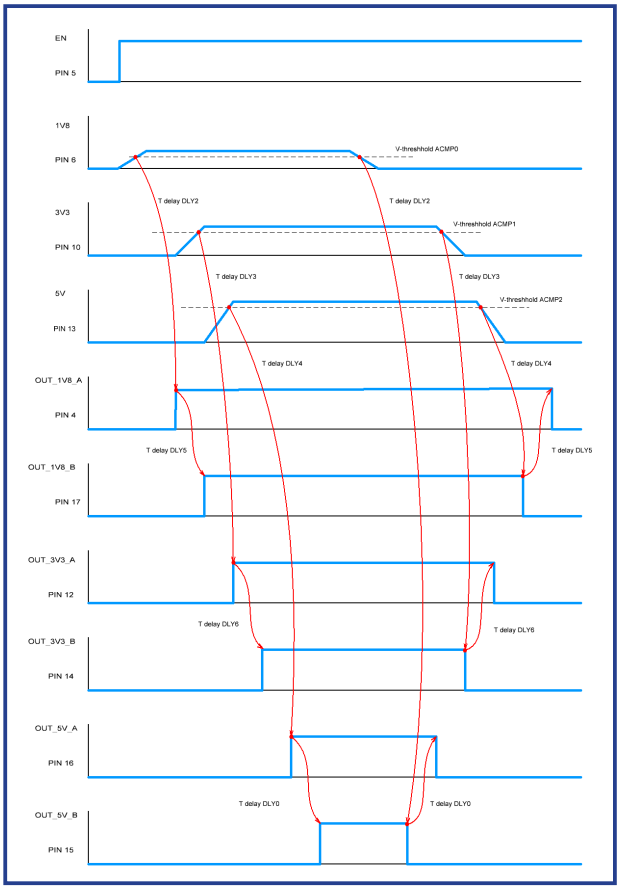


Ingredients

- Any GreenPAK with 3 ACMPs
- Six resistors

GreenPAK Diagram



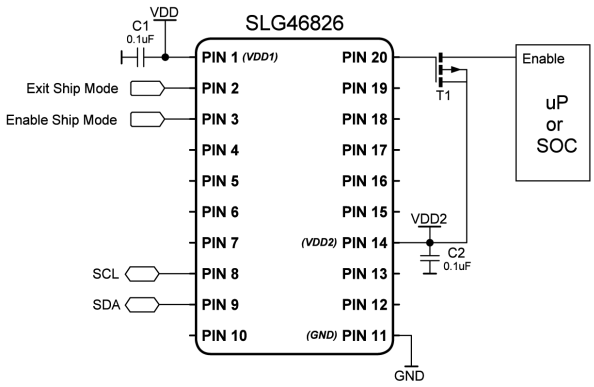


Design Steps

1. Configure input pins for the EN and voltage monitoring.
2. Configure output pins to sequence system.
3. Power on ACMPs, connecting POR to PWR UP and configure the IN- source of each with the desired voltage threshold levels.
4. Configure DLYs with the desired delay times.
5. Configure LUTs with the proper logic functions.

Application: Ship Mode Controller

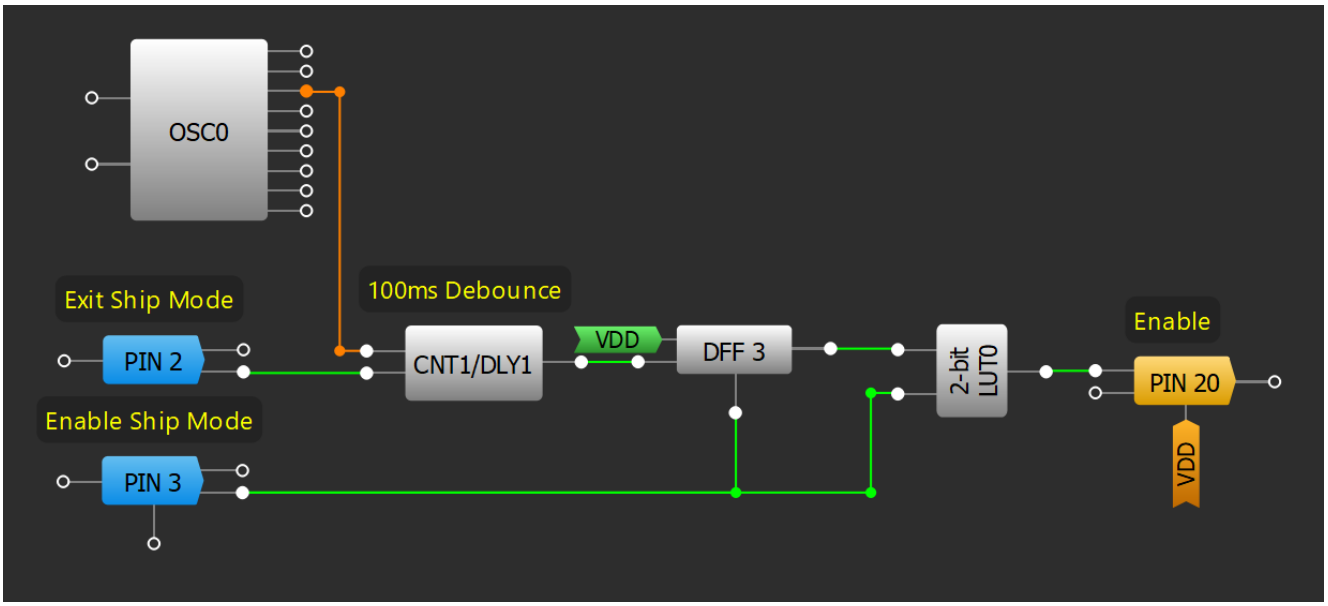
An ultra-low power button monitor can save battery life while a product is not yet with the end user. This enables a better first experience by the user.



Ingredients

- Any GreenPAK
- External PMOS load switch

GreenPAK Diagram

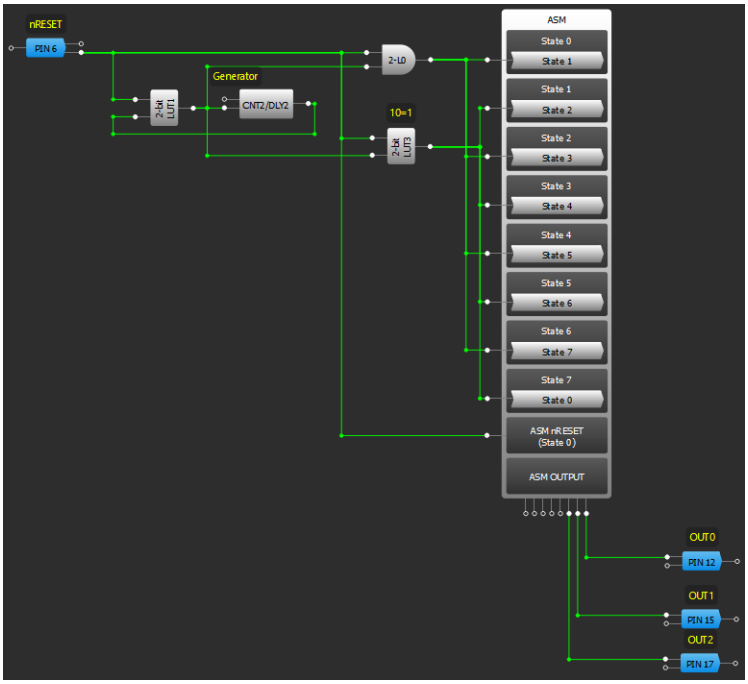


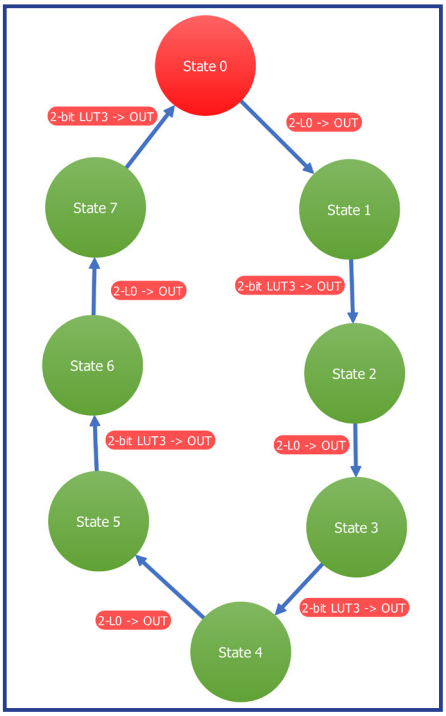
Design Steps

1. Configure PIN2 as an input with 1MΩ pullup.
2. Set desired Button delay time in CNT1/DLY1.
3. Modify LUT contents for correct polarity in and out of ship mode.

Technique: Creating a Synchronous State Machine from an ASM

Synchronous state machines (SSM) transitios on the edge of an incoming clock if the transition condition is met. The generic approach to convert a GreenPAK Asynchronous State Machine (ASM) macrocell into a SSM uses a clock signal with a pulse width greater than the ASM transition time.





Consider the SSM in the 3-bit counter example above. CNT2 and 2-bit LUT1 are used to generate the clock. The ASM uses 8 states connected in series. 2-bit LUT0 and 2-bit LUT3 are used to prevent a logic high signal on two near state transitions. The value of the ASM output for every state is shown below.

RAM

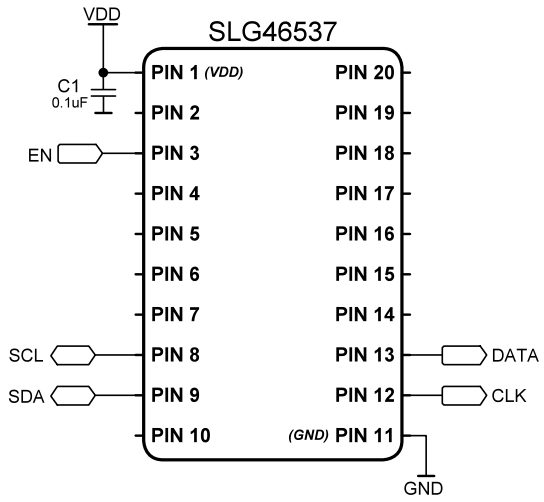
State name	Connection Matrix Output RAM							
	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0
State 0	0	0	0	0	0	0	0	0
State 1	0	0	0	0	0	0	0	1
State 2	0	0	0	0	0	0	1	0
State 3	0	0	0	0	0	0	1	1
State 4	0	0	0	0	0	1	0	0
State 5	0	0	0	0	0	1	0	1
State 6	0	0	0	0	0	1	1	0
State 7	0	0	0	0	0	1	1	1

The ASM changes from the reset state (State 0) to the next state (State 1) when PIN6 goes high. The following transitions through the states occur as CNT2 toggles first high, then low, and so on.

For more a detailed information about the process of creating an SSM with the ASM see AN-1126 ASM to Synchronous Conversion.

Application: N-Length Bitstream

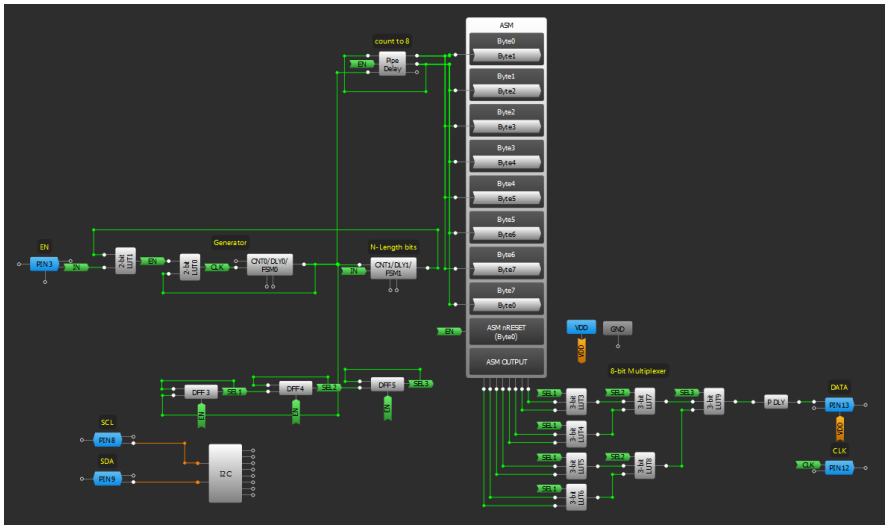
A bitstream is a sequence of bits transmitted continuously over a communications path. The GreenPAK can create a repeating string of up to 64 bits.

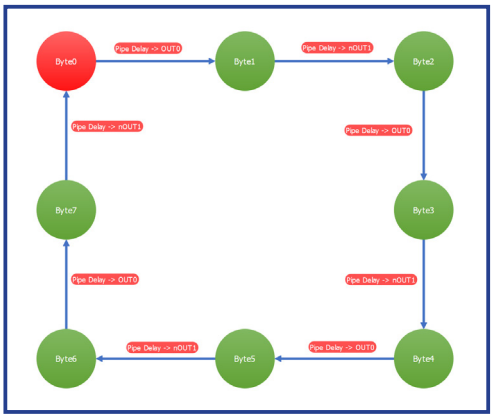


Ingredients

- Any GreenPAK with an ASM

GreenPAK Diagram





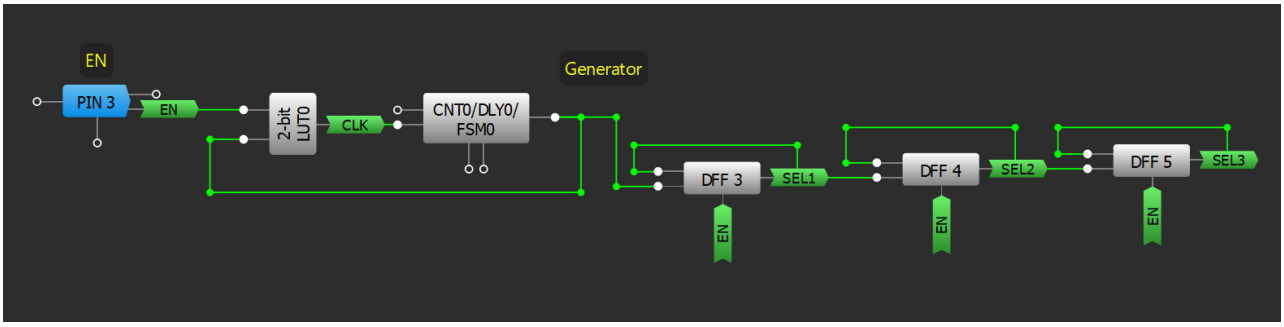
Design Steps

1. Configure a generator and 8-bit Multiplexer using Application: 8-bit Multiplexer.
2. Configure CNT1 to determine the length of the bitstream.
3. Configure the ASM using Technique: Creating a Synchronous State Machine from an ASM.
4. Connect the outputs of the 8-bit multiplexer and generator to the desired output PINs.
5. The length of the bitstream (counter data of CNT1) can be changed using I2C.
6. The DATA stored in the ASM output RAM can be changed using I2C.

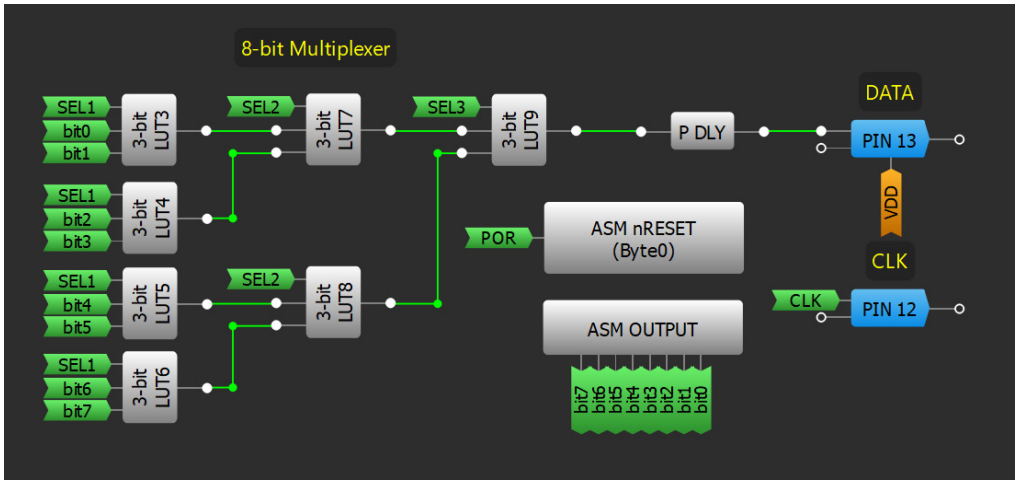
Technique: Multiplexing a Bitstream

This technique can be used in any GreenPAK.

GreenPAKs are often used to transfer a data pattern. If the data is transferred from the GreenPAK or the data is transmitted along several lines from the SoC, they must be amalgamated for transmission on one line. Below is an example of the GreenPAK multiplexing a bitstream originating from the ASM output RAM.



The generator circuit is shown above. The generator implements the operation of the CLK line for the synchronous data transmission, enabled by the EN signal. The generator also implements the multiplexer operation algorithm for the correct combination of the transmitted data on one line.

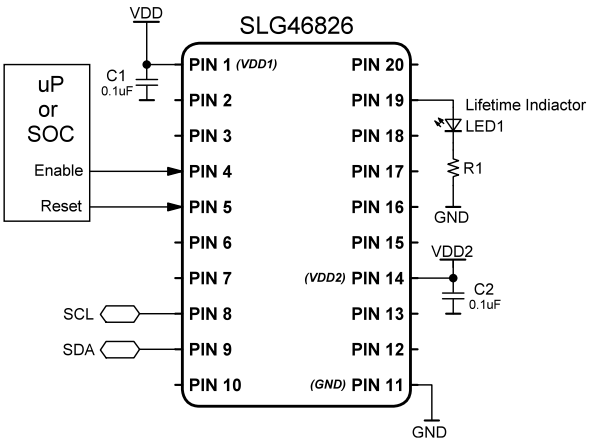


IN3	IN2	IN1	IN0	OUT
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

The 8-bit multiplexer circuit is shown above. All LUTs are configured collectively as an 8-bit multiplexer (see AN-1003) the MUX truth table is also shown above. The multiplexer outputs the combination of bits from the ASM block outputs according to the algorithm of the generator and the data is transmitted on one line to DATA. The DATA output will always be the same as the MSB of the ASM output RAM. when EN is LOW. The ASM output RAM can be changed via I2C. logic can also be implemented to change the state of the ASM in order to change the data bits. If the ASM isn't available, the inputs can be pulled high or low for the data values.

Application: 10 Year Counter

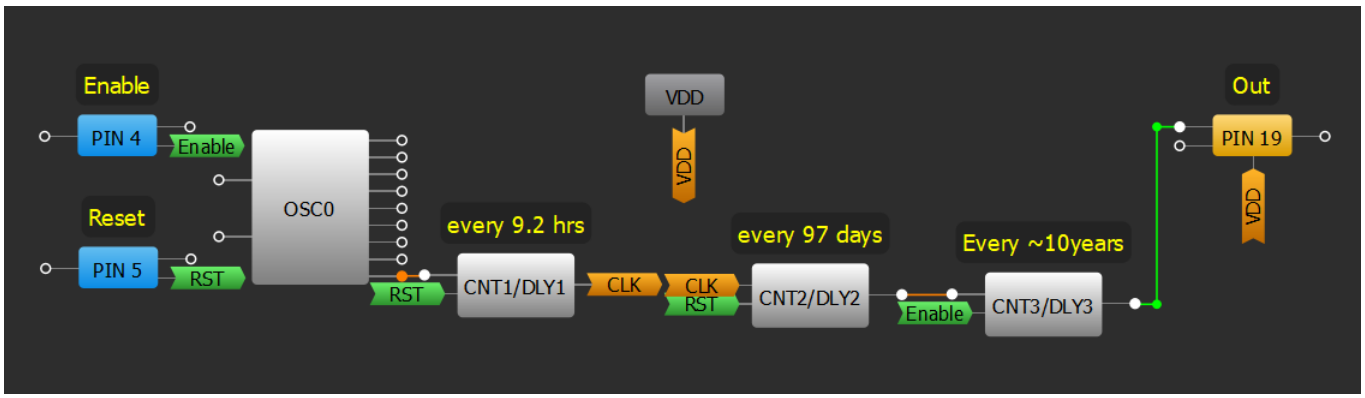
Ultra-long counters can be used to determine the lifetime of a product without requiring a large tax on the power budget.



Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram

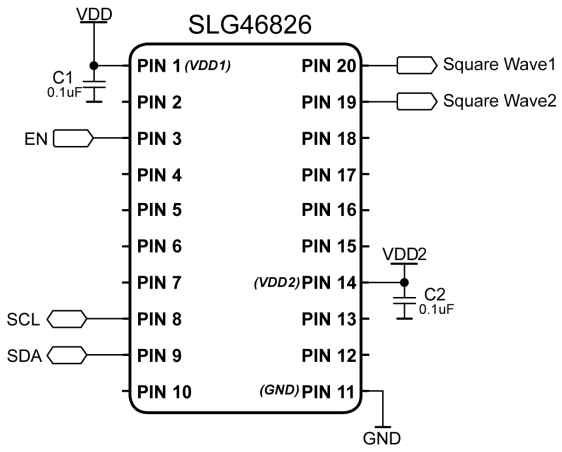


Design Steps

1. Chain the Counters using Technique: Sequencing CNT/DLY Blocks.
2. Connect input pins and output pin.
3. Set timing in CNT properties.

Application: Square Wave Generator

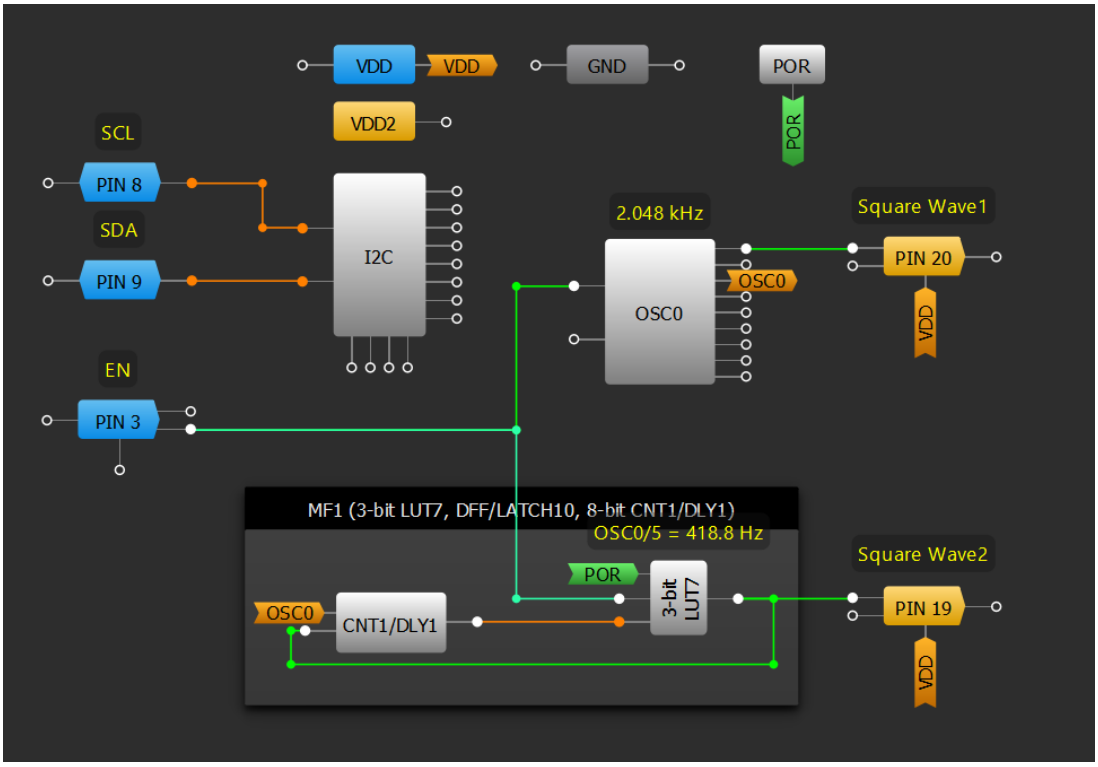
Square waves are essential for clocking digital systems. They can easily be implemented in GreenPAK with the oscillator blocks or a delayed logic for a customized frequency.



Ingredients

- Any GreenPAK
- No other components needed

GreenPAK Diagram



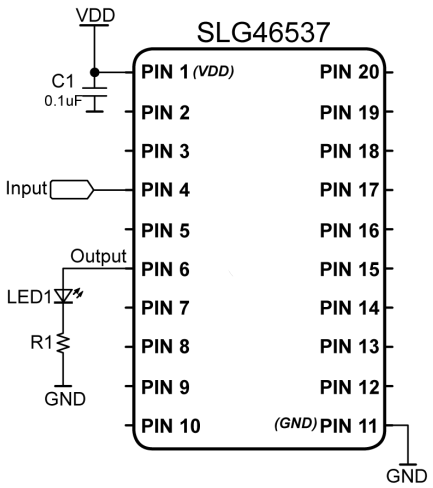
Design Steps

1. Configure the EN input and the square wave outputs.
2. Use an internal oscillator to generate a square wave on PIN20. The 'CLK' predivider and 'OUT0' second divider can be altered to customize the frequency.
3. Use a both edge delay and a LUT to generate a square wave on PIN19. This configuration allows the user to divide down the frequency of the square wave by a finer adjustment.

$$Division\ Coefficient = Counter\ Data + 2$$

Application: Two Event Button Press

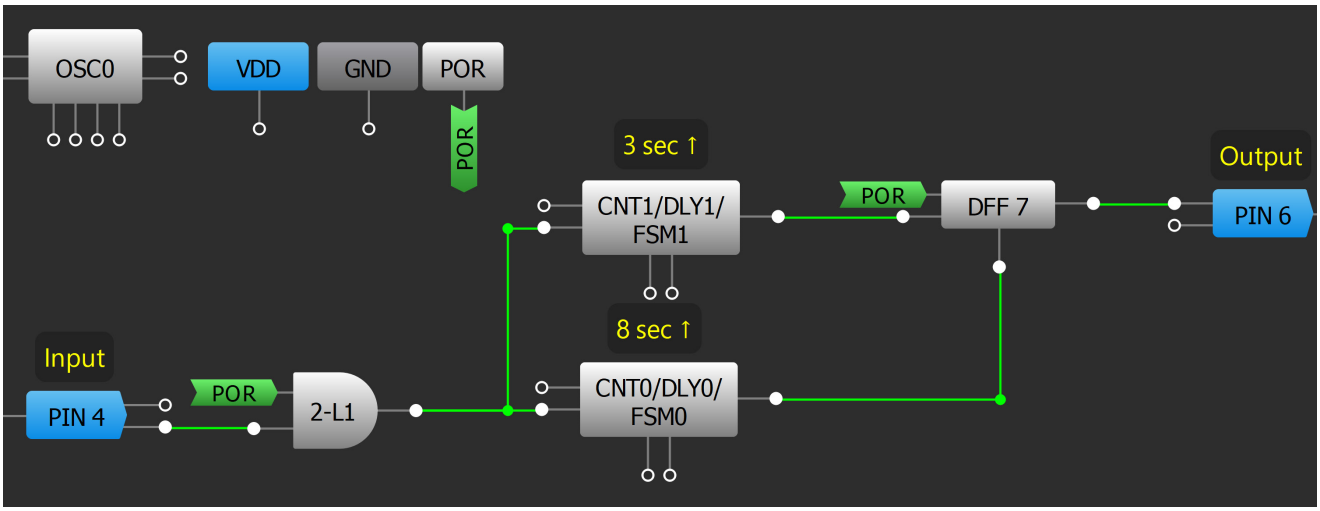
Using a single button to generate several events is a common solution to save on external control components. Using one button and predetermined time intervals you can organize control of an LED flashlight.



Ingredients

- Any GreenPAK
- One LED
- One resistor

GreenPAK Diagram



Design Steps

1. Configure GPIO pins as an input for button and an output for LED control.
2. Add CNT0/DLY0, CNT1/DLY1, and a DFF to remember the last state.
3. Configure the CNT/DLY blocks to “Delay” mode, with Edge select configured to “Rising.”
4. Configure the CNT0/DLY0 output as “nOUT.”

Chapter 3
Signal Conditioning

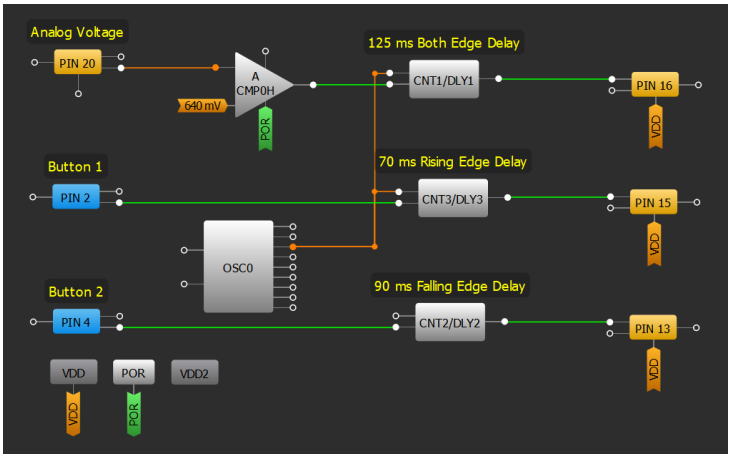
This chapter presents applications that interpret an external signal and condition it to be useful for an operation within a system. Some applications that involve this are frequency division/multiplication, filters, and sensor controllers.

Technique: Using a CNT/DLY Block as a Deglitch Filter

This technique can be used in any GreenPAK

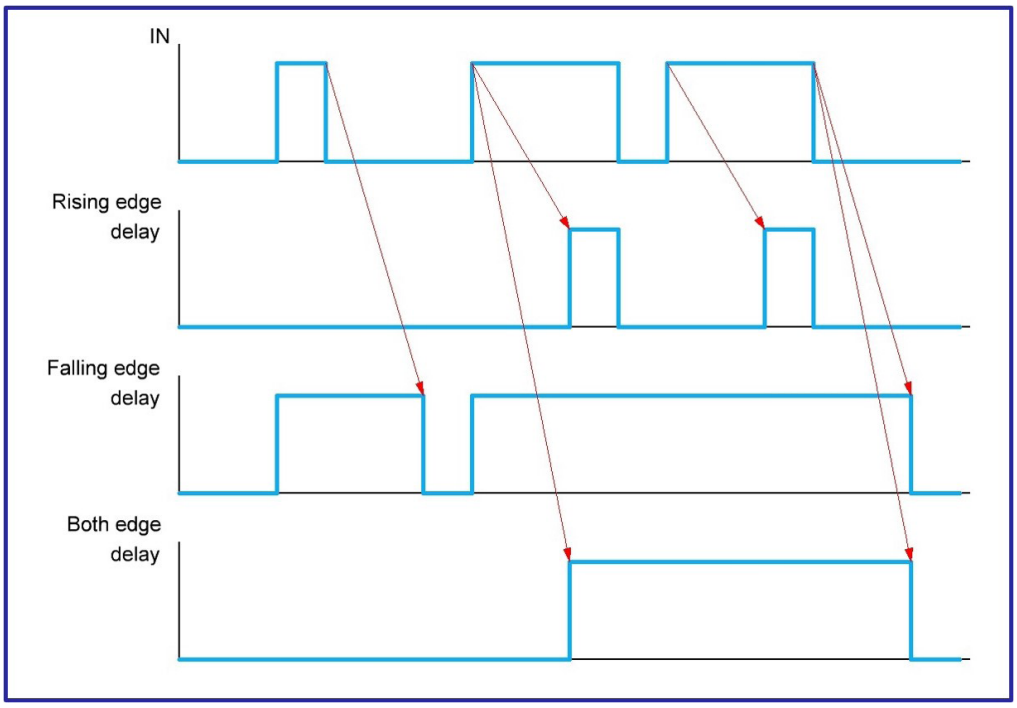
Deglitch / debounce filters are used to eliminate glitches - spurious signal transitions. Glitches occur in situations like a button being pressed/released or when a voltage is very close to a threshold of a hysteresis-less PIN.

There are 3 possible edge-triggered options to configure a deglitch delay: rising, falling or both. In this case delay block will filter pulses shorter then delay value with corresponding polarity: active high for rising edge, active low for falling edge, both High and Low for both edge delay.



Example Deglitch Delay Options

See diagram below:

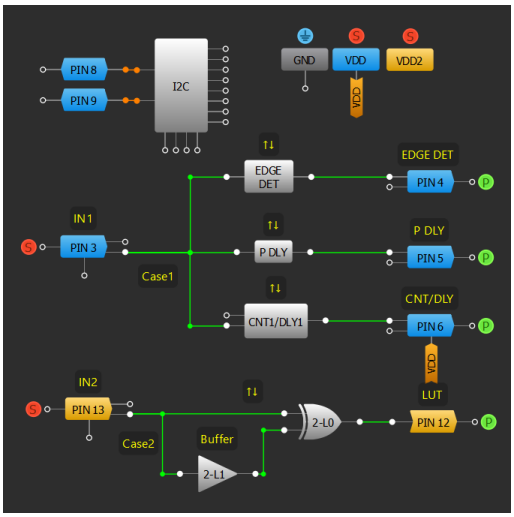


Edge Delay Behavior

Technique: Edge Detector

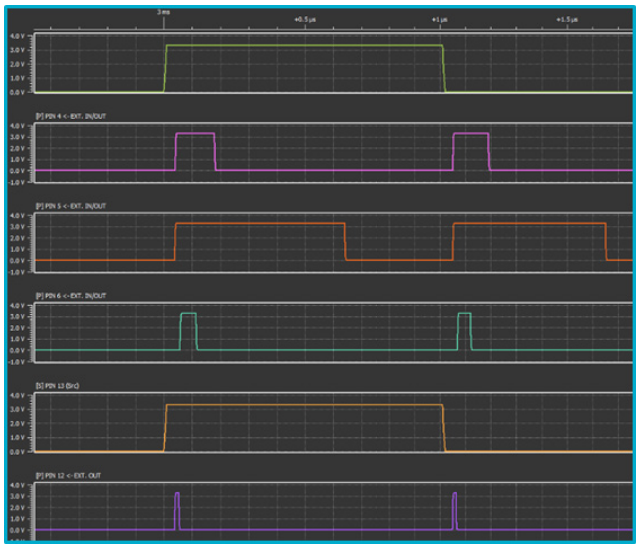
This technique can be used in any GreenPAK.

Edge detectors are important components in digital electronics. It is a simple circuit with one input and one output. Edge detectors create a short pulse when a defined edge (rising, falling, or both) is detected. It's useful for implementing a reset function, watchdog timer, or other edge-dependent applications. There are several ways to implement an edge detector (see figure below). To read a detailed description on how to build edge detectors and see more examples see AN-1046 Various Edge Detector Circuits.



Edge Detector Implementation

Option 1 uses the built-in edge detector feature in EDGE DET, P DLY, and CNT/DLY blocks. These blocks have the benefit of producing a longer duration pulse compared to Option 2 (figure below).

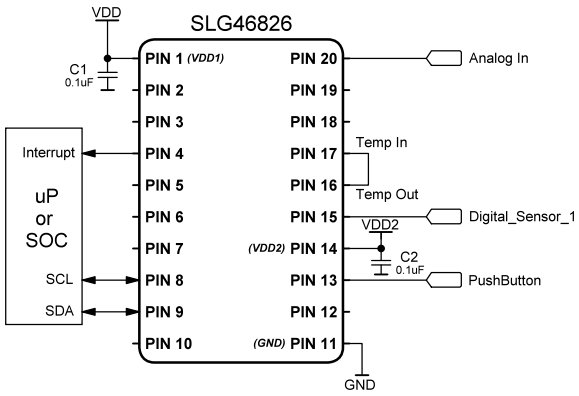


Different Edge Detectors Timings

Option 2 uses the small delay caused by signal propagation through a buffer. This delayed signal is compared by a 2-bit XOR to the original input, whose propagation time is exceptionally small. The short delay through the buffer causes a difference between the XOR's inputs, which generates a short pulse on its output. Because of the internal structure of the LUTs their inputs have different propagation delay time.

Application: Interrupt Controller

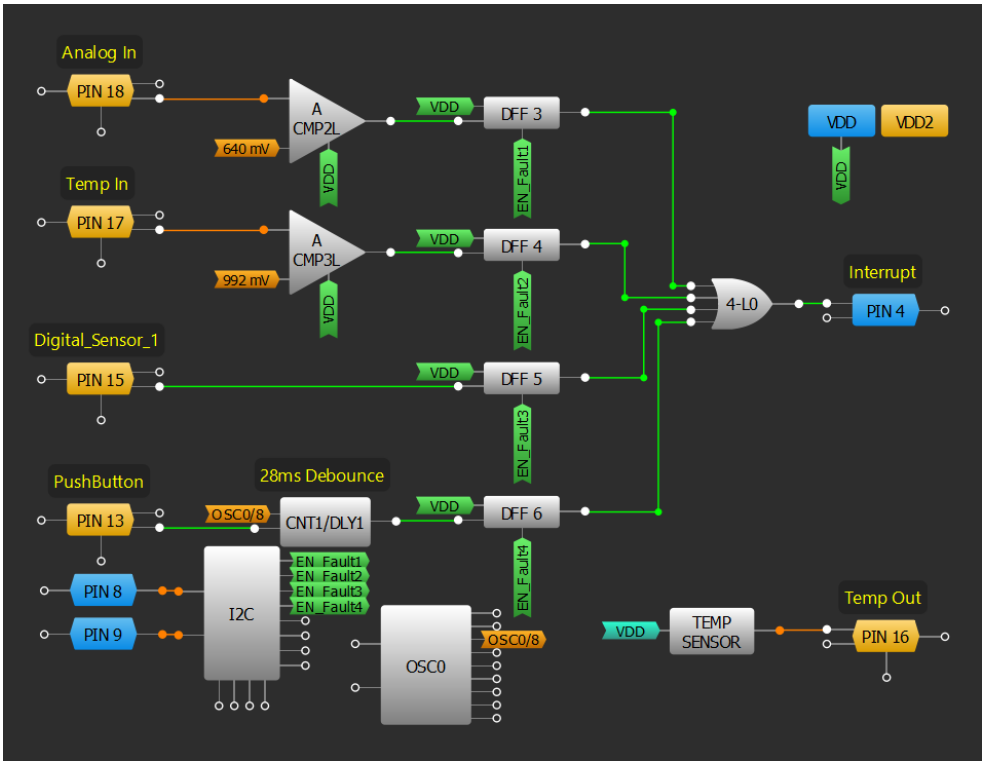
The GreenPAK can be configured to monitor multiple different interruptible signals and aggregate that information for the host processor to act on. The microprocessor or SOC can read the output of each DFF via I2C to determine the source of the fault.



Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram



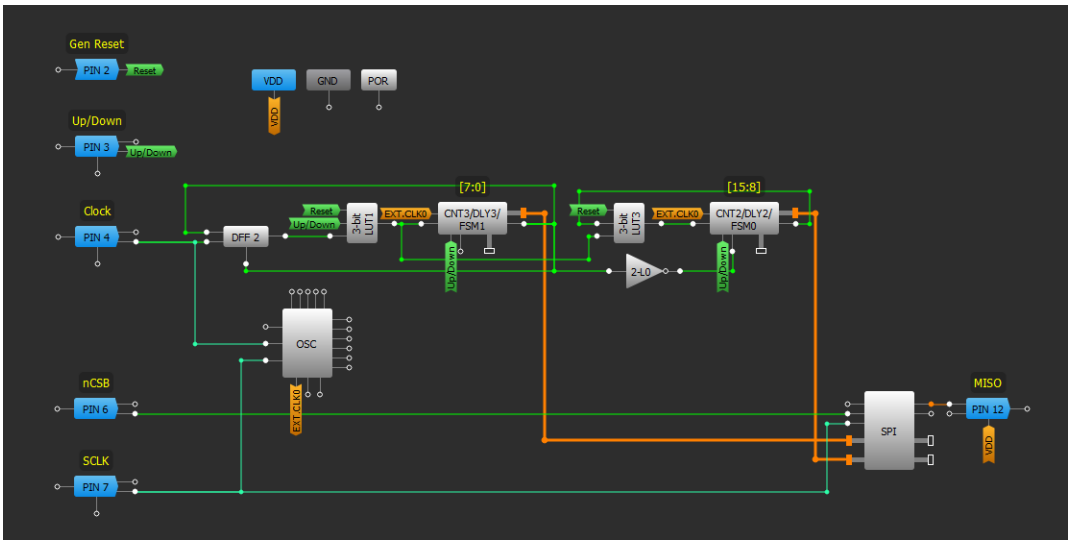
Design Steps

1. Configure PINs 16, 17, and 18 as “Analog Input/Output.”
2. Configure levels of ACMPs to proper threshold.
3. Wire DFFs to OR gate and set PIN4 as output.
4. Configure DLY1 to desired debounce time.

Technique: Creating a Bi-directional Counter

This technique can be used within any GreenPAK that includes an SPI interface. Alternatives to this technique can be accomplished using other GreenPAKs with an FSM block and storing the counter information using an I2C read command, parallel output, or other method.

A Counter is a basic digital circuit used for counting input events (pulses, edges), often constructed using a cascade of digital flip-flops. In GreenPAK some CNT/DLY blocks are more robust, and can be used as a finite state machine (FSM) that is not only capable of incrementing but can decrement or hold the current value, dictated by interconnects in the GreenPAK matrix. This technique exemplifies this behaviour by using two FSM blocks in GreenPAK to monitor a pulse input (Clock) and output the corresponding 16-bit sequence via the SPI macrocell.



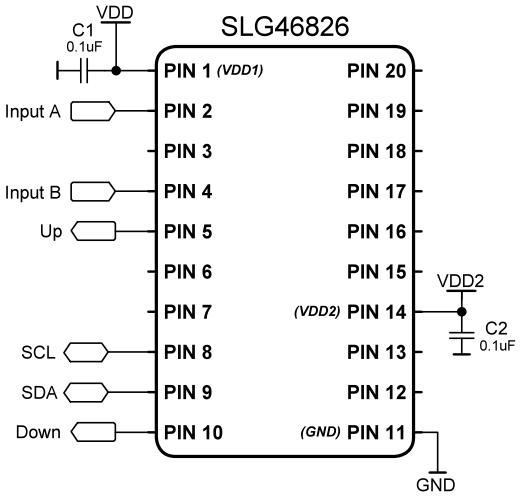
16-bit FSM with SPI Output

The 16-bit FSM with an output to the SPI block counts input clock pulses in a constructed 16-bit register (FSM0, FSM1). At any time a user can read the value via SPI, reset the 16-bit register, or change the count direction.

The 16-bit counter is implemented using two counters (FSM0 and FSM1 blocks) with additional logic. Bits [15:8] are stored in FSM0, [7:0] in FSM1. Both FSMs are connected to the SPI block, which can output serial data via SPI. The count direction is controlled by an Up/Down pin, directly connected to the FSM blocks' UP matrix output. If this pin is HIGH, the system counts UP, if this pin is LOW, the system counts DOWN. Gen Reset pin is used to reset both counter values (active HIGH).
The Clock input pin is applied simultaneously at the CLK input of FSM1 and FSM0. FSM1 counts each clock, whereas FSM0 counts only when FSM1 counter value is 255 and Up/Down signal is HIGH or when FSM1 counter value is 0 and Up/Down signal is LOW. This functionality is achieved using the KEEP input of the FSM0. When this signal is HIGH the counter value of the FSM0 is not changing despite the clock signal. KEEP is connected to FSM1 output through an inverter. In turn, FSM1's output is only HIGH when counter value is 0 and Up/Down signal is LOW, or when counter value is 255 and Up/Down signal is HIGH.

Application: Encoder

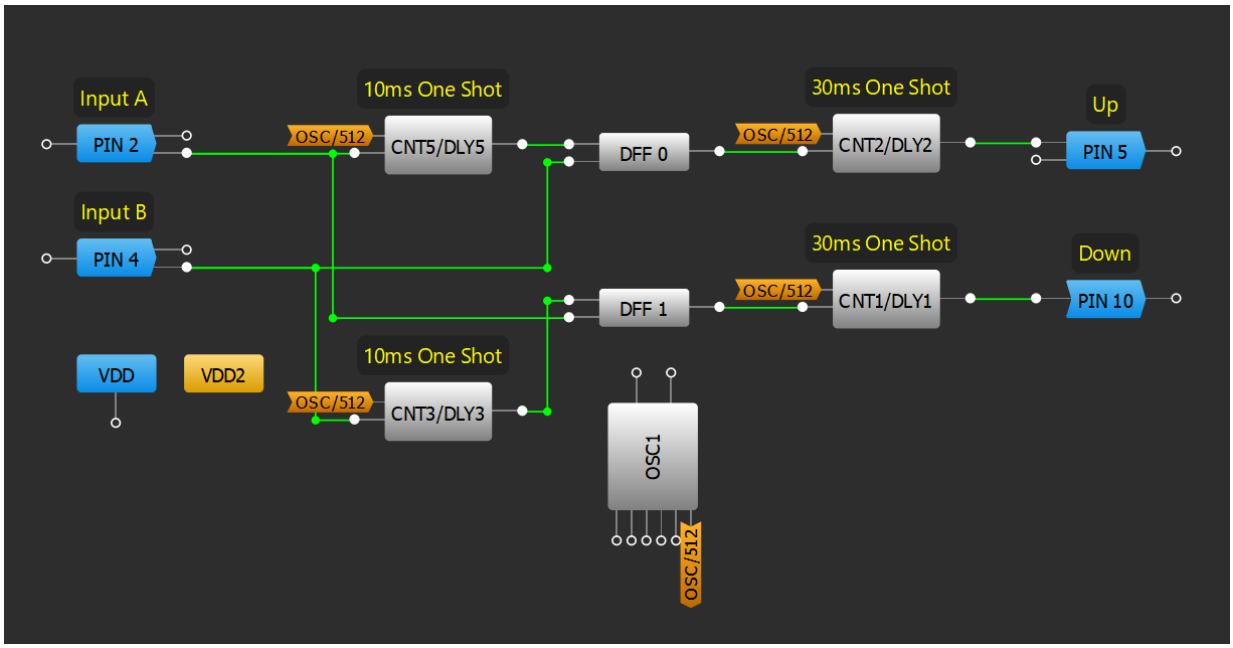
Encoders are used to convert rotary or linear motion to a digital signal. This design is optimized for mouse wheels and volume control in headsets.



Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram

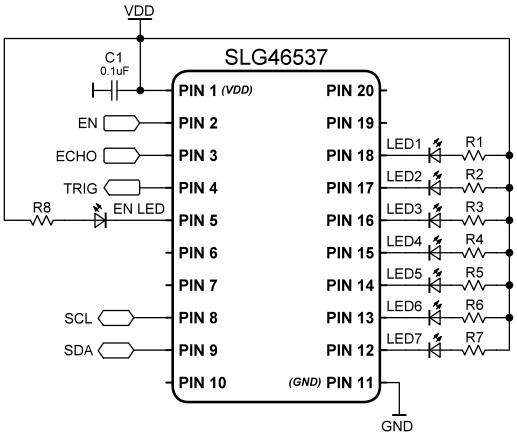


Design Steps

1. Configure pins as digital inputs.
2. Configure two pins as output to designate direction of encoder.
3. Set CNT3/DLY3 and CNT5/DLY5 to the “One shot” mode with the desired filter time.
4. Configure DFFs to detect direction (Up or Down).
5. Set CNT1/DLY1 and CNT2/DLY2 to the “One shot” mode with the desired output pulse width.

Application: Distance Sense

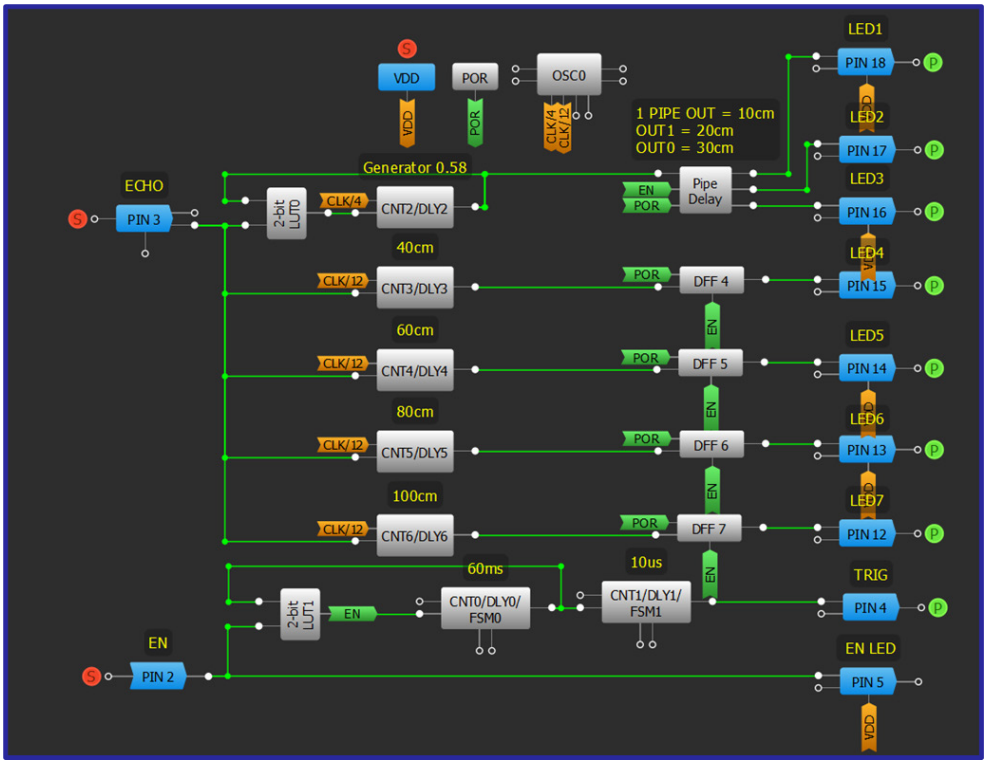
Ultrasonic ranging modules provide a non-contact measurement function. This design is a controller for an ultrasonic rangefinder based on the HC-SR04.



Ingredients

- Any GreenPAK
- LED for each distance measurement
- Resistor for each distance measurement

GreenPAK Diagram

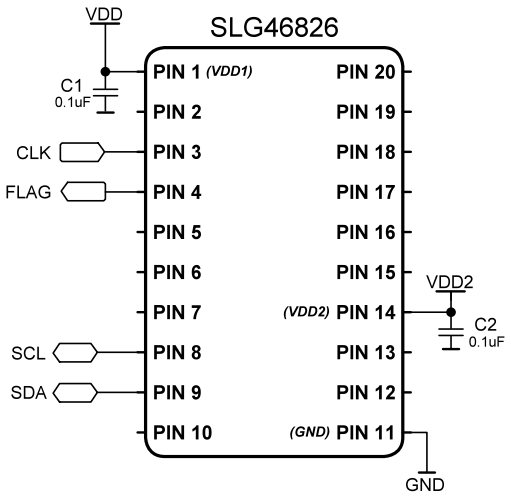


Design Steps

1. Configure GPIO input for Echo and output for Trig.
2. Add LUT logic and CNT/DLY0 to create a generator with ENABLE signal.
3. Add Pipe Delay and CNT/DLY2 to create a generator for detect distance.
4. Configure CNT/DLY blocks as a rising edge delay to measure varied distances.
5. Add and configure DFFs to latch distance data.
6. Connect each DFF output to the desired output pins and configure as open drain.

Application: Frequency Range Detector

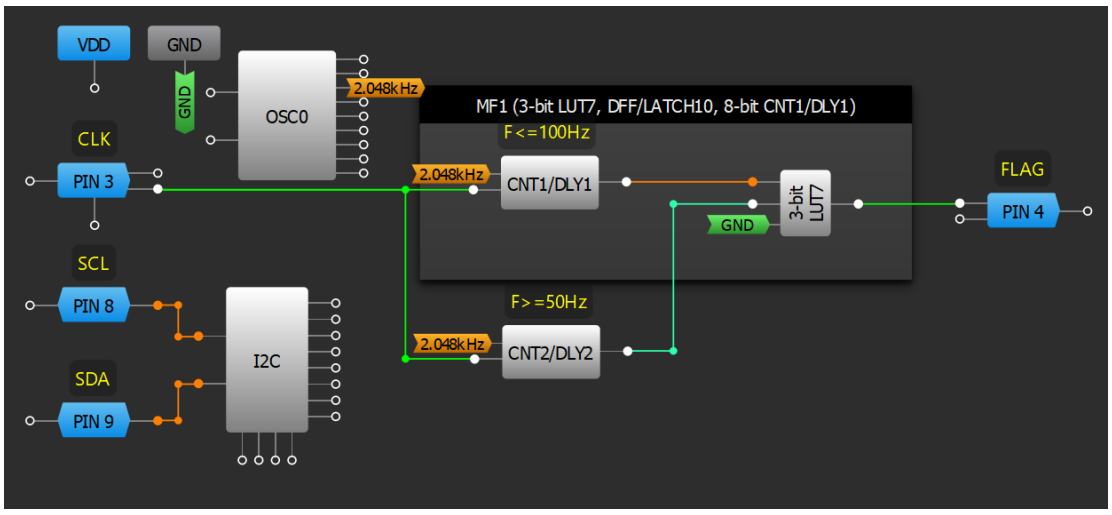
Many devices have a specific frequency range in which they operate and require the input clock to stay within this range. This application is used to detect if the input clock frequency is within the desired range.

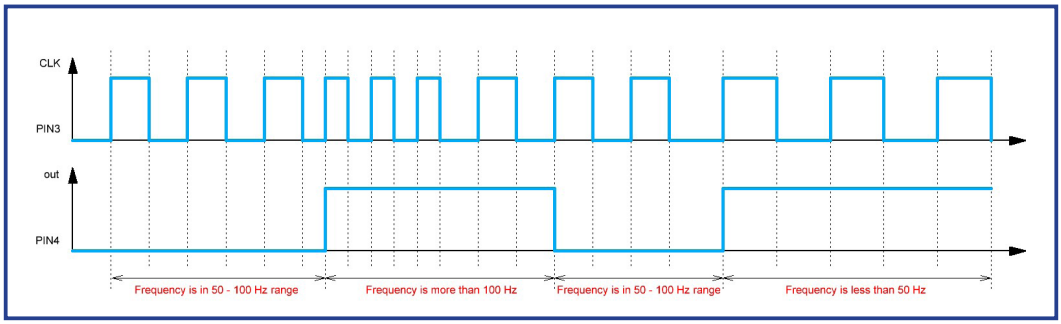


Ingredients

- Any GreenPAK

GreenPAK Diagram



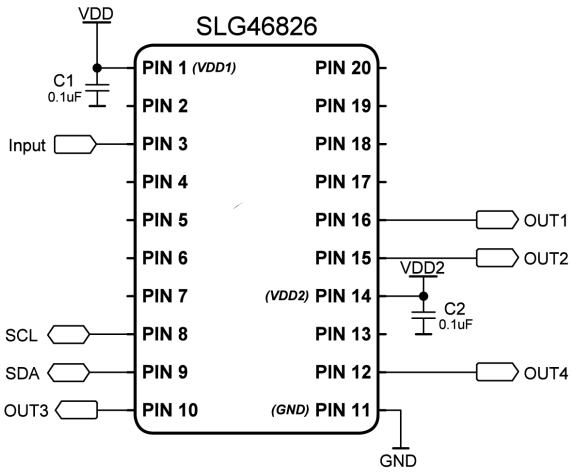


Design Steps

1. Configure GPIO pins as an input for the clock and output for the flag.
2. Configure CNT/DLY blocks to the "Frequency detect" mode with a rising edge detect.
3. Set each CNT/DLY block respectively to the minimum and maximum frequency values.
4. Configure a LUT to go high when the frequency is outside the desired frequency range.

Application: Frequency Divider

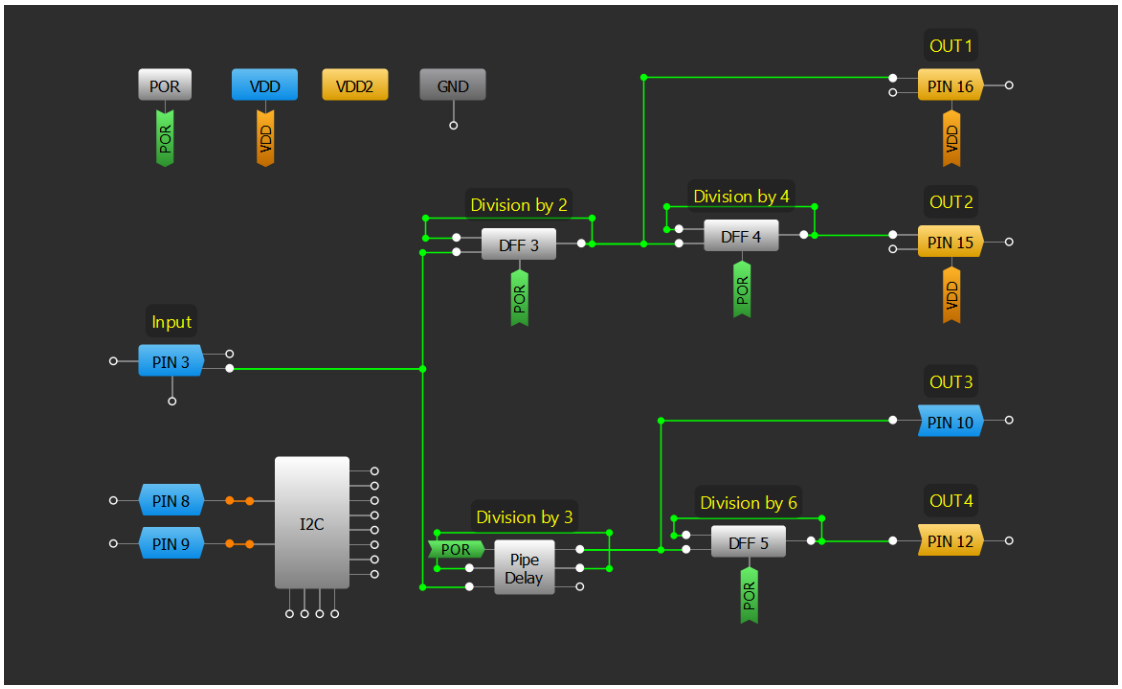
Frequency dividers are used to divide the input frequency into different coefficients. It can used for improving the performance of electronic countermeasure equipment, communication systems and laboratory instruments.



Ingredients

- Any GreenPAK
- No other components are needed

GreenPAK Diagram



Design Steps

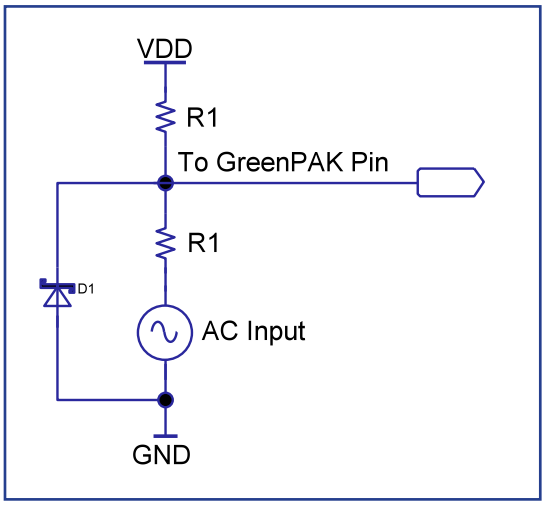
1. Configure inputs for each inputs signal (Input frequency, coefficient).
2. Use DFF and Pipe Delay to divide the frequency for the first stage.
3. Use another DFF to divide the signal by a factor for the second stage.
4. Configure LUT logic to decide outputs into specified coefficient.

Technique: Zero-Voltage Cross Detection

This technique can be used in GreenPAKs that include ACMP's.

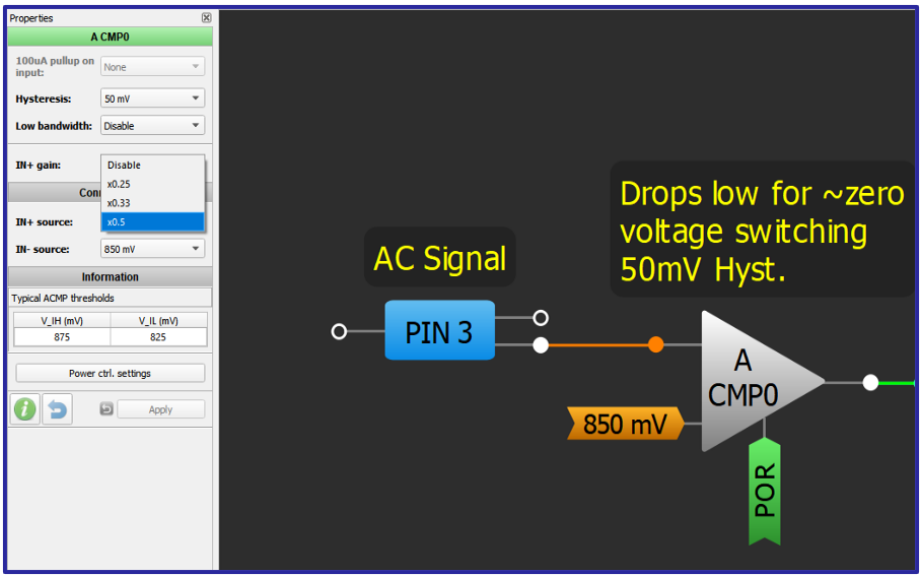
Zero-voltage cross detection is commonly used as an accurate method of detecting AC characteristics, such as frequency and phase.

GreenPAKs have a pin voltage range of 0V to a VDD value of 5.5V. To interpret an AC signal that crosses at the 0V point using the GreenPAK a DC offset shift should be implemented between the AC signal and the GreenPAK pin. This can be accomplished by a 1:1 resistor divider between VDD of the GreenPAK and the AC signal, shown in in the Basic DC Offset figure.



Basic DC Offset

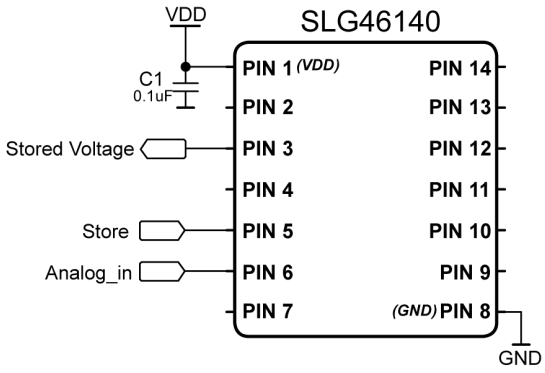
Zero-voltage cross detection requires, at minimum, one or two comparators and a counter. The comparators check the incoming AC signal against a reference voltage, which can either come from the GreenPAK's available reference voltages or an external reference point. If a desired ZVCD voltage is greater than the available GreenPAK reference voltages the AC signal may instead be reduced by using the IN+ gain option within the comparator's property settings and comparing the reduced AC value to a similarly scaled scaled reference.



Using the IN+ Gain to Measure a 1.7V Crossing

Application: Analog Storage Element

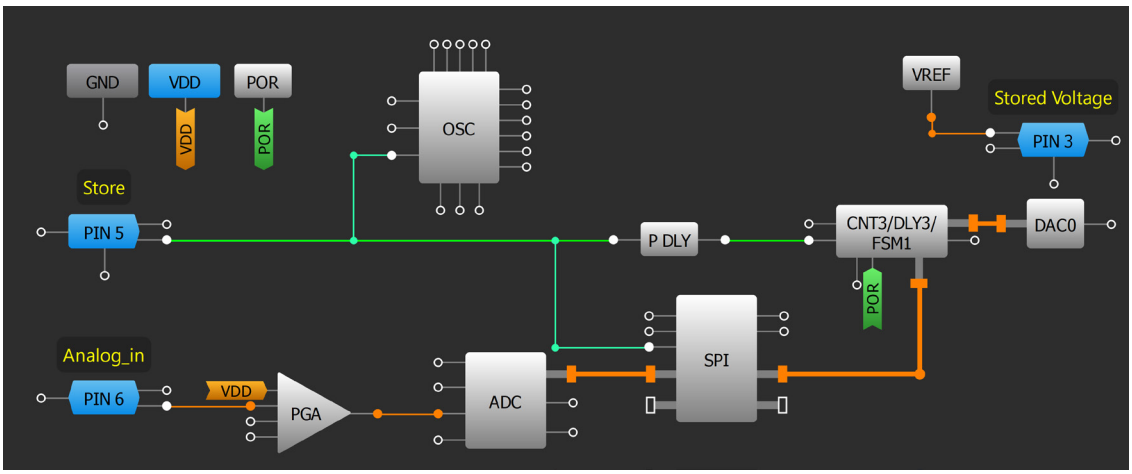
This application can be used to store the analog voltage on an output until the rising edge is applied to the Store input. The input and output analog voltages lie within the range of 0-1V.

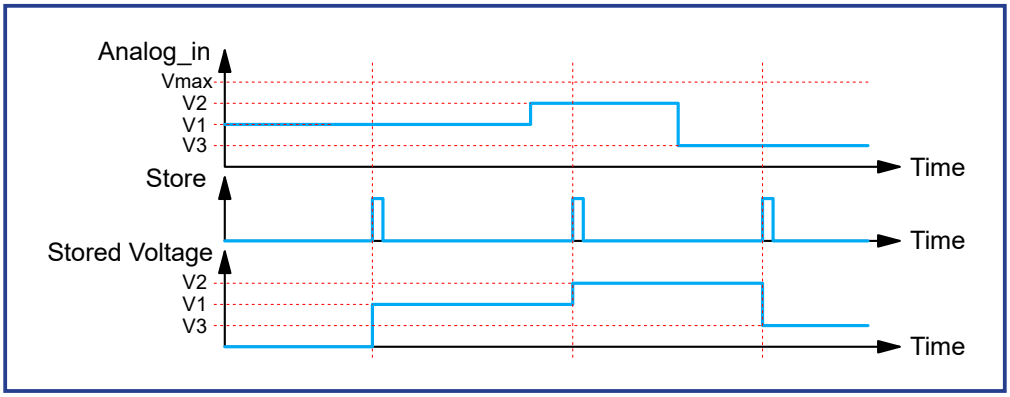


Ingredients

- Any GreenPAK with an ADC, SPI, and DAC

GreenPAK Diagram





Design Steps

1. Configure SPI to “ADC/FSM buffer” mode, change the PAR input data source to “ADC”.
2. Configure FSM0 to “Set (counter value = FSM data)” and change the FSM data source to “SPI [7:0].”
3. Configure DAC Input selection to “From DCMP1’s input” and VREF Source selector to “DAC0 out”.
4. Connect Store input directly to SPI SCLK and FSM1 SET IN through P DLY set as a both edge delay.

Chapter 4
Safety Features

This chapter presents applications that are intended to respond to fault conditions in a system and protect it from damage. Some applications that provide safety to electronic systems are battery indicators, watchdog timers, and temperature sensors.

Technique: Reducing ACMP Power Consumption

This technique can be used in GreenPAKs that include ACMP's. The reduction in power consumption will vary.

GreenPAKs are often used in projects to reduce the system's current consumption. However, several components within GreenPAKs, when active, can cause a noticeable change in current consumption. Amongst the most consumptive macrocells are the analog comparators. Table 1 is taken from the SLG46826 datasheet to highlight the ACMP's consumption.

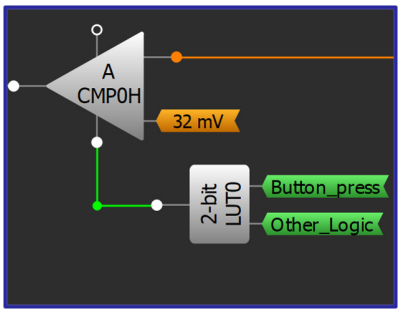
Luckily, ACMPs can be shut down when not in use. This is done in two ways:

1. Through the PWR UP input of the ACMP.
2. Enabling a wake-sleep controller (WS Ctrl) for the ACMP.

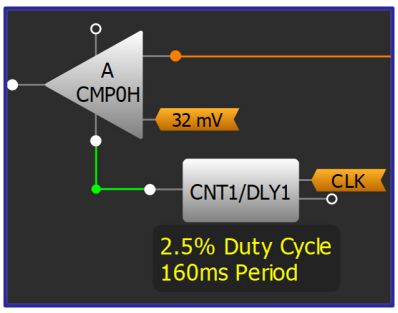
Note	V_{DD} = 2.5 V	V_{DD} = 3.3 V	V_{DD} = 5.0 V	Unit
Chip Quiescent	0.39	0.43	0.53	µA
Vref OUT0 (Source none, Source Temp Sensor, Buffer On)	12.79	12.95	13.57	µA
Vref OUT0 (Source none, Source Temp Sensor, Buffer Off)	7.62	7.67	7.87	µA
Vref OUT1 (Source none, Buffer On)	6.53	6.61	7.02	µA
Vref OUT1 (Source none, Buffer Off)	1.40	1.44	1.54	µA
Vref (ACMPxH, 0.32 mV, Buffer On)	12.24	12.59	12.21	µA
Vref (ACMPxL, 0.32 mV, Buffer On)	6.93	7.01	7.43	µA
ACMP0H, 1H, 2L, 3L,hysteresis disabled, gain = 1, +IN - IO11, 12, 13, 14 Pull Up 1M, Vref = 32 mV	65.86	67.12	70.77	µA
ACMP0H, 1H, 2L, 3L,hysteresis disabled, gain = 1, +IN - IO11, 12, 13, 14 Pull Down 1M, Vref = 32 mV	37.34	38.05	40.29	µA
ACMP0H, 1H, 100 µA disabled, hysteresis disabled, gain = 1, +IN - IO13, 14 Pull Up 1M	63.85	65.11	68.71	µA
ACMP0H, 1H, 100 µA disabled, hysteresis disabled, gain = 1, +IN - IO13, 14 Pull Down 1M	35.97	36.68	38.87	µA
ACMP0H, 100 µA disabled, hysteresis disabled, gain = 1, +IN - VDD, Vref = 32 mV	36.30	36.96	38.85	µA
ACMP0H, 100 µA enabled, hysteresis disabled, gain = 1, +IN - IO14 Pull Up 1M, Vref = 32 mV	46.77	47.31	49.23	µA
ACMP0H, 100 µA enabled, hysteresis disabled, gain = 1, +IN - IO14 Pull Down 1M, Vref = 32 mV	49.02	50.29	53.75	µA

Table 1 SLG46826 Current Consumption

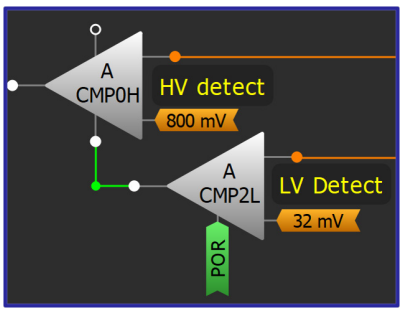
Wake-sleep control requires a dedicated counter configured to "Wake sleep controller" mode. This is available in many (but not all) GreenPAKs. PWR UP control can be used in any GreenPAK with ACMPs. When the signal is HIGH the ACMP is on; using logic, counters or other macrocells to turn off the ACMP can drastically reduce power consumption. For example, if two different voltage thresholds are needed the higher-threshold ACMP can be kept inactive until the lower threshold is met.



(a) Logic



(b) Duty Cycle



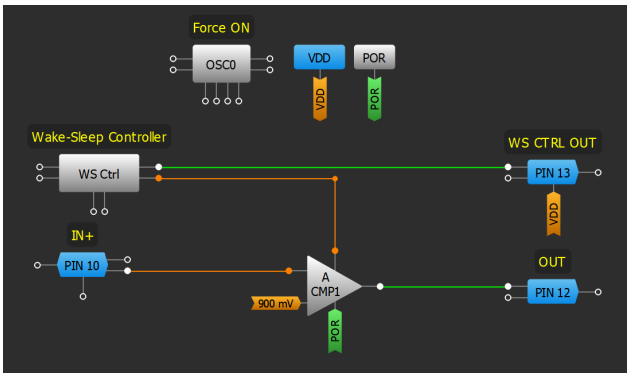
(c) Cascaded

Common PWR UP Configurations

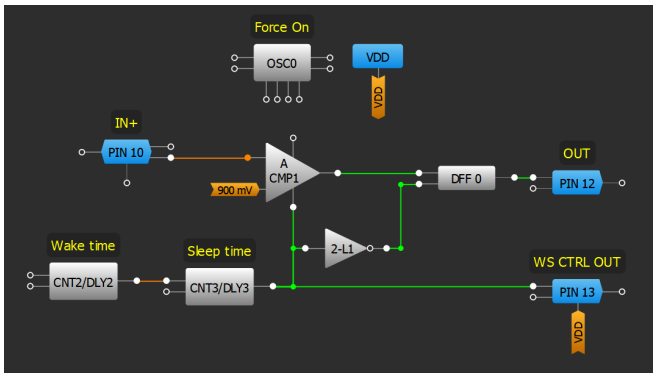
Technique: Wake-Sleep Controller

Waking and sleeping analog macrocells is useful for reducing power consumption. It is possible to accomplish this with the wake-sleep controller for analog macrocells like ACMPs and ADCs.

Wake-sleep involves switching analog macrocells on and off periodically. For some GreenPAKs this function can be implemented using the WS Ctrl block. For those that don't have this block, it can be implemented using two counters (one counter if there is no need to change the wake time), a D flip-flop, and an inverter. The figures below display examples using these respective methods.



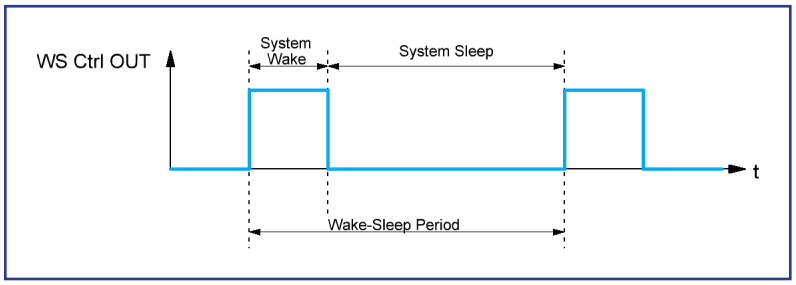
WS Ctrl Method



Two Counter Method

Without wake -sleep implemented, the total current consumption consists of:

- Quiescent current
- ACMP current



Behavior of Wake-Sleep

With wake-sleep implemented, the quiescent current is approximated as follows:

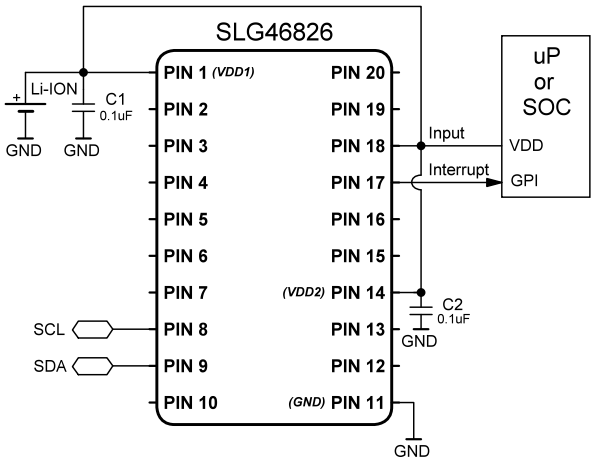
$$I_{WS} = \frac{System\ Wake}{System\ Wake + System\ Sleep} * I_{without\ WS} = \frac{System\ Wake}{WS\ Period} * I_{without\ WS}$$

The total current with wake-sleep implemented is:

$$Total\ Current = I_{Quiescent} + I_{OSC} + I_{Wake\ Sleep}$$

Application: Window Comparator

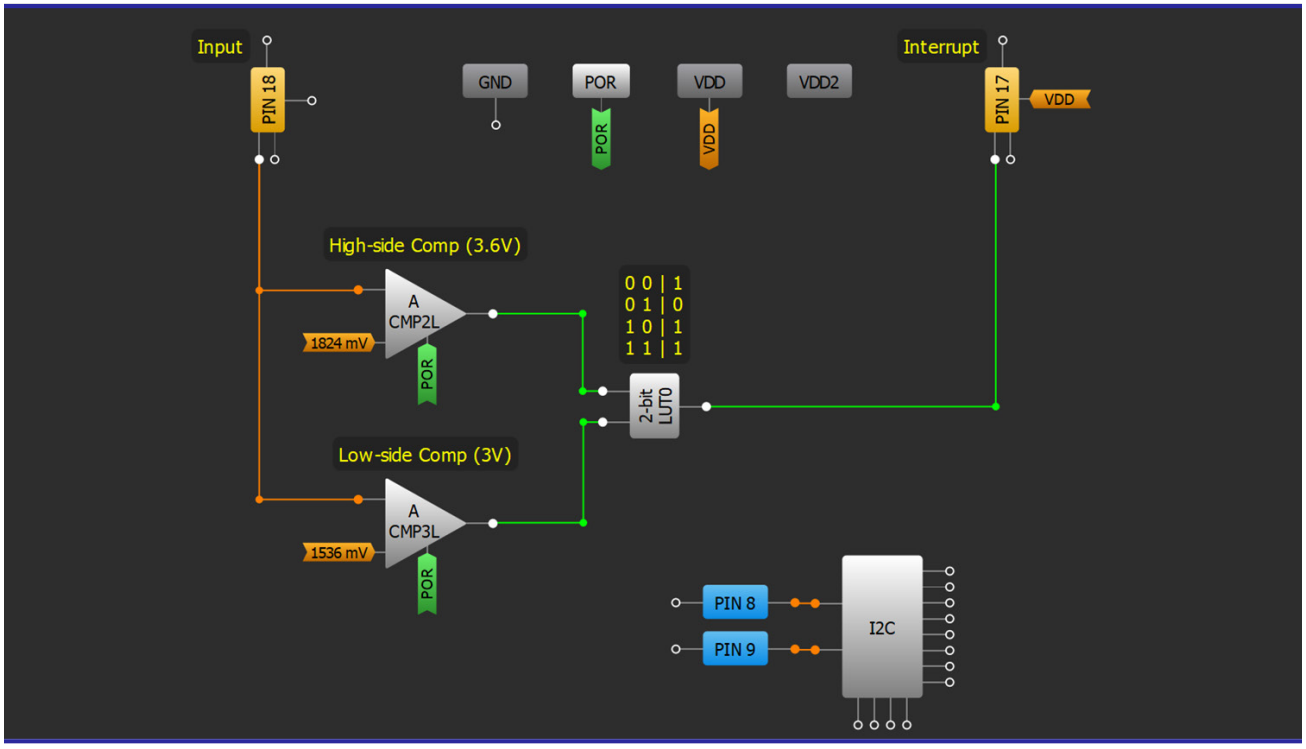
Window comparators are an essential part of any design that runs off a depletable power source, like a battery or supercapacitor. By monitoring battery voltage, a device can opt to stop using nonessential resources at low battery levels. This can prevent permanent damage to the device.



Ingredients

- Any GreenPAK w/ ACMP's
- No other components are needed

GreenPAK Diagram

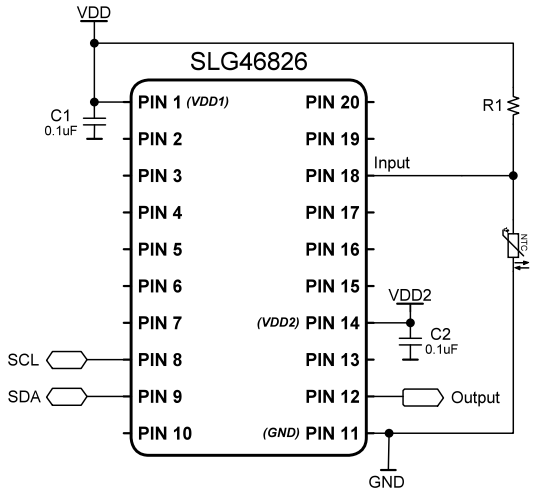


Design Steps

1. Configure the High-side ACMP2L by using the IN- source and IN+ gain options to set the desired high-side threshold.
2. Repeat step 1 for the low side ACMP with the low-side threshold.
3. Change the IN+ source for the second comparator to ACMP2L IN+ source.
4. Add the LUT logic to trigger an interrupt when the LOW-side comp is low or high-side is high.

Application: Over Temperature Protection

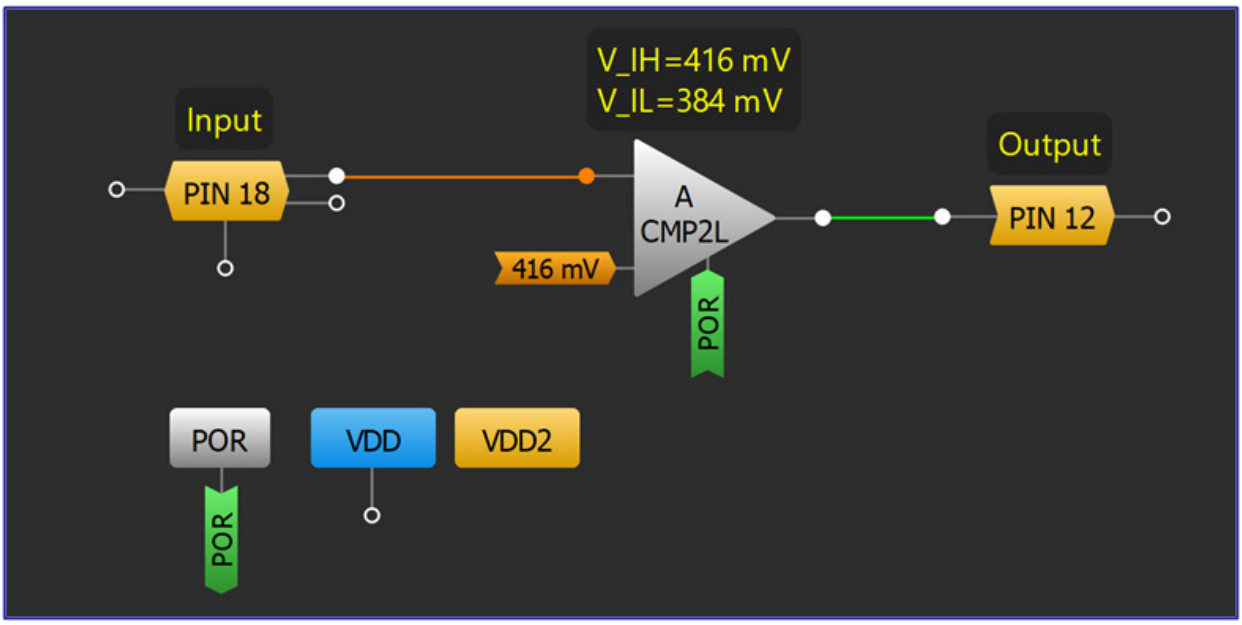
An over temperature protection circuit is widely used to alert a system of high temperatures. This circuit protects the system from overheating when the internal temperature exceeds a safe threshold.



Ingredients

- Any GreenPAK w/ ACMP's
- One resistor
- One NTC thermistor

GreenPAK Diagram

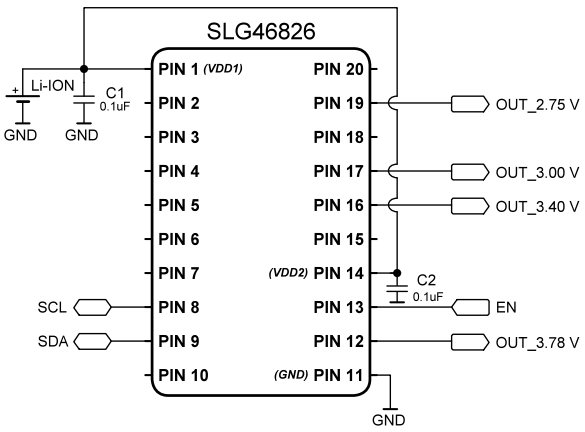


Design Steps

1. Set the IN+ source of ACMP2L to PIN18 and IN- source to the desired threshold.
2. Connect one node of the resistor to VDD and the second node to PIN18.
3. Connect one node of the NTC thermistor to PIN18 and the second node to GND.

Application: Battery Charge Indicator

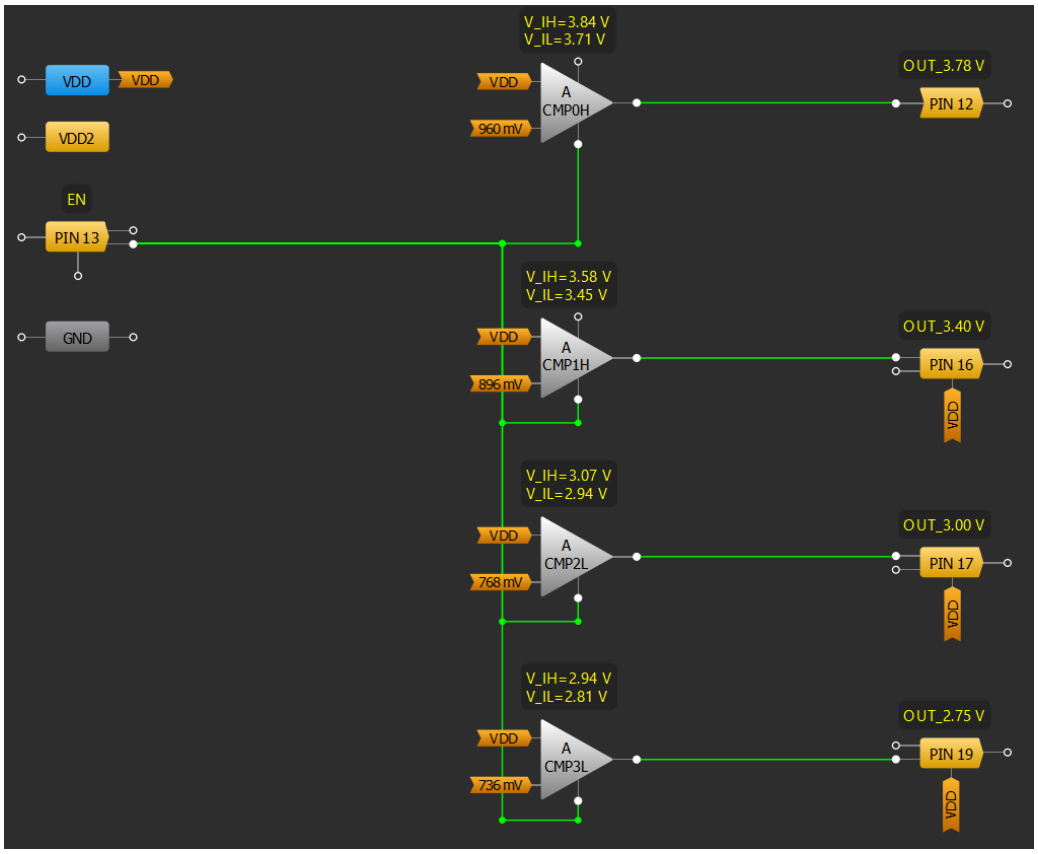
Battery charge indicators are used in battery-powered devices to indicate the state of charge. This design is optimized for a lithium-ion battery.



Ingredients

- Any GreenPAK w/ ACMP's
- No other components are needed

GreenPAK Diagram

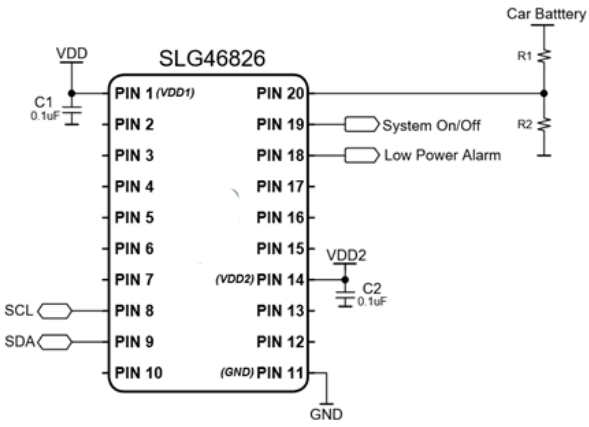


Design Steps

1. Connect pin to PWR UP pin of ACMP0H, ACMP1H, ACMP2L, and ACMP3L.
2. Set IN+ source of all the ACMPs to "VDD/PIN20" and each IN- source to the desired threshold level.

Application: Low Voltage Indicator for Infotainment

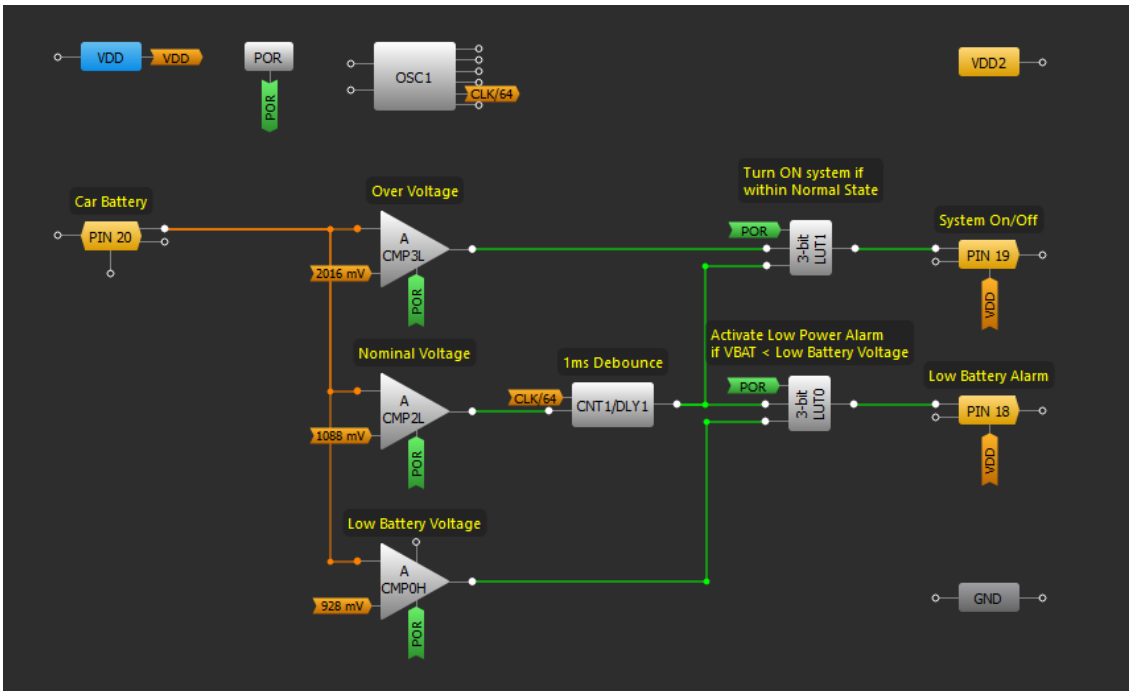
Voltage indicators are used in battery-powered devices to indicate the state of charge. This device monitors the voltage levels of a car battery and adjusts infotainment activities as needed to conserve power.



Ingredients

- Any GreenPAK
- Two resistors for voltage divider

GreenPAK Diagram

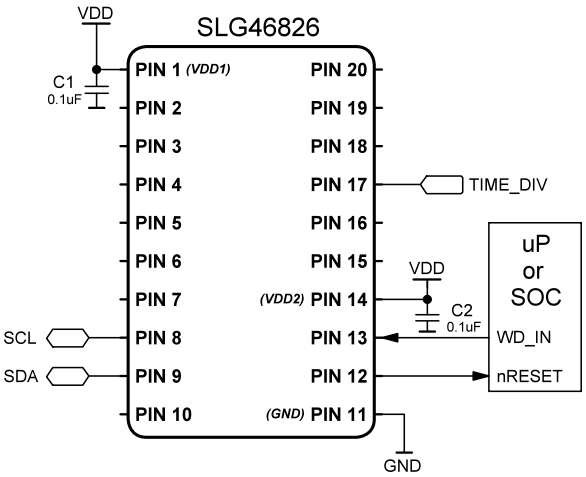


Design Steps

1. Configure the ACMP0H IN+ source as “PIN 20” and the other ACMPs as “ACMP0H IN+ source”.
2. Add a voltage divider on PIN 20 to handle high voltage from the car battery.
3. Configure the IN- source to the desired voltage threshold values.
4. Configure logic to determine the voltage level windows for the outputs.
5. Add a debounce delay in between ACMP2L and 3-bit LUT0.

Application: Watchdog Timer

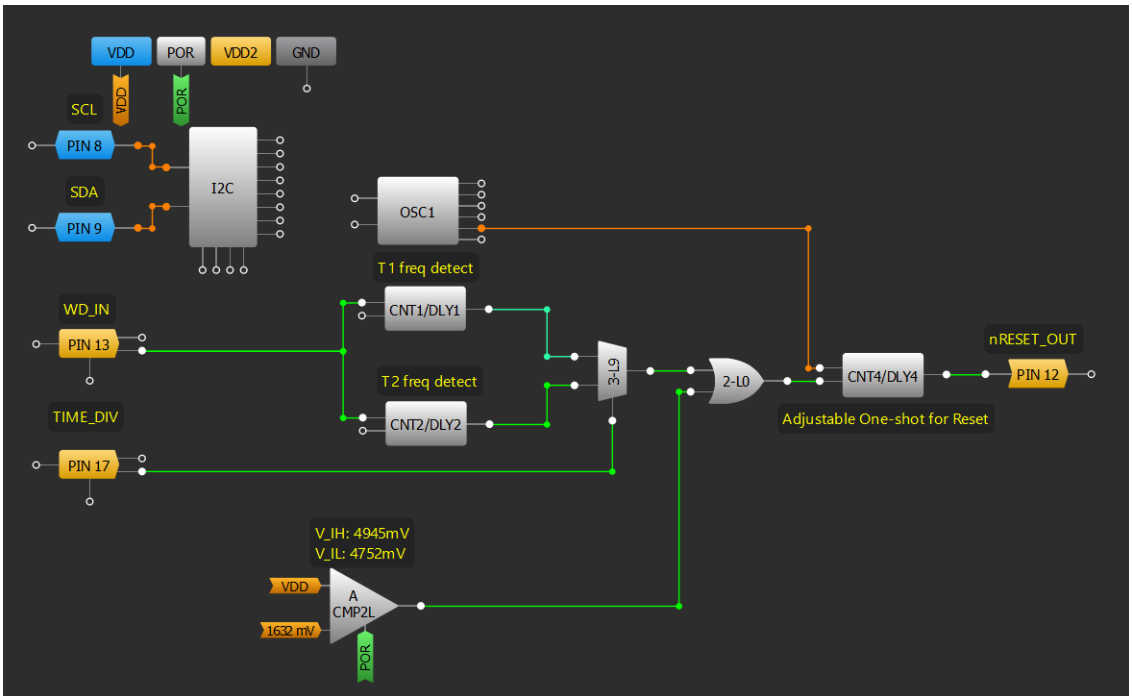
Watchdog timers are used for automatically generating a system reset signal if the microcontroller or microprocessor neglects to periodically send a pulse. Monitoring for low supply voltage is an additional, common feature of watchdog ICs.



Ingredients

- Any GreenPAK w/ ACMP's
- No other components are needed

GreenPAK Diagram

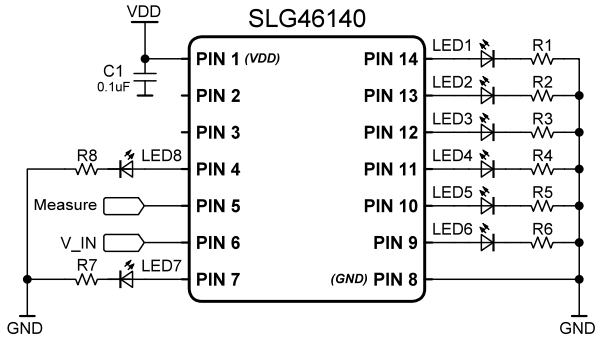


Design Steps

1. Set undervoltage threshold with an ACMP.
2. Configure two CNT/DLY blocks to the "Frequency detect" mode.
3. Design the digital logic to combine active signals from undervoltage and watchdog timeout.
4. Add a one shot block to trigger the reset pulse. It can be inverted to be active low.

Application: Voltage Level Detection

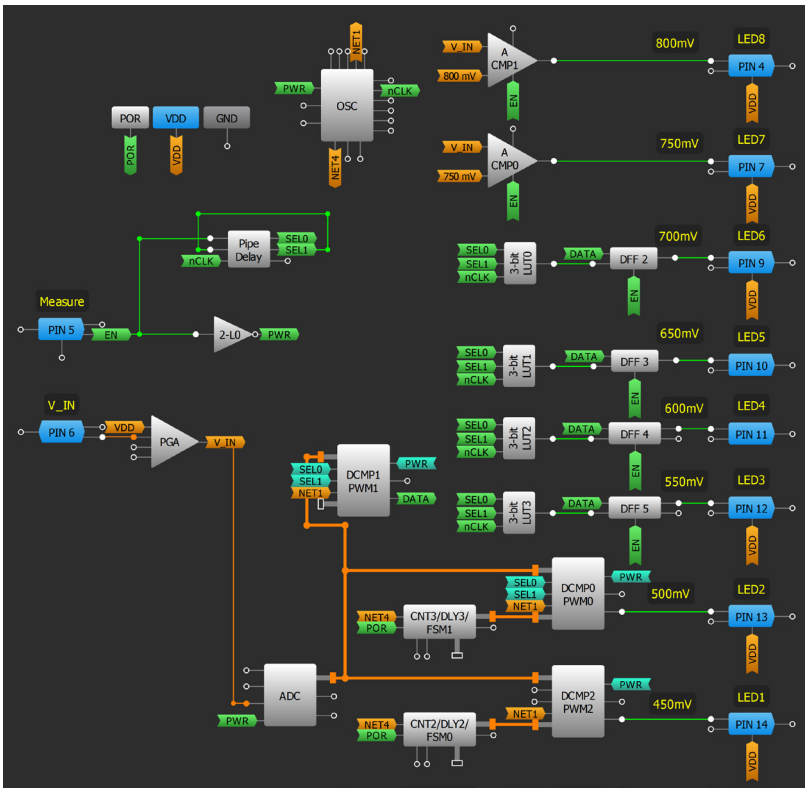
Some applications require multiple voltage levels to be evaluated, rather than a few distinct levels. This application shows how to use of ACMPs, DCMPs, and an ADC to monitor the voltage amplitude.



Ingredients

- Any GreenPAK with ACMPs, DCMPs, and an ADC
- Up to eight LEDs and resistors

GreenPAK Diagram

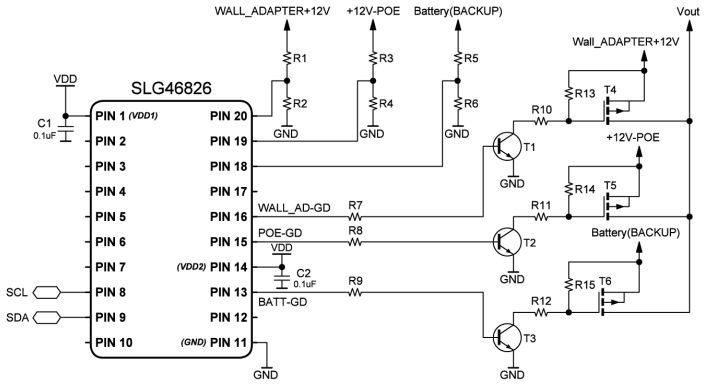


Design Steps

1. Power on the ADC, DCMPs, and ACMPs.
2. Configure the DCMPs using Technique: Using DCMP/PWM Macrocell in PWM Mode.
3. Set the IN- of each ACMP and DCMP with the desired voltage threshold levels.
4. Add LUT and DFF logic to select and write data of the amplitude of the analog voltage from DCMP1.

Application: Power Backup Management

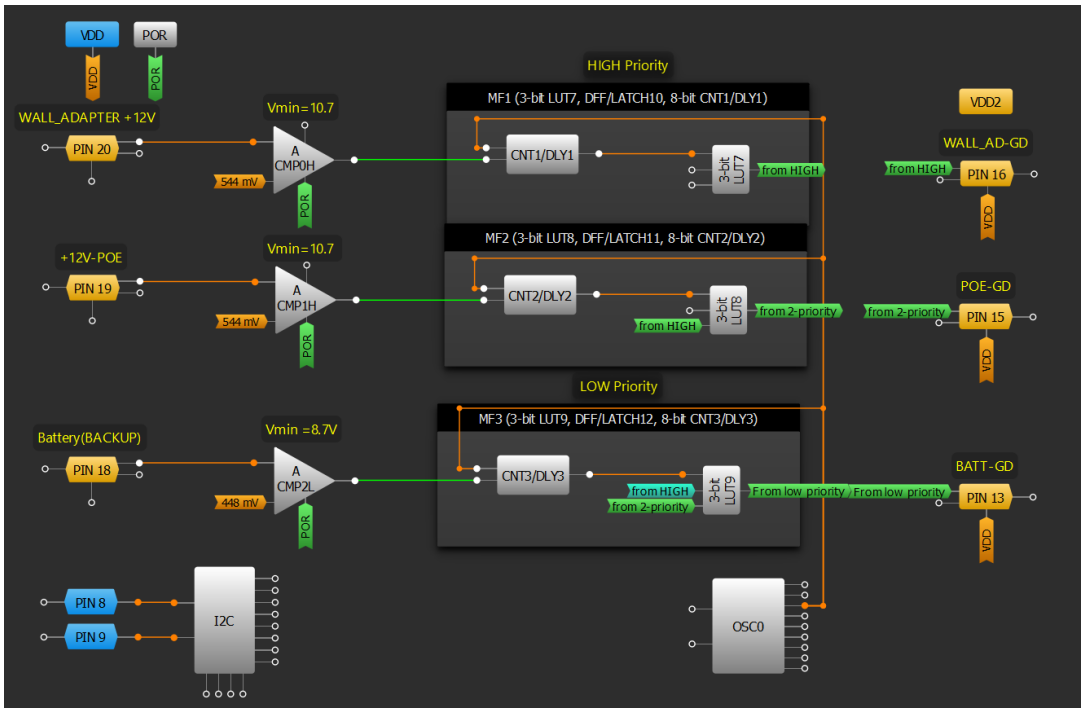
Power Backup management is used when a designer must guarantee the non-interrupted power supply of a system from different sources.



Ingredients

- Any GreenPAK with three ACMPs.
- External resistor dividers to attenuate input signal to the operating value range of ACMPs.

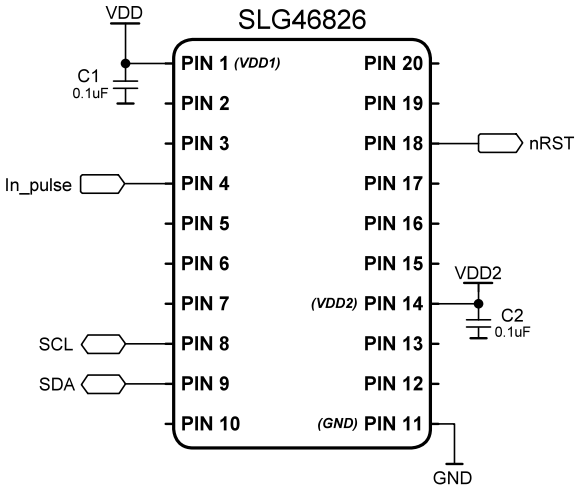
GreenPAK Diagram



Design Steps

1. Use three ACMPs to detect power input signals.
2. Use CNT/DLY blocks configured as a delay to implement a debounce filter.
3. Add logic cells to create switching priority between input sources.

Application: N-pulse Presence Watchdog

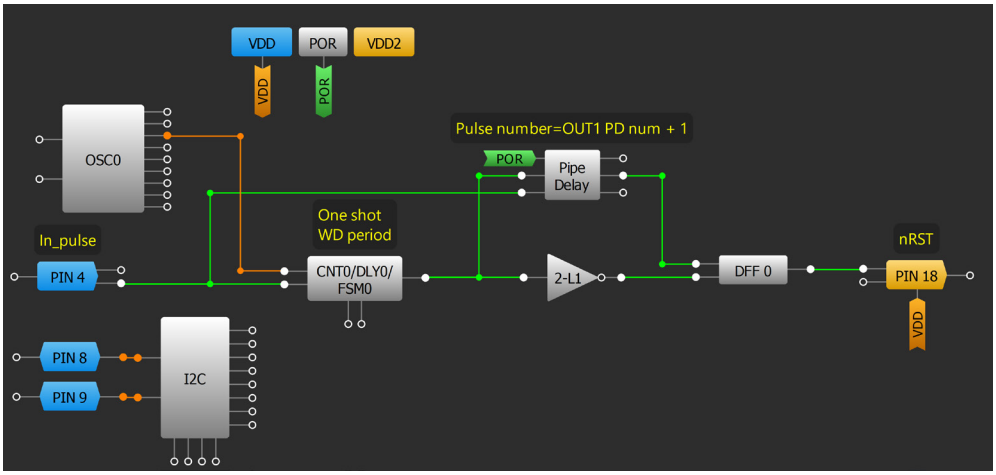


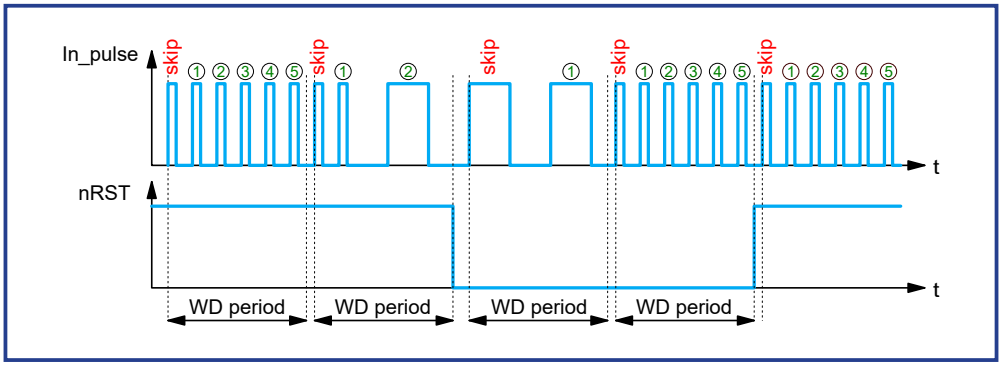
Watchdog timers are used for automatically generating a system reset signal if the microcontroller or microprocessor neglects to periodically send a pulse. This application monitors the number of pulses that go to the GreenPAK during a watchdog period. If the number is less than the predefined pulse number, a system reset will be triggered.

Ingredients

- Any GreenPAK

GreenPAK Diagram





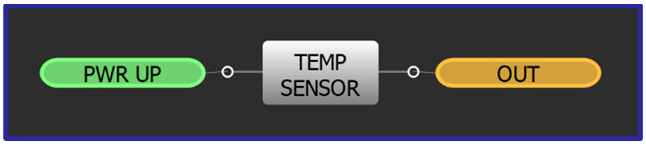
Design Steps

1. Configure CNT0/DLY0/FSM0 as a one shot with the desired watchdog period.
2. Define the pulse number in the Pipe Delay (Note: Pulse number = OUT1 PD num + 1).
3. Invert the output of the one shot and connect it to the CLK input of DFF0.
4. Connect nOUT1 of the Pipe Delay to the D input of DFF0.

Technique: Using the Temperature Sensor Block

This technique can be used with any GreenPAK that has a Temperature Sensor macrocell inside.

Some ICs have an analog Temperature Sensor (TS) with an output voltage linearly proportional to the Centigrade temperature. The TS is rated to operate in a temperature range of -40°C to 85°C. The error in the whole temperature range does not exceed ±0.85%. The TS output can be connected directly to the Analog Output or to the ACMP positive input. The TS may have two output voltage ranges and a Power Up input. The Power Up optionally can be activated using the matrix input or from the register. The TS can also be activated and the range can be changed via I2C. The TS output voltage at a constant temperature has very low variations over VDD changes (for example, in the SLG46826, the output voltage error is less than ±0.08% at all temperatures).

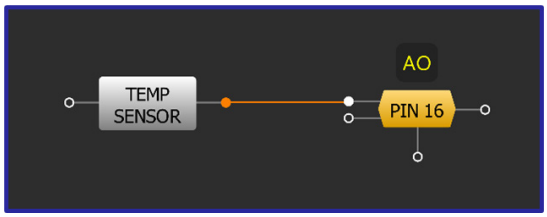


Temperature Sensor Macrocell

The TS output voltage can be calculated using the following formula:

$$Vts = K \times T + V0$$

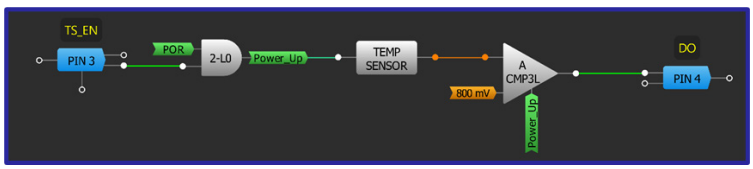
Where:
Vts - TS Output Voltage;
K - Coefficient;
T - Temperature in °C;
V0 - Output Voltage at 0 °C.



Temperature Sensor Connected to Analog

The temperature proportional voltage signal can be applied to the Analog Output (PIN16). The Power down source configuration should be set to “From register”.

Temperature Sensor output signal can be compared to the reference voltage in the ACMP block (See the figure below), which generates a twostate signal at the discrete output. In order to decrease the power consumption TS_EN is used. TS_EN enables the Temperature Sensor and switches the ACMP3L on. Power down source should be set to “From matrix.”



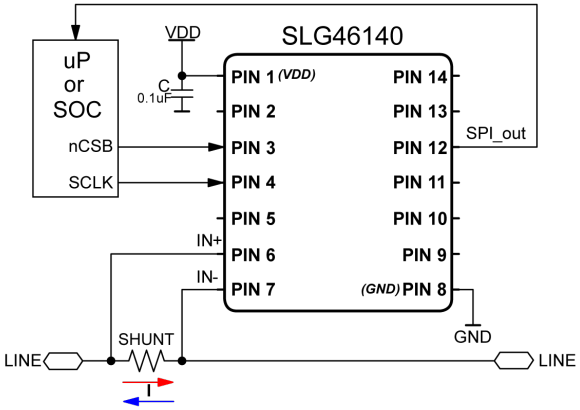
Temperature Comparator

GreenPAK with TS can:

- measure PCB components temperature
- measure FET or BJT case temperature
- create an alarm signal for SoC or in closed-loop applications
- minimize errors for ADCs, DACs, OpAmps and other temperature dependent schematics

Application: Current Detection Through External Sense Resistor

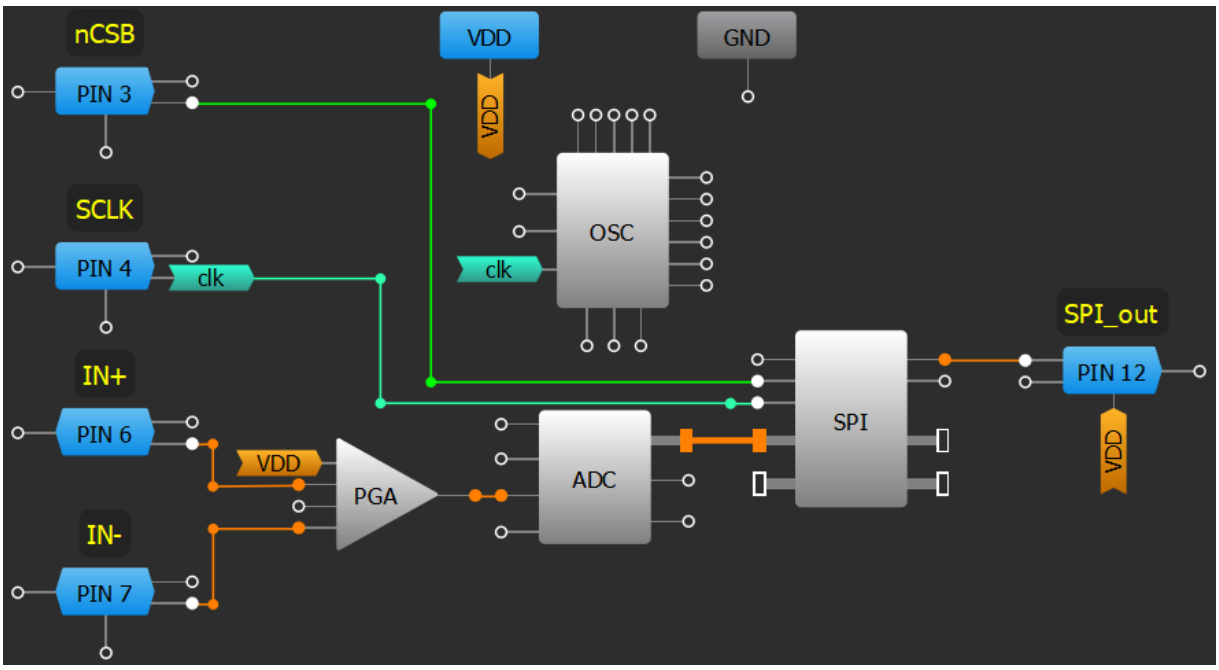
The GreenPAK can be used to sense the current going through a device by sensing the voltage across a sense resistor. This application outputs a serial code to represent the value it has sensed.



Ingredients

- Any GreenPAK with a PGA, ADC, and SPI
- One resistor

GreenPAK Diagram

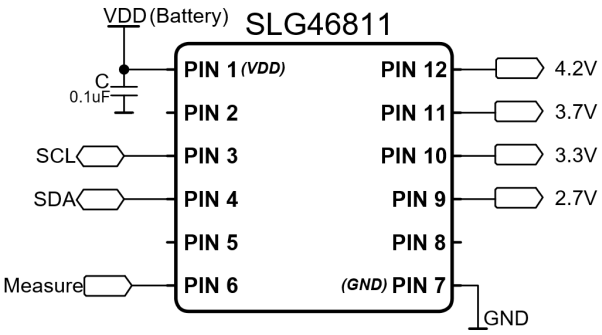


Design Steps

1. Power up the ADC by removing VDD from the PWR DOWN input.
2. Configure the PGA to “Differential” mode. It will automatically connect to the ADC, PIN6, and PIN7.
3. Configure SPI to the “P2S” mode and change the PAR input data source to “ADC.”

Application: Monitor Four Levels for One Analog Signal With One MS ACMP

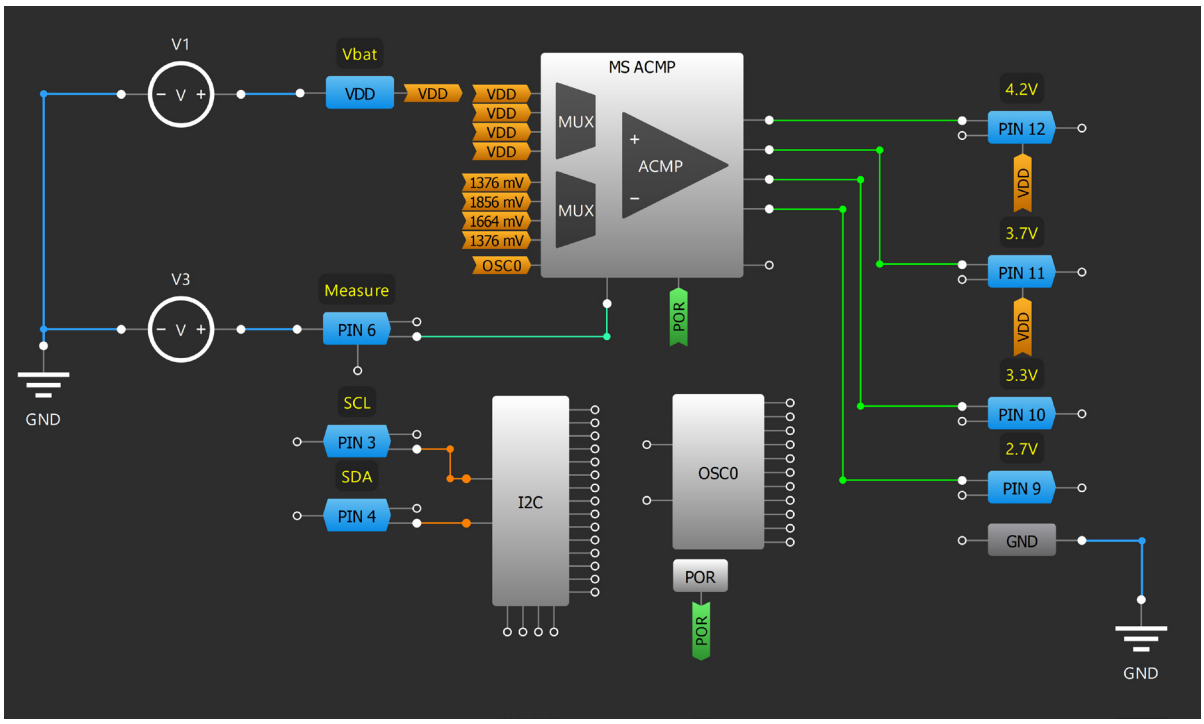
Monitoring four levels for one analog signal with a GreenPAK can be useful in a variety of applications. For instance, it can be used for battery management, fluid level control, temperature, light, proximity, pressure, humidity sensing, etc



Ingredients

- SLG46811V or any GPAK with appropriate number of ACMPs

GreenPAK Diagram

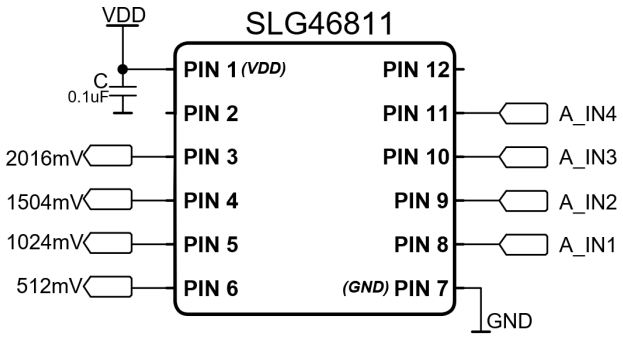


Design Steps

1. Configure MS ACMP to Multi-channel mode and choose 4-channels.
2. Choose Rising Edge Activation to Enable MS ACMP.
3. Adjust the IN- source for Channel 0 – Channel 3.
4. Configure PIN6 as digital input and connect it to Enable input of MS ACMP. By every rising edge applied to PIN6 MS ACMP will measure the VDD voltage output the result to PIN9-PIN12.
5. I2C can rewrite the ACMPs' threshold values.

Application: Monitor Four Separate Analog Signals With One MS ACMP

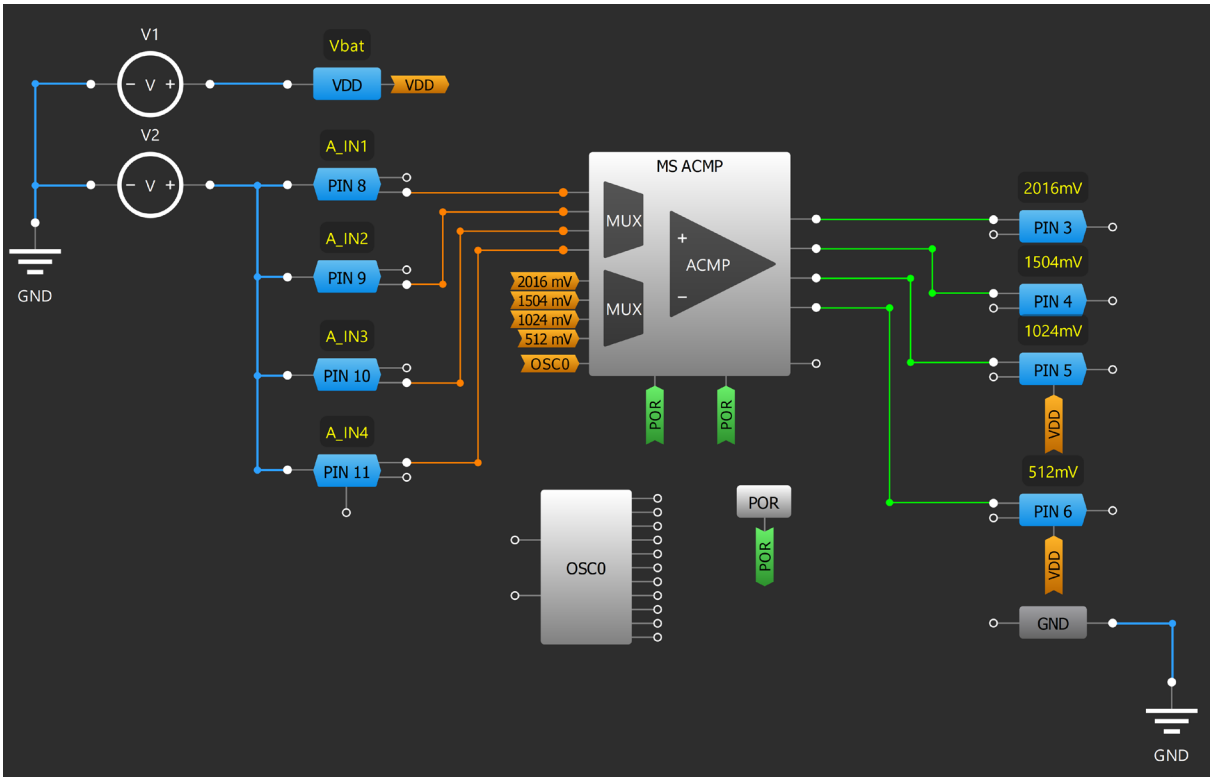
Monitoring four separate analog signals with a GreenPAK can be useful in a number applications, where four independent voltage signals change slowly.



Ingredients

- SLG46811V or any GPAK with appropriate number of ACMPs

GreenPAK Diagram



Design Steps

1. Configure MS ACMP to Multi-channel mode and choose 4 channels.
2. Choose High-Level Activation and connect POR (MS ACMP will be continuously sampling) to ENABLE input of MS ACMP.
3. Adjust the IN-source for Channel 0 – Channel 3. The voltage at PIN8 - PIN11 is compared to the reference, and the output to PIN3-PIN6 accordingly.
4. I2C can rewrite the ACMPs' threshold values.

Chapter 5 Communication Protocols

This chapter presents applications that involve communication between devices. The following applications and techniques involve I2C, serial, parallel communication protocols.

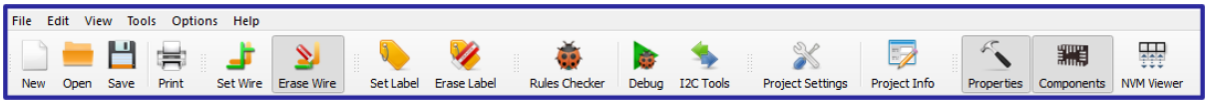
Many of the techniques and applications available in this section rely upon a GreenPAK's I2C capability. To learn about I2C within a GreenPAK please consult the chip's Datasheet.

Technique: Changing Your Design with I2C

This technique can be used in any I2C compatible device.

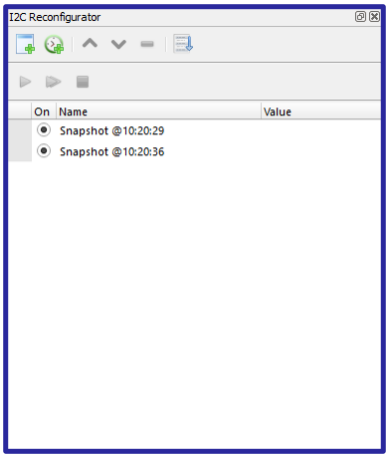
If a GreenPAK device is I2C compatible its behavior can be edited even after it has been programmed. However, a device must be MTP-compatible and undergo In-System Programming [ISP] to retain design changes after it has lost power. This technique outlines a fast way to determine which I2C commands need to be performed to change a design.

1. Complete your initial design. This is the design the IC will use whenever it's booted up.
2. In GreenPAK Designer, select the I2C Tools Button to open the I2C Reconfigurator.

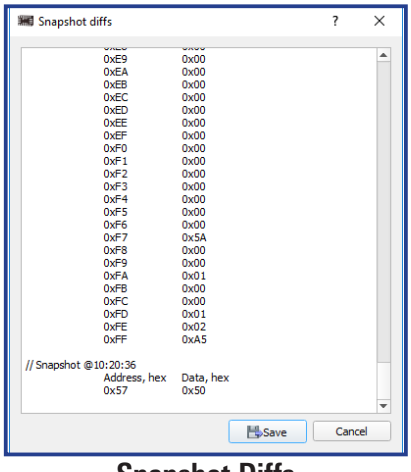


I2C Tools Button

3. In the I2C Reconfigurator select the snapshot button (boxed in red), or press SHIFT+A. This will take an I2C commandlist "snapshot" of your current design.
4. Change your design to the next configuration.
5. Take a snapshot using the method in step 3 to create the second snapshot.
6. Click the Snapshot diffs button (boxed in green). This will show the I2C commands necessary to create this design. It is not necessary to program these, since they are instantiated on the boot-up of the GreenPAK.
7. Scroll down the Snapshot diffs list, where you will find the second snapshot. This will show only the values that have changed between the first and second snapshot.
8. These values, shown in the red box on the right, correlate to the hexadecimal address and data value that need to be sent in I2C to change the threshold value of the ACMP (or any other change that may occur).



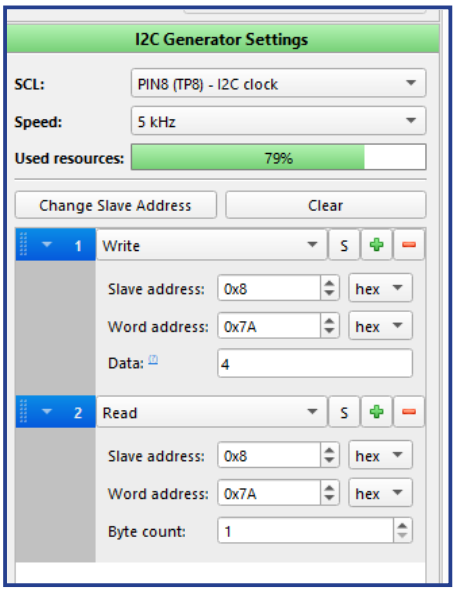
I2C Reconfigurator



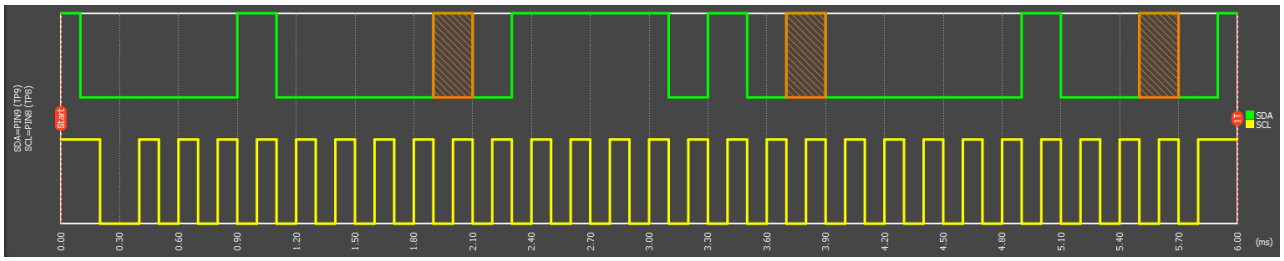
Snapshot Diffs

Technique: Creating an I2C command

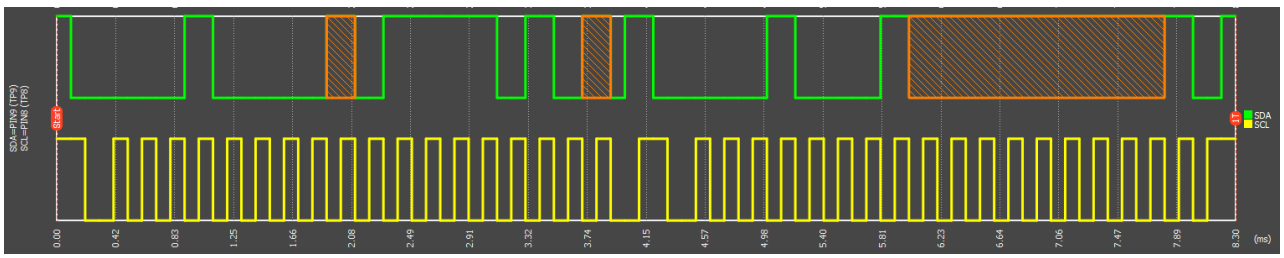
This technique can be used in any GreenPAK with an I2C macrocell With the I2C generator a user can create an I2C signal based on logic generators. It consists of two logic generators acting as SDA and SCL lines. The user can combine predefined I2C primitives to generate the needed waveform and choose an SCL frequency: 1k, 2.5k, and 5 kHz for the GreenPAK Advanced Development Platform and 1k, 2.5k, 5k, 10k, 20k, 50k, 100k, 200k, 400k, 1000 kHz for the GreenPAK Pro Development Platform.
To create an I2C signal using the I2C Generator:



1. Select the Debug button.
2. Select "I2C generator" on the SDA input external connector setting of the I2C block.
3. Go to Signal Wizard by clicking EDIT.
4. Select PIN8 as SCL and set the speed of the clock.
5. Choose Read or Write composite commands.
6. Open composite command and set the Slave address and Word address. For a Read command set byte count. For a Write command set data to write.



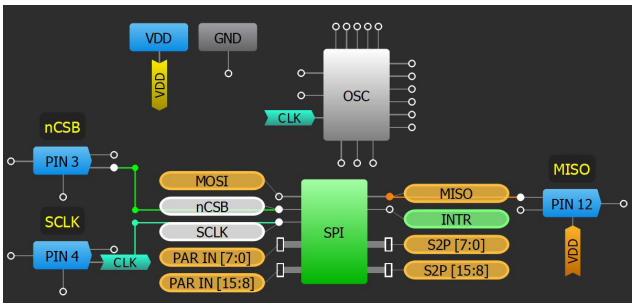
I2C Write Command



I2C Read Command

Technique: Using the Serial to Parallel Interface (SPI) Block

This technique is for the Serial to Parallel Interface (SPI) block, available in the SLG46140, SLG46620, and SLG46621.
The block is a special macrocell that can be used for communication between a GreenPAK and a SOC. It can either translate serial data to parallel or parallel data to serial. The inputs are standard SPI I/O connections (MOSI, MISO, nCSB, SCLK, and INTR). nCSB is an active low chip select. SCLK is the serial clock which clocks the SPI macrocell.



SPI macrocell

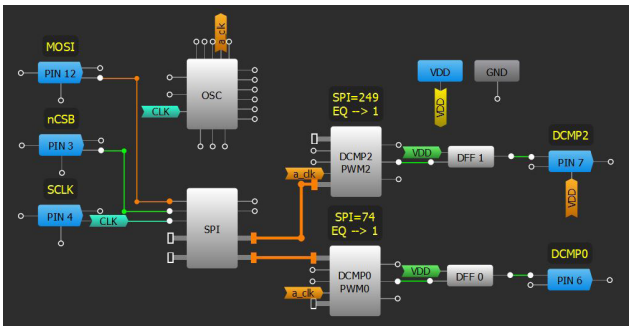
The SPI can be used to transfer data to such blocks as:

- FSM
- DCMP
- DAC (through DCMP)

The same SPI can be used to transfer data from:

- ADC
- FSM

All this can be used with other macrocells for functionality like:



SPI Macrocell in Serial to Parallel Mode

- Pulse Width Modulation
- Analog to Digital Comparison
- Digital to Analog Comparison
- Comparing two results with DCMP
- SDIO and LCD

The SPI can be selected to work in either 8-bits or 16-bits. Remember that the SPI macrocell cannot send and receive serial data in the same program file. It must be set up either in the “S2P” or “P2S” mode.
In the “P2S” mode the INTR pin pulses high for one clock period each time data after transmission completes. Otherwise, the SPI implemented in GreenPAK meets the generally accepted standard. It is possible to set the clocking frequency up to 2 MHz. It is also possible to configure the clock polarity with the CPOL bit and clock phase with the CPHA bit. When CPHA = 0, data can only be transmitted from serial to parallel, not from parallel to serial.
When CPHA = 1, data can be transmitted both from serial to parallel and from parallel to serial.

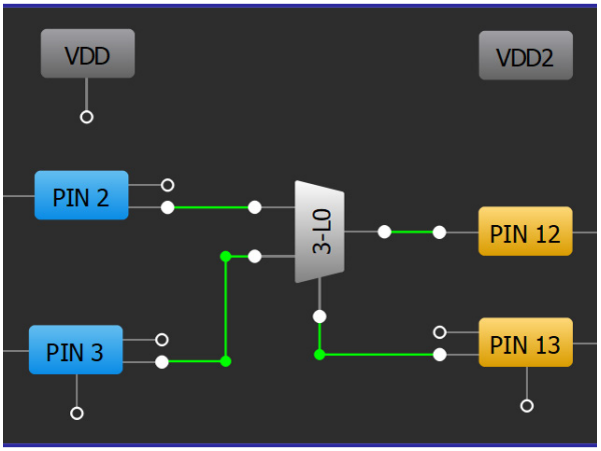
Technique: Level Shifting

This technique will work with any GreenPAK that has dual voltage rails, such as the SLG46826V.

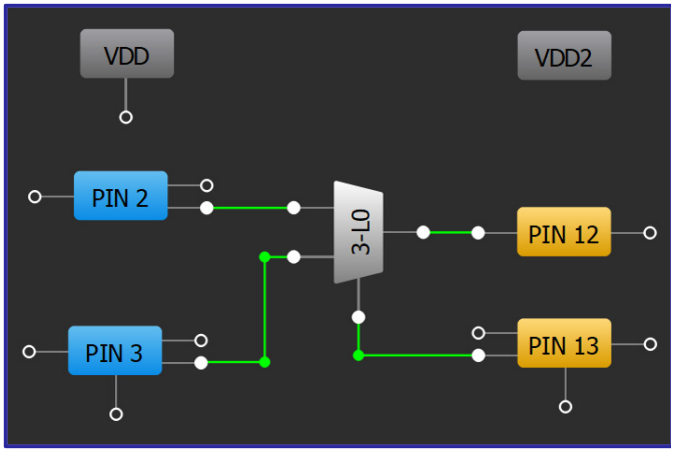
Often in a system-level design it's necessary to combine the data from two signals that operate at different voltage levels. For example, the analog rails in a system might operate at 5.0V while the digital rails operate at 3.3V.

Many GreenPAK ICs solve this problem by using dual voltage rails: signals that operate at different rails can be input into the GreenPAK, manipulated, then output from the GreenPAK at either of the voltage rail levels.

When starting a new GreenPAK Designer design using a dual rail part you'll be asked to input the voltage range of both rails. The available ranges of both rails will vary from part-to-part. The higher voltage rail should be designated as VDD, not VDD2.



Dual Rail Project Info



Dual Rail Logic Example

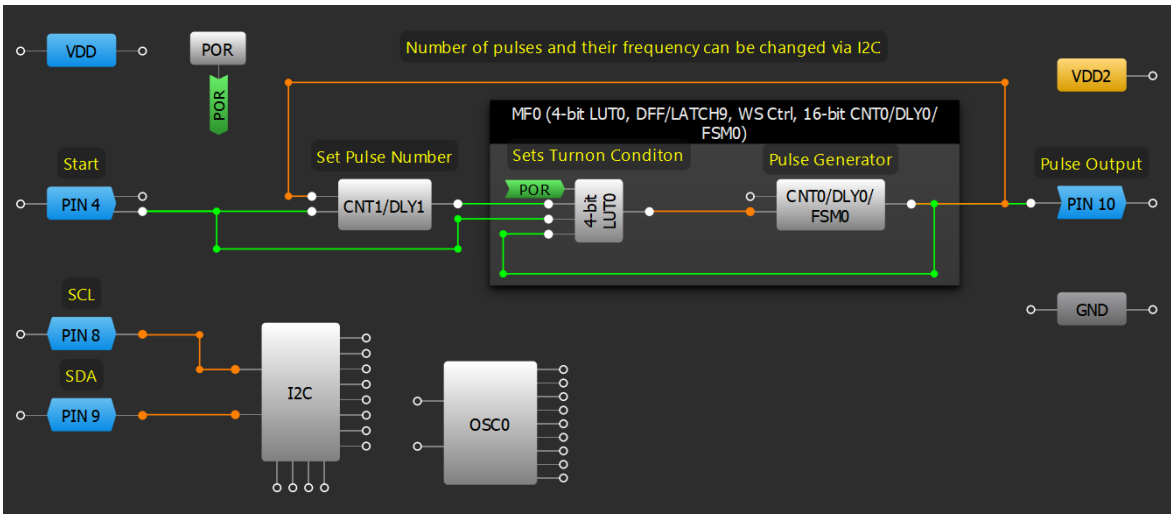
In dual rail parts the GPIO connections to the first and second rails are indicated by the color of the IO PIN within GreenPAK Designer. VDD will be indicated by blue pins and VDD2 will be indicated by amber pins. Once inside the GreenPAK matrix the signals from the different voltage levels will behave identically.

Technique: Sending a Preset Number of Pulses

This technique can be done in any GreenPAK. Multi-function blocks within some GreenPAKs help reduce the component count.

In many communication protocols a set number of bits must be sent to or received by another IC. Typically, this requires the GreenPAK must track the number of pulses sent or received. For example, in a shift register receiving data, the number of bits must be monitored to ensure that the expected data is in the correct register, rather than skewed incorrectly or relentlessly continuing to shift.

There are many ways to set a predetermined number of pulses in GreenPAK. A scalable, efficient way is described in this technique. This method also has the added benefit of limiting clock skew between the other IC and the GreenPAK by resetting the clock skew after every transaction. The figure below shows a series of blocks to send a preset number of pulses. There are a pulse number stage and a pulse generator stage.



Preset Pulse Generator Design

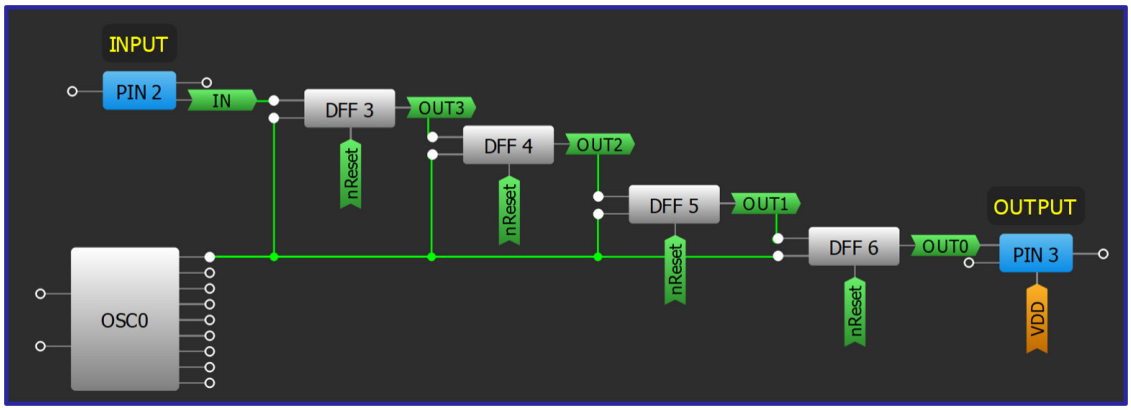
The pulse number stage consists of a one shot block that is clocked by the output pulses of the pulse generator. On a rising edge on PIN 4 (Start), the CNT1/DLY1 output will be set HIGH until it is clocked by the number of pulses set in the Counter data. After the set number of pulses, it will return LOW.

The pulse generator is made by MF0. Within MF0, CNT0/DLY0 is a both edge delay with an inverted output. Its delay time sets the period of the pulse generator. The output is fed back to 4-bit LUT0 that is configured to only invert the signal from CNT0/DLY0 while CNT/DLY1's output is HIGH. After the one shot pulse is finished the pulse generator will stop sending pulses.

Technique: Building a Shift Register

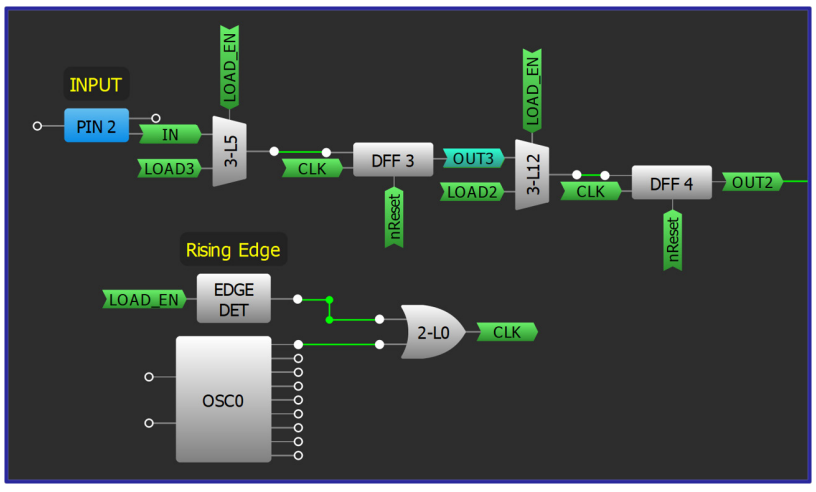
This technique can be used within any GreenPAK. The size of the shift register is dependent upon the components available within the specific GreenPAK.

Shift registers are a critical component for serializing or deserializing data. A shift register is a chain of flip-flops that can be sequentially linked and whose outputs can be individually accessed. Each has a connection to a shared clock; on the rising edge of the clock the registers will "shift" their data to the next flip-flop in the sequence. The figure below shows a basic, 4-bit shift register. The D flip-flops can be globally reset using a shared reset signal.



Basic Shift Register

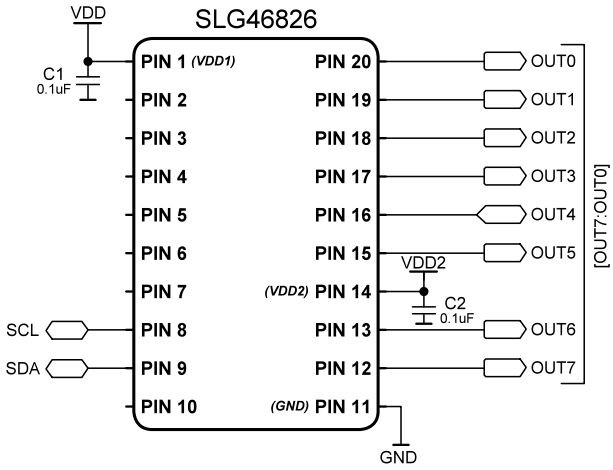
Often, shift registers must be loaded within the GreenPAK. This can be done by adding a MUX standard logic cell before each DFF. When data is ready to be loaded the MUX select input (S in GreenPAK Designer) is toggled and the DFF clock input is triggered to commit the value for each register. The figure below shows the addition of a MUX on 2 of the bits in above figure's basic shift register. LOAD_EN shares the CLK input to commit the loaded values to DFF3 and DFF4.



Loading a Shift Register

Application: I2C GPIO Expansion

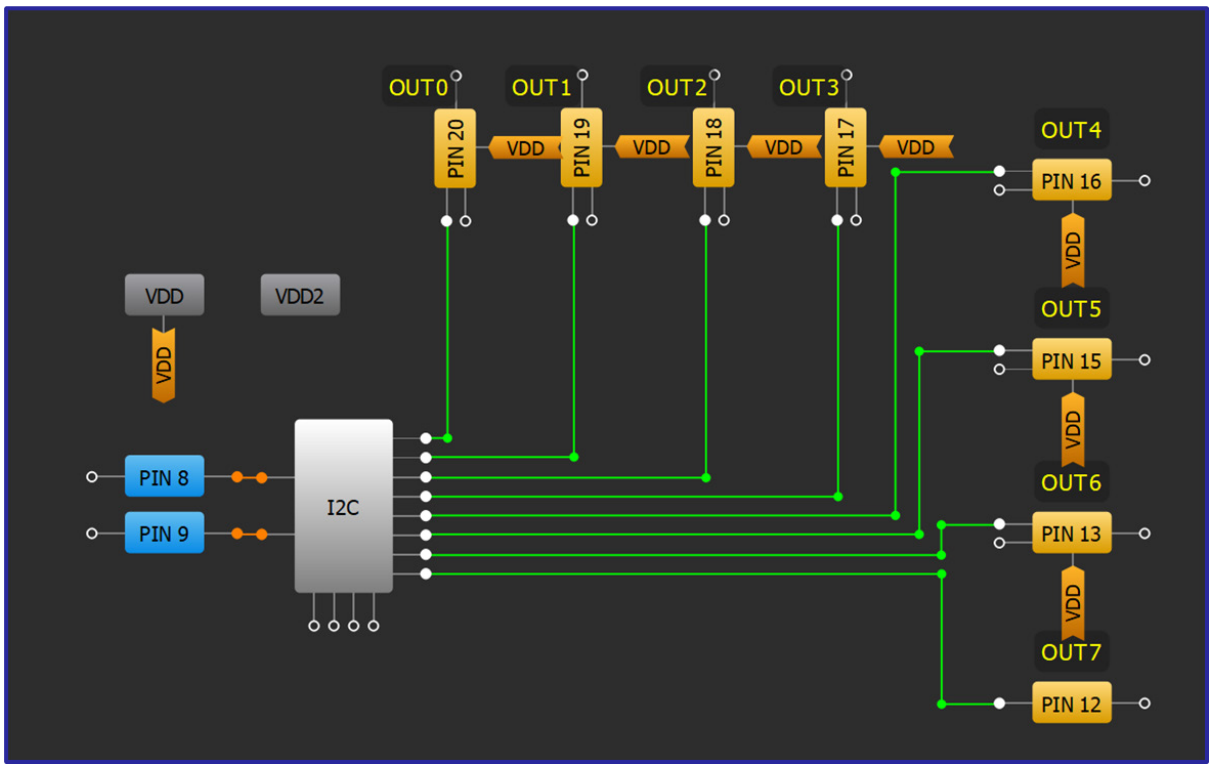
GPIO expansion is used to change the data originating from a few lines into data sent across many. I2C is a common input for this type of application, since one or more addresses can be changed into multiple, dedicated lines used by different ICs.



Ingredients

- Any GreenPAK with I2C
- No other components are necessary

GreenPAK Diagram

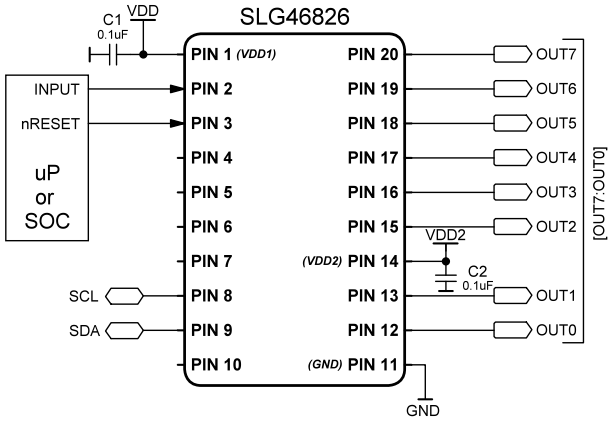


Design Steps

1. Configure GPIO pins as output.
2. Connect to I2C virtual inputs.
3. I2C virtual inputs can be changed individually or simultaneously using the I2C virtual output address, found in the GreenPAK's datasheet.

Application: Serial to Parallel (External Clock)

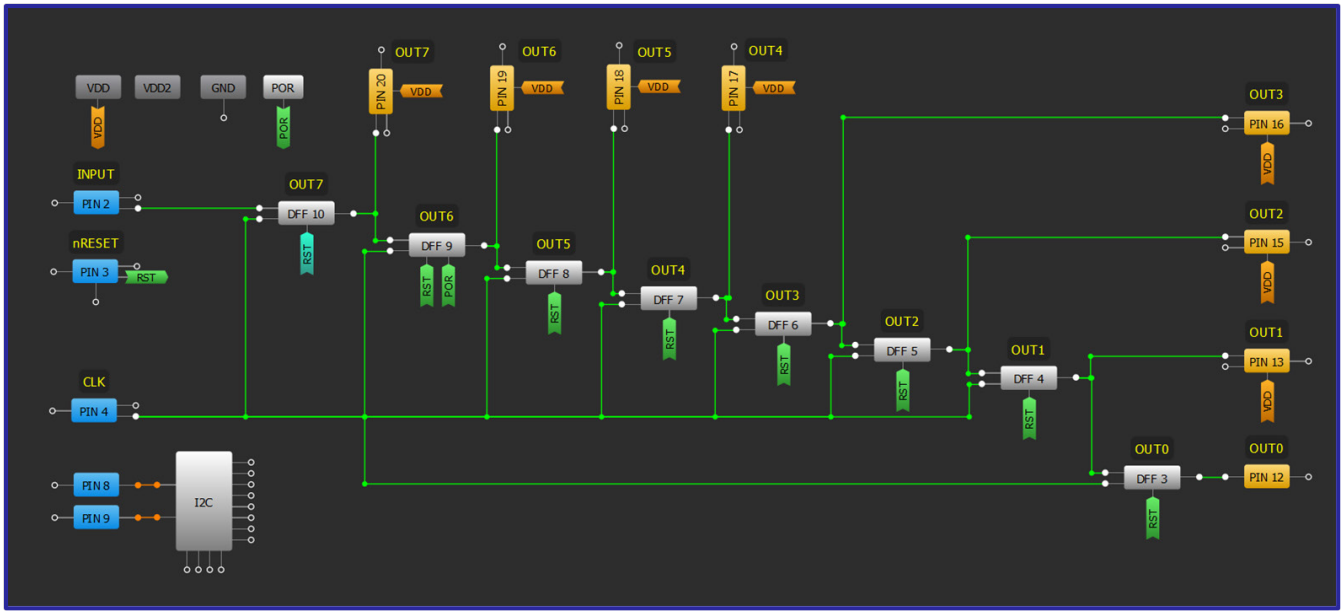
Deserialization ICs are used for data to be sent across one wire and read as multiple bits of data by the receiver. They often use an external clock tied to the same device as the data line to avoid clocking the data at an incorrect time.



Ingredients

- Any GreenPAK
- An IC an external clock output

GreenPAK Diagram

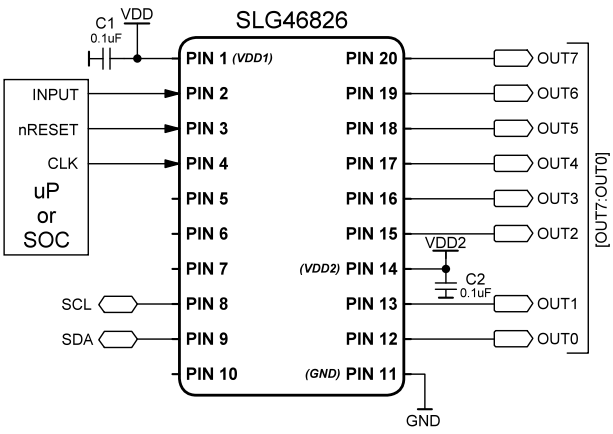


Design Steps

1. Configure the shift register as shown in Technique: Building a Shift Register.
2. Add an Input connection for the internal clock and connect it to the CLK input of the shift registers.
3. Add an Input connection for the reset function and connect it to the nRESET of the shift registers.

Application: Serial to Parallel (Internal Clock)

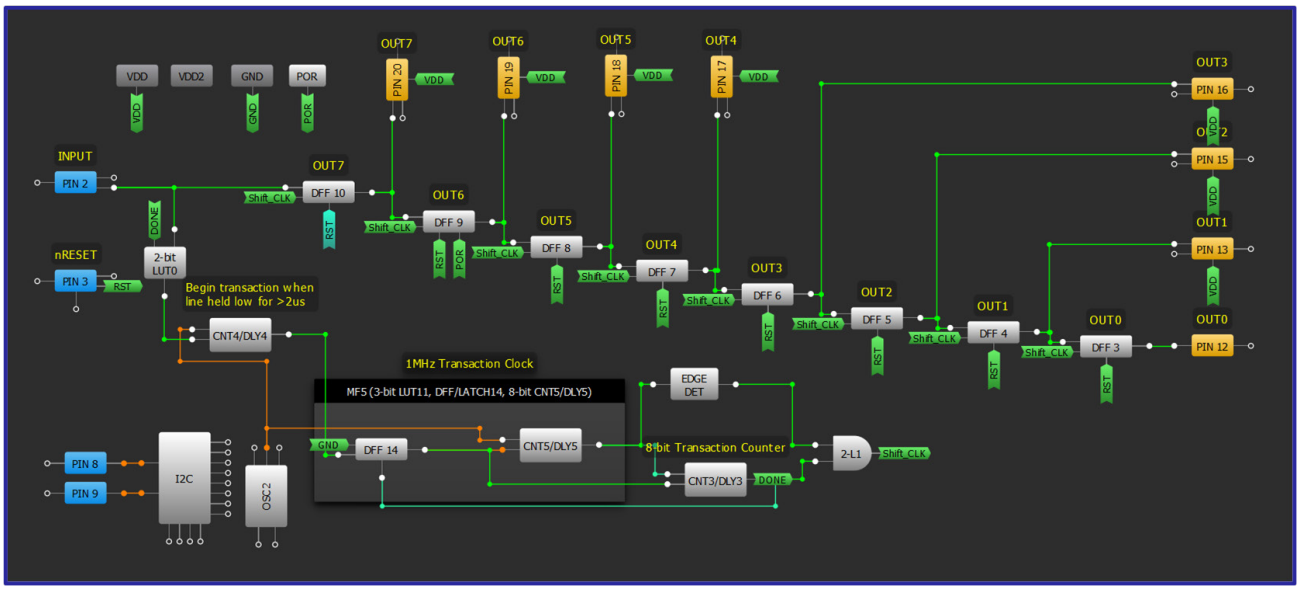
Deserialization ICs are used for data to be sent across one wire and read as multiple bits of data by the receiver. When adding an external clock isn't realistic, the GreenPAK can use an internal oscillator triggered from an action on the input line, such as a LOW signal that's held for a pre-determined period.



Ingredients

- Any GreenPAK
- An IC an external clock output

GreenPAK Diagram

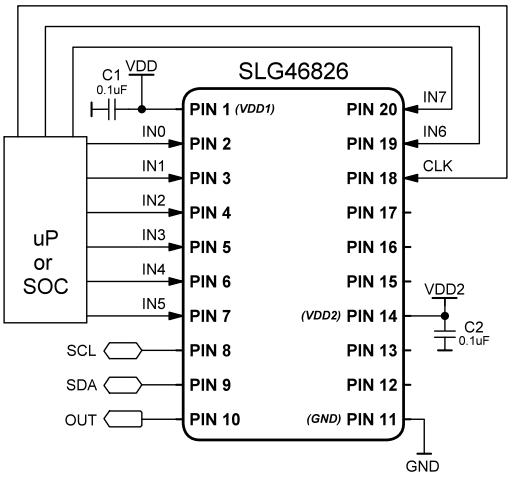


Design Steps

1. Configure the shift register as shown in Technique: Building a Shift Register.
2. Add an Input connection for the reset function and connect it to the nRESET of the shift registers.
3. Configure a preset number of pulses to match the number of shift registers. This is outlined in Technique: Sending a Preset Number of Pulses.

Application: Parallel to Serial

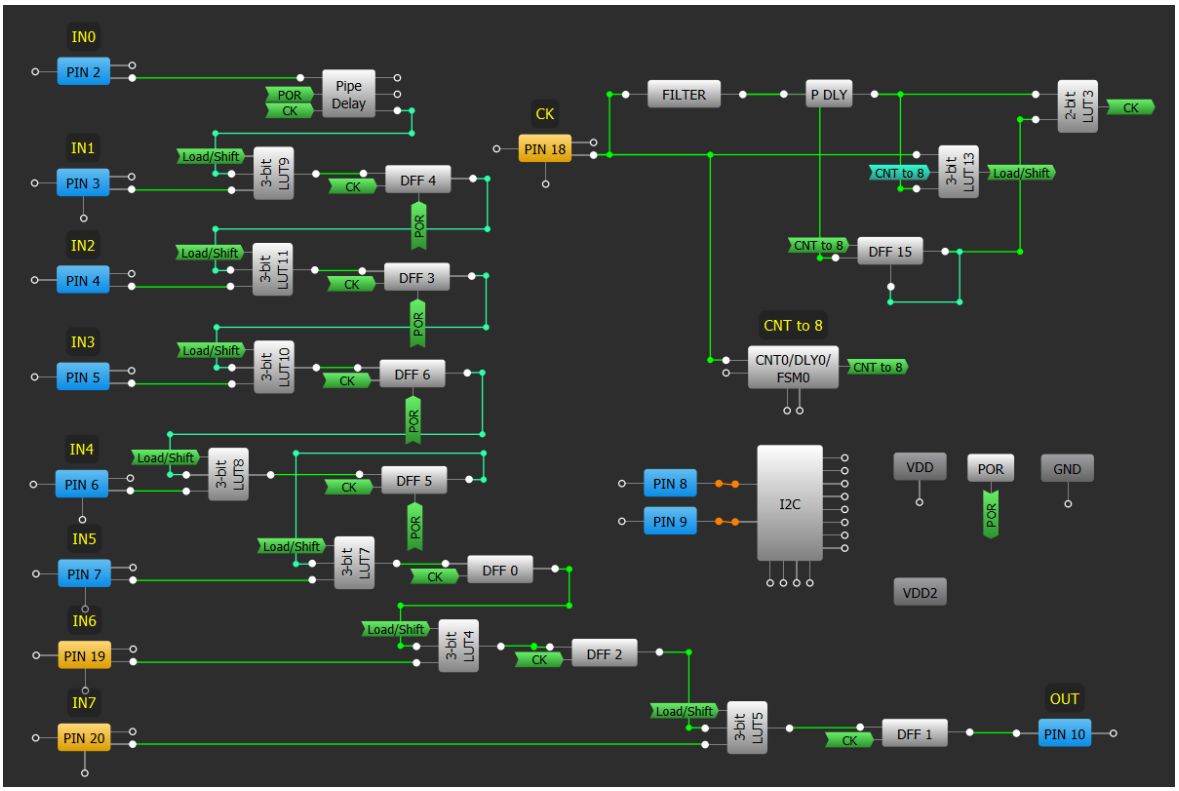
Parallel to Serial convertors are commonly used to send data from one IC to another using one or two wires (Data and Clock). Internal or external clock can be used here, and number of parallel bits is limited by number of GPAK I/Os and internal blocks available.



Ingredients

- Any GreenPAK
- An IC an external clock output

GreenPAK Diagram

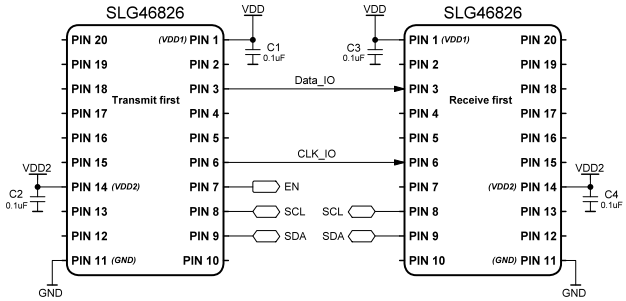


Design Steps

1. Configure the shift register as shown in Technique: Building a Shift Register.
2. Add and configure 3-bit LUTs for each DFF to operate as MUX using Technique: Configuring Standard Logic w/ LUT Macrocells.
3. Configure Load/Shift function and connect internal blocks to inputs, output.

Application: Bi-Directional Communication (Transmit First)

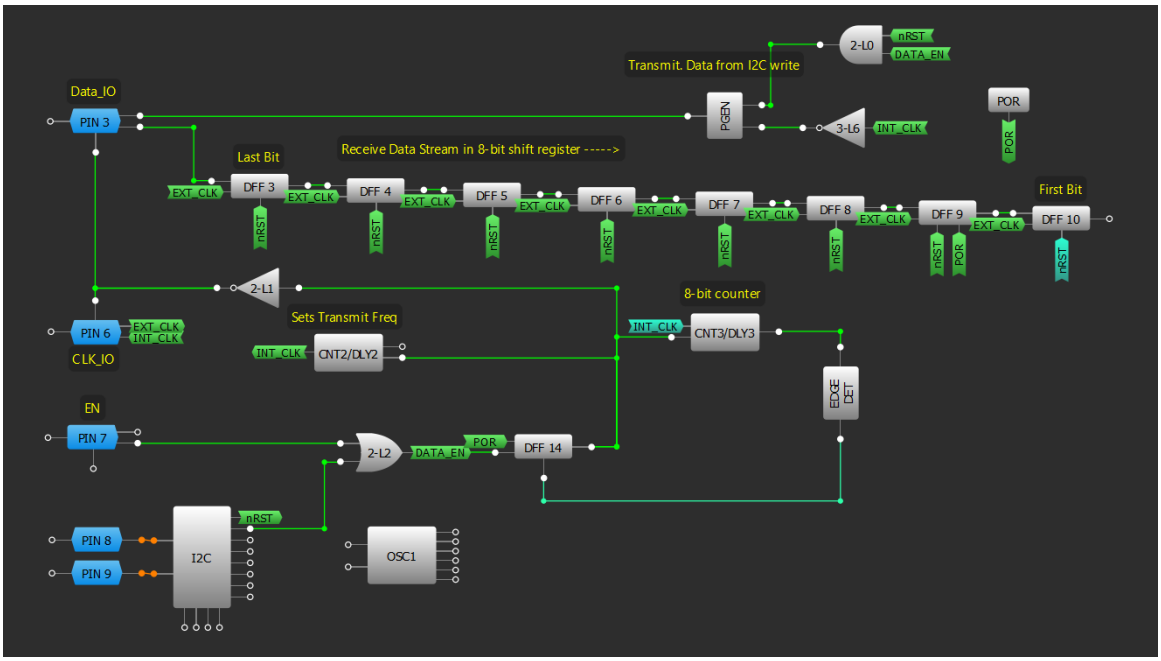
Bi-directional communication systems are critical in applications where board area is scarce, or limited contacts are used to connect between devices (for example, a wearable and charger). A transmit-first design indicates that this device will always be enabled first in the transaction.



Ingredients

- Any GreenPAK
- An IC an external clock output

GreenPAK Diagram

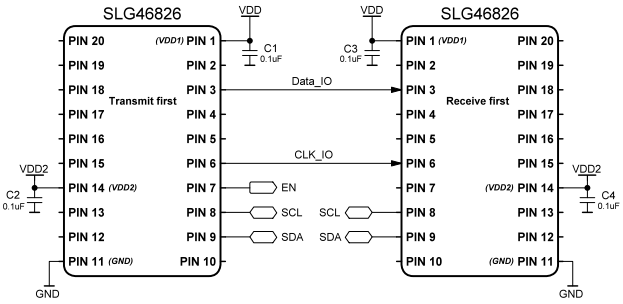


Design Steps

1. Configure two GPIO pins as a “Digital input/output,” one for the data line and one for the clock.
2. Create an internal clock signal using a CNT/DLY block configured as a “Reset counter.”
3. Configure a shift register to store incoming data, shown in Technique: Building a Shift Register. Shift registers can be read with I2C.
4. Add PGEN to send outgoing data. The PGEN contents can be changed using I2C.
5. Configure DFF to enable transmission. Connect output to transmit clock signal and OE of I/O pins.
6. Add CNT/DLY block to disable transmission, with CLK input from DFF. Connect Edge Det block to reset DFF after set number of clock pulses.
7. Connect the external clock to the shift register and connect the internal clock to PGEN and I/O pins.

Application: Bi-Directional Communication (Receive First)

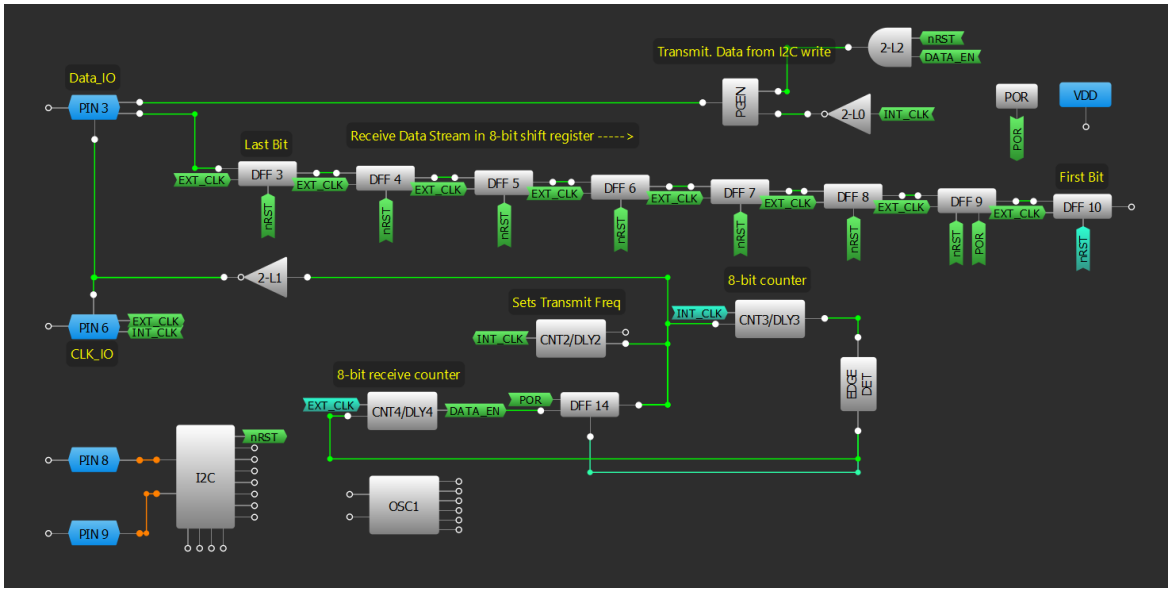
Bi-directional communication systems are critical in applications where board area is scarce, or limited contacts are used to connect between devices (for example, a wearable and charger). A receive-first design indicates that this device will always be enabled second in the transaction.



Ingredients

- Any GreenPAK with OE pins
- An IC an external clock output

GreenPAK Diagram

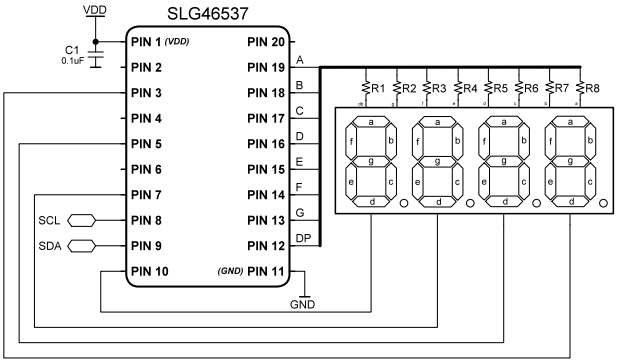


Design Steps

1. Configure two GPIO pins as a “Digital input/output,” one for the data line and one for the clock.
2. Create an internal clock signal using a CNT/DLY block configured as a “Reset counter.”
3. Configure a shift register to store incoming data, shown in Technique: Building a Shift Register. Shift registers can be read with I2C.
4. Add PGEN to send outgoing data. The PGEN contents can be changed using I2C.
5. Configure DFF to enable transmission. Connect output to transmit clock signal and OE of I/O pins.
6. Add CNT/DLY block as a “Reset counter” to disable transmission, with CLK input from DFF. Connect Edge Det block to reset DFF after set number of clock pulses.
7. Add CNT/DLY block as a “Reset counter,” with input as external clock and output to enable transmission DFF.
8. Connect the external clock to the shift register and connect the internal clock to PGEN and I/O pin.

Application: 7-Segment Display Using ASM and I2C

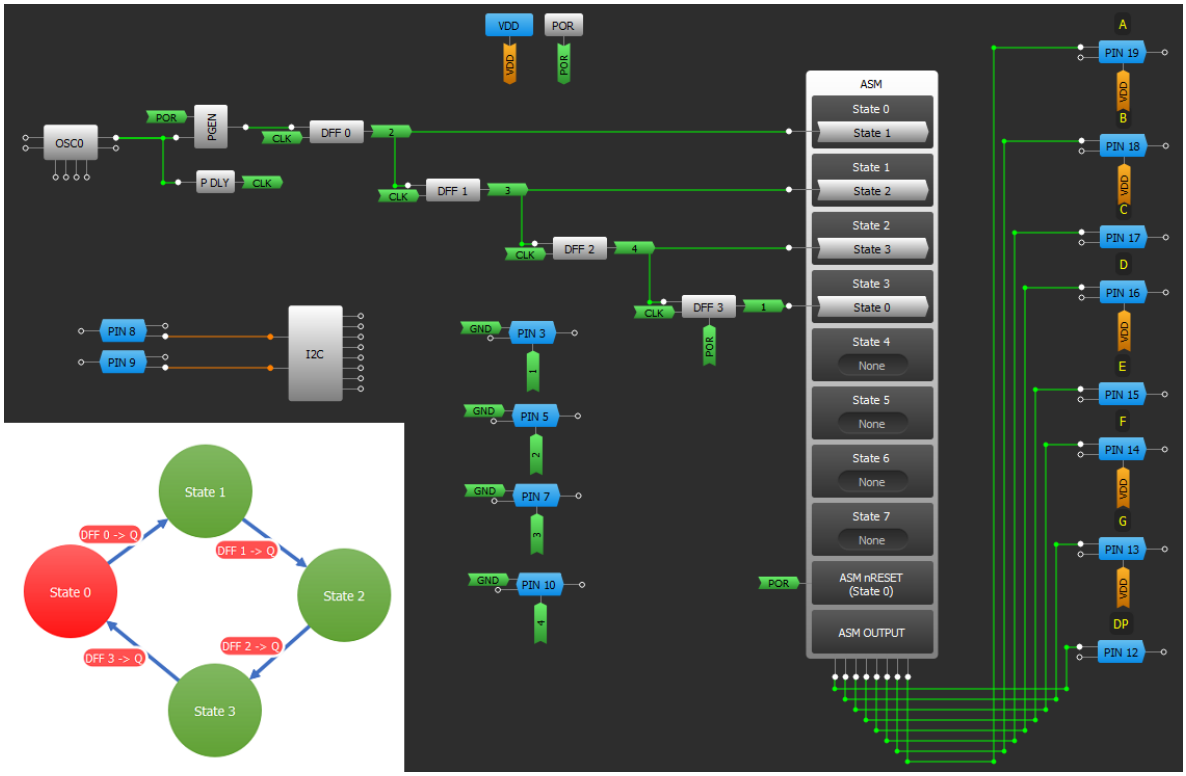
A 7-segment indicator is a common numerical display. The GreenPAK asynchronous state machine and I2C can be used to provide directions to the segments as to which number should be displayed. The provided example is compatible with a 4-digit, 4 decimal display.



Ingredients

- Any GreenPAK with I2C and ASM
- 7-segment display
- Eight resistors

GreenPAK Diagram

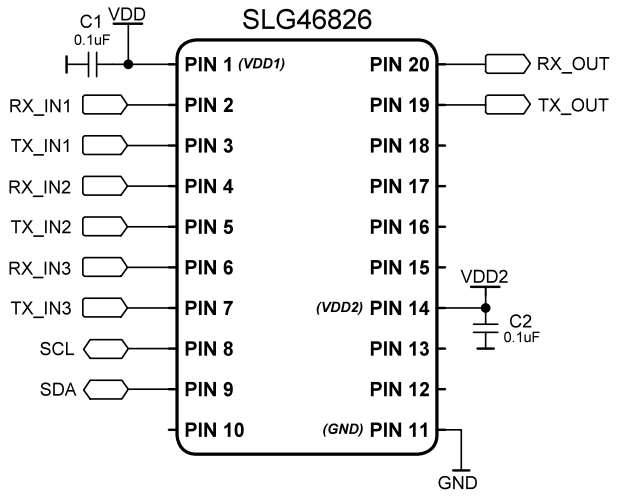


Design Steps

1. Configure GPIO pins as output and connect them to the ASM output.
2. Add shift register using Technique: Building a Shift Register.
3. Create a logic generator using the PGEN block, corresponding the required number of digits.
4. Add and configure ASM to match the initial digital sequence.
5. Update one or more digits to an ASM state via I2C.

Application: Communication MUX Using I2C

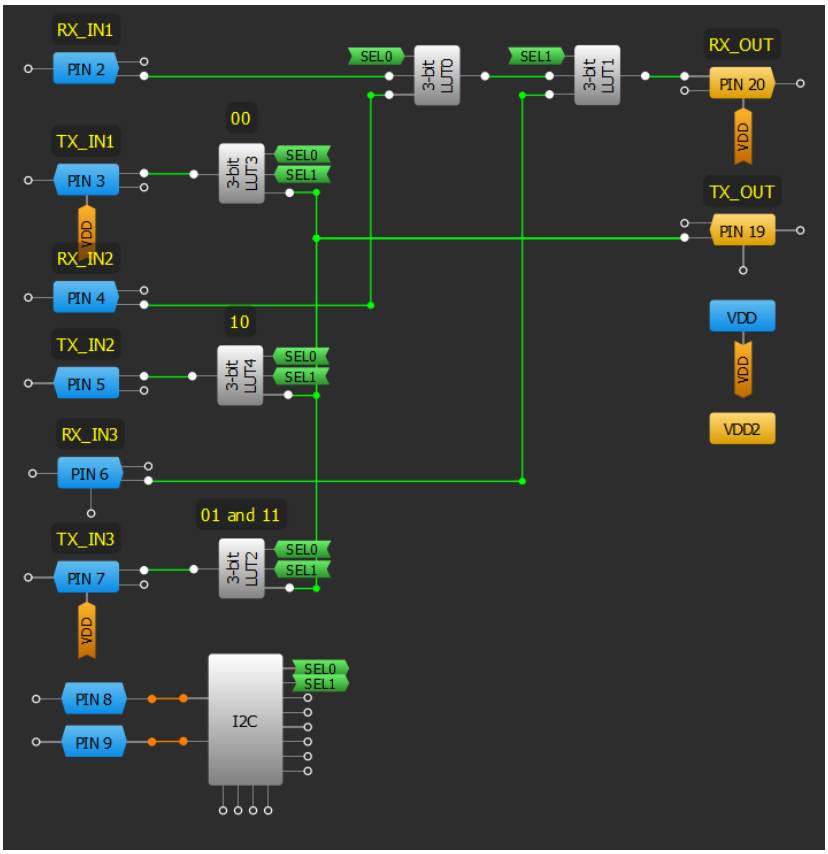
An I2C Communication MUX is used when a designer needs to combine several I2C input signals and forward them into a single output line.



Ingredients

- Any GreenPAK with I2C
- Two resistors
- Two capacitors

GreenPAK Diagram

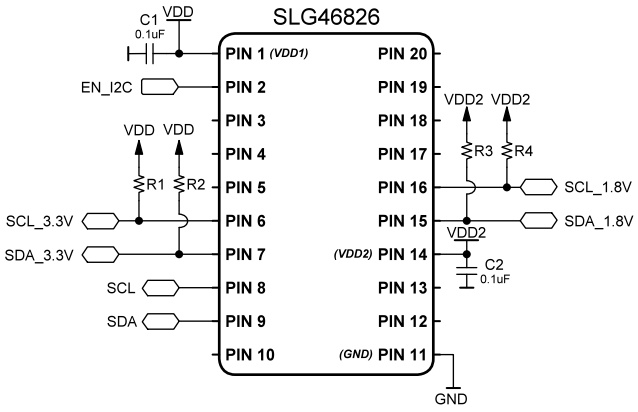


Design Steps

1. Configure a 1:3 demux with LUTs to share the TX signal from the MCU, sent to external UART ports.
2. Configure a 3:1 mux with LUTs to receive an RX signal from the external UART ports to the MCU.
3. Use the I2C to select the input port.

Application: I2C Level Shifter

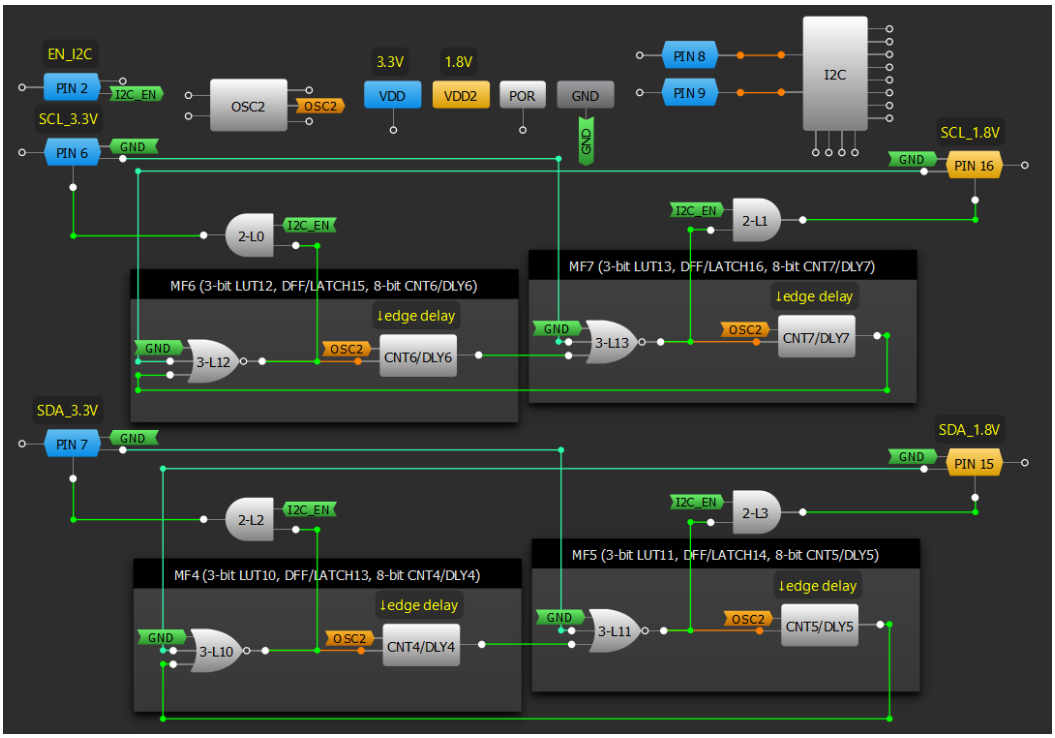
An I2C Communication MUX is used when a designer needs to combine several I2C input signals and forward them into a single output line.



Ingredients

- Any GreenPAK
- Four resistors

GreenPAK Diagram

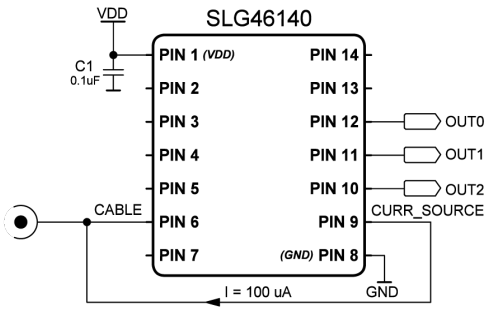


Design Steps

1. Configure four GPIO pins as digital input/outputs with the output mode set to open drain NMOS.
2. Configure one pin as a digital input for the enable signal.
3. Add an AND gate to each input/output.
4. Configure four multi-function blocks (or four LUTs and four CNT/DLY blocks if Multi-function blocks aren't available) as a NOR gate feeding into a falling edge delay.
5. Select OSC2 as the clock source for the delay blocks and set it to "Force Power On."

Application: Connection Detect

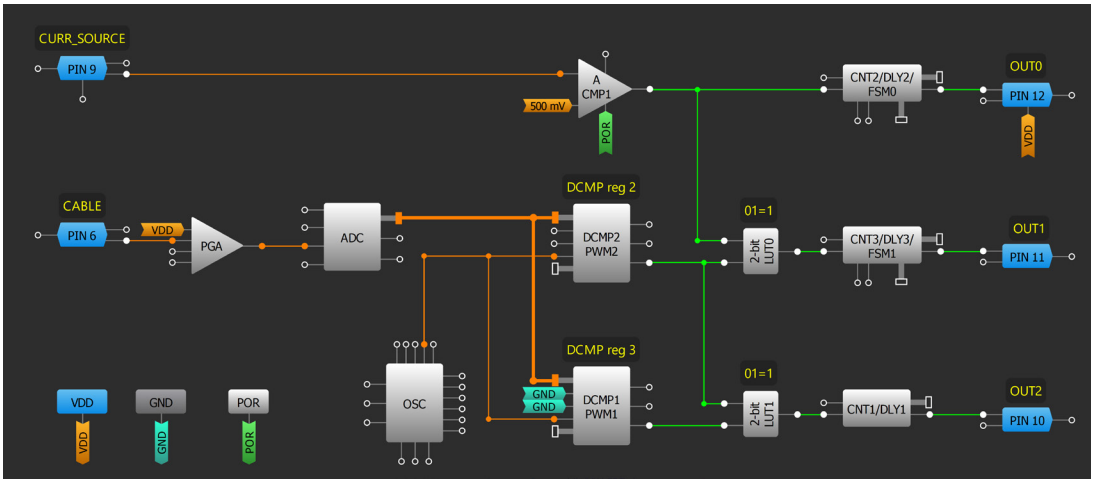
This application detects the presence of a cable by measuring the voltage proportional to its resistance. The ACMP's 100uA current source is used to produce the voltage drop across the cable. This configuration is also able to determine which type of connection is being made based on different wire lengths or connected loads.



Ingredients

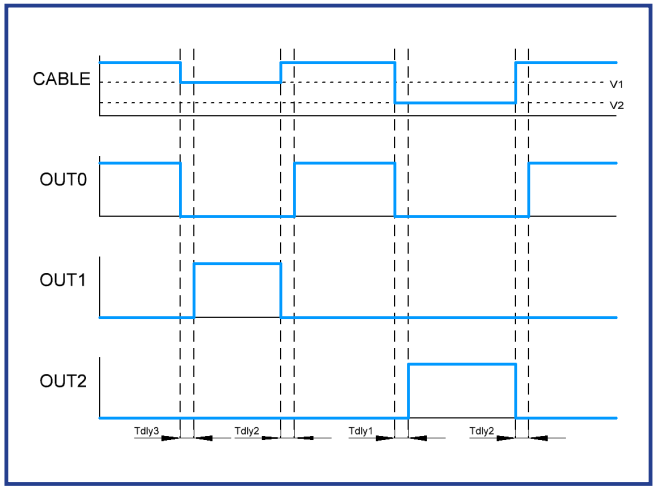
- Any GreenPAK
- Four resistors

GreenPAK Diagram



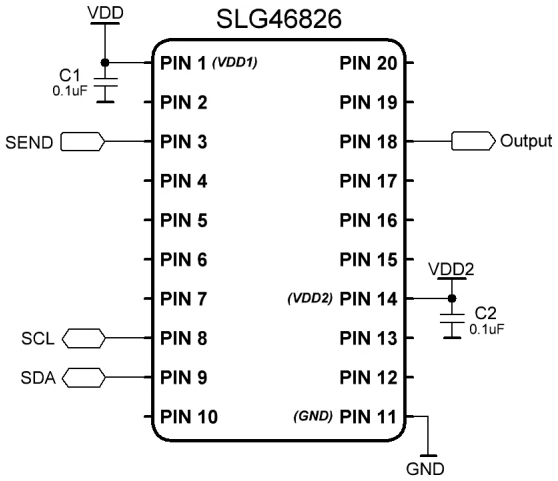
Design Steps

1. Enable the input 100uA current source in ACMP1 and configure the IN- source.
2. Configure ADC and PGA blocks.
3. Enable DCMP/PWM blocks (delete VDD from SHARED PD input). Set DCMP/PWM power register to Power on. Make sure that IN+ selector is connected to ADC and IN- selector is connected to an internal register.
4. Set needed value of register in DCMP/PWM blocks and configure MTRX SEL inputs.
5. Configure 2-bit LUT0 and 2-bit LUT1;
6. Set CNT1-CNT3 to rising edge delay mode.
7. Configure PIN10-PIN12 as Digital output 1x push pull.



Application: Custom Pattern Generator

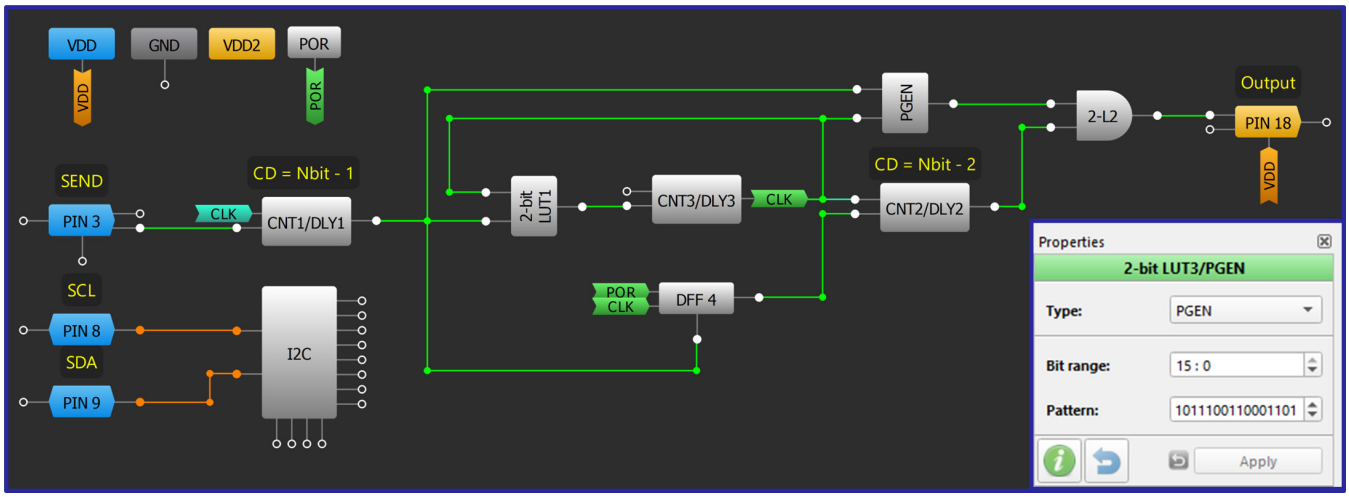
The pattern generator (PGEN) in the GreenPAK stores a pattern of logic 1's and 0's (up to 16 bits) that can be sent to the internal matrix serially. This design outputs a 16-bit code after a rising edge on an input. It can be used in sequential logic applications.



Ingredients

- Any GreenPAK with PGEN
- No other components needed

GreenPAK Diagram



Design Steps

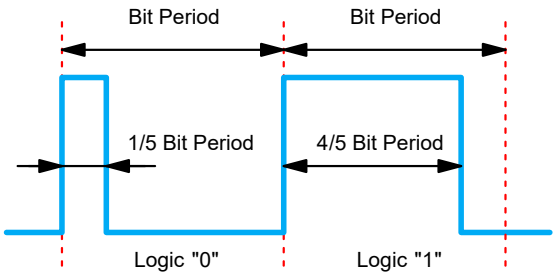
1. Configure the custom N_bits pattern in the PGEN.
2. Configure CNT1/DLY1 mode as a One Shot. Set Counter Data to Counter Data=N_bits-1.
3. Configure CNT2/DLY2 mode as a One Shot and set Counter Data to Counter Data=N_bits-2.
4. Configure CNT3/DLY3 as rising edge delay and create the generator using 2-bit LUT1.
5. The design can be improved by using a GreenPAK with I2C to dynamically modify the PGEN data, clock frequency and Counter Data.

Technique: Sending Serial Protocols Using Duty Cycle Detection

This technique can be used with any GreenPAK consisting of PGEN, Counter/Delay blocks and I2C. Read Technique: Reading Serial Protocols with a Shift Register and Technique: Reading Serial Protocols with a Pipe Delay for how to read the protocol discussed here with the GreenPAK.

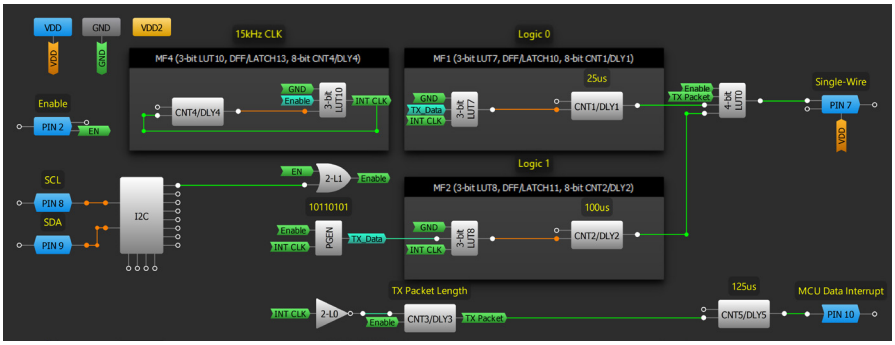
Serial data transmission has a wide variety of communication applications including power lines, wireless systems, and MCUs. A single-wire data transmission can transfer data using distinct duty cycles for another GreenPAK or MCU to read.

The duty cycle detection topology is shown in the figure below. Each bit is represented by a periodic pulse with a particular duty cycle range. Within this design a transacted bit is set "0" when the pulse duration is ≤1/5 of the period and "1" when ≥4/5 of the period. Other duty cycle ranges can be used, provided there is sufficient distance between the duty cycle ranges for Logic "0" and Logic "1".



Logic 0 and Logic 1 Detection

In the design displayed below, data transmission is initiated by Enable, which can be supported through an external GPIO or an I2C Virtual Input. CNT4/DLY4 sets the bit period for data transmission. MF1 and MF2 blocks determine the duty cycle to transmit. Data to be transmitted is written in the PGEN via I2C or non-volatile memory. The PGEN can transmit up to 16 bits, but this design demonstrates 8-bit data transmission on the single-wire output. For more information on how to write different patterns through PGEN via I2C, refer to Application: Custom Pattern Generator.



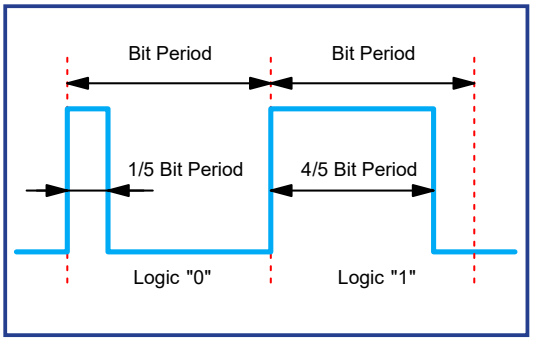
The data packet length is determined by CNT3/DLY3, which in this case it is set to transmit 8-bits. 4-bit LUT0 transmits the data on the Single-Wire output after the data is embedded into the duty cycle of the transmission signal. After data transmission is completed, PGEN can be re-written through I2C.

If the user wishes to bit bang the transmission, CNT5/DLY5 generates an interrupt signal to indicate to an external MCU that the transmitted package is completed.

Technique: Reading Serial Protocols with a Shift Register

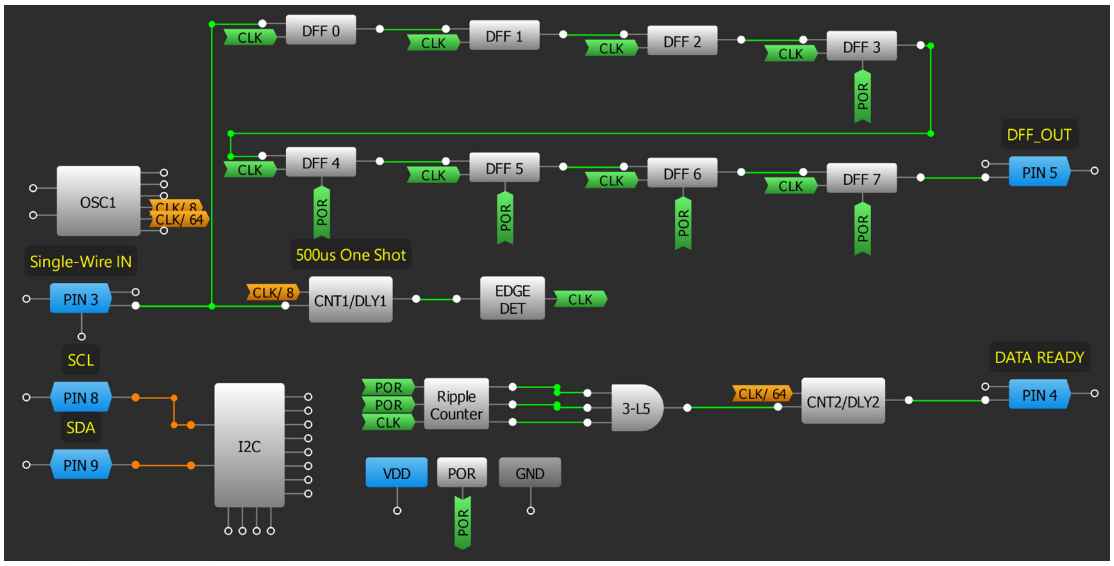
This technique can be used within any GreenPAK equipped with enough DFF blocks, a Ripple Counter, and three CNT/DLY blocks. It explains a method to read the single-wire protocol discussed in Technique: Sending Serial Protocols Using Duty Cycle Detection.

Serial data transmission has a wide variety of communication applications including power lines, wireless systems, and MCUs. A shift register can be used as a highly versatile serial receiver for a single-wire transmission.



The figure above shows the single-wire topology discussed in Technique: Sending Serial Protocols Using Duty Cycle Detection. A delayed edge detect on the rising edge is used to sample halfway into the signal period (bit length). When the shift register of this particular design is clocked, the signal will be LOW at points where the pulse is ≤1/5 of a bit period and HIGH where the pulse is ≥4/5 of a bit period.

The figure above shows a method for obtaining the single-wire data in 8-bit increments. Each time the shift register is clocked, the Ripple Counter is also incremented. When the Ripple Counter receives 8 pulses, CNT2/DLY2 outputs a high pulse on DATA READY to signify to an MCU that the complete single-wire transmission has been read and is ready to be read by I2C. This design reads transmissions with a 1ms period since CNT1/DLY1 delays the rising edge by 500µs, but this can be easily changed by adjusting its counter value.



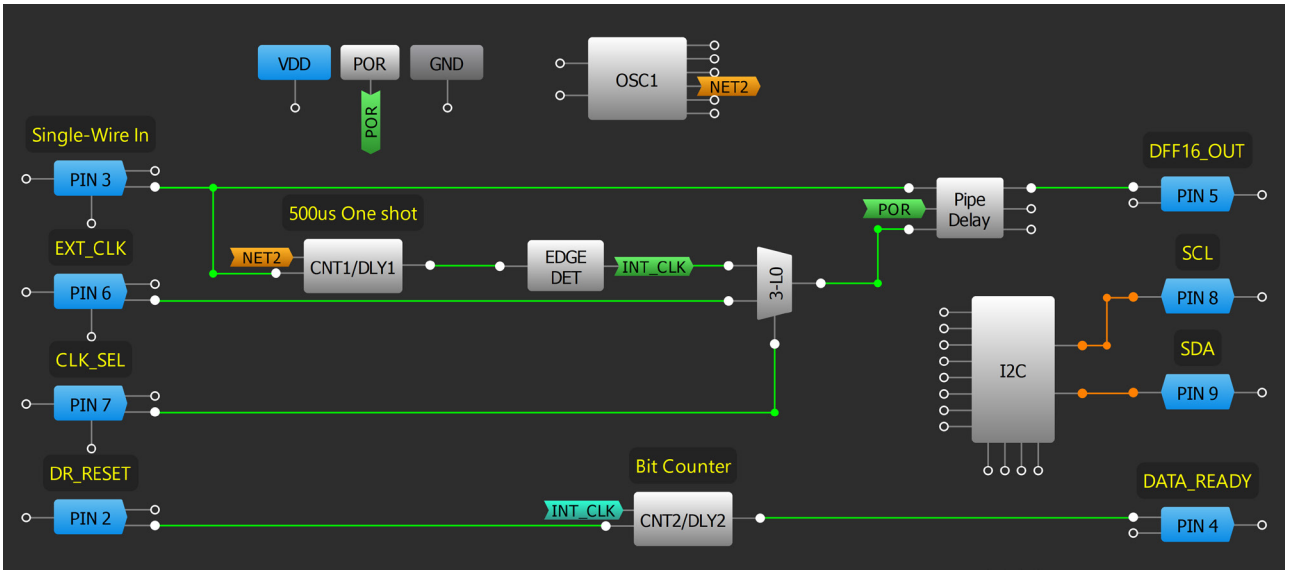
Technique: Reading Serial Protocols with a Pipe Delay

This technique can be used within any GreenPAK. It explains a method to read the single-wire protocol discussed in Technique: Sending Serial Protocols Using Duty Cycle Detection.

Serial data transmission has a wide variety of communication applications including power lines, wireless systems, and MCUs. A single Pipe Delay block can be used in the place of a resource heavy shift register as a serial receiver for a single-wire transmission. However, it must be noted that unlike the shift register method, this Pipe Delay method can only output its data serially and doesn't have parallel output capability.

The Pipe Delay can be thought of as a 16-bit shift register with three available outputs. Two are configurable to output any of the Pipe Delay's sixteen internal D flip-flops. For this 16-bit implementation, shown below, OUT0 of the Pipe Delay is set to represent the 16th DFF output. An input for an MCU to externally clock the Pipe Delay is multiplexed with the Single-Wire In input to allow it to unload the Pipe Delay's data without incrementing the bit counter.

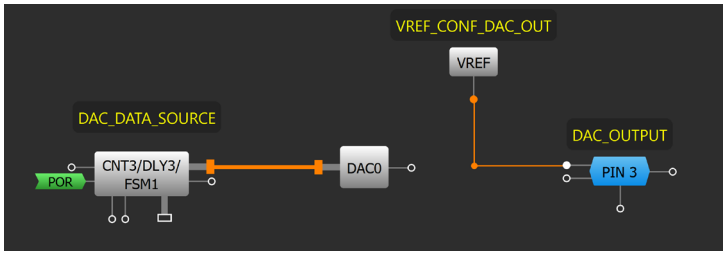
This design utilizes the same delayed data sampling mechanism as Technique: Reading Serial Protocols with a Shift Register, but it instead keeps track of its data bits with a counter block rather than a Ripple Counter. This counter increments each time a data bit is detected. Upon reaching 16 bits, it will set the DATA_READY pin HIGH until the next data bit is sent. It is important to note that this counter block must be reset before any data is sent to ensure proper operation, which can be done internally by POR or externally by an input (as in this design).



Technique: Using the Digital-to-Analog Converter (DAC)

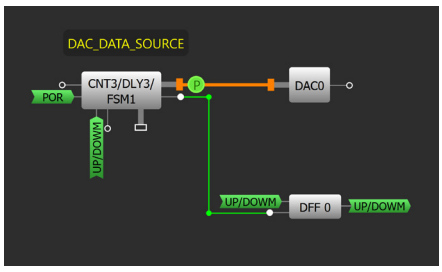
This technique can be used with SLG46140, SLG46620, and SLG46621 GreenPAKs.

Some GreenPAK devices contain Digital-to-Analog Converters (DAC). They are 8-bit DACs which operate at maximum sampling speed of 100 ksps. The DAC's differential non-linearity is less than 1LSB and integral non-linearity is less than 1LSB. The DAC output-to-PIN resistance is 1 kΩ. Load resistance is recommended to be no less than 10kΩ and load capacitance to be no more than 100 pF. Typically, the DAC output range is from 0V to 1V, but in the SLG46620/1 the DAC1 output range is from 50mV to 1.05V.
Either the register, SPI, or FSMs can be configured as an input for the DAC. The DAC's output can be configured to the VREF's output pin, PGA, or ACMP.
In some ICs DAC0 is used as a part of the pseudo-differential mode of the PGA macrocell. Therefore, DAC0 is not available when the PGA is in pseudo-differential mode. Also, DAC1 is shared with an ADC macrocell. Therefore, it is impossible to use DAC1 when ADC is used. To connect the DAC macrocell output to a VREF macrocell, it is necessary to configure this pin as analog input/output and configure the Source selector for the VREF to be a DAC.



Sawtooth Generator

The DAC can be used to create a simple Sawtooth Generator shown above. In this case DAC0 uses FSM1 as its data source. The DAC output is connected to the VREF which is connected to PIN3 (DAC_OUTPUT). The output signal period and resolution are set by the counter data and clock frequency of FSM1. Also, you can add a toggling DFF and connect it to the UP input of an FSM to produce a triangle wave generator shown below. The counter data value changes up and down in time, depending on DFF0's output.



Triangle Wave Generator

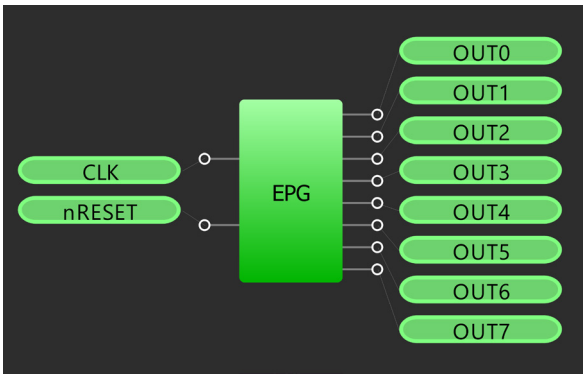
The DAC can be used as a reference for an ACMP or PGA negative input in "Differential" and "Pseudo-differential" mode. If the ACMP reference uses a DAC, the user can compare the analog signal to discrete data without using the ADC block.
The GreenPAK DAC:

- can be used to create waveform generators;
- create reference voltage source controlled via SPI;
- can be used in different converters as it converts temperature, humidity and other digitized values to analog voltage.

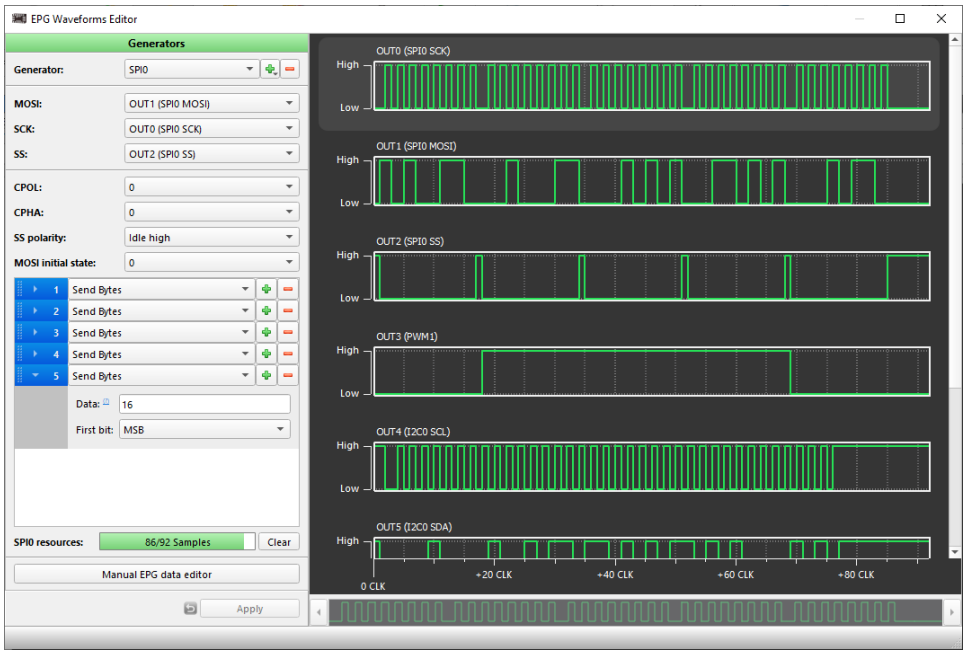
Technique: EPG

The Extended Pattern Generator (EPG) is capable of producing a range of outputs that is 92 bytes in size. It retrieves data from non-volatile memory (NVM) and outputs one byte at a time every time the CLK input signal encounters a rising edge. Additionally, the EPG shares its outputs with I2C Virtual Inputs. The maximum clock frequency is up to 1 MHz.

Upon power-up of the system, the EPG behaves differently based on the signal received at the nReset input. Specifically, when the nReset input is active LOW, the EPG will display the initial value at the output. Conversely, if the nReset input is active HIGH, the EPG will display user-defined patterns at the output. This feature enables the user to customize the output of the EPG to their specific needs.
EPG can work continuously when CLK is being applied in Overflow mode or keep at the last byte in Stop at boundary mode.



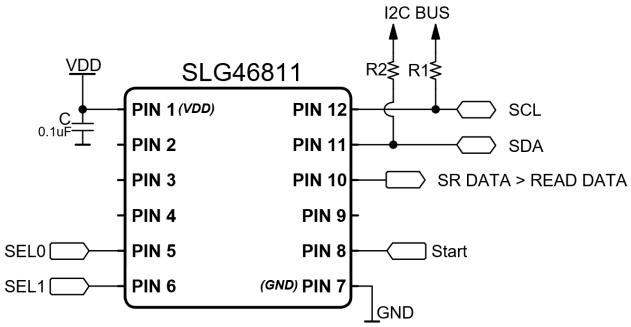
The EPG Waveforms Editor lets a user select from a variety of predefined Generators, such as SPI, I2C, PWM, or manual, and assign the output accordingly. Each of these generators offers a range of tunable settings, making it easier for the user to build their desired pattern.



The resource bar is a visual representation that indicates the number of bits used in the pattern. It provides the user with a clear understanding of the amount of memory and resources that have been allocated to the current pattern.

Application: I2C Master Read Command with ACK Check and Data Comparison

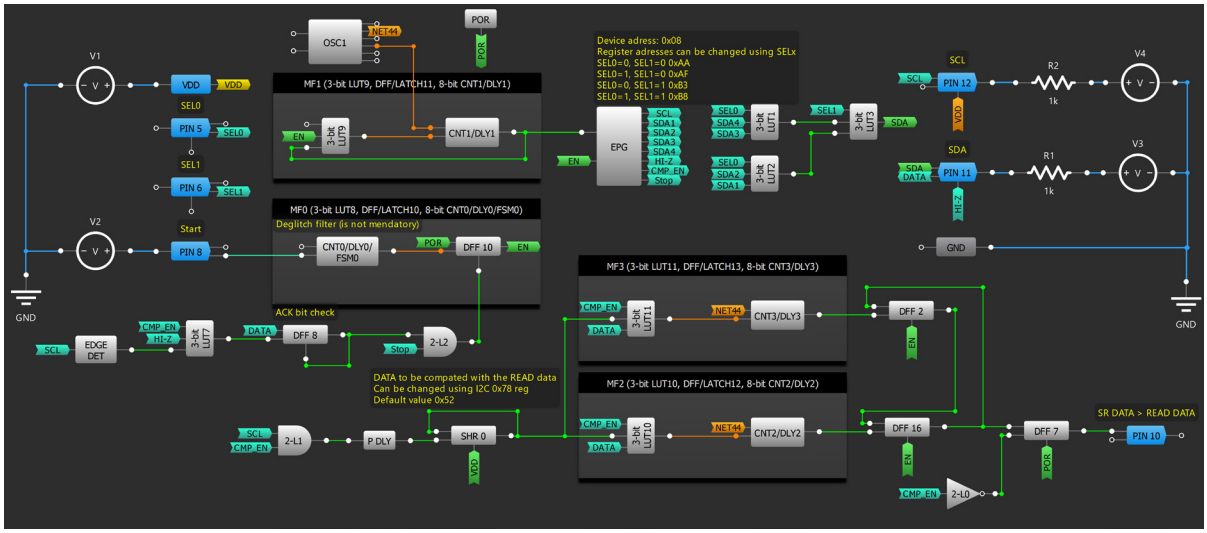
This application demonstrates how to build a simple I2C master which can read the slave data and compare it with the reference data using SLG46811.



Ingredients

- SLG46811V
- Two resistors

GreenPAK Diagram

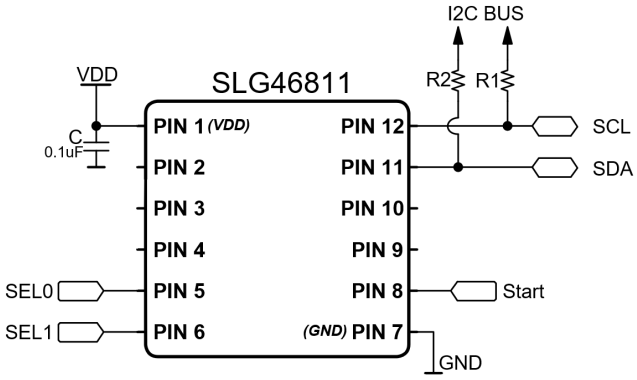


Design Steps

1. Configure EPG generator (see EPG technique) to create I2C Read command and set several SDA and SCL patterns.
2. Add one channel in EPG to check the ACK bit every 9th clock at SCL
3. Create a 4-bit Multiplexer, as was described in Application: 8-bit Multiplexer. SEL0 and SEL1 choose which data (SDA) will be output.
4. Create frequency generator based on CNT1/DLY1 and LUT9, which performs clocks for EPG
5. Deglitch filter based on CNT0/DLY0 which triggers the I2C pattern.
6. Design monitors if the ACK bit check was present during the I2C command, if the I2C slave doesn't respond, the transmission stops.
7. When the I2C slave starts sending the data back, they are compared with the reference data stored in SHR 0. At every clock during comparison the data in SHR0 is shifted and compared with the data received. DFF7 stores the comparison result until the read command.

Application: I2C Master Write Command with ACK Check

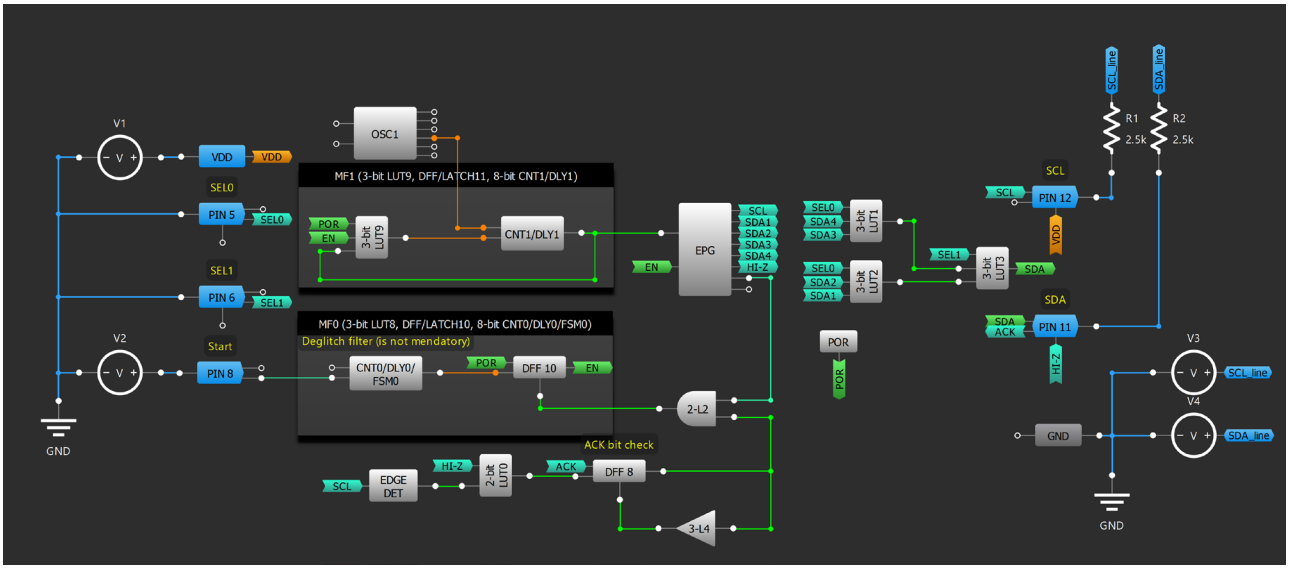
This application demonstrates building a simple I2C master using the SLG46811 device. The I2C master built using this device is capable of rewriting one or multiple bytes, making it a great solution for systems that do not require a complex implementation.



Ingredients

- SLG46811V
- Two resistors

GreenPAK Diagram

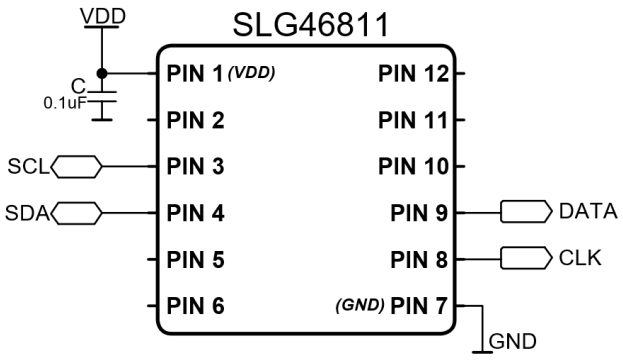


Design Steps

1. Configure EPG generator (see EPG technique) to create I2C Write command and set several SDA and SCL patterns.
2. Add one channel in EPG to check the ACK bit every 9th clock at SCL.
3. Create a 4-bit Multiplexer, as was described in Application: 8-bit Multiplexer. SEL0 and SEL1 choose the data (SDA) to be outputted.
4. Create frequency generator based on CNT1/DLY1 and LUT9, which performs as clocks for EPG.
5. Deglitch filter based on CNT0/DLY0 which triggers the I2C pattern.
6. Design monitors if the ACK bit check was present during the I2C command if the I2C slave doesn't respond, the transmission stops.

Application: I2C Programable Pattern Generator Using Shift Registers

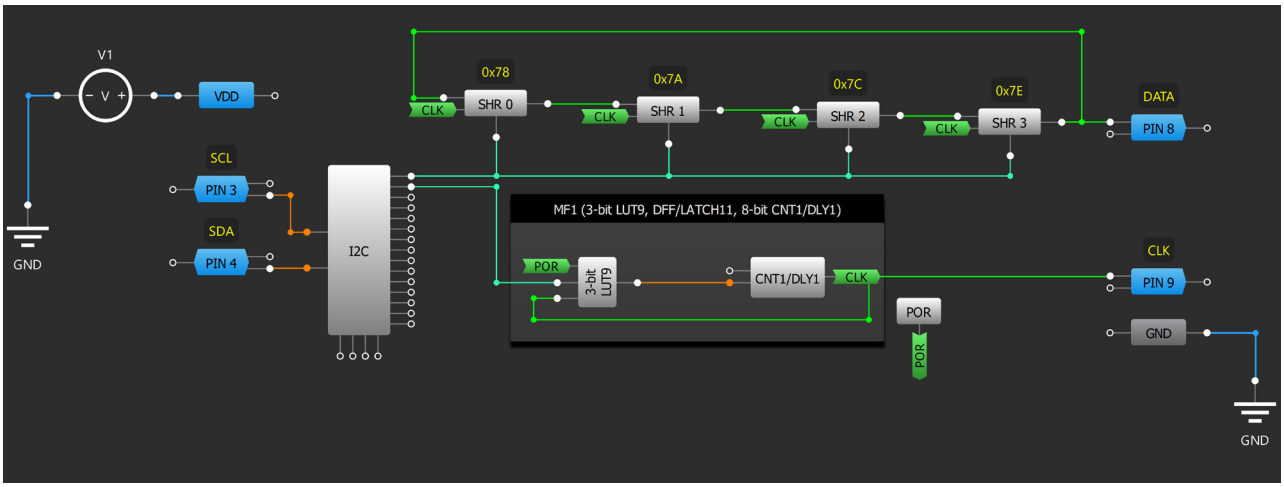
This application demonstrates the construction of a simple pattern generator with a capacity of up to 32 bits, utilizing the SLG46811 device. Such an approach can prove to be beneficial for cost-effective and energy-efficient applications.



Ingredients

- SLG46811V or any GreenPAK with Shift Registers

GreenPAK Diagram

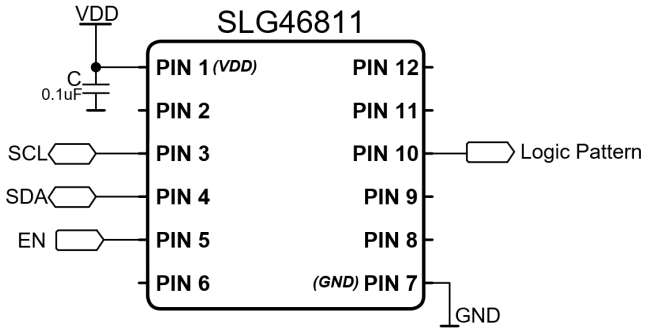


Design Steps

1. Configure Shift Register blocks and connect them serially.
2. Connect the output of SHR3 with the D input of SHR0
3. Create clock generator using CNT1/DLY1 and LUT9 in MF1
4. Use I2C virtual inputs to clear the SHR0-SHR3 and start sending the pattern
5. Add PIN9 to output CLK to distinguish the data

Application: Long Length Pattern Using EPG

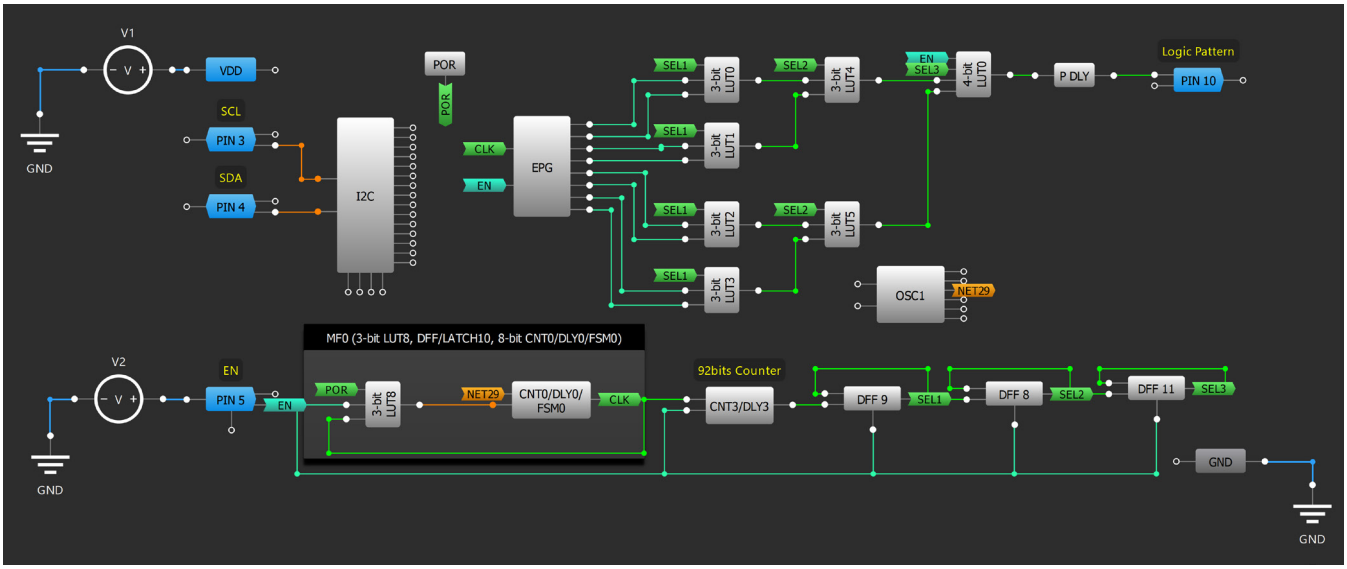
This application demonstrates the construction of a pattern generator with a capacity of up to 736 bits, utilizing the SLG46811 device. Such an approach can prove to be beneficial for cost-effective and energy-efficient applications.



Ingredients

- SLG46811V

GreenPAK Diagram

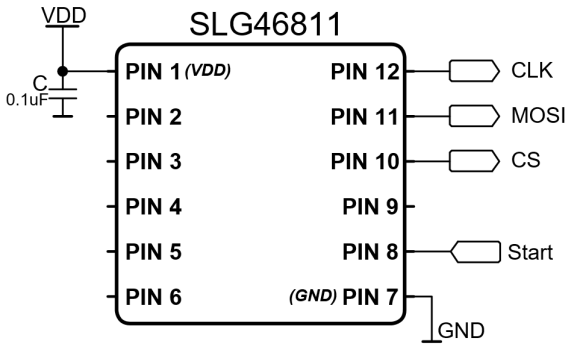


Design Steps

1. Configure the EPG generator (see EPG technique) and set the data that should be outputted
2. Create an 8-bit Multiplexer, as was described in Application: 8-bit Multiplexer, and connect to appropriate outputs of EPG
3. Create a frequency generator based on CNT0/DLY0 and LUT0, which performs clocks for EPG.
4. Configure CNT3/DLY3 to count 92 clocks generated by the frequency generator.
5. 3-bit counter-based DFF8, DFF9, and DFF11 select data outputted from EPG see Technique: Multiplexing a Bitstream.
6. Add EN PIN5 to start the pattern when HIGH and stop when LOW.

Application: Basic SPI Master

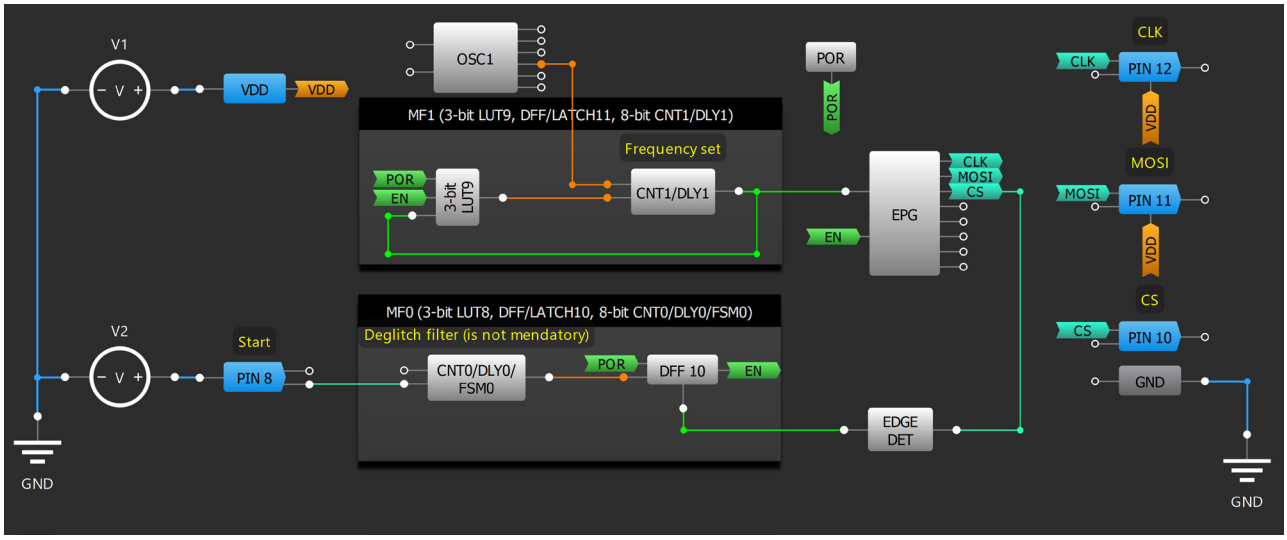
This application demonstrates how to build a simple SPI master using the SLG46811 device. With this device, the SPI master is capable of rewriting one or multiple bytes, making it a good solution for systems that do not require a complex implementation.



Ingredients

- SLG46811V

GreenPAK Diagram



Design Steps

1. Configure the EPG generator (see EPG technique) to create an SPI generator, choose the outputs for MOSI, SCL, and CS signals.
2. Create a frequency generator based on CNT1/DLY1 and LUT9, which performs as clocks for EPG.
3. Deglitch the filter based on CNT0/DLY0, which triggers the SPI pattern by clocking DFF10.
4. To reset DFF10, use EDGE DET with inverted output polarity. When CS goes high, EDGE DET performs inverted peak, which resets the EN signal to stop the pattern.

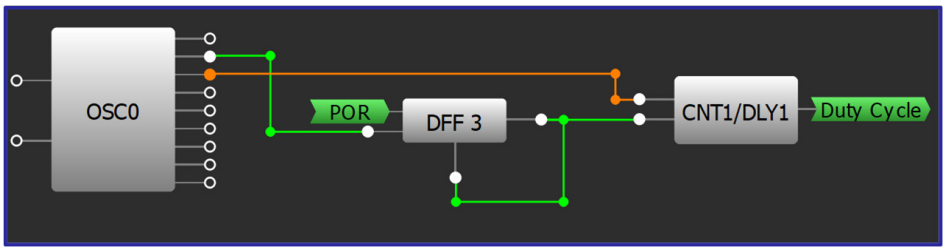
Chapter 6
Pulse-based Control

This chapter presents applications that control the pulse width of a signal. This most commonly involves PWM, which is commonly used for LED controllers, motor controllers, sound.

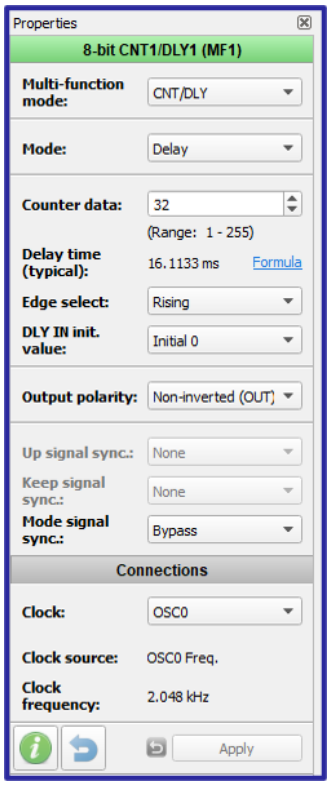
Technique: Setting a Constant Duty Cycle

This technique will work with any GreenPAK.

Setting an immutable duty cycle requires one CNT/DLY block, an oscillator, and a DFF. The macrocells should be configured as shown in figure below.



Simple Duty Cycle Configuration



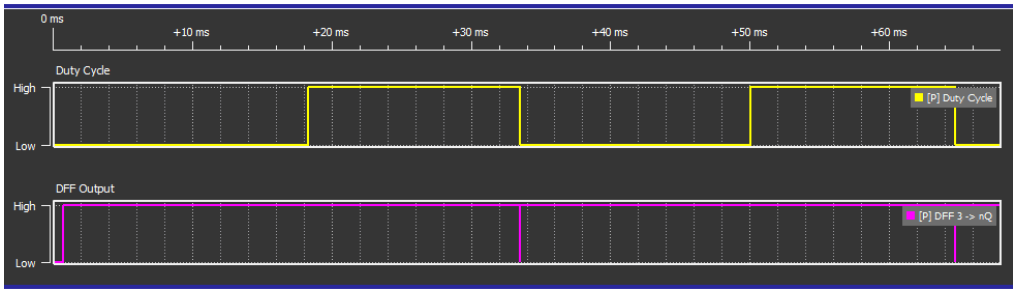
CNT/DLY Config

The oscillator determines the period, the DFF is a rising edge detector, and the CNT/DLY block determines the duty cycle. When the rising edge from the oscillator is registered by the DFF it will send a LOW pulse to the CNT/DLY block. This will set the CNT/DLY output LOW, and the output will only rise after the Delay Counter data has been met.

From the DFF's initial configuration change the Q output polarity to "Inverted (nQ)" and connect the output of the DFF. This will allow it to operate as a rising edge detector; it will remain HIGH until a rising edge is detected on the clock, whereupon it will briefly drop LOW. The FILTER/EDGE DET block can also be configured for this purpose.

The oscillator OUT0 or OUT1 is connected to the DFF's clock input to generate the period. The period should always be greater than the duty cycle. In this example the period, set by 'OUT1' second divider by, is set to "OSC/64."
The CNT/DLY block's Counter data option sets the duty cycle. The delay time sets the duration of the low signal.

The duty cycle is calculated as: $D=(T_{period} - T_{delay})/T_{period}$

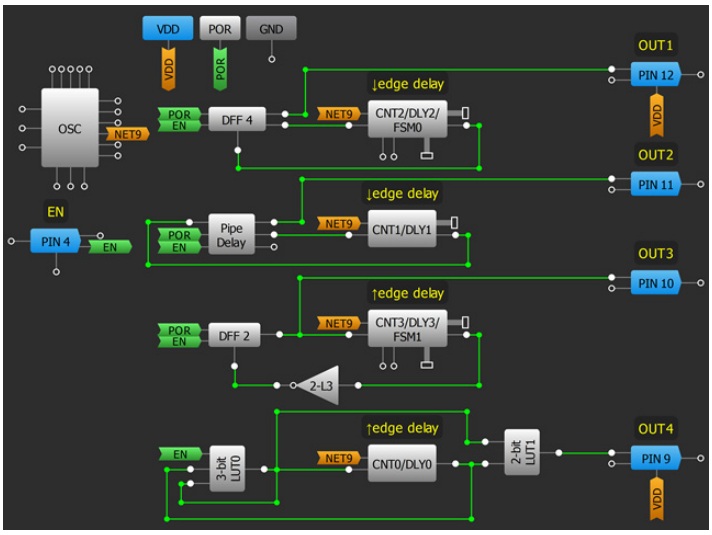


Simulation of Duty Cycle = 50%

Technique: One Shot Implementation

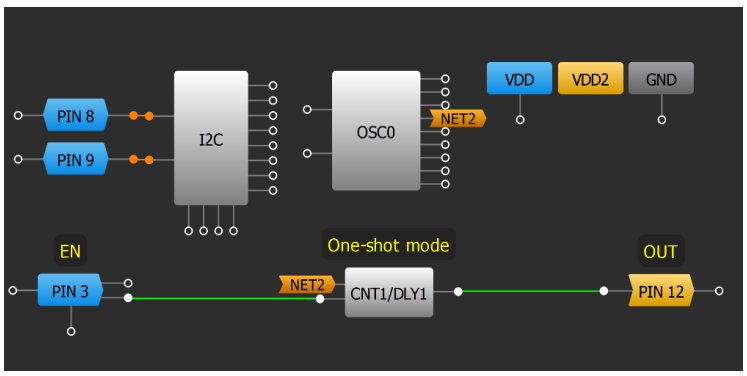
This technique can be used in any GreenPAK.

A one shot circuit generates an output pulse with a pre-defined duration. After the circuit produces a pulse, it returns to its stable state and produces no more pulses until it is triggered again. It is a very important component for reset functions, watchdog timers, and many other applications. One shot can be easily implemented in the GreenPAK. Figure below shows a few methods of creating a one shot impulse triggered by the rising edge of input EN. The pulse duration can be adjusted by changing the counter data value in the DLY blocks used.



Different One-shot Circuit Implementation

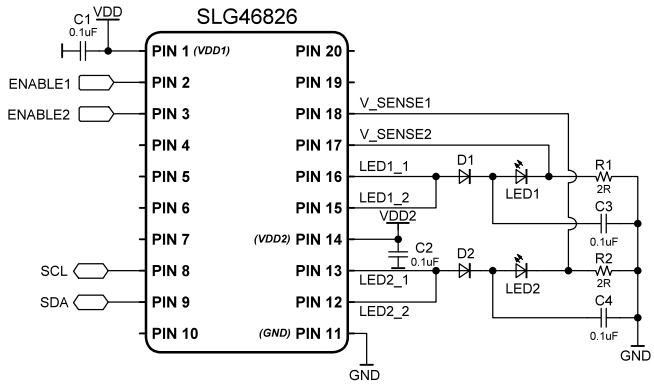
In many GreenPAKs (see figure below), implementing one shot only takes one DLY block. The only thing that the user needs to do is to switch the mode in block properties window to “One shot” and to select the edge it will detect. It can be either rising, falling, or both.



One-shot Implementation in SLG46826

Application: Constant Current LED Driver

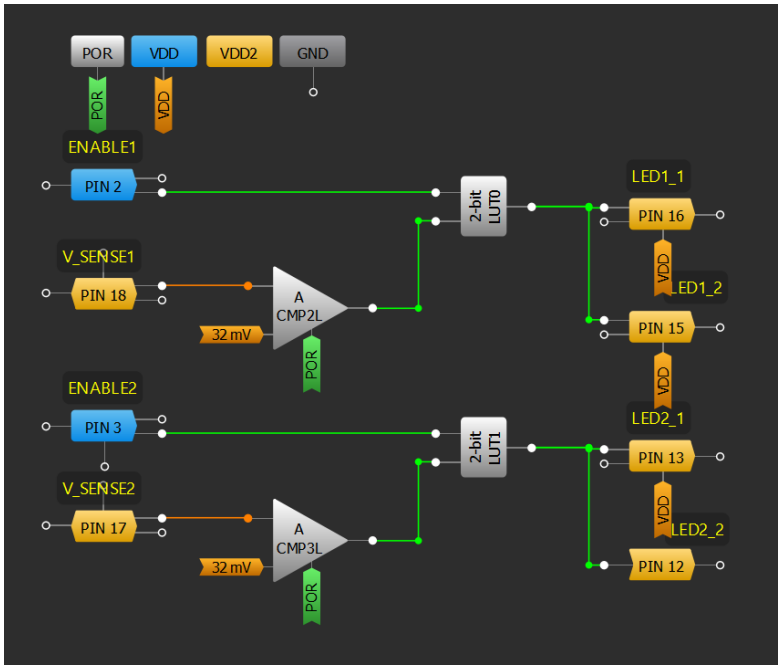
This application detects the presence of a cable by measuring the voltage proportional to its resistance. The ACMP's 100uA current source is used to produce the voltage drop across the cable. This configuration is also able to determine which type of connection is being made based on different wire lengths or connected loads.



Ingredients

- Any GreenPAK
- Four capacitors
- Four diodes (2 LEDs, 2 silicon diodes)
- Two resistors

GreenPAK Diagram

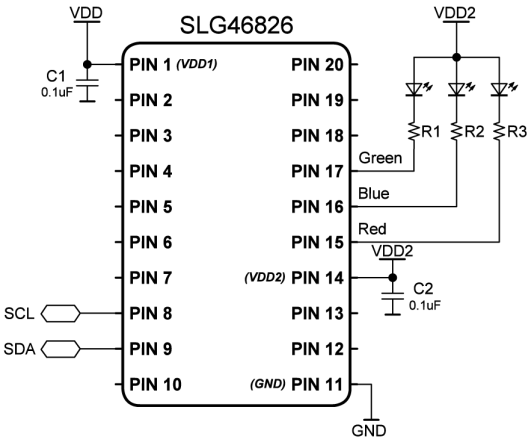


Design Steps

1. Configure two ACMPs, each with their IN- source set to the desired threshold.
2. Configure LUTs to enable the LED outputs.
3. Connect LED1_1 and LED1_2 to the anode of a silicon diode and connect the cathode of the silicon diode to the anode of an LED.
4. Connect a resistor between the cathode of the LED and ground.
5. Connect a capacitor between the anode of the LED and ground.
6. Repeat steps 3-5 for the LED2 outputs.
7. Connect V_SENSE1 and V_SENSE2 to the cathode of LED1 and LED2 respectively.

Application: RGB LED Control via I2C

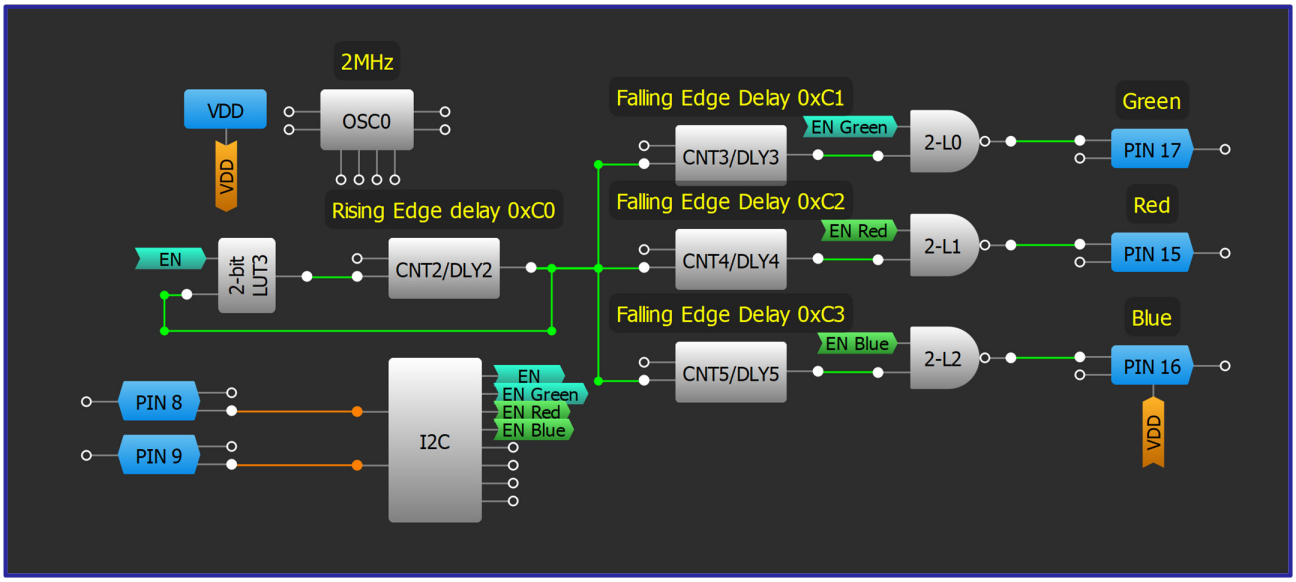
RGB LEDs are used to add complexity to LED indication systems and can be controlled with a GreenPAK. I2C is used in this application as an easy way to change the duty cycle to produce different colors.



Ingredients

- Any GreenPAK with I2C
- RGB LED
- Three resistors

GreenPAK Diagram



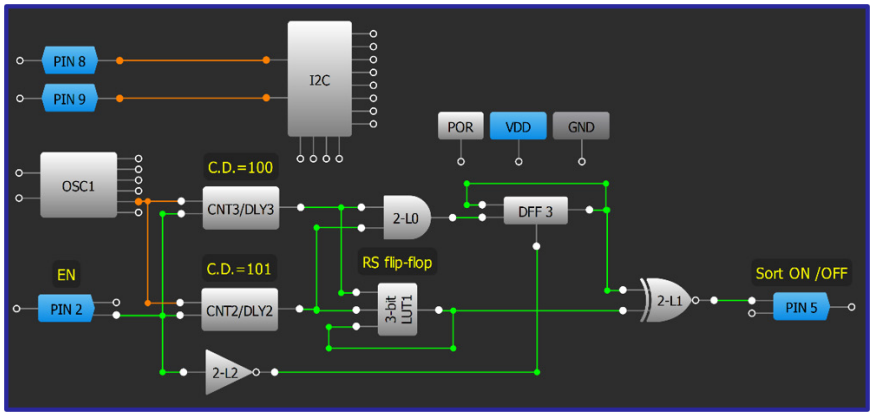
Design Steps

1. Configure GPIO pins as open-drain outputs for RGB cathode connection.
2. Add LUT logic and CNT/DLY2 to create a generator with EN signal.
3. Configure a CNT/DLY block to rising-edge delay.
4. Add and configure LUTs for each output using Technique: Configuring Standard Logic w/ LUT Macrocells.
5. Connect each LUT output to the desired output pins.
6. I2C virtual inputs can be changed individually or simultaneously using the I2C virtual output address.
7. Counter data of CNT/DLY blocks can be changed individually or simultaneously using the I2C.

Technique: Creating a Breathing LED Pattern

This technique can be used within any GreenPAK. The quantity of independent Soft ON/OFF channels depends on the number of counters available within the particular part.

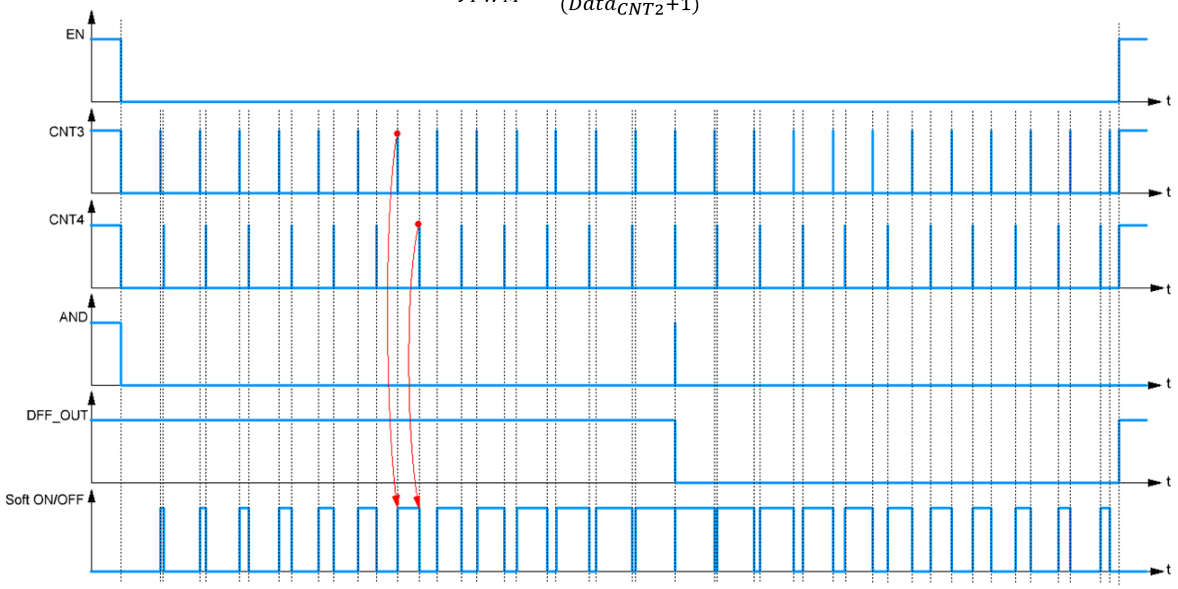
A breathing LED pattern can be generated through a consistent difference in pace between two counters. Each counter outputs a high pulse for one clock cycle of their programmed period. Two CNT/DLY blocks are programmed with different counter data settings to provide a small offset between their outputs. These output signals are used to set and reset a flip-flop within the device. The figure below depicts a basic implementation, wherein CNT2/DLY2 sets the ON period and CNT3/DLY3 sets the duty cycle.



LED Breathing Implementation

In the implementation within the figure above the frequency of the PWM is set by CNT2 and can be calculated with the equation:

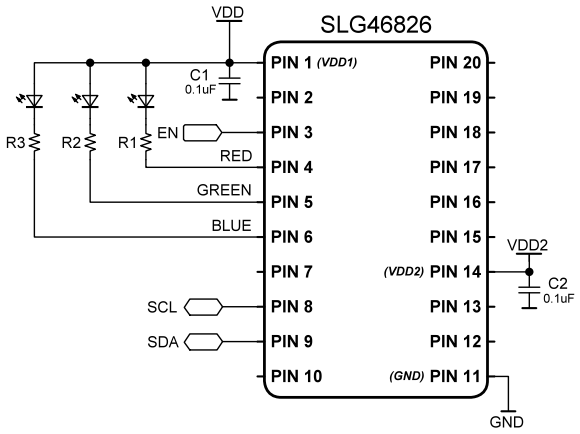
$$f_{PWM} = \frac{fosc}{(Data_{CNT2}+1)}$$



The effect of the small offset is shown through the waveforms of the figure below. The PWM cycle ends when the counters' outputs coincide. This causes a short high impulse on AND gate and DFF flops. The NXOR gate makes the inversion of the PWM, which provides a soft OFF. PIN2 is the enable signal and while it is HIGH the counters are in high level reset.

Application: Breathing RGB LED

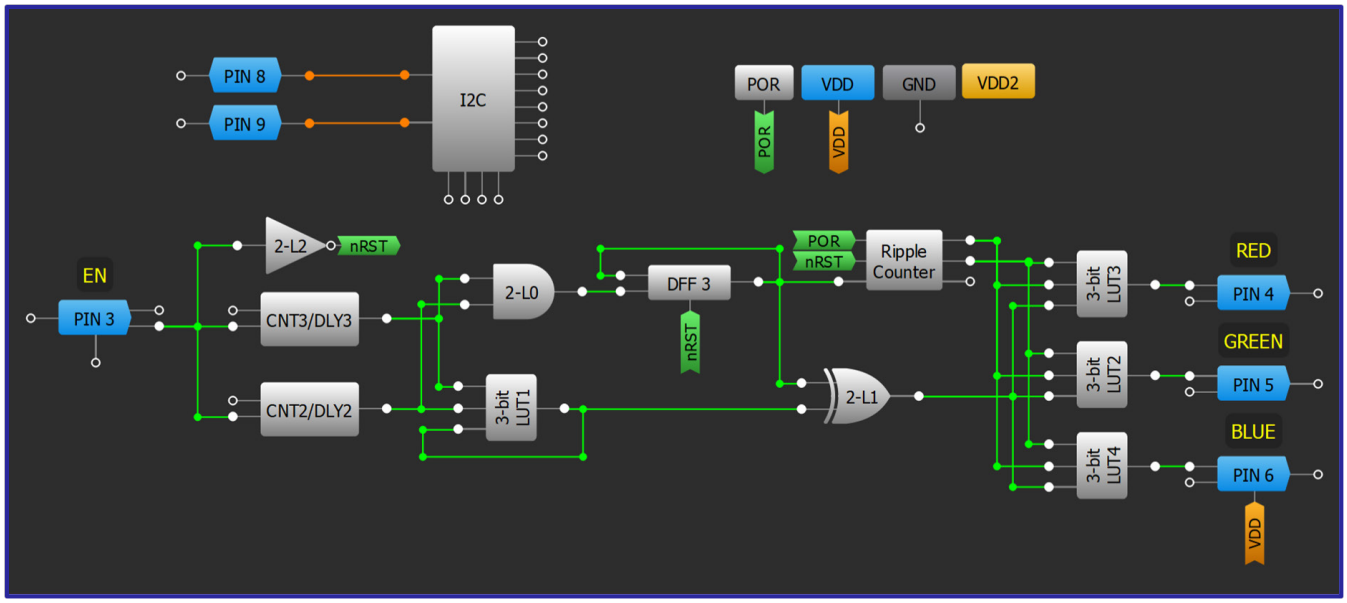
RGB LEDs are used to add complexity to LED indication systems and can be controlled with a GreenPAK. They can be paired with a soft ON/OFF circuit for a breathing pattern.



Ingredients

- Any GreenPAK
- One RGB LED
- Three resistors

GreenPAK Diagram

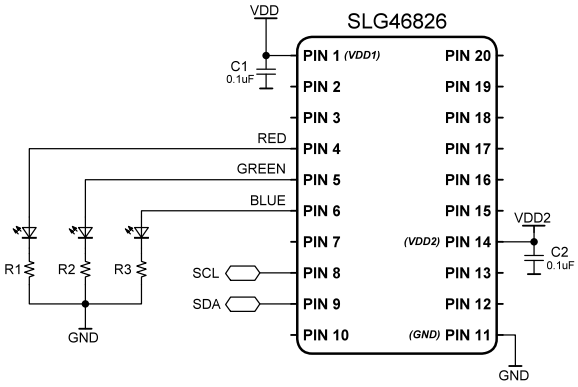


Design Steps

1. Configure GPIO pins as open drain NMOS outputs.
2. Create soft ON/OFF circuit as shown in Technique: Creating a Breathing LED Pattern.
3. Configure Ripple Counter - set Functionality mode to Range: SV-EV cycles (SV=1, EV=3).
4. Configure LUTs collectively as a demultiplexer.
5. Add enable (EN) signal to start/stop RGB breathing.

Application: Breathing RGB LED Control with I2C

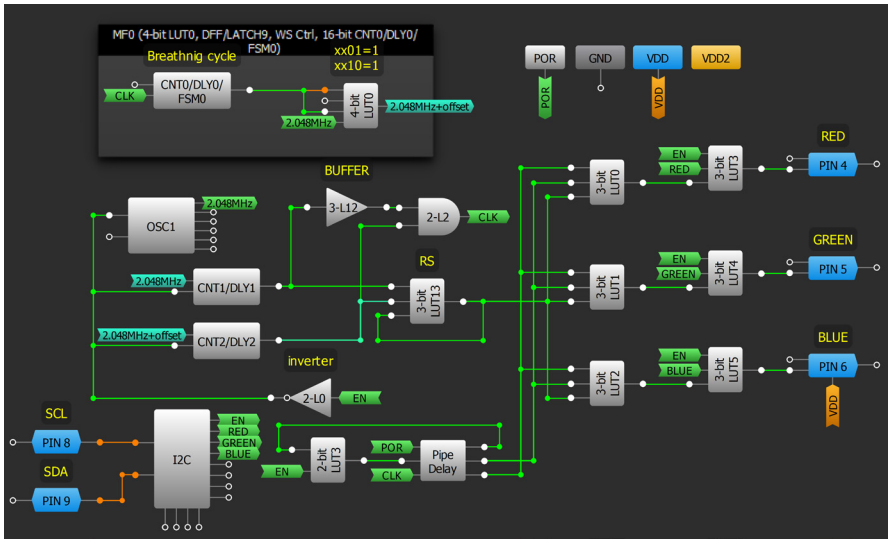
RGB LEDs are used to add complexity to LED indication systems They can be paired with a soft ON/OFF circuit for a breathing pattern and further controlled with I2C. Changing the CNT0 counter data changes the breathing period.



Ingredients

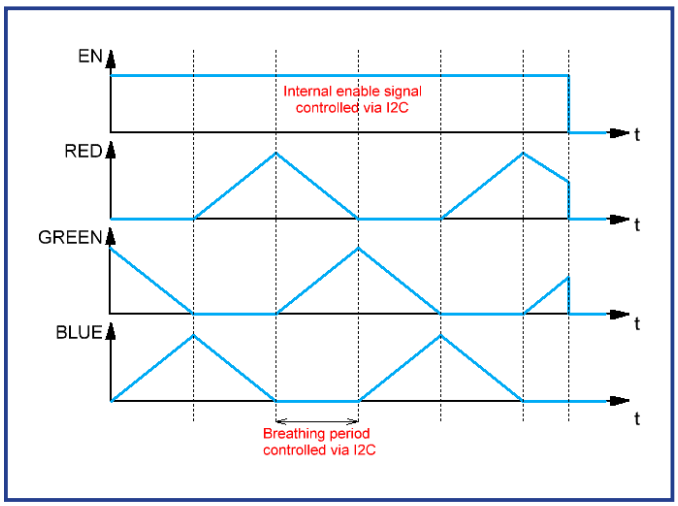
- Any GreenPAK with I2C
- One RGB LED
- Three resistors

GreenPAK Diagram



Design Steps

1. Create a soft ON circuit as shown in Technique: Creating a Breathing LED Pattern but instead set the counter data for CNT1 and CNT2 to the same value.
2. Add MF0 which adds a small offset between CNT1 and CNT2.
3. Configure LUT3-LUT5 as a multiplexer switched by the EN signal controlled via I2C.
4. Configure LUT0-LUT2 as a demultiplexer which passes the breathing signal according to the timing diagram.

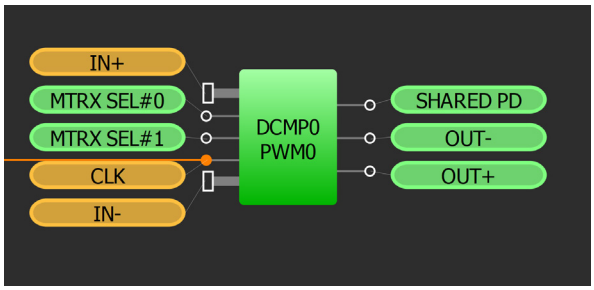


Technique: Using DCMP/PWM Macrocell in PWM Mode

This technique is for the DCMP block, available in the SLG46140, SLG46620, and SLG46621.

Overview of the DCMP/PWM Macrocell

The DCMP/PWM macrocell is used to compare two 8-bit values or generated PWM signals. There are three DCMP/PWM blocks per IC that can operate independently, and each DCMP/PWM has two 8-bit inputs (IN+, IN-) that can be used to generate a PWM signal. Inputs MTRX SEL#0 and MTRX SEL#1 are used during static PWM generation to select one of the four available registers. Input SHARED PD is used to power on or off the device. The PWM output duty cycle range can be configured to range from 0% to 99.61% or 0.39% to 100%.

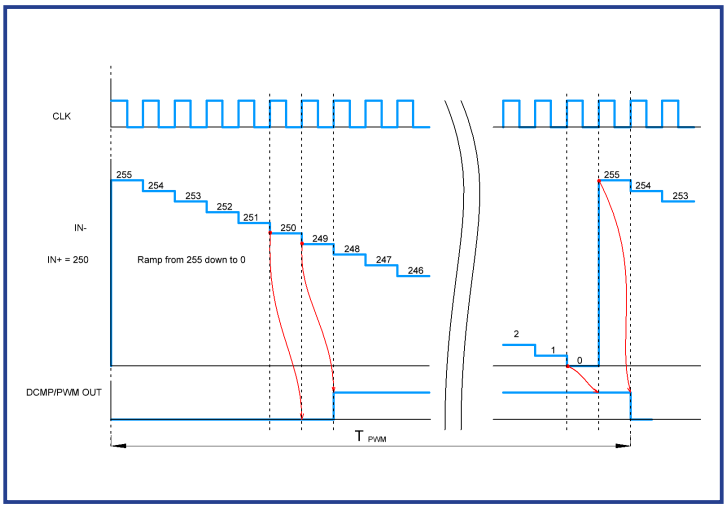


Creating the PWM Signal

One input of the PWM generator is a linearly cycled ramp of data. This is from a counter that counts from 255 down to 0 or vice versa.

The other input should be stable for at least 1 PWM signal period (PWM ramp counter period). It could be data from the SPI, ADC, FSM blocks, or from an internal register of the DCMP/PWM.

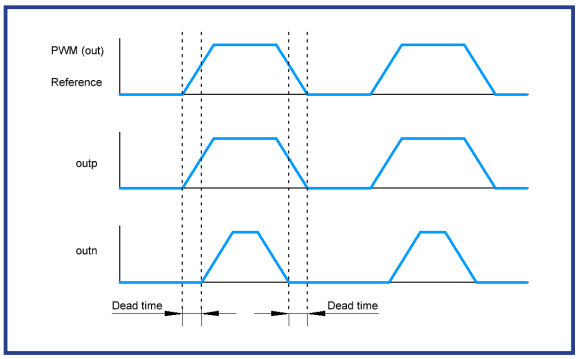
The figure shown to the right displays the operation of a DCMP/PWM when IN- is connected to a CNT/DLY block that counts from 255 down to 0 (PWM ramp counter) and the IN+ source is an internal register set to 250.



DCMP/PWM in PWM mode

The IN+ setting is the key to the operation of the macrocell. Static PWM values can be made using the internal registers. PWM dynamic feedback can be made using the ADC. An MCU-controlled PWM can be set by using the SPI interface.

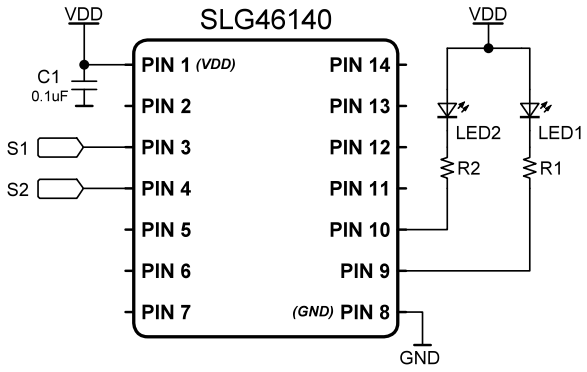
The output OUT- and OUT+ has dead band time that can be from 10 to 80 ns and set in the properties pane of the DCMP/PWM.



Deadband time OUT- and OUT+

Application: PWM Selection

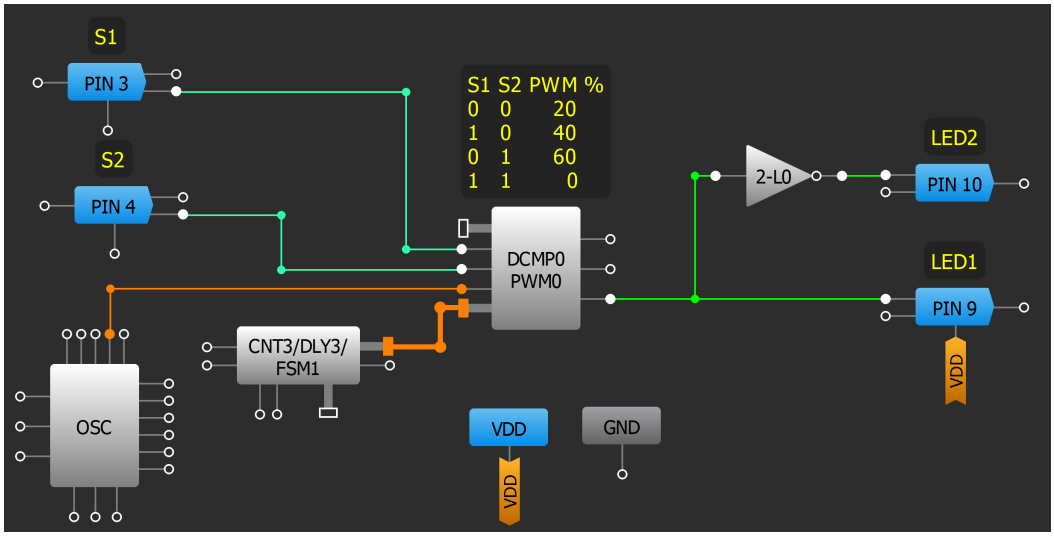
PWM selection is commonly used for functions like adjusting LED brightness and controlling fan speed. In this implementation two LEDs are adjusted to discrete brightness levels based upon the input of two switches.



Ingredients

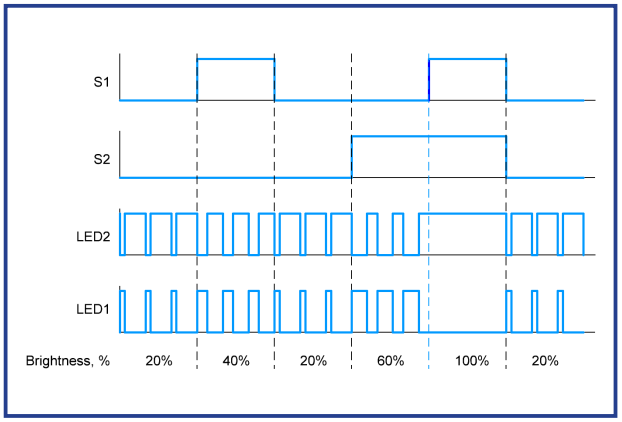
- GreenPAK with DCMP
- No other components are needed

GreenPAK Diagram



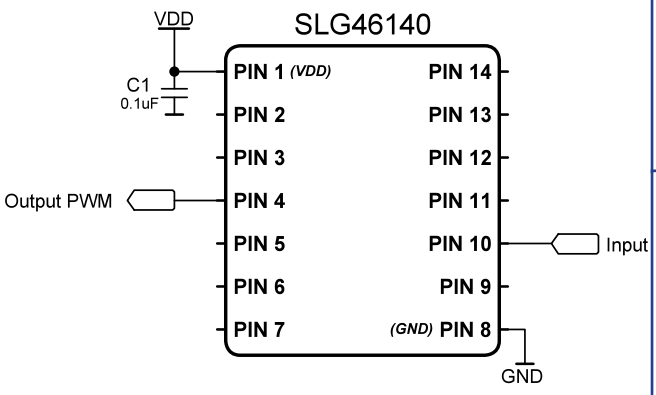
Design Steps

1. Enable the DCMP by removing VDD from SHARED PD input. Set DCMP/PWM power register to “Power on”. Set IN+ selector to “Register selected through the matrix.” Set IN- selector to “FSM1 [7:0].” Set the registers of DCMP0: register 0 – 51; register 1 – 102; register 2 – 154; register 3 – 0.
2. Add CNT/DLY block configured as Counter/FSM. Set counter data to 255.
3. Configure RC OSC power mode to “Force power on.”
4. Configure PIN10 and PIN9 to open drain NMOS.
5. Add LUT as an inverter.
6. Connect input pins to the MTRX SEL pins of a DCMP/PWM block.



Application: PWM Generator Using ACMP and DAC

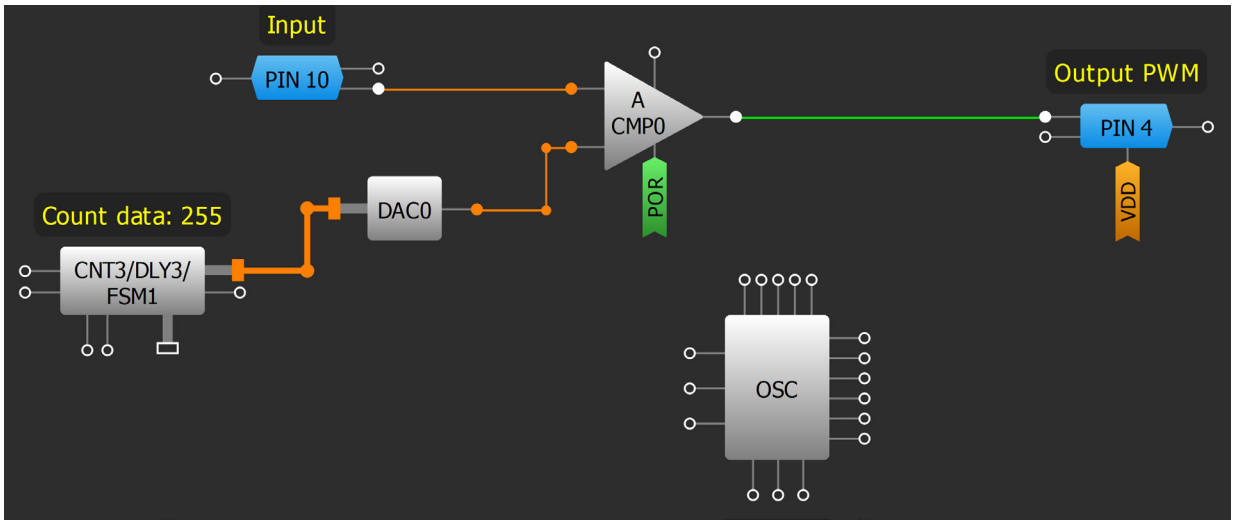
PWM generators can be used to control devices such as DC motors and LEDs. This implementation uses an analog signal that an ACMP compares with the signal of DAC0. CNT3 is used to generate the value for the DAC.



Ingredients

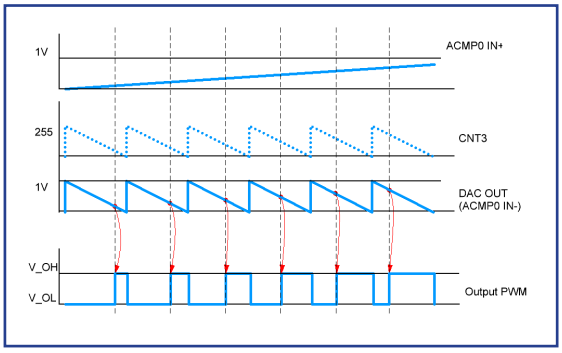
- GreenPAK with DAC
- No other components are needed

GreenPAK Diagram



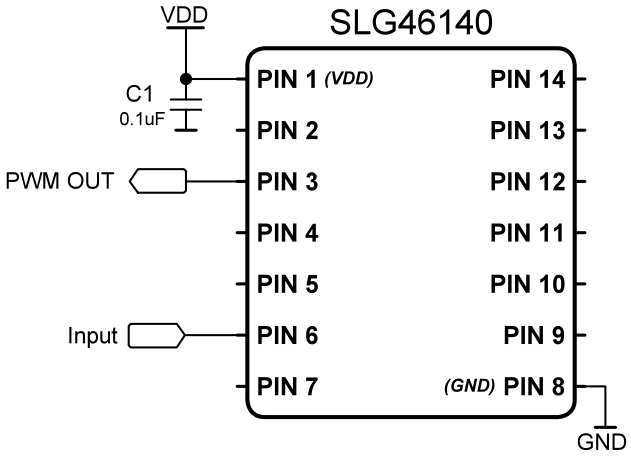
Design Steps

1. Connect ACMP0 PWR UP input to POR. Set IN- source to "Ext. Vref (DAC0 out)."
2. Set DAC0 power on signal to "Power on" and Input selection to "From DCMP1's input."
3. Configure FSM-compatible CNT/DLY to "Counter/FSM" mode with counter data = 255.
4. Set RC OSC power mode to "Force power on."



Application: PWM Generator Using ADC

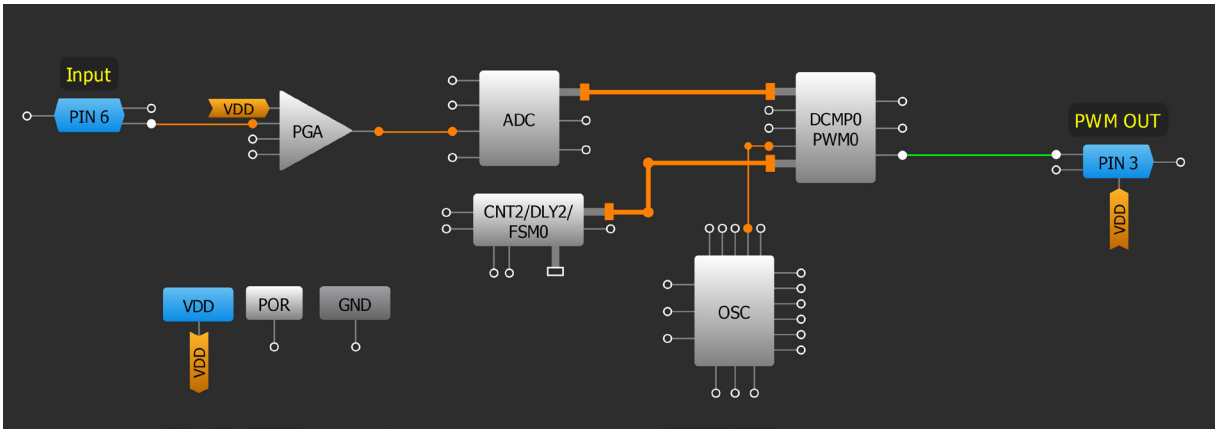
PWM generators can be used to control devices such as DC motors and LEDs. This implementation uses an analog signal connected to an ADC to compare with the value of CNT2 in PWM0. If the CNT2 value is less than the digitized analog signal, the output of PWM0 is high. After CNT2 value is 0 the output of PWM0 is low.



Ingredients

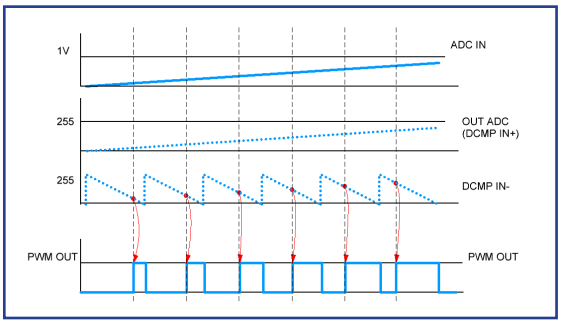
- GreenPAK with ADC
- No other components are needed

GreenPAK Diagram



Design Steps

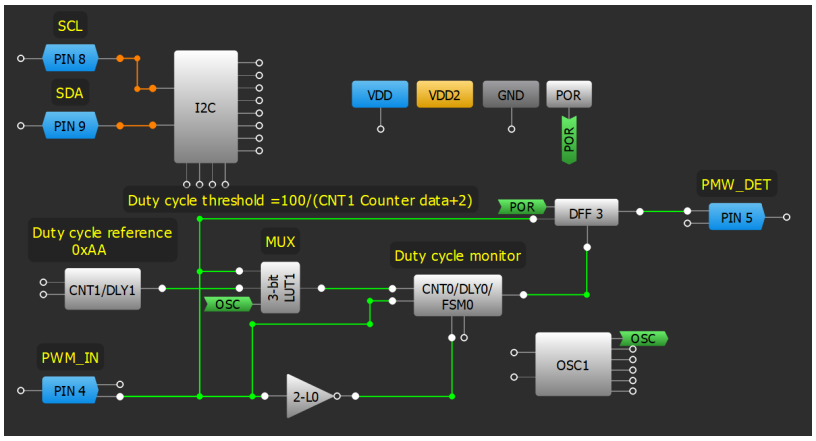
1. Remove VDD from the ADC's PWR DOWN input and set PGA Power on signal to "Power on."
2. Set PIN6 to "Analog input/output."
3. Configure DCMP0/PWM0 by deleting VDD from SHARED PD input. DCMP/PWM power register set Power on. Check that IN+ selector connect to "ADC [7:0]" and IN- selector connect to "FSM0 [7:0]."
4. Configure 4-bit LUT1/14-bit CNT2/DLY2/FSM0 as "Counter/FSM" with counter data equal to 255.
5. Connect DCMP0/PWM0 OUT+ to output pin.



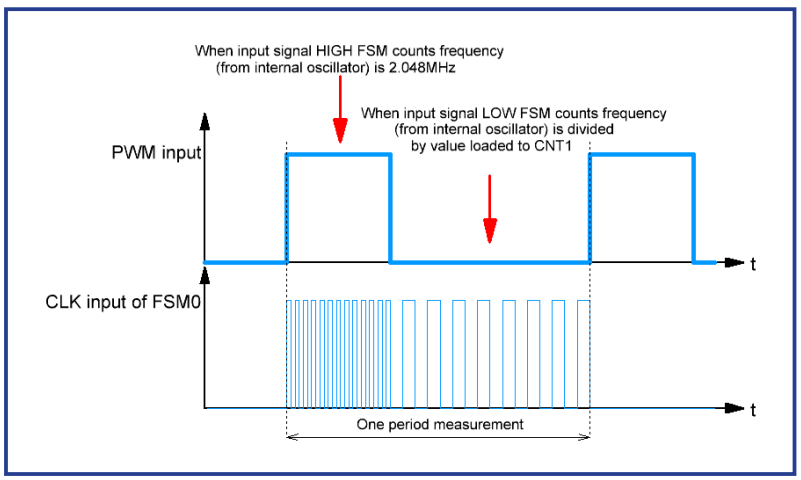
Technique: Duty Cycle Detection

This technique can be used within any GreenPAK. Since the input frequency range is limited by the maximum FSM counter data, it is better to use a 16-bit FSM. The PWM detection input frequency should be much slower than the duty cycle reference frequency to improve the accuracy.

Duty cycle detection is important for functions like overload protection, DC/DC conversion, servo motor control, and protocol detection. This design can be easily implemented using a GreenPAK with an FSM block (For the below example a SLG46826 chip is used, see the figure below).



In the implementation shown above, when PIN4 goes HIGH, FSM0 starts counting DOWN, clocked by the internal oscillator. FSM0 is set to 65535 by a rising edge on PIN4. When PIN4 goes LOW, FSM0 starts counting UP with the frequency from internal oscillator divided by CNT1's data value. If FSM0 reaches 65535, DFF3 will be set LOW by the next edge rising edge from PIN4, which flags that the duty cycle is below the set threshold.

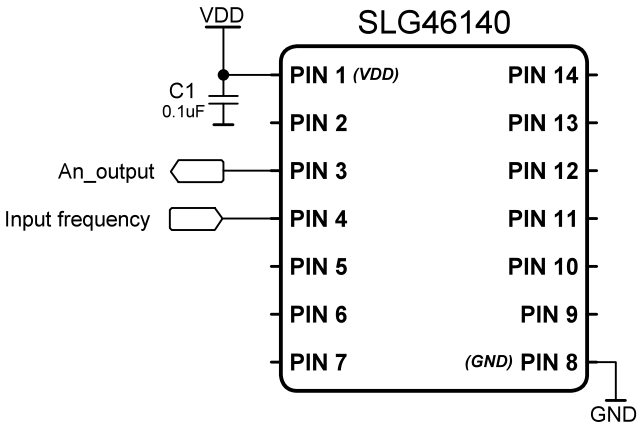


The duty cycle reference frequency can be adjusted by changing the CNT1 counter data value via I2C. To calculate the duty cycle threshold, the following formula is used:

$$Duty\ Cycle\ Threshold = \frac{100}{\text{CNT1 Counter data} + 2}\ (\%)$$

Application: Frequency to Analog Voltage Converter

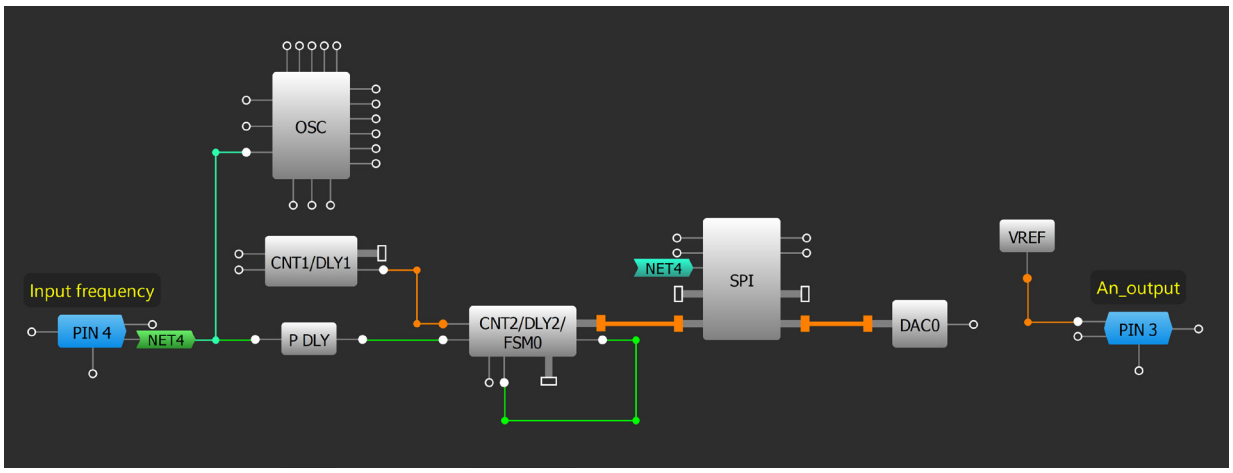
PWM generators can be used to control devices such as DC motors and LEDs. This implementation uses an analog signal connected to an ADC to compare with the value of CNT2 in PWM0. If the CNT2 value is less than the digitized analog signal, the output of PWM0 is high. After CNT2 value is 0 the output of PWM0 is low.

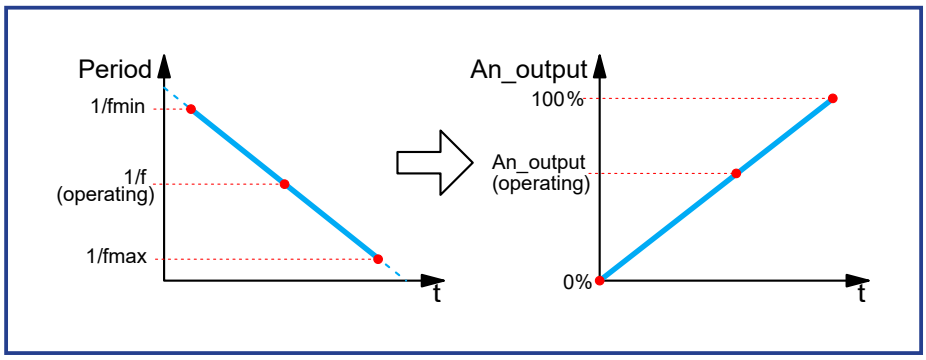


Ingredients

- Any GreenPAK with SPI and a DAC

GreenPAK Diagram





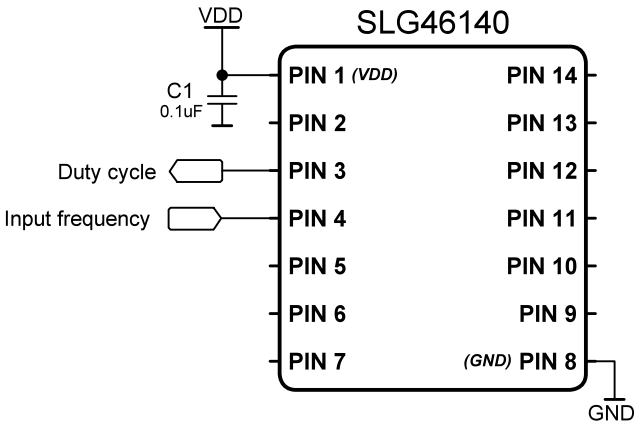
Design Steps

1. Configure SPI to the "ADC/FSM Buffer" mode, change the PAR input data source to "FSM0 [15:8]."
2. Configure FSM0 block to "Set (counter value)," change the clock source to CNT1.
3. Configure the DAC Input selector "From DCMP1's input" and VREF Source selector to "DAC0 out."
4. To find input frequency range and output analog voltage use the following formulas:

$$f_{min} = \frac{f_{osc}}{(CNT1+1)\cdot FSM0} \qquad f_{max} = \frac{f_{osc}}{(CNT1+1)} \qquad An\_output_{operating} = V_{ref\_max} - \frac{f_{min}\cdot V_{ref\_max}}{f_{operating}}$$

Application: Frequency to Duty Cycle Converter

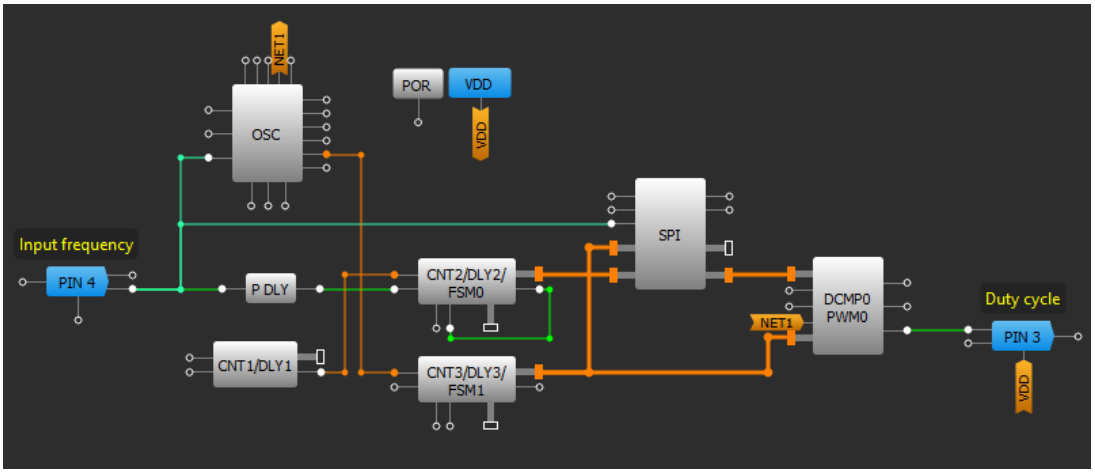
This application can be used to convert input frequency to a certain duty cycle. The input frequency is within some predefined range and can be changed by adjusting the counters. The output PWM frequency is constant but can be changed according a given requirement.

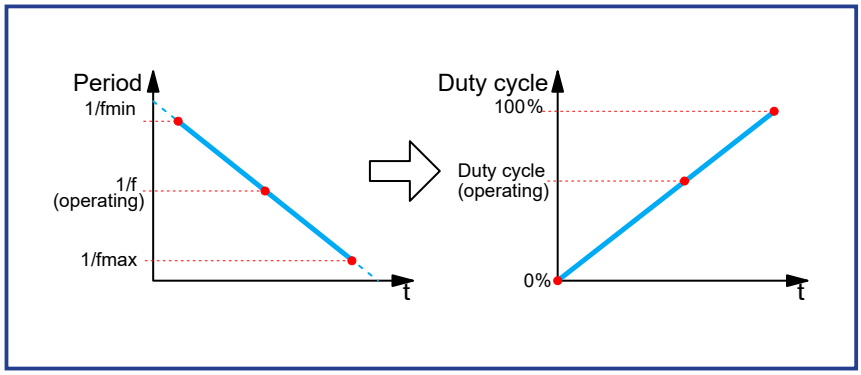


Ingredients

- Any GreenPAK with SPI and DCMP

GreenPAK Diagram





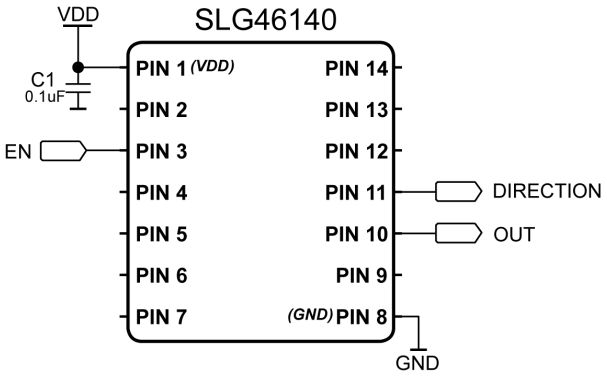
Design Steps

1. Configure SPI to the "ADC/FSM buffer" mode and change the PAR input data source to "FSM0[15:8] FSM1[7:0]."
2. Configure FSM0 block to "Set (counter value)", change the clock source to "8-bit CNT1/DLY1 (OUT)."
3. Configure DCMP0 to compare "SPI [15:8]" data" and "FSM1 [7:0]" data.
4. Calculate the input frequency range and operating duty cycle using the following formulas:

$$f_{min} = \frac{f_{osc}}{(CNT1+1)\cdot FSM1} \qquad f_{max} = \frac{f_{osc}}{(CNT1+1)} \qquad Duty\ Cycle_{operating} = \left(1 - \frac{f_{min}}{f_{operating}}\right) \cdot 100\%.$$

Application: Linear Frequency Modulation

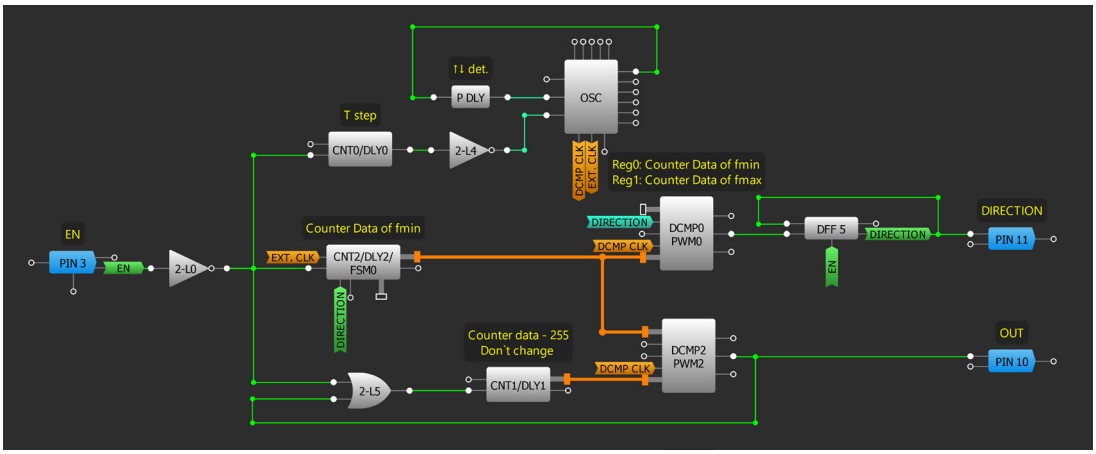
This application can be used to produce a frequency, which can be changed gradually from fmin to fmax and vice versa during a specific time. The frequency does not change linearly for large frequency range.

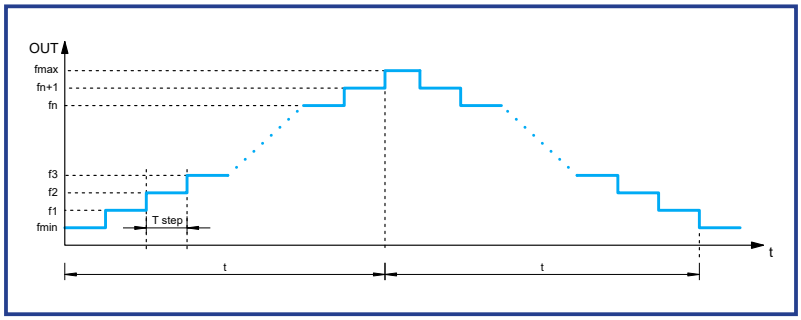


Ingredients

- Any GreenPAK with DCMPs

GreenPAK Diagram





Design Steps

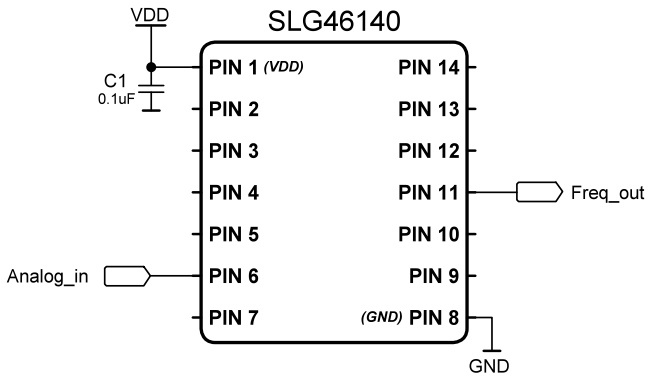
1. Configure desired GPIO pins.
2. Add, connect, and configure LUTs, P DLY, DFF, DCMPs and CNT/DLY/FSM blocks as shown above.
3. Counter data for minimum and maximum frequency and the period of rising/falling are calculated as follows:

$$Counter\ Data_{\max\,(min)} = 255 - \left(\frac{f_{osc}}{f_{\max\,(min)}} - 1\right) \qquad T_{step} = \frac{t \cdot f_{max} \cdot f_{min}}{f_{osc} \cdot (f_{max} - f_{min})}$$

4. Set the appropriate counter data for particular DCMPs selector.
5. Set the appropriate CNT/DLY0 counter data for step time Tstep.

Application: Voltage-Controlled Oscillator

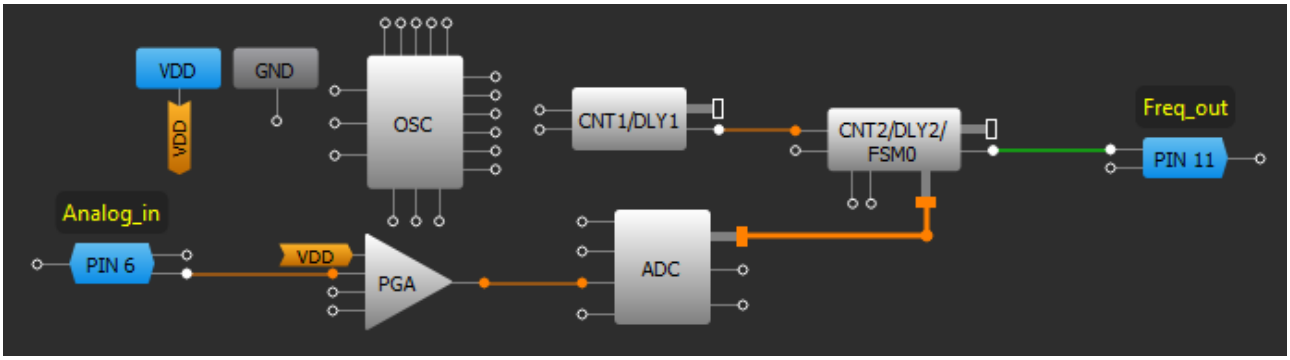
This application can be used to produce a frequency, which can be changed gradually from fmin to fmax and vice versa during a specific time. The frequency does not change linearly for large frequency range.

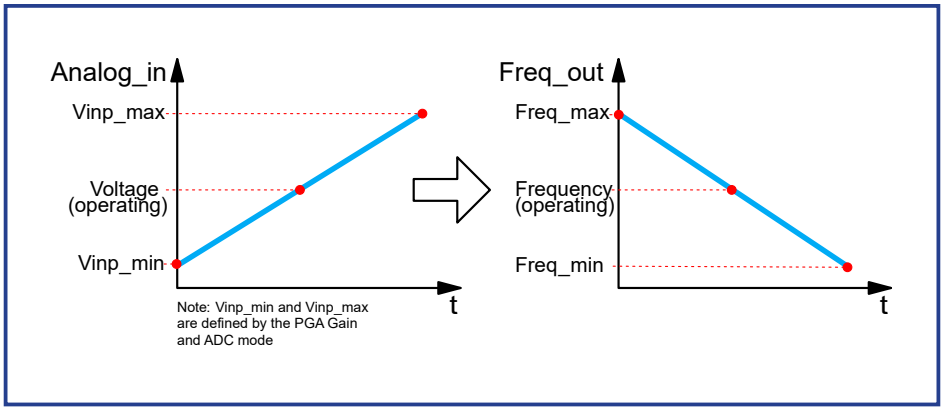


Ingredients

- Any GreenPAK with ADC

GreenPAK Diagram





Design Steps

1. Configure PIN6 as "Analog input/output."
2. Change the connections of FSM0: FSM data source to "ADC" and Clock to "8-bit CNT1/DLY1(OUT)."
3. Power up the ADC block and change the PGA Gain from "x0.25" to "x1."
4. In order to find output frequency range and operating frequency use the following formulas:

$$fmax = \frac{fosc \cdot Vinp\_max}{2 \cdot (CNT1+1)} \qquad fmin = \frac{fosc}{256 \cdot (CNT1+1)} \qquad Frequency(operating) = fmax \cdot \frac{Vinp\_max}{Voltage(operating)}$$

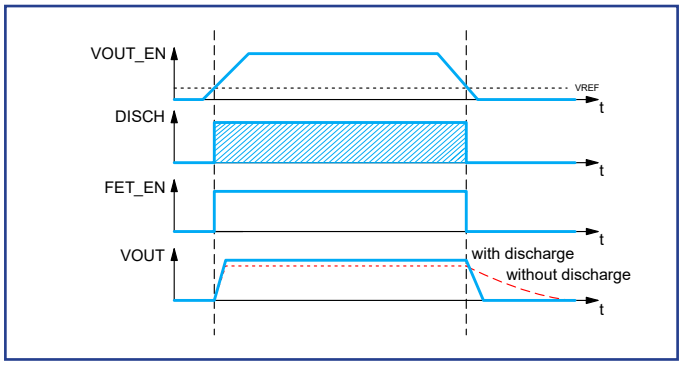
Chapter 7
Power Management

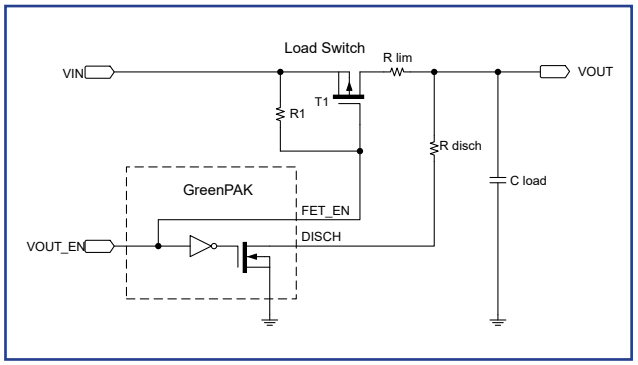
This chapter presents applications that manage the power usage of an electronic system. Some applications that involve power management are charge pumps, LDOs, discharge circuits.

Technique: Output Discharge

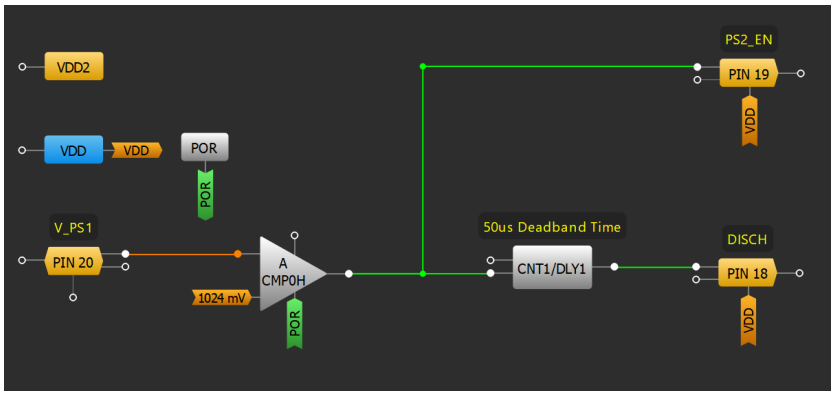
This technique can be used within any GreenPAK.

An output discharge circuit is a method to prevent an output pin from floating or a brownout of the system. It ensures that the output pin is set to a known "zero" state when there is a condition that should disable the device but that there is no leakage during operation.





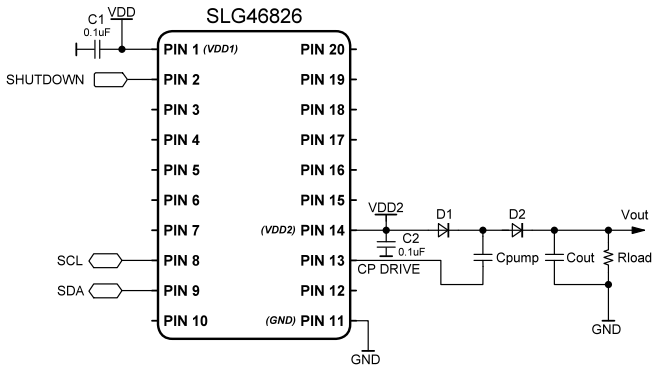
The figure above shows the implementation of a quick discharge circuit for a load switch output VOUT. An open drain NMOS output in the GreenPAK has it's drain connected to VOUT. The gate of the NMOS is inverted within the GPIO structure, so there is no need to invert the VOUT_EN signal within the matrix. When VOUT_EN is HIGH, the load switch is turned on and the discharge path to GND is open to prevent leakage. When VOUT_EN is LOW the load switch is turned off and the discharge path to GND is closed to quickly discharge VOUT. To limit the current through the FETs and control the discharge two resistors Rlim and Rdisch are added.



The figure above shows an example of a conditional output discharge circuit within a power sequencer that monitors the level of the power supply that preceded it. Configure PS2_EN as a push pull output and DISCH as an open drain NMOS output. When V_PS1 drops below 1024mV, this circuit will turn off the power supply and will discharge the output to GND. A 50µS deadband time is added with CNT/DLY1.

Application: Charge Pump

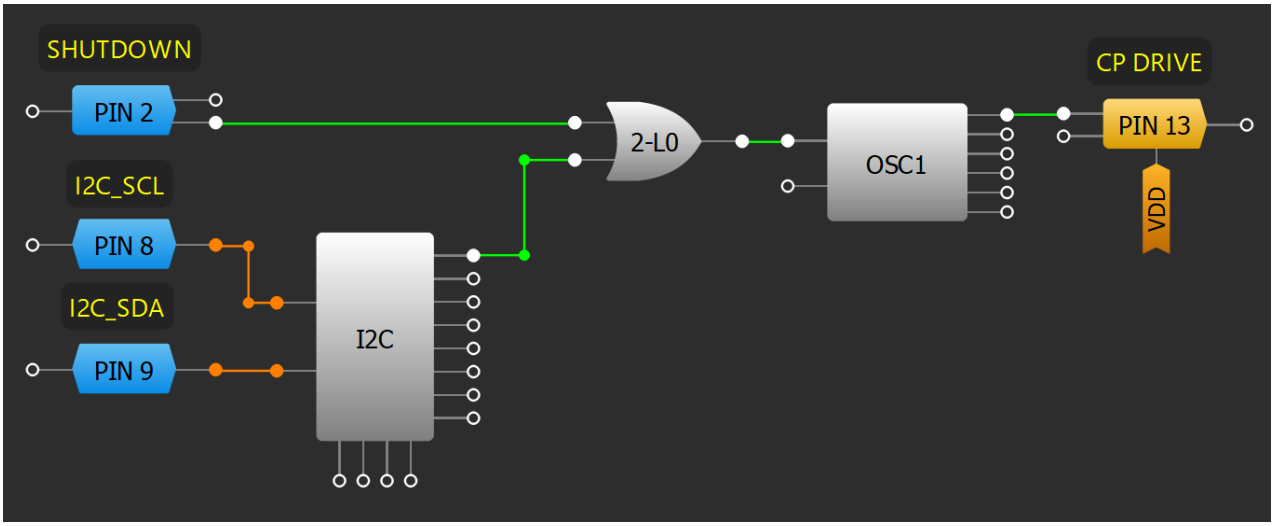
A charge pump is a DC-DC converter that uses capacitors for energetic charge storage to create different voltage levels. It can be used to provide additional voltage levels for powering specific interface circuits, sensors etc. Schottky diodes are recommended for best performance.



Ingredients

- Any GreenPAK
- Two capacitors
- Two diodes

GreenPAK Diagram

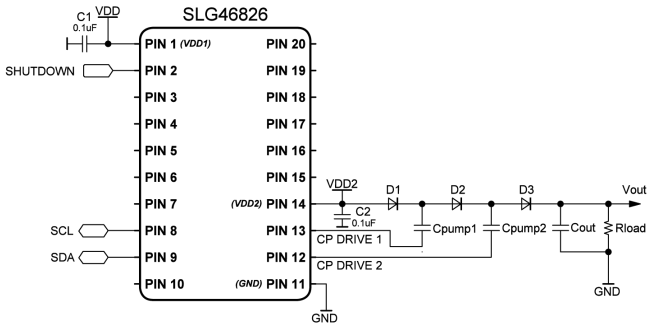


Design Steps

1. Set the divider in OSC1 obtain the desired output frequency.
2. Configure logic to provide a shutdown function, either through IO or I2C (if available)
3. Connect a diode (D1) and Cpump between VDD and CP_DRIVE.
4. Connect the anode of D2 to the cathode of D1.
5. Connect Cout and Rload in parallel between the cathode of D2 and ground.

Application: Two-Stage Charge Pump

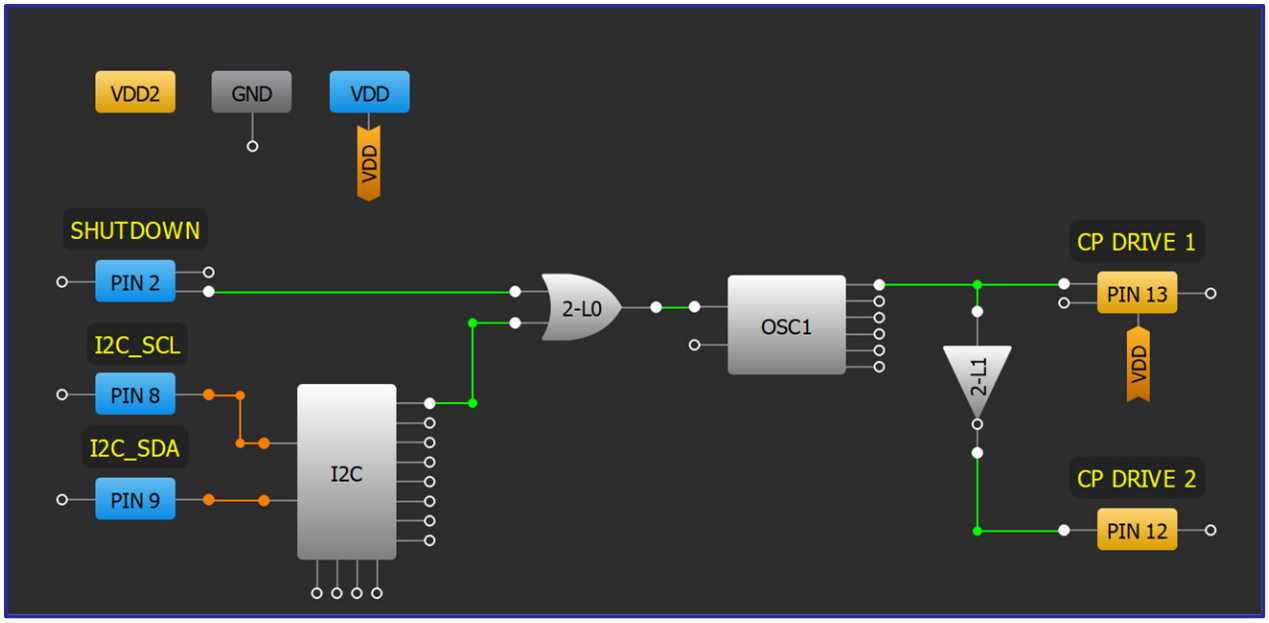
A charge pump is a DC-DC converter that uses capacitors for energetic charge storage to create different voltage levels. It can be used to provide additional voltage levels for powering specific interface circuits, sensors etc. Schottky diodes are recommended for best performance.



Ingredients

- Any GreenPAK
- Three capacitors
- Three diodes

GreenPAK Diagram

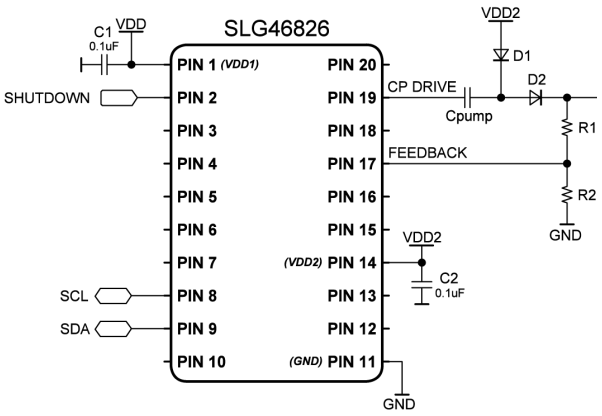


Design Steps

1. Create a basic charge pump design using the steps in Application: Charge Pump.
2. Configure logic to add an inverter. Connect the inverter between an output pin and the oscillator.
3. Connect an additional diode and capacitor between the first stage and the output stage of the charge pump. The new capacitor and diode should be the same type and value as the first stage's components.

Application: Charge Pump with Output Regulation

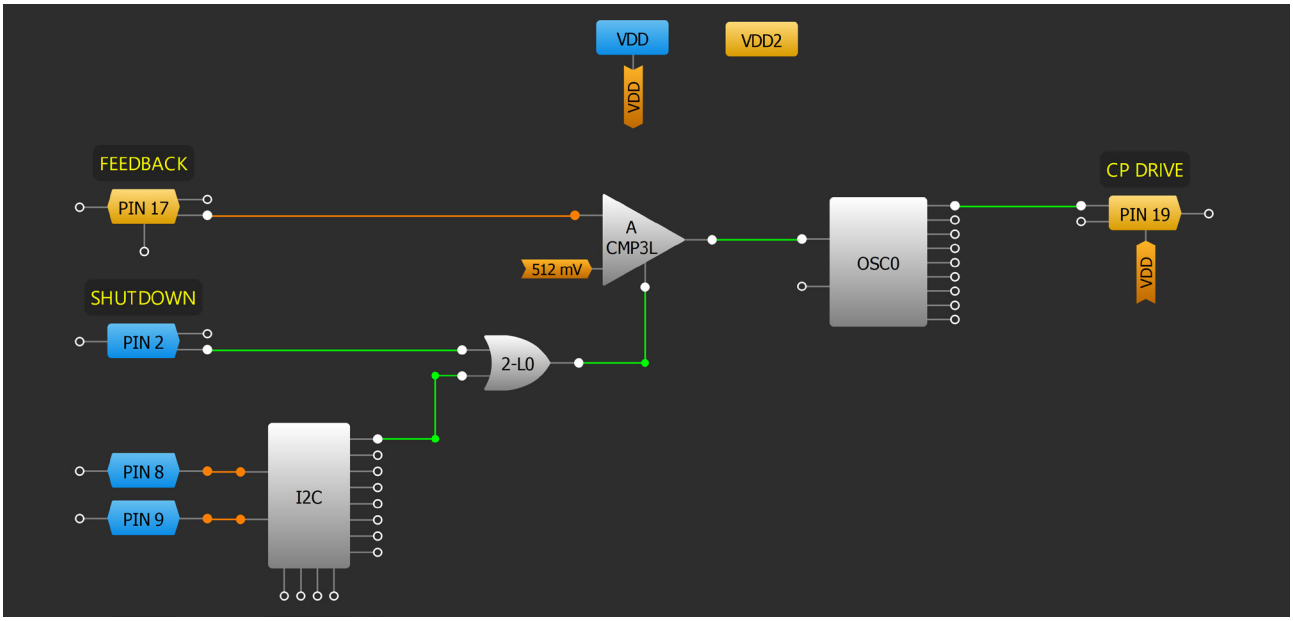
A charge pump is a DC-DC converter that uses capacitors for energetic charge storage to create different voltage levels. This charge pump with output regulation circuit can change the output voltage level via I2C. Schottky diodes are recommended for the best performance.



Ingredients

- Any GreenPAK w/ ACMPs
- Two capacitors
- Two diodes
- Two resistors

GreenPAK Diagram



Design Steps

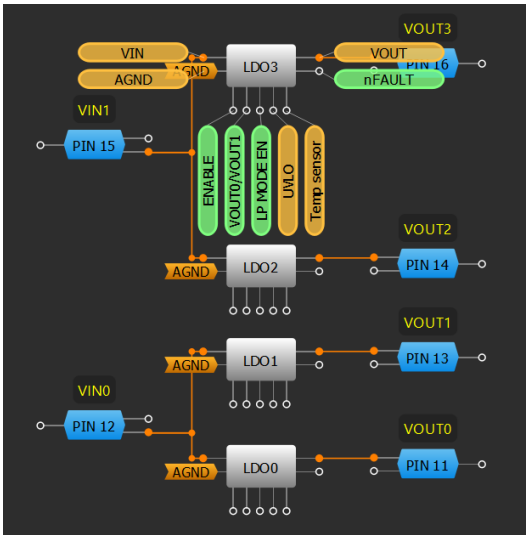
1. Create a basic charge pump design using the steps in Application: Charge Pump.
2. Set the IN+ source of ACMP3L to PIN17 (FEEDBACK) and IN- source to the desired Vref.
3. Using two resistors R1 and R2 create a voltage divider and connect the divider output to PIN17.
4. The output voltage can be calculated using the formula: V_out= V_ref· R1/R2 .
5. The GreenPAK can change the output voltage with I2C by setting the Vref for the IN- source of ACMP3L. The feedback resistor divider coefficient can also be changed to regulate the output voltage to a different value.

Technique: Using the LDO Regulators

This technique describes the low dropout (LDO) regulator macrocells, available in the SLG46580, SLG46582, SLG46583, and SLG46585.

Low dropout (LDO) regulators maintain the output voltage of a supply at a stable value despite changes in load impedance or supply voltage variations with a minimal dropout voltage. This makes them useful for portable devices that rely on dissipating batteries and RF systems which must handle periodic ripples. Some GreenPAK devices are equipped with LDO regulator macrocells.

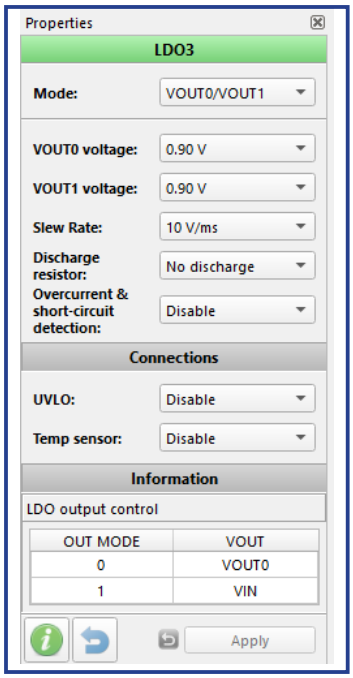
Each LDO macrocell has access to 32 possible output voltage levels ranging from 0.90V to 4.35V. Two different output voltages (VOUT1 and VOUT2) can be programmed. The state of the VOUT1/VOUT2 input selects which voltage level is outputted. Refer to the SLG46580 datasheet for the minimum VIN and VDD values for each output voltage level. Setting the LP MODE EN input LOW (default) selects the High Power (HP) Mode and setting it HIGH selects the Low Power (LP) mode. HP Mode can handle the highest rated output currents, but the LP Mode has a smaller quiescent current which provides a higher efficiency within its smaller rating. The LDOs are only stable in HP Mode when a >2µF capacitor is attached to each LDO's VOUT.



In the properties pane, the slew rate of each LDO can be changed to set a soft start. Each LDO has the option to enable a 300 Ohm discharge resistor on their output. Each LDO can also collectively enable a 210mA over-current limit and a short-circuit detection that limits the current to 20mA if the output voltage drops below 0.5V while it is in HP Mode.

Enabling the UVLO connection in an LDO sets an ACMP to probe its VIN and shut down the LDO if it drops below a certain undervoltage lockout (UVLO) threshold (ACMP IN- level).

The thermal limitations of the device must be considered when setting up the LDO regulator. The devices are rated at 0.6W of power dissipation at 85°C ambient. Enabling the Temp sensor connection sets ACMP2 to probe the GreenPAK's temperature sensor so it can shut down the LDO if it surpasses the properly programmed temperature and restarts the LDO after it cools down.

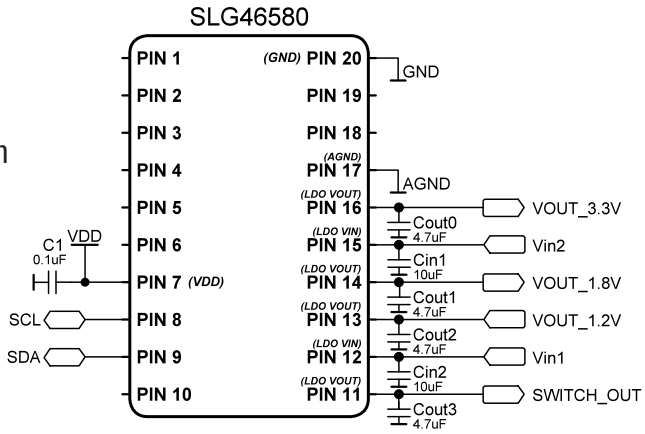


Device Specifications

- SLG46580/SLG46585:
- 4x LDO regulators;
- Imax: HP Mode = 150mA, LP Mode = 100µA
- SLG46582:
- 2x LDO regulators;
- HP Mode Imax = 300mA, LP Mode Imax = 200µA
- SLG46583:
- 1x LDO regulator
- HP Mode Imax = 600mA, LP Mode Imax = 400µA

Application: Flexible Power Island

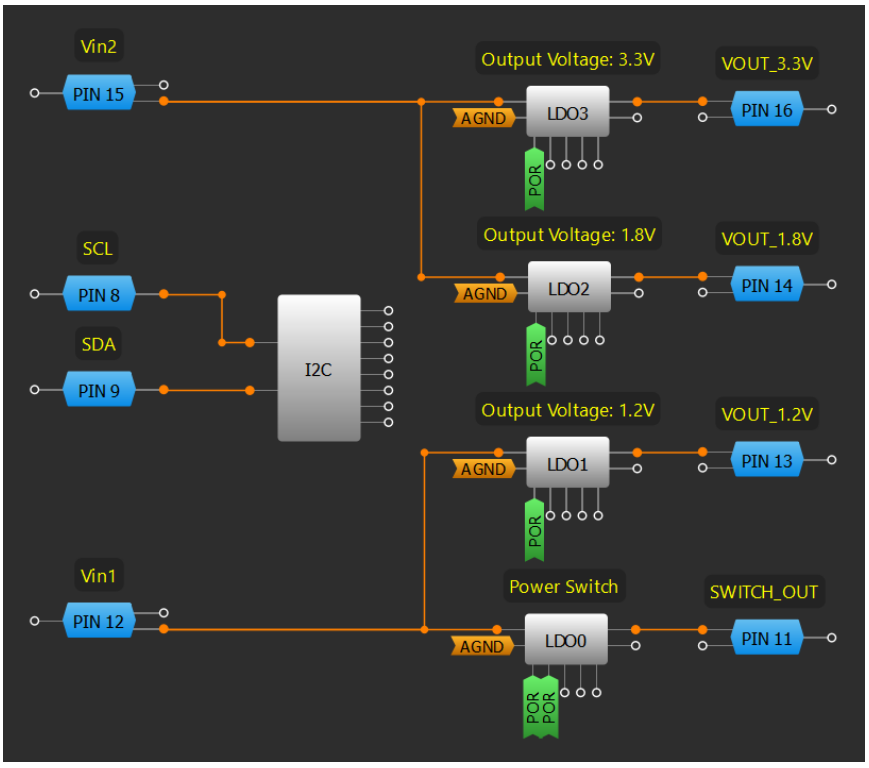
A flexible power island can help a designer divide up a power system into small "islands" of power regions that can be spread across a system. It can provide different level regulated voltages to fulfill specific system requirements.



Ingredients

- Any GreenPAK with internal LDOs
- 7 capacitors

GreenPAK Diagram

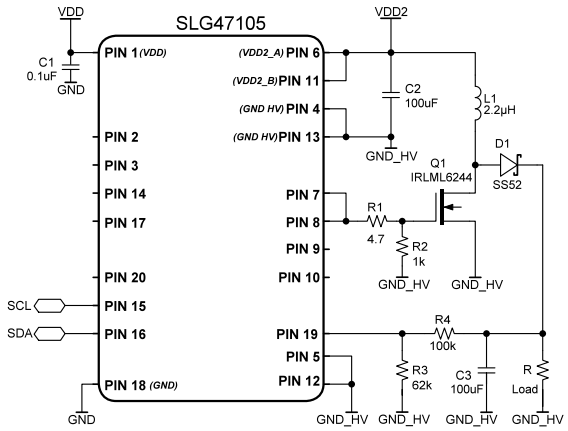


Design Steps

1. Set the VOUT0 voltage and VOUT1 voltage for each LDO channel.
2. Connect POR to each LDO's ENABLE pin.
3. Configure an LDO to the "VOUT0/PWR switch" mode and set the OUT MODE pin HIGH to set it as a Power Switch Output.

Application: Boost (Step-up) Converter

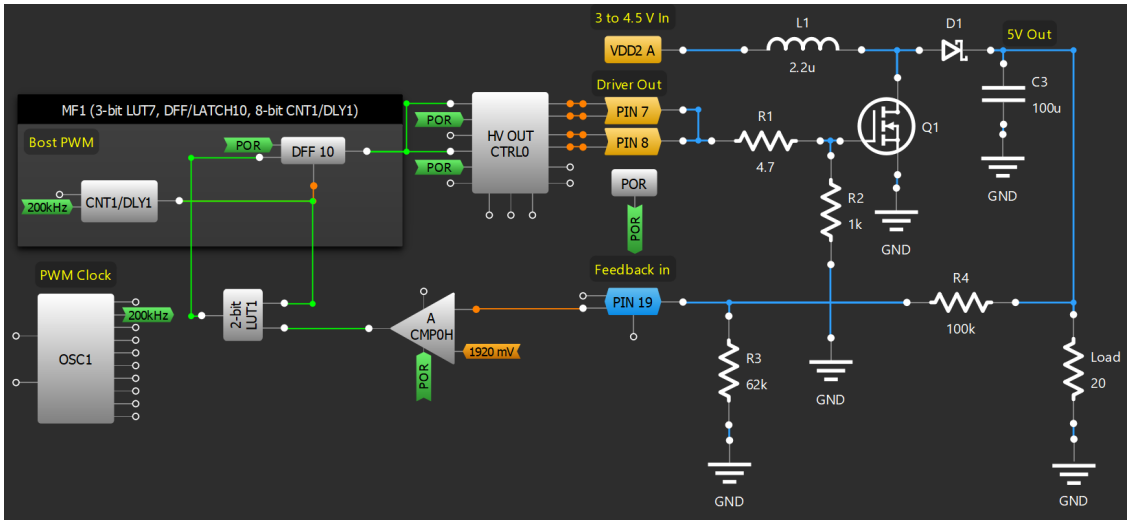
In this application, the SLG47105 is used as a driver for the Brushed DC Motor with Push-to-Start/Hold-to-Stop functions. This feature can be used for electrical toys, tools, and others.



Ingredients

- SLG47105
- MOSFET
- Schottky diode
- Inductor
- Three capacitors
- Four resistors

GreenPAK Diagram

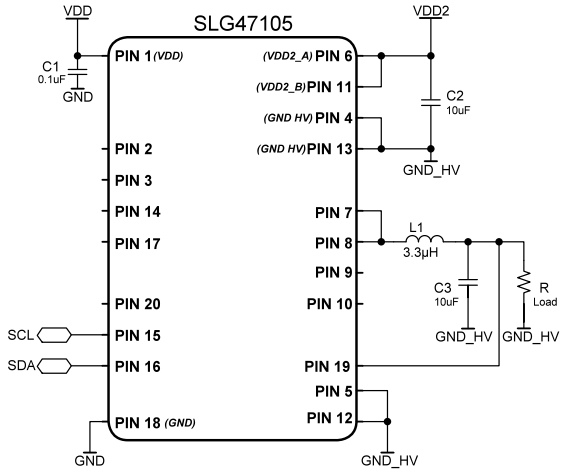


Design Steps

1. Complete the circuit. Connect all components as shown on the circuit diagram.
2. Enable HV OUT CTRL0 connecting OE0/1 to POR and set their Modes as “Fast for pre-drive” and “Half-Bridge”.
3. Configure PIN7 and PIN8 as “HIGH and LOW side on”
4. Enable ACMP0H. Set its IN- source to “1920 mV” to regulate the output voltage to 5 V. Connect its output to IN0 of the 2-bit LUT2.
5. Configure CNT1/DLY1 as a Delay and connect its output to DFF10’s nRESET and IN1 of the 2-bit LUT2. Connect DLY IN to Flex-Div OUT of the OSC1. Set the Flexible divider to 125.
6. Connect DFF’s D to POR and its CLK to 2-bit LUT2’s output.
7. Connect DFF’s output to both IN0 and IN1 of the HV OUT CTRL0.

Application: Buck (Step-down) Converter

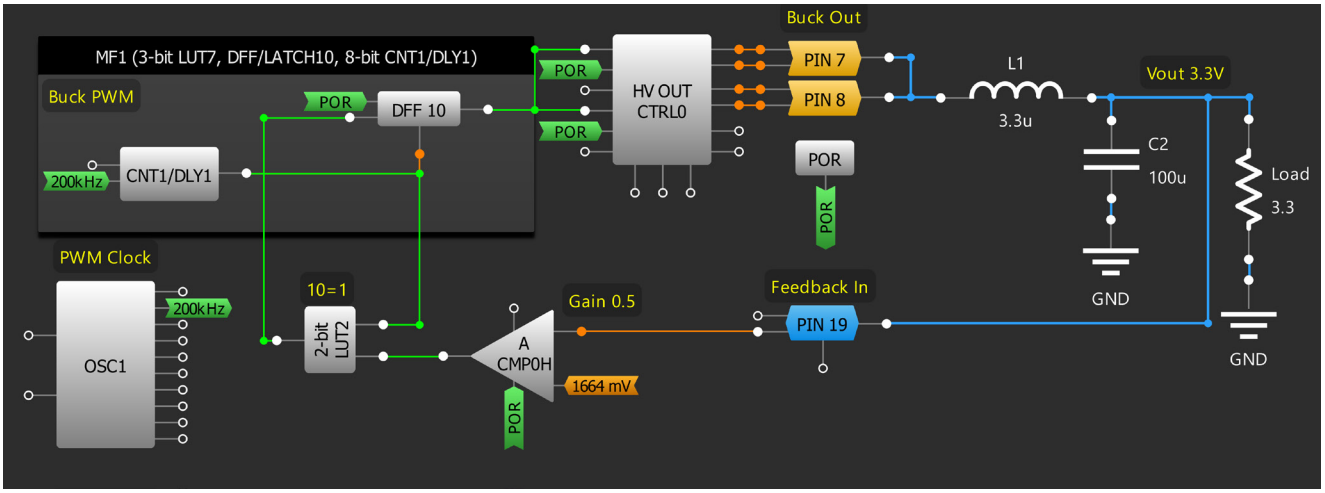
In this application, the SLG47105 is used as a buck converter. In this design example, the converter steps down VDD2 voltage to 3.3 V. Also, this design shows how to build an analog PWM block.



Ingredients

- SLG47105
- Inductor
- Three capacitors

GreenPAK Diagram



Design Steps

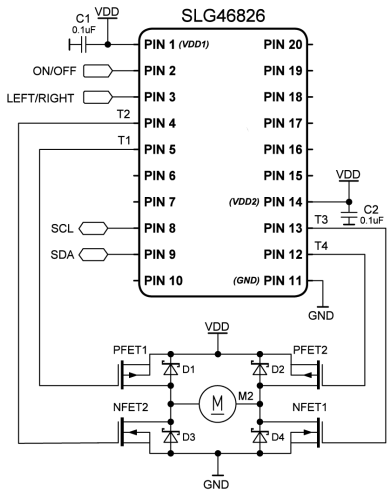
1. Complete the circuit: connect one pin of L1 to both PIN7 and PIN8, connect PIN5 to GND; connect PIN19 to the other pin of L1 and C2, connect the other C2 pin to GND; connect the load in parallel to C2.
2. Enable HV OUT CTRL0 connecting OE0/1 to POR and set their Modes as “Fast for pre-drive” and “Half-Bridge”.
3. Configure PIN7 and 8 as “HIGH and LOW side on”
4. Enable ACMP0H by connecting to POR. Set its IN + gain to x0.5 and IN - source to “1664 mV” to regulate the output voltage to 3.3 V. Connect its output to IN0 of the 2-bit LUT2.
5. Configure CNT1/DLY1 as a Delay and connect its output to DFF10’s nRESET and IN1 of the 2-bit LUT2. Connect DLY IN to Flex-Div OUT of the OSC1. Set the Flexible divider to 125.
6. Connect DFF’s D to POR and its CLK to 2-bit LUT2’s output.
7. Connect DFF’s output to both IN0 and IN1 of the HV OUT CTRL0.

Chapter 8
Motor Control

This chapter presents applications that control DC motors. It centers on using the integrated H-Bridge of the SLG47105 and its accompanying blocks to provide current and voltage regulation.

Application: H-Bridge Control

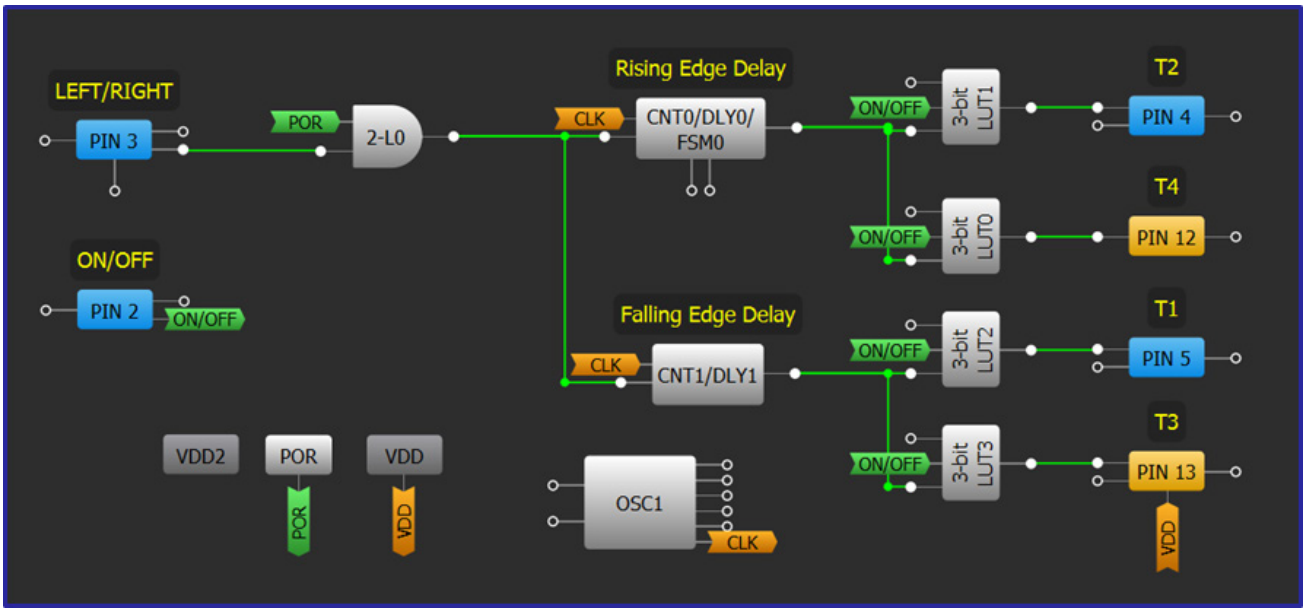
An H-bridge is an electronic circuit constructed of four transistors that reverses the polarity of the voltage across a load. They are often used to control DC motors.



Ingredients

- Any GreenPAK
- Four transistors
- Four diodes

GreenPAK Diagram

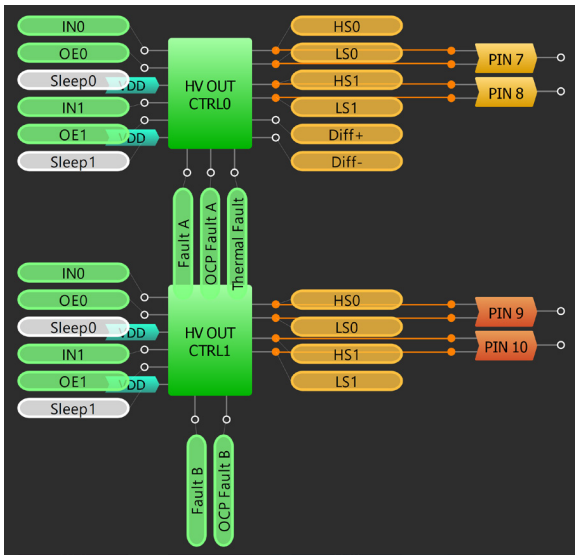


Design Steps

1. Add and configure inputs and outputs.
2. Add delay blocks using Technique: Optimizing CNT/DLY Accuracy.
3. Add and configure LUTs for each output using Technique: Configuring Standard Logic w/ LUT Macrocells.

Technique: Using the HV OUT CTRL Blocks

This technique describes the use of HV OUT CTRL blocks in the SLG47105V.



HV OUT CTRL0/1

SLG47105V consists of 2 HV OUT CTRL macrocells namely HV OUT CTRL0 and HV OUT CTRL1. Each macrocell can be used to drive 2 unidirectional DC motors or 1 bidirectional DC motor. Both HV OUT CTRL0/1 can be used to drive a stepper motor.

To enable HV OUT CTRL0/1 connect Sleep 0/1 to an active LOW. Each Sleep pin can be activated separately. To drive unidirectional select HV OUT mode as "Half-Bridge" and for bidirectional motors select "Full-Bridge". In "Full-Bridge" mode select Mode control as "IN-IN" or "PH-EN". Table 2 displays the "Half-Bridge" control logic. Table 2 and Table 4 respectively describe "IN-IN" and "PH-EN" mode controls. Pin 7/8/9/10 (Hi-Z by default) are used to control motor speed using PWM. These pins can be pulled up/down externally. In "Full Bridge" mode, Pin 7/9 and Pin 8/10 can be connected in parallel externally.

To control DC motor with "IN-IN" mode control using PWM0/1, connect IN1 to an active LOW signal in fast mode and an active HIGH signal in slow decay mode. Connect IN0 to PWM0/1 output. Fast decay mode is used to instantly reduce inductive current and coast motor towards zero velocity. The slow decay mode causes a slow reduction in inductive current and rapid deceleration.

IN-IN Mode Logic for Full Bridge Mode

Sleep 0/1	IN0	IN1	Pin 7/9	Pin 8/10	Function
1	X	X	Hi-Z	Hi-Z	Off
0	0	0	Hi-Z	Hi-Z	Coast
0	0	1	L	H	Reverse
0	1	0	H	L	Forward
0	1	1	L	L	Brake

Half-Bridge Mode

Sleep 0/1	OE	IN0/1	Pin 7/8	Function
1	X	X	Hi-Z	Off
0	0	X	Hi-Z	Off (Coast)
0	1	0	L	Brake
0	1	1	H	Forward

PH-EN Mode Logic for Full Bridge Mode

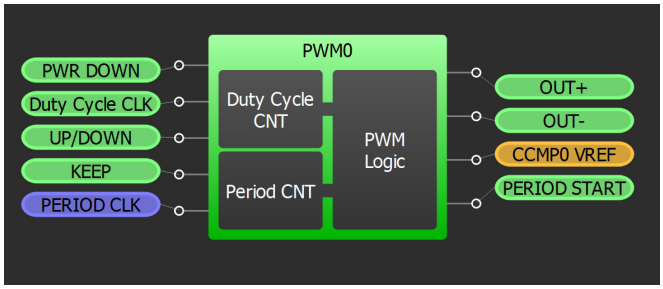
Sleep 0/1	Decay	EN/PWM	PH/Direct	Pin 7/9	Pin 8/10	Function
1	X	X	X	Hi-Z	Hi-Z	Off (Coast)
0	0 (fast)	0	X	Hi-Z	Hi-Z	Coast
0	1 (slow)	0	X	L	L	Brake
0	X	1	0	H	L	Forward
0	X	1	1	L	H	Reverse

When any fault occurs, Fault A/B pins go HIGH and HV OUT CTRL0/1 are disabled. When the fault pins go LOW, normal operation is restored. Fault A and Fault B consist of all fault signals for VDD2_A and VDD2_B separately. When an overcurrent condition occurs, OCP Fault A/B goes HIGH. An OCP deglitch time can be enabled on Pin 7/8/9/10. The OCP deglitch time enable is shared among Pin 7/8 and Pin 9/10. The retry time for OCP is user selectable and separate for each pin. When die temperature exceeds safe limits, Thermal Fault turns HIGH. VDD2_A and VDD2_B have separate UVLO (undervoltage lock out) enable.

Technique: Using the SLG47105 PWM Blocks in Regular Mode

This technique is for the PWM blocks, available in the SLG47105.

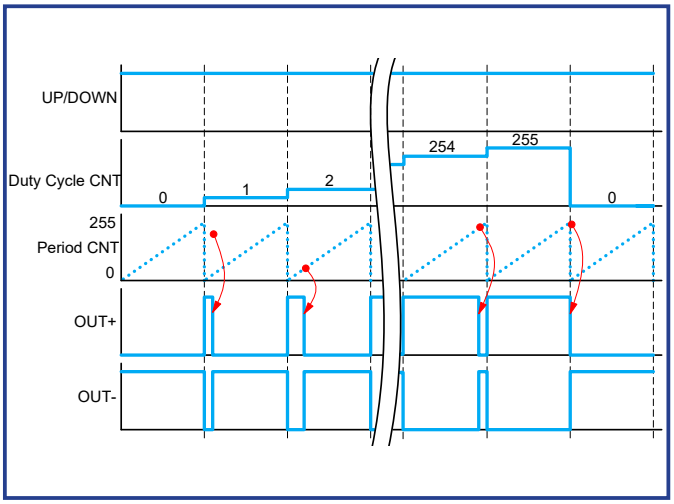
PWM is commonly used in DC motor control, LED brightness control, and other applications. SLG47105 is equipped with advanced PWM blocks to handle higher voltage level dedicated PINs. PWM blocks have already been implemented in other GreenPAKs as discussed in Technique: Using DCMP/PWM Macrocell in PWM Mode, but the two PWM blocks in the SLG47105 integrate the counters involved in their operation while in "Regular Mode" and include more advanced settings.



The first 8-bit counter included in the block is PWM Period CNT, which sets the frequency of the PWM signal and counts from 0 to 255 and so on. There are two PWM outputs: "OUT+" and "OUT-". OUT+ is logic HIGH at the start of the period and once the PWM period counter reaches the duty cycle value, the output changes to logic LOW until the PWM period ends, as shown in the figure below. OUT+ is the positive PWM output and OUT- is the negative PWM output that is inverted to OUT+ and shifted for deadband time if defined. Both can invert their output by register settings.

In Regular Mode (described in this article) the Duty cycle source is set to "Duty Cycle CNT". The second 8-bit counter included, named Duty Cycle CNT, increments or decrements the duty cycle value for the next PWM period dependently on UP/DOWN input. The Duty Cycle CLK is an external clock from the matrix by default. It changes the duty cycle value by the rising edge. It can also be set to be clocked by the period counter overflow or every 2nd or 8th pulse of the overflow.

The PWM block has an 8-bit resolution by default, but a 7-bit resolution can be selected instead to allow for a higher PWM frequency. The PWM duty cycle changes at a step of 0.4 % for the 8-bit resolution and 0.8 % for the 7-bit resolution. The duty cycle can change from true 0% to true 100%. PWM starts from the Initial duty cycle value. The block has an UP/DOWN internal connection that defines the direction of the duty cycle change. The duty cycle will increase if set HIGH and decrease if set LOW.



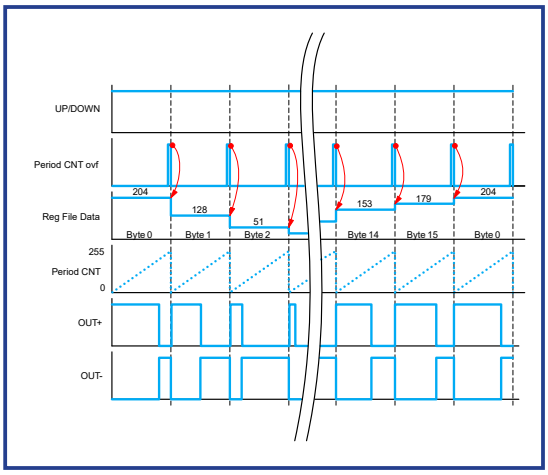
The Keep/Stop connection can either be selected to hold the duty cycle ("Keep" setting) or to hold the duty cycle and the OUT+ and OUT- outputs constant ("Stop" setting) when it is set HIGH. The Continuous/Autostop mode is either set to "Continuous" where the PWM output duty cycle overflows when it reaches the full range value (default setting) or to" Autostop" where the PWM output stops when it reaches 0% or 100% of the duty cycle. When "Autostop" is selected, the "Boundary OSC disable" option can be activated. This allows disabling Oscillator, used by PWM cell, automatically when 0% or 100% of the duty cycle is reached.

Technique: Using the SLG47105 PWM Blocks in Preset Registers Mode

This technique is for the PWM blocks, available in the SLG47105. More information on how the block works can be found in Technique: Using the SLG47105 PWM Block in Regular Mode.

The previous technique explained how to use the SLG47105 PWM block in "Regular Mode" with the Duty Cycle Source set to "Duty Cycle CNT." This technique will instead explain how to use the block in "Preset Registers Mode" where the duty cycle cycles through 16 predefined values (Reg File). Using the "Preset Registers Mode" allows for non-linear PWM patterns for motor control (i.e. sinusoidal or logarithmic).

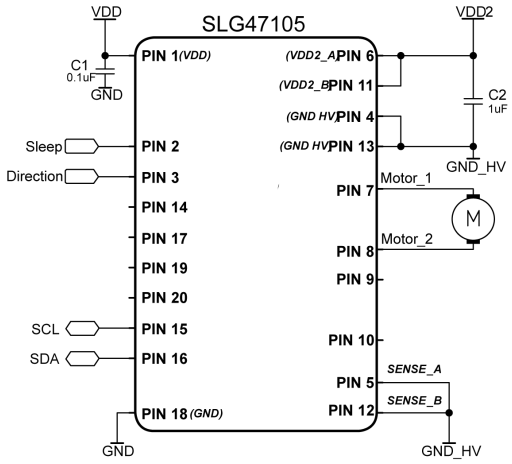
Selectable preset registers are reserved to determine 16 different PWM duty cycle values. Duty Cycle CLK can be selected to Clock from Matrix or PWM period CNT ovf (overflow). A clock on the Duty Cycle CNT CLK input changes which register's value is applied to compare with the Period CNT. Reg File is shared between the two PWM macrocells. Either all 16 bytes, the least significant 8 bytes, or the most significant bytes can be used. The initial byte is limited by the unique ranges of each setting.



The internal connections have a similar function as Regular Mode but apply to the 16-byte structure rather than the 8-bit counter. The polarity of Up/Down decides whether the next register (HIGH) or the previous (LOW) is applied. The Keep/Stop operates the same way but halts the sequence of registers.

Application: Constant Voltage Brushed DC Motor Driver

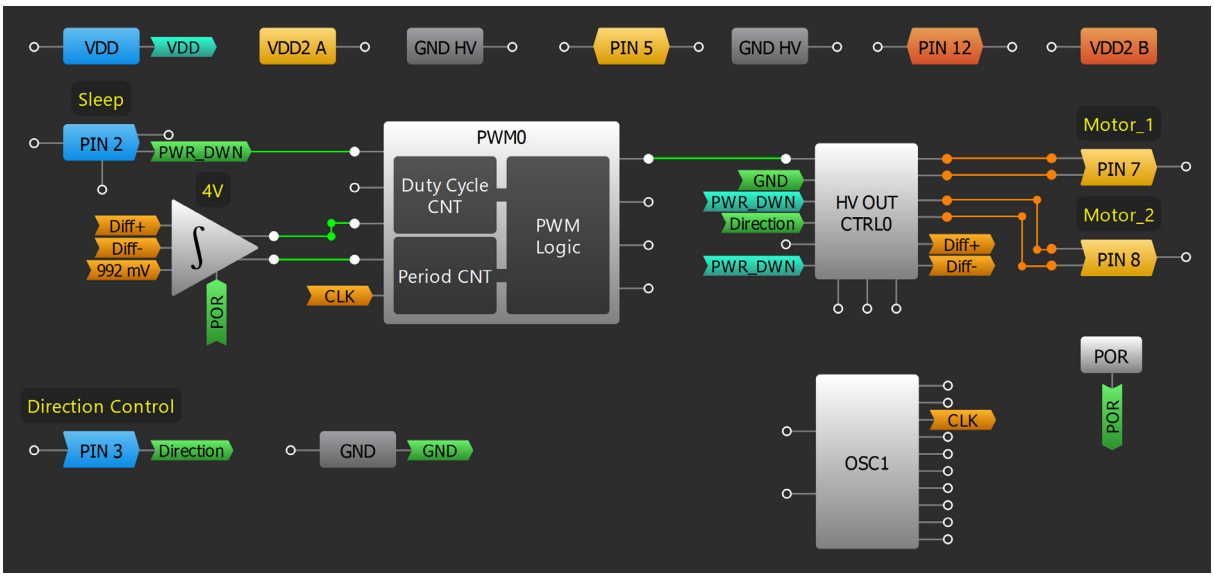
Maintaining a constant voltage over a brushed DC motor ensures it maintains a constant speed. In this design, Differential Amplifier with Integrator and Comparator controls the PWM block to regulate the voltage across the load.



Ingredients

- SLG47105V
- Brushed DC Motor

GreenPAK Diagram

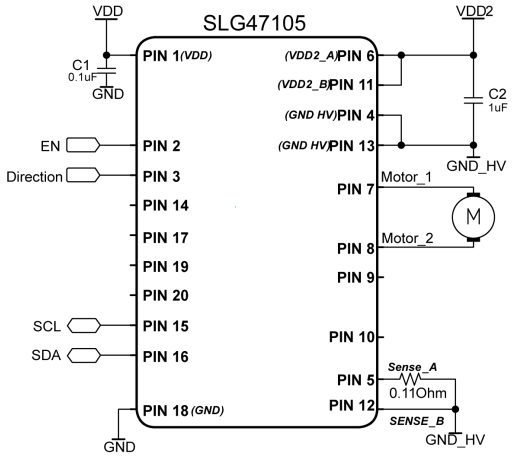


Design Steps

1. Enable HV OUT CTRL0 with PIN2 and set its Mode as “Full bridge” and Mode control as “PH-EN”.
2. To change motor direction, connect PIN3 to PH of HV OUT CTRL0.
3. Enable Differential Amplifier with Integrator and Comparator and select integrator reference voltage to the desired threshold (Threshold=V_REF·4).
4. For correct Differential Amplifier with Integrator and Comparator operation, enable PWM0 through PIN2 and set Duty Period CLK to “OSC1”. PWM frequency must be 44 kHz or higher to make sure that Integrator operates correctly. Connect UPWARD and Equal outputs to UP/DOWN and Keep of PWM0 respectively.
5. Set PWM0 Resolution to “8-bits”, Duty Cycle Source to “Duty Cycle CNT”, Duty Cycle CLK to “Period CNT ovf/8”, and Initial Duty Cycle Value to “50%”.
6. Connect PWM0 OUT+ to EN of HV OUT CTRL0 to drive the motor at a constant speed.

Application: Constant Current Brushed DC Motor Driver

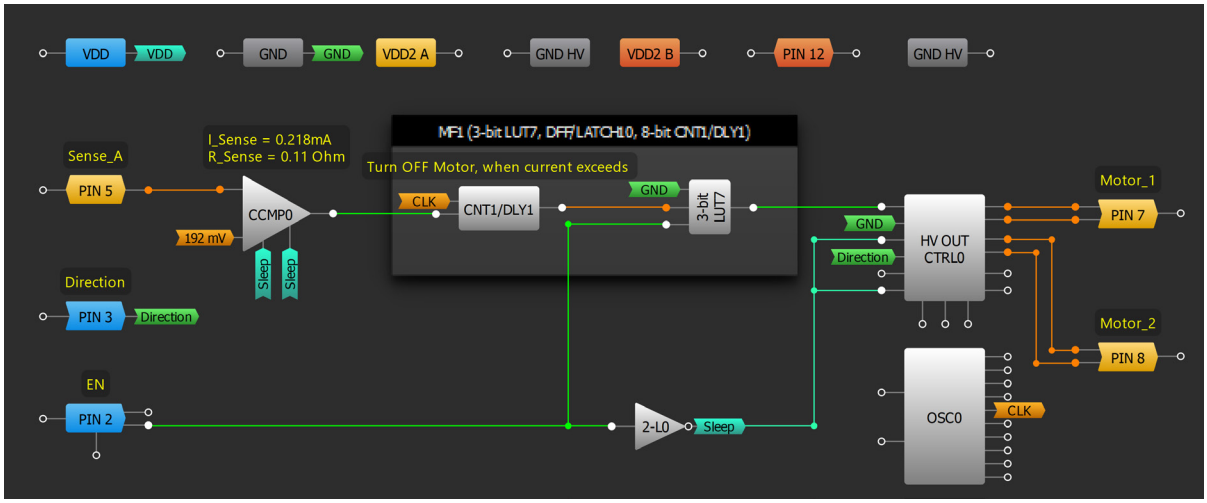
Maintaining a constant current over a brushed DC motor ensures it maintains a constant torque. In this design, the current sense comparator (CCMP0) is used to limit the current through the sense resistor to turn OFF motor when the current exceeds a specified limit.



Ingredients

- SLG47105V
- Brushed DC Motor
- One resistor

GreenPAK Diagram

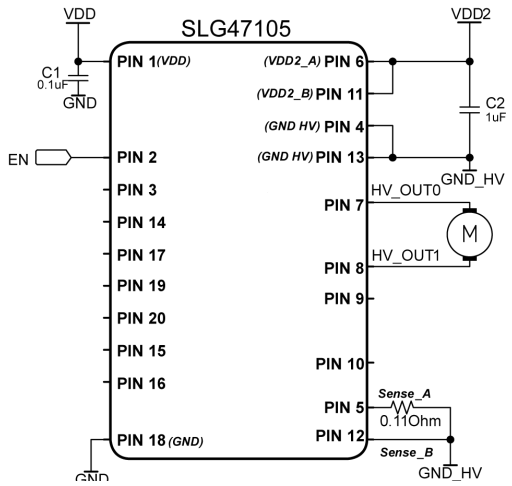


Design Steps

1. Connect a brushed DC motor across PIN7 and PIN8, and sense resistor from PIN5 to GND
2. Enable HV OUT CTRL0 by connecting inverted PIN2 signal to its Sleep 0/1.
3. Configure HV OUT CTRL0's mode as "Full bridge" and Mode control as "PH-EN."
4. To allow change of motor direction, connect Pin 3 to PH of HV OUT CTRL0.
5. Enable CCMP0 by changing Sleep CTRL to "Auto."
6. Select CCMP0's IN- source to "192mV" to limit current to 0.218mA;
7. Configure CNT1/DLY1 to turn OFF motor for 100ms when current exceeds the specified limit.
8. Configure 3-bit LUT7 to turn ON the motor only when current is within range and PIN2 is HIGH.
9. Connect the 3-bit LUT7 output to EN of HV OUT CTRL0.

Application: Constant Current Using the PWM Chopper

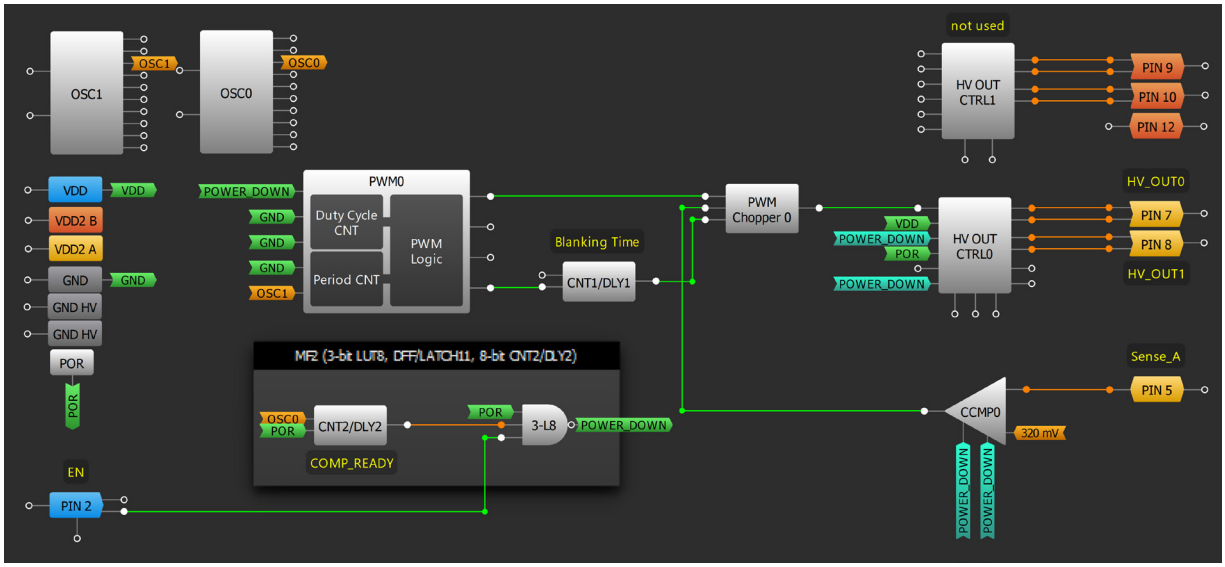
Maintaining a constant current over a brushed DC motor ensures it maintains a constant torque. This application shows how to limit the current with the PWM chopper block.



Ingredients

- SLG47105V
- Brushed DC Motor
- One resistor

GreenPAK Diagram

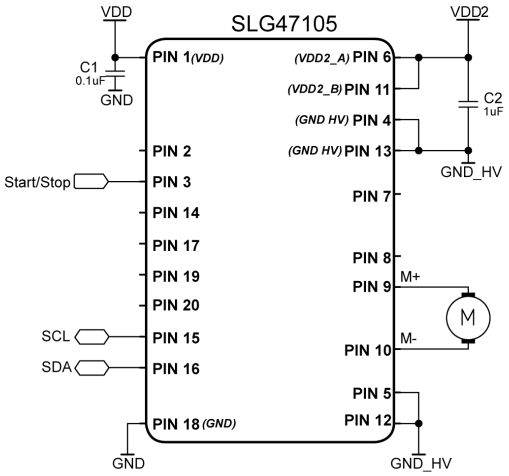


Design Steps

1. Connect a brushed DC motor across PIN7 and PIN8, and sense resistor from Pin 5 to GND
2. Configure HV OUT CTRL0's mode as "Full bridge" and Mode control as "PH-EN"
3. Enable CCMP0 by changing Sleep CTRL to "Auto"
4. Select CCMP0's IN- source to "320mV" to limit current to ~0.36mA
5. Add PWM0 block and set the initial duty cycle to 230 and adjust the OSC1 predivider
6. Configure CNT1/DLY1 as a falling edge delay to set the Blanking Time
7. Add PWM Chopper 0 and make the appropriate connections with PWM0, CNT1/DLY1, and CCMP0 to create the duty cycle chopper and limit the motor current
8. Add PIN2 as an enable button to start/stop the motor and internal motor controlling blocks.
9. Connect the 3-bit LUT7 output to EN of HV OUT CTRL0.

Application: Unidirectional DC Motor Control with Soft ON/OFF

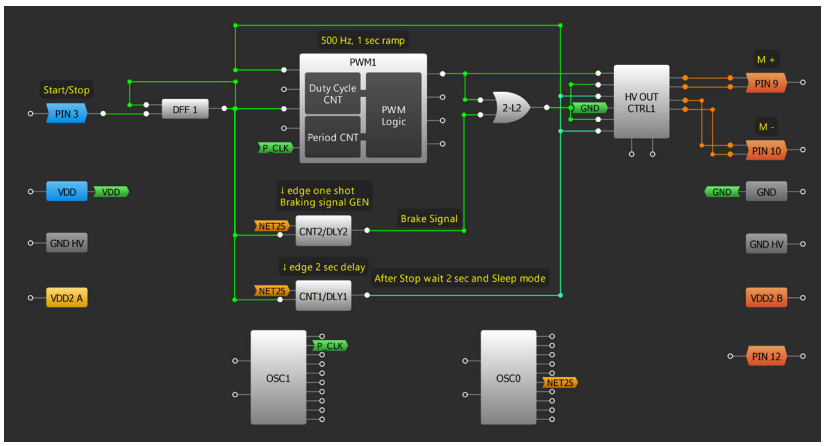
Soft ON/OFF can be used to decrease the starting current and load torque of a brushed DC motor. This application is for unidirectional DC motors that allow a single direction for mechanical elements, and either run or stop. This regulation is important because the load torque must not exceed the torque on the motor shaft as it starts and stops.

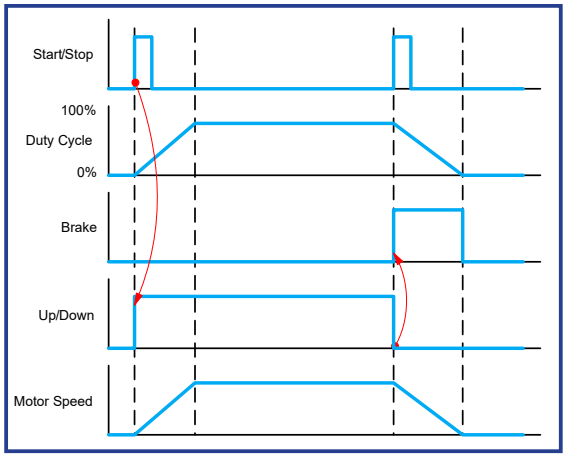


Ingredients

- An SLG47105
- Unidirectional brushed motor
- Two capacitors

GreenPAK Diagram



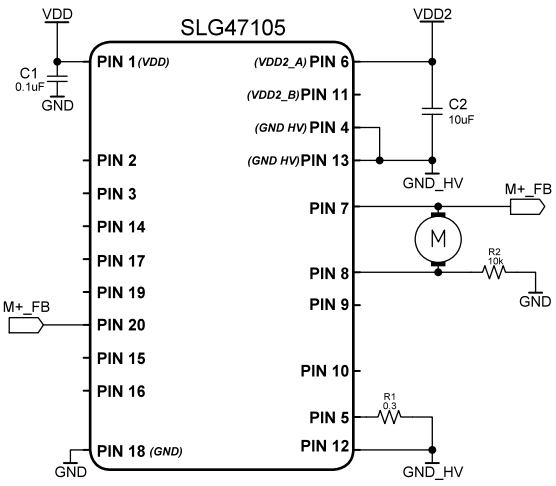


Design Steps

1. Set PIN9 and PIN10 Output mode to “High and Low side.” HV OUT CTRL1 set to “Half Bridge” HV OUT mode.
2. Set Duty Cycle CLK in PWM1 block to “Period CNT ovf/2”. Period CLK set to “Ext.Clk.” Connect OSC1 Flex-DIV OUT to PWM1 Period Clock input. Set the value of flexible divider in OSC1 properties.
3. Configure DFF1 Initial Polarity to High and the Q Output Polarity to “Inverted (nQ).” Connect the output of the DFF to the PWM1 UP/DOWN input.
4. Add CNT1/DLY1 to design. Configure that to the “Falling Edge Delay” with an inverted output. The signal from this delay disables the PWM1 block and enables the sleep mode of HV OUT CTRL1 after the Stop signal comes.
5. 2-bit LUT2 used to forming Hi-Z and Stop signals on HV Outputs.

Application: Push-to-Start/Hold-to-Stop

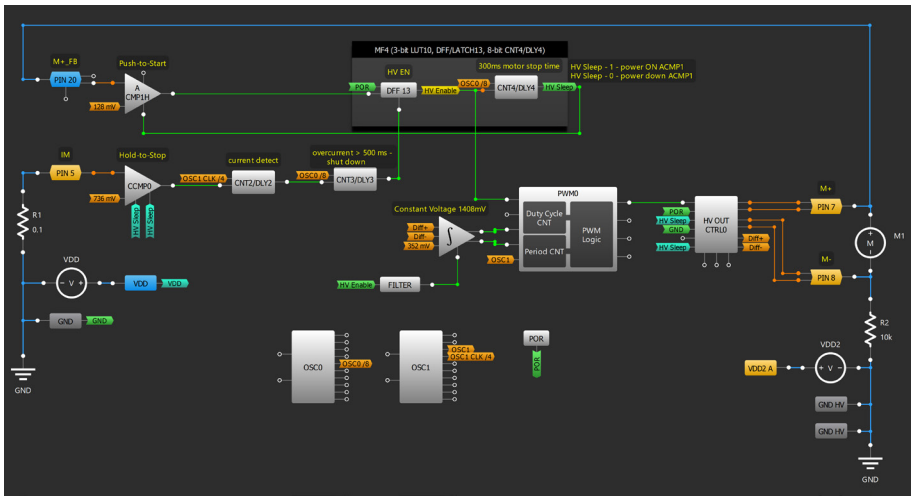
In this application, the SLG47105 is used as a driver for the Brushed DC Motor with Push-to-Start/Hold-to-Stop functions. This feature can be used for electrical toys, tools, and others.



Ingredients

- An SLG47105
- A Brushed DC Motor
- Two resistors

GreenPAK Diagram

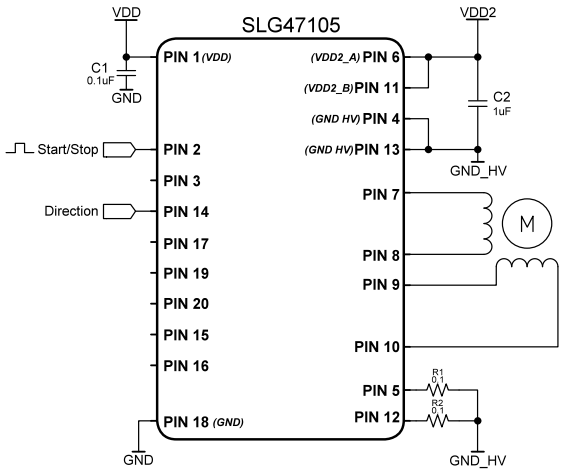


Design Steps

1. Connect a Brushed DC Motor: first winding across PIN7, second winding across PIN8 with a Pull-down resistor of 10 kΩ, and sense resistor from PIN5 to GND. Configure Output modes of PIN 7 and 8 as “HIGH and LOW side”.
2. Enable HV OUT CTRL0 connecting Sleep0/1 to an active LOW (HV Sleep output of CNT4/DLY4) and set its Mode as “Full Bridge” and Mode control as “PN-EN”. Connect POR to Decay and GND to PH.
3. Enable CCMP0 by changing Sleep CTRL to “Auto”. Select its IN - source to “736 mV” to adjust the Hold-to-Stop function.
4. Configure CNT2/DLY2 as a Delay, and connect its output to CNT3/DLY3 (Delayed edge detect Mode) to ensure the system shut down after more than 500 ms of overcurrent – Hold-to-Stop.
5. Configure PIN20 as Analog input/output and connect it to ACMP1H IN - source, select 128 mV In - source, and connect comparator’s output to DFF13 (Q Output Polarity – Inverted (nQ), Initial Polarity – Low) – Push-to-Start.
6. Configure CNT4/DLY4 as a Delay to ensure 300 ms motor stop time (HV Sleep signal).
7. Add PWM0 and make the appropriate connections with Differential Amplifier (reference voltage – 352 mV) to maintain a constant voltage over a brushed DC motor (see Application Constant Voltage Brushed DC Motor Driver), DFF13, and HV OUT CTRL0 to create the duty cycle.

Application: Bipolar Stepper Motor Driver

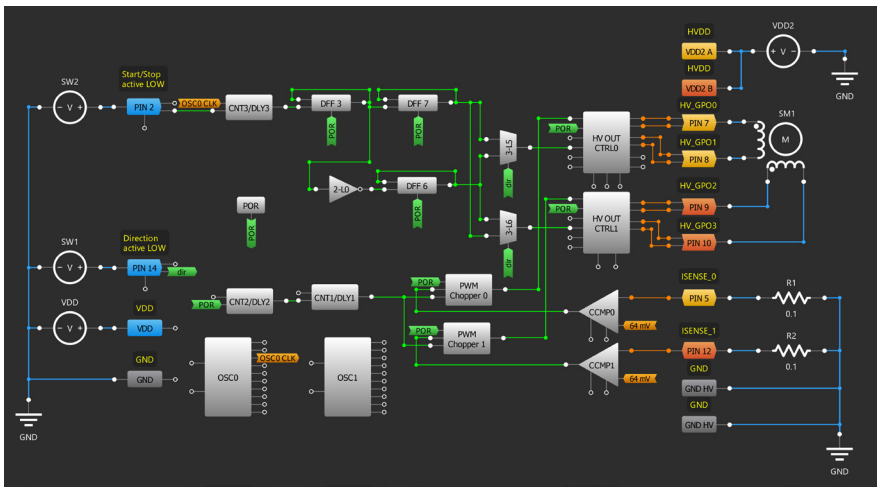
In this application, the SLG47105 is used as a driver for the stepper motor. In this design example, the driver has full step mode in both directions. Also, this design shows how to limit the current with the PWM Chopper block



Ingredients

- SLG47105
- Stepper Motor
- Two resistors

GreenPAK Diagram



Design Steps

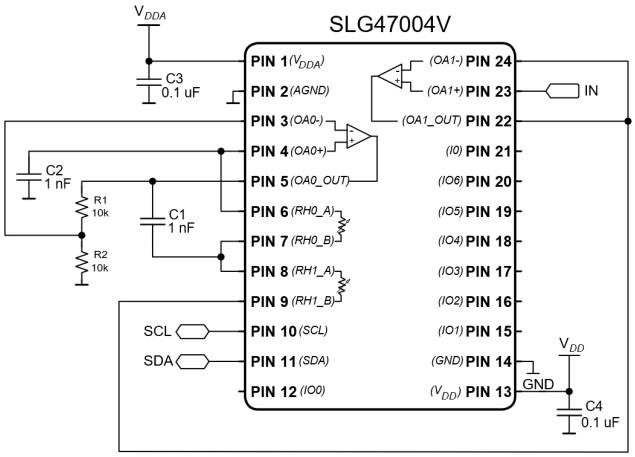
1. Connect a Stepper Motor:
- First winding across PIN7 and PIN8, and sense resistor from PIN5 to GND;
- Second winding across PIN9 and PIN10, and sense resistor from PIN12 to GND.
2. Enable HV OUT CTRL0/1 connecting Sleep0/1 to an active LOW and set their Modes as “Full Bridge” and Mode controls as “PN-EN”.
3. Enable CCMP0/1 by changing Sleep CTRL to “Auto”. Select its IN-source to “64 mV” to limit current to ~ 80 mA.
4. Configure CNT2/DLY2 as a Reset Counter, and connect its output to CNT1/DLY1 (Delay Mode) to set the Blanking Time.
5. Add PWM0/1 Chopper and make the appropriate connections with POR, CNT1/DLY1, and CCMP0/1 to create the duty cycle chopper and limit the motor current. Connect its output to EN of HV OUT CTRL0/1.
6. Configure PIN2 (Start/Stop) and PIN14 (Direction) as Digital Inputs and make appropriate connections with Delay on CNT3/DLY3, DFF3, DFF7, and DFF6 (Q Output Polarity – Inverted (nQ), Initial Polarity – Low), 2-bit LUT0 set as Invertor, and 3-bit Multiplexers LUT5 and LUT6. Connect LUT5/6’s Multiplexer Output to PH of HV OUT CTRL0/1 to Start/Stop the motor and to change the motor`s direction.

Chapter 9
Advanced Analog Features

This chapter presents applications that manage analog features using components in AnalogPAK. Applications involve the most common circuit topologies using built-in operational amplifiers, digital rheostats, Chopper ACMP, etc.

Application: Adjustable Active Filter Using OpAmp

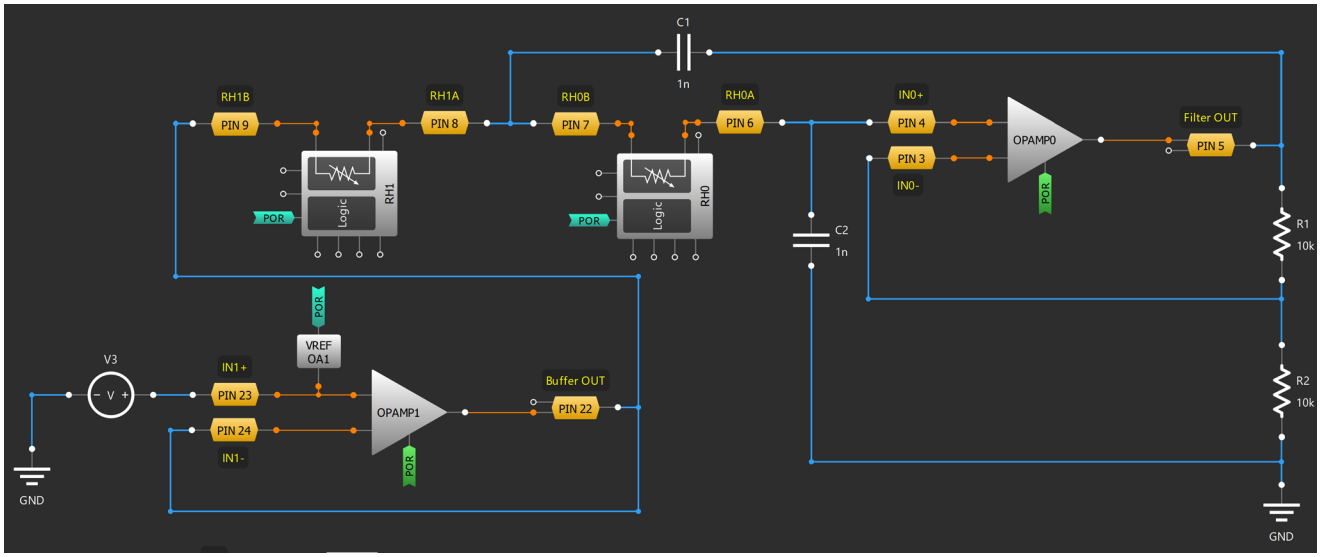
There are many applications where signals from different sources (i.e. sensors) can be sensed with one ADC. These systems require an analog multiplexer with analog filters for each channel because each signal source may have its own set of filter requirements (for example, cutoff frequency). In this design, one filter serves many analog inputs and provides different cutoff frequencies to them. I2C master can write data to rheostat registers and adjust the cutoff frequency of the filter.



Ingredients

- SLG47004V
- Two resistors
- Two capacitors

GreenPAK Diagram

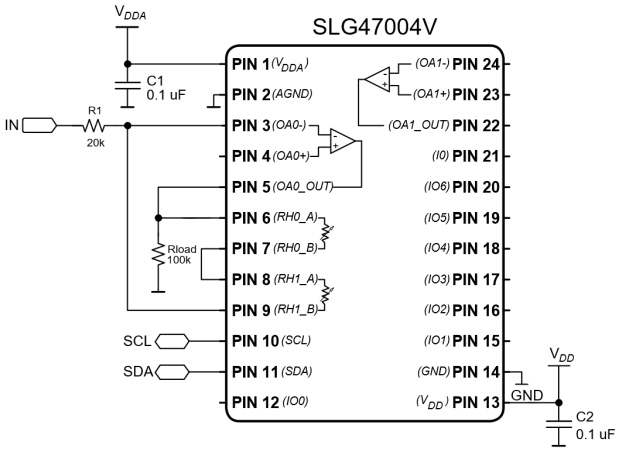


Design Steps

1. Enable OpAmp0 and Opamp1. Set bandwidths to 8 MHz and enable Charge Pumps. Set OpAmp1 Vref connection to IN+.
2. Enable Vref OA1, set input voltage to VDD and output selection to VDD * (16/64).
3. Set Digital Rheostat 1 in Rheostat mode. Disable Auto-Trim, connect FIFO nRST input to POR and set desired resistance (initial data) for both rheostats.

Application: Adjustable Inverting OpAmp

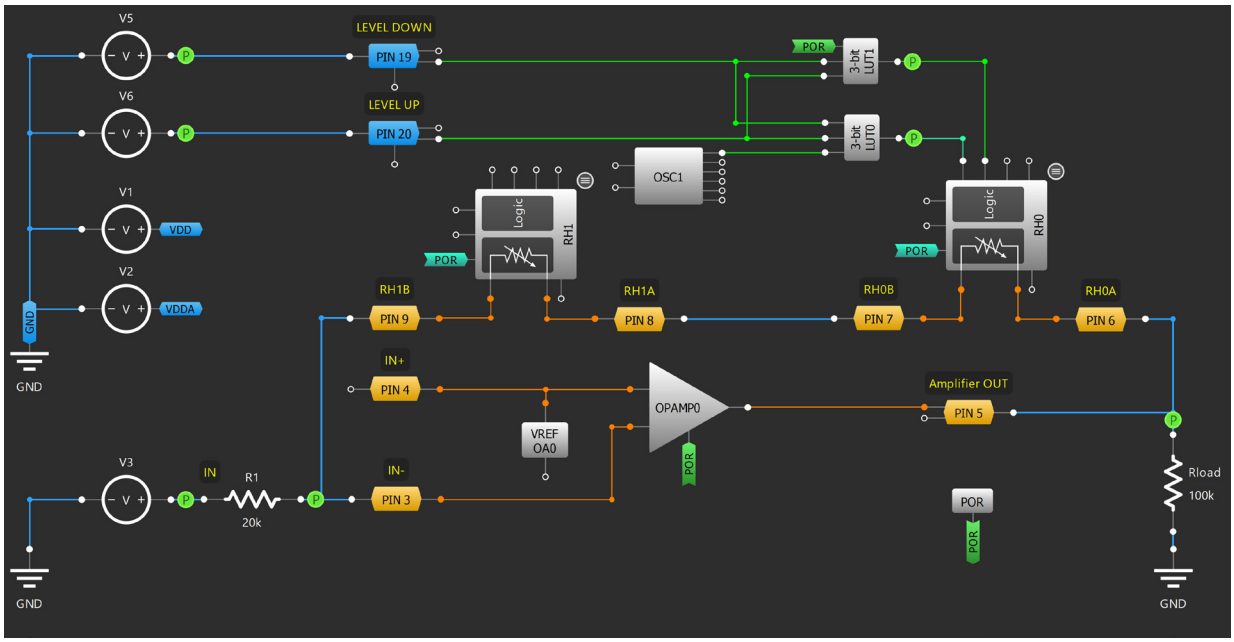
An inverting amplifier is a type of operational amplifier circuit which produces an output that is out of phase for its input by 180o. It means that if the input pulse is positive, the output pulse will be negative, and vice versa External control signals allow a user to adjust the gain of the inverting amplifier both downward and upward.



Ingredients

- SLG47004V
- One resistor

GreenPAK Diagram



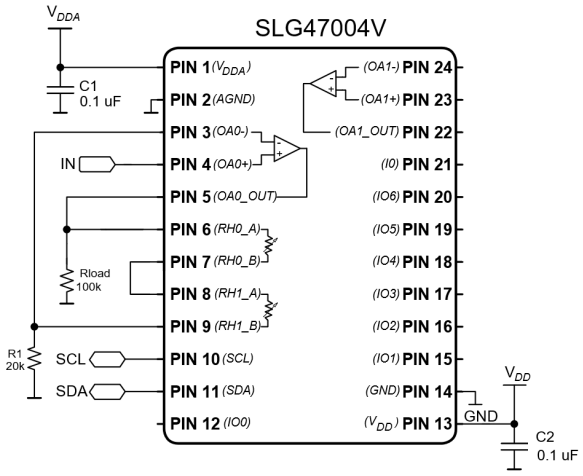
G=V_out/V_inp =-(RH0+RH1)/R_1

Design Steps

1. Enable OpAmp0. Set bandwidth to 8 MHz, enable Charge Pump and set Vref connection to IN+.
2. Enable Vref OA0, set input voltage to 2.048 V and output selection to 512 mV.
3. Set Digital Rheostat 1 in Rheostat mode. Disable Auto-Trim, connect FIFO nRST input to POR and set desired resistance (initial data) for both rheostats.
4. Configure LUT0 to output the OSC1 clock signal when LEVEL UP or LEVEL DOWN signals are HIGH. Configure LUT1 to output a High-level signal when LEVEL UP is HIGH.

Application: Adjustable Non-Inverting Op Amp

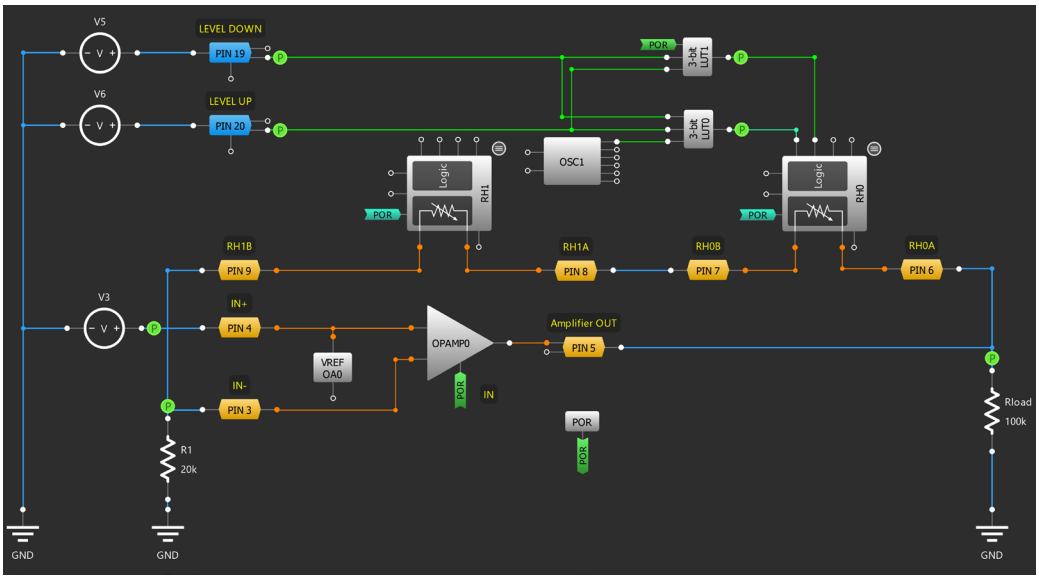
A non-inverting amplifier is an op amp-based amplifier with positive voltage gain. When you apply any signal to the non-inverting input, it does not change its polarity when it gets amplified at the output terminal. So, in that case, the gain of the amplifier is always positive. External control signals allow a user to adjust the gain of the non-inverting amplifier both downward and upward.



Ingredients

- SLG47004V
- One resistor

GreenPAK Diagram



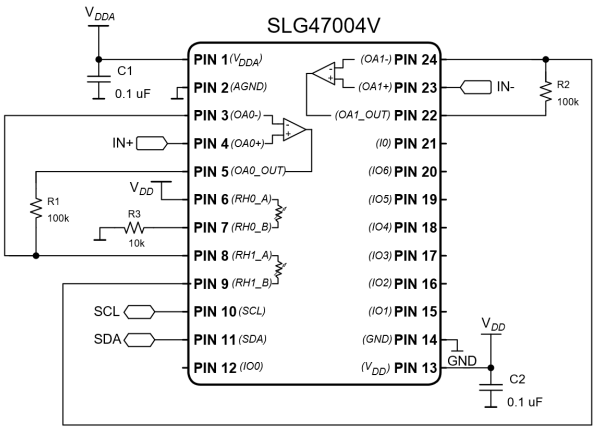
G=V_out/V_inp =(1+(RH0+RH1)/R_1).

Design Steps

1. Enable OpAmp0. Set bandwidth to 8 MHz, enable Charge Pump and set Vref connection to IN+.
2. Enable Vref OA0, set input voltage to 2.048 V and output selection to 256 mV.
3. Set Digital Rheostat 1 in Rheostat mode. Disable Auto-Trim, connect FIFO nRST input to POR and set desired resistance (initial data) for both rheostats.
4. Configure LUT0 to output the OSC1 clock signal when LEVEL UP or LEVEL DOWN signals are HIGH. Configure LUT1 to output a High-level signal when LEVEL UP is HIGH.

Application: Instrumentation Amplifier

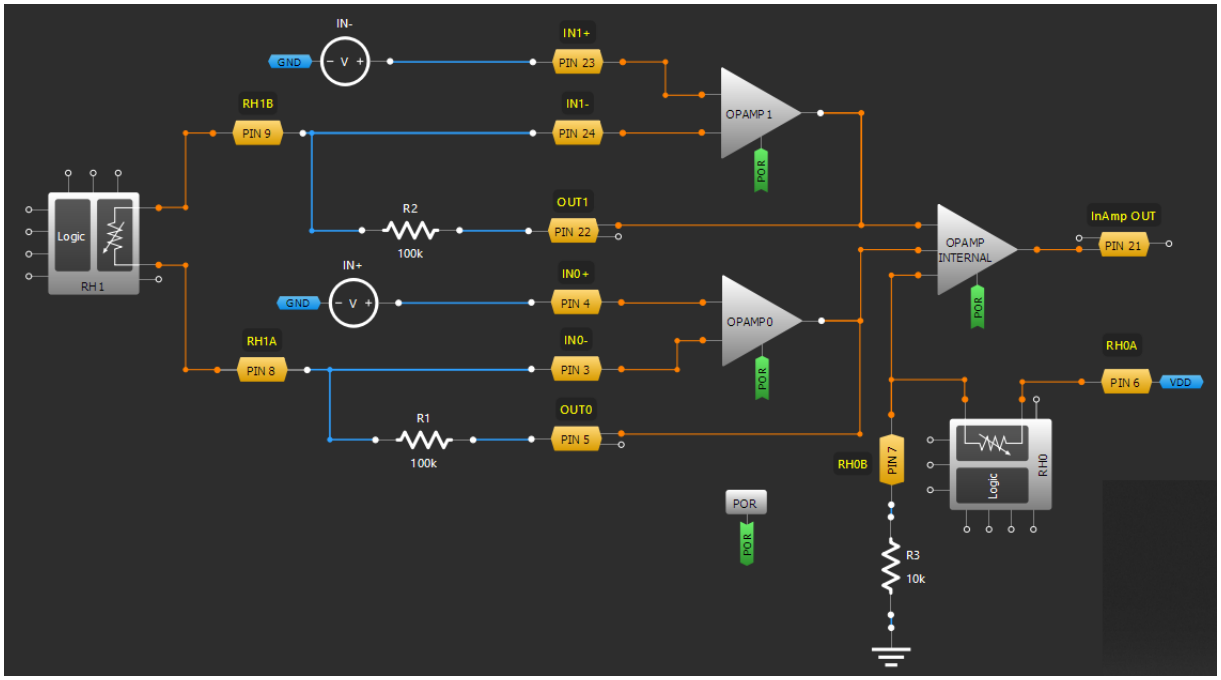
An instrumentation amplifier is a type of differential amplifier equipped with input buffer amplifiers, eliminating the need for input impedance matching. Anong other characteristics are a very low DC offset, low drift, low noise, a very high open-loop gain, very high common-mode rejection ratio, and very high input impedance.



Ingredients

- SLG47004V
- Three resistors

GreenPAK Diagram



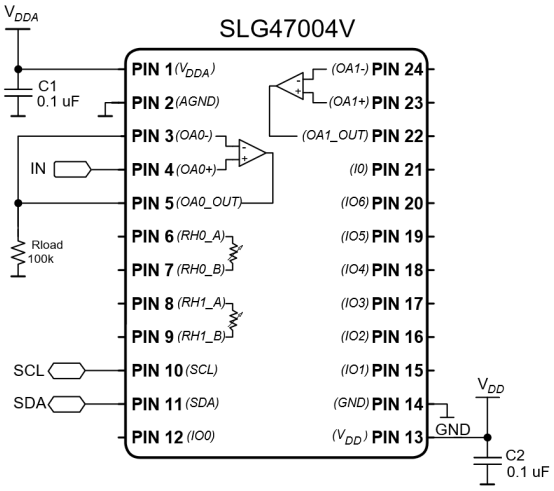
$$V_{OUT} = (1 + \frac{R_1+R_2}{R_{H1}})(V_{IN+} - V_{IN-}) + V_{DD}\frac{R_3}{R_{H0}+R_3}.$$

Design Steps

1. Enable OpAmp0, OpAmp1 and OpAmp Internal. Set bandwidths to 128 kHz and enable Charge Pumps. Set Vref connection for OpAmp Internal to RH0 PIN B.
2. Set Digital Rheostat 1 to Rheostat mode. Disable Auto-Trim, connect FIFO nRST input to POR and set desired resistance (initial data) for both rheostats.

Application: Voltage Follower Using OpAmp

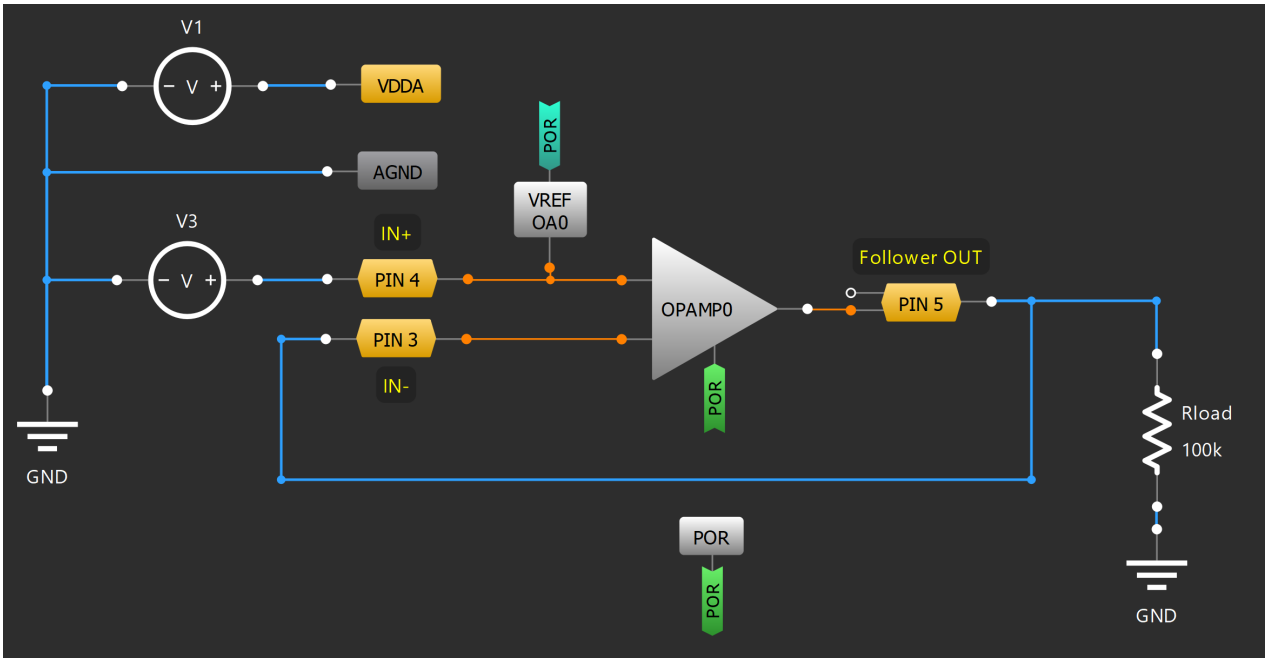
A voltage follower (also known as a buffer amplifier) is an op amp circuit whose output voltage is equal to the input voltage. Hence a voltage follower op amp does not amplify the input signal and has a voltage gain of 1. A voltage follower circuit has a very high input impedance. This characteristic makes it a popular choice in many different types of circuits that require isolation between the input and output signal.



Ingredients

- SLG47004V

GreenPAK Diagram

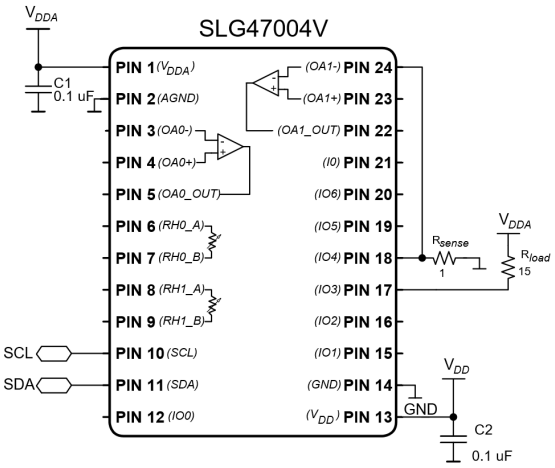


Design Steps

1. Enable OpAmp0. Set bandwidth to 8 MHz, enable Charge Pump and set Vref connection to IN+.
2. Enable Vref OA0, set input voltage to VDDA and output selection to VDDA*(8/64).

Application: Current Sink Using OpAmp and N-channel FET

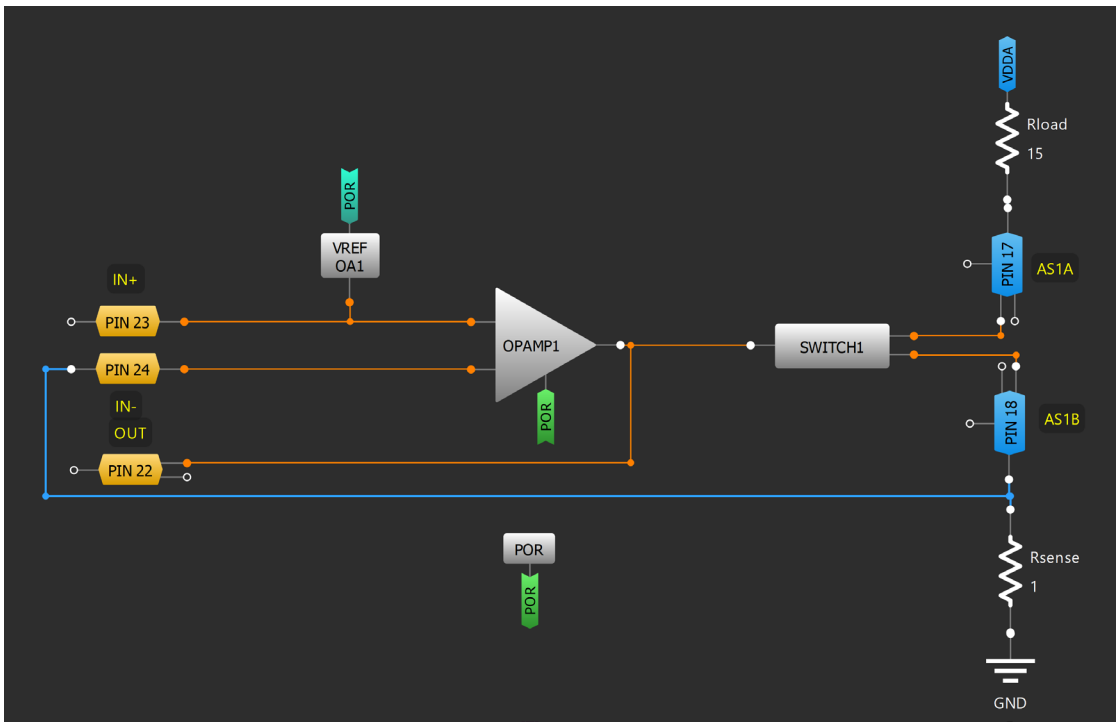
With a particular connection an operational amplifier can work as a current sink. This sink keeps up constant voltage across a constant sense resistor. As a result, the current flowing through the load, irrespective of the load resistance, is constant as well.



Ingredients

- SLG47004V
- One resistor

GreenPAK Diagram

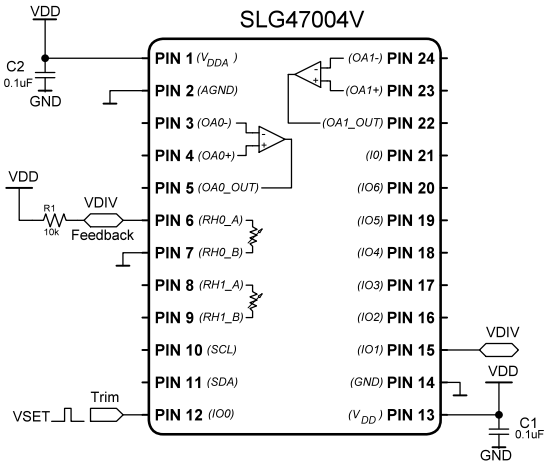


Design Steps

1. Enable OpAmp1. Set bandwidth to 128 kHz, enable Charge Pump and set Vref connection to IN+.
2. Enable Vref OA1. Set input voltage to 2.048 V and output selection to 96 mV. Reference voltage defines current via the load
3. Set Switch1 mode to Analog Switch and Big NMOS Control to setting by OpAmp.

Application: Auto-Trim

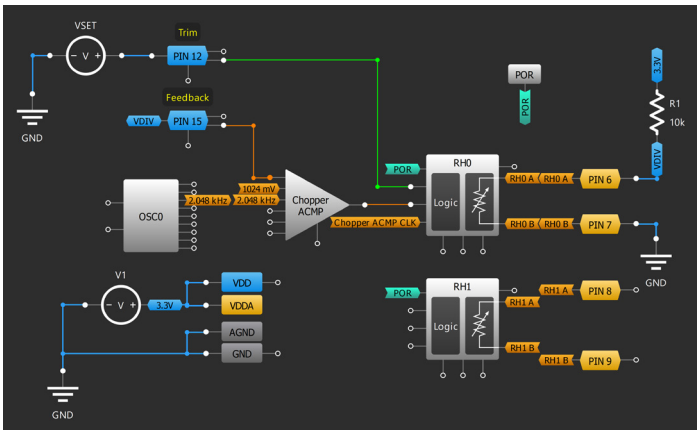
In this application, the SLG47004 is configured as an Auto-Trim Function Example For One-Point Trim.

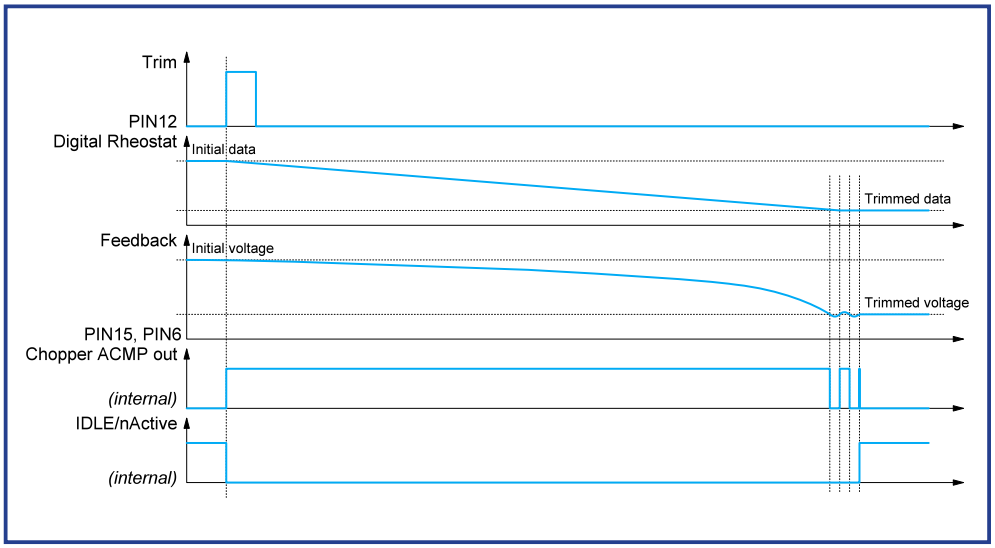


Ingredients

- SLG47004
- Resistor

GreenPAK Diagram



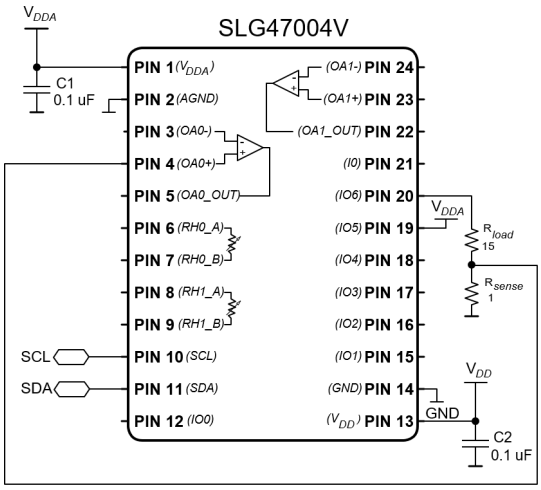


Design Steps

1. Set VSET pulse to “SET” input of RH0 via PIN12 “Trim”.
2. Connect PIN7 to the GND and resistor R1 (gas sensor, thermistor, etc.) to PIN6 and VDD. Ensure the feedback connection VDIV from PIN6 to PIN15.
3. Set internal Vref value for Chopper ACMP reference of 1024 mV. Select Channel0, CH0 clock – OSC0, and IN+ CH0 source – external Vref PIN 15. Connect Chopper ACMP output to nUP/Down of RH0.
4. Configure the RH0: Auto-Trim - Enable, Active level for UP/DOWN - Up when LOW, Resistance (initial data) – 511 (~50.7096 kΩ).

Application: Current Source Using OpAmp and P-channel FET

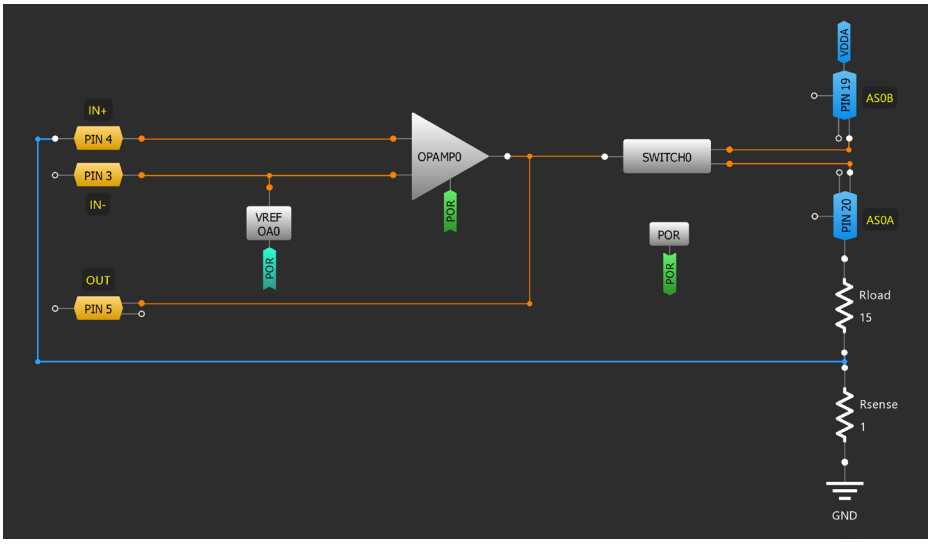
An operational amplifier at a certain connection can work as a current source. This source keeps up a constant voltage across the constant sense resistor. As a result, the current flowing through the load, irrespective of the load resistance, is constant as well.



Ingredients

- SLG47004
- Resistor

GreenPAK Diagram



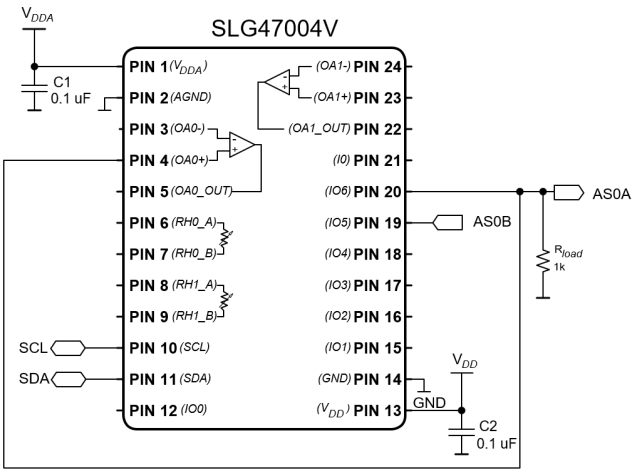
$$I_{load} = \frac{V_{ref}}{R_{sense}}$$

Design Steps

1. Enable OpAmp0. Set bandwidth to 128 kHz, enable Charge Pump and set Vref connection to IN-.
2. Enable Vref OA0. Set input voltage to 2.048 V and output selection to 96 mV.
3. Set Switch0 mode to Analog Switch and Big PMOS Control to setting by OpAmp.

Application: Voltage Regulator Using OpAmp

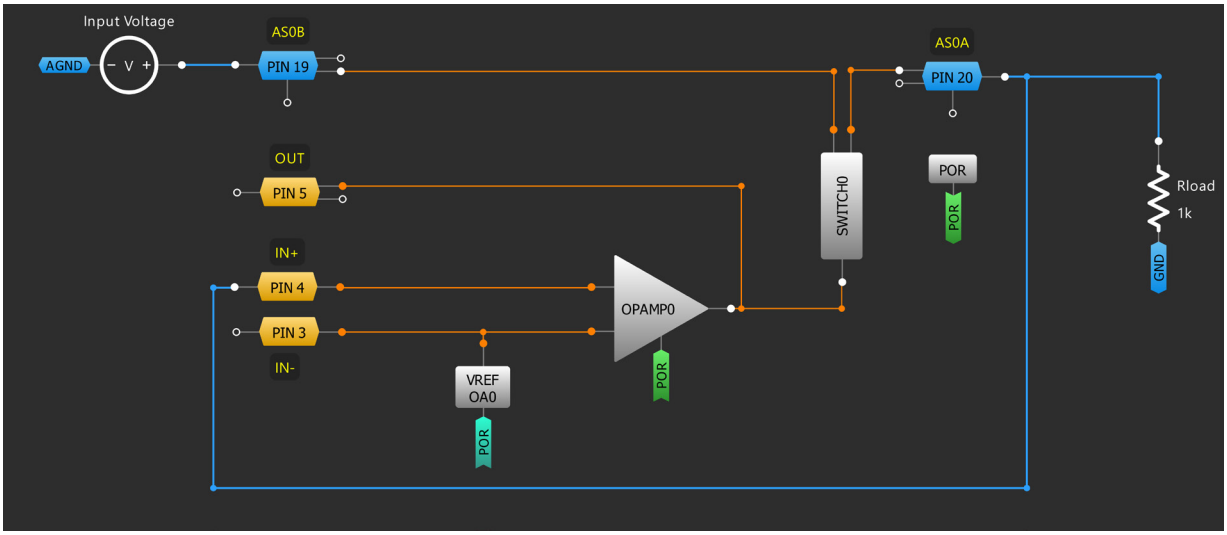
An OpAmp-based voltage regulator is a circuit that uses operational amplifier to regulate and stabilize the output voltage. By comparing the output voltage to a reference voltage, the OpAmp adjusts its output to maintain a constant voltage level. This feedback mechanism allows the voltage regulator to compensate for input voltage variations and provide a consistent and reliable output voltage.



Ingredients

- SLG47004V
- No other components needed

GreenPAK Diagram

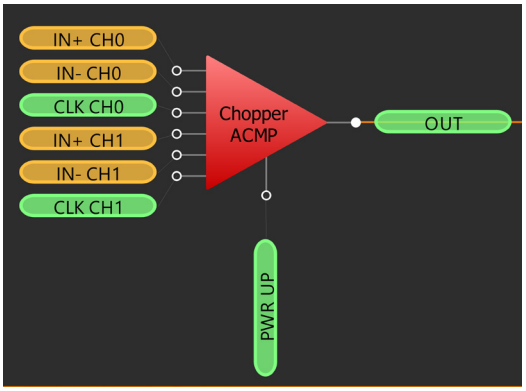


Design Steps

1. Enable OpAmp0. Set bandwidth to 128 kHz, enable Charge Pump and set Vref connection to IN-.
2. Enable Vref OA0. Set input voltage to 2.048 V and output selection to 1792 mV.
3. Set Switch0 mode to Analog Switch and Big PMOS Control to setting by OpAmp.
4. After applying the input voltage, the output voltage remains stable and equal to VOUT = VREF, regardless of input voltage variations.

Technique: Using Chopper ACMP with Digital Rheostats

There is one Chopper Rail-to-Rail Analog Comparator (ACMP) macrocell in the SLG47004. It is possible to use Chopper ACMP to do in-system trim by changing the Rheostat resistance in Auto-Trim mode (see Application: Auto-Trim function).

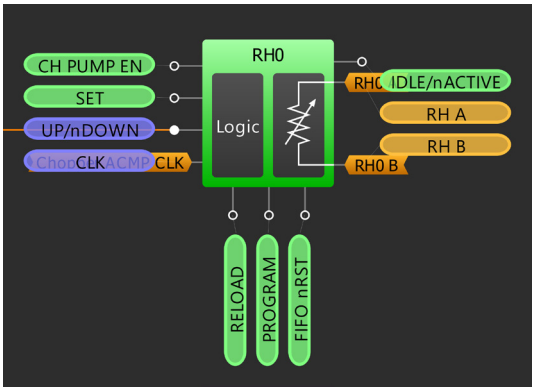


Activation

When used in Auto-Trim mode, the Chopper ACMP is automatically enabled by the Auto-Trim control logic when Rheostat calibration is in progress.

For using the Auto-Trim function the following preliminary steps must be taken:

- Enable Auto-Trim in Digital Rheostat0/1 Properties.
- Choose IN+ CH0/1 of the Chopper ACMP. It can be user system voltage feedback. If the Auto-Trim function is used for two rheostats, IN+ CH0/1 must be configured for both rheostats (for cases when SET0 is latched and when SET1 is latched).
- Configure IN- CH0/1. It can be user desired set point threshold. If the Auto-Trim function is used for two rheostats, IN- CH0/1 must be configured for both rheostats (for cases when Set0 is latched and when Set1 is latched).



- Select Channel in Chopper ACMP Properties to work with Chopper ACMP.
- Configure inverting or non-inverting Chopper ACMP output.
- Select clock source (internal clock from internal pre-dividers or from connection matrix). Note that in Auto-Trim mode clock source frequency is limited by the Chopper Comparator time response. Therefore, the clock source frequency must not be greater than <fChACMP> kHz.
- Start the Auto-Trim process by setting the SET0/1 input of the RH0/1 block to a HIGH level. The Trim process starts with a rising edge on Set input. This Set signal is latched until the end of the Auto-Trim process. The SET signal will enable the Chopper ACMP and the Vref if they were not enabled earlier. The counter starts to count up or down depending on the level at the UP/nDOWN input of the RH0/1 (Chopper ACMP output). If the user selected the “Internal Clock” option for Clock input, these clock pulses are generated automatically during the trim time. Each rising edge of the Clock pulse changes the value of the counter and, consequently, the value of the rheostat.

The Auto-Trim process stops if one of three stop conditions occur:

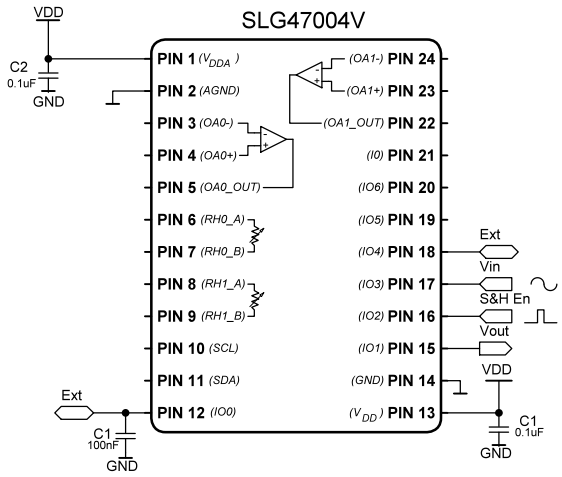
1) A subsequent change on Up/Down input in the moment of rising edge on Clock input.
2) the value of rheostat reaches its maximum (1023).
3) the value of rheostat reaches its minimum (0).

Stop conditions result in a change of the IDLE/nACTIVE signal, which resets the internal Auto-Trim logic. Note that the SET input is edge sensitive, but if the user keeps a HIGH logic level at this input after reaching the set point, the Programmable Trim block will continue to operate and continue to switch rheostat around the set point.

- To start a new Auto-Trim process user should reapply a HIGH level on Set input.

Application: Sample and Hold Circuit

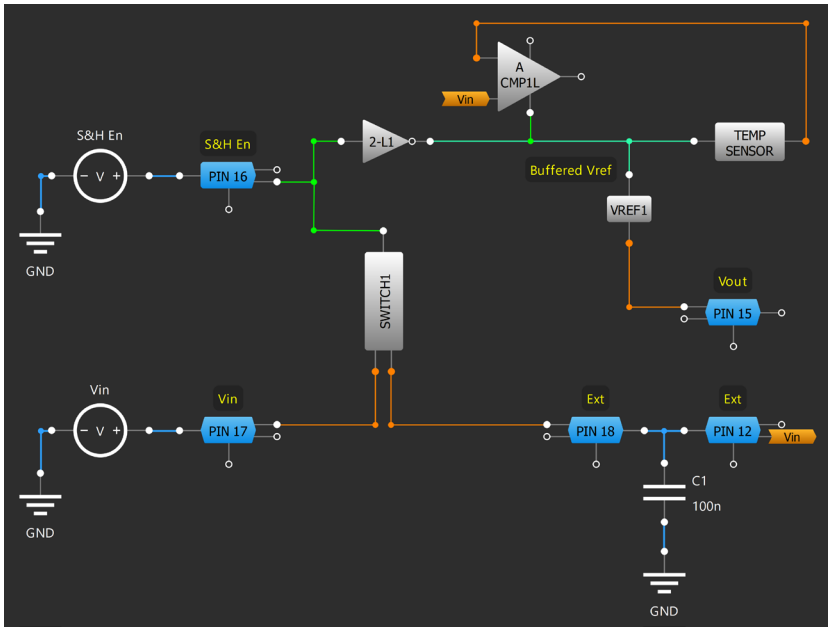
In this application, the SLG47004 is configured as a Sample and Hold Circuit which consists of an analog switch, capacitor, and buffer.



Ingredients

- SLG47004
- Capacitor

GreenPAK Diagram



Design Steps

1. Configure “Enable by Matrix” Small PMOS enable of SWITCH1.
2. Set the S&H En signal and connect it to PIN16 (Digital Input, Pull Down 1M).
3. Enable SWITCH1 by connecting to PIN16.
4. Connect PIN17 (Analog In/Out, Floating) to the Vin signal source, PIN18 to PIN 12 (both Analog In/Out, Floating), and provide the connection with a 100 nF capacitor.
5. Configure 2-L1 as an Inverter and connect its IN0 to PIN16 and OUT to PWR UP of ACMP1L and TEMP SENSOR.
6. Configure the ACMP1L: Vref source selection – VDDA, IN+ source – TEMP SENSOR output, In- Low to High/High to Low source – Ext. Vref PIN12.
7. Configure the power down source of TEMP SENSOR as “From matrix”.
8. PIN15 is Vout of the Sample and Hold Circuit.

Document No.: R11TB0003CE0000