Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN2115

Generate Triangle and Trapezoid Waveforms with a Switched Capacitor Integrator

Author: Sigurd Peterson

Associated Project: Yes

Associated Part Family: CY8C24/27/29xxx

Software Version: PSoC® Designer™ 5.4

Related Application Notes: AN2041

Trapezoid and triangle waveforms can be generated by modulating the input to a switched capacitor block configured as an integrator. This application note shows how to configure the SCBlock User Module as an integrator. It also demonstrates the analog modulator feature of the PSoC® device to generate trapezoid and triangle waveforms with this integrator.

Introduction

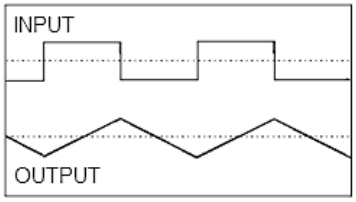
Many applications require ramped waveforms. These are often generated by controlling a digital to analog converter (DAC) with software. However, this places a significant load on the CPU. Another technique that produces a controlled ramp is driving an integrator with a waveform generated by simple digital logic. The PSoC device can be configured with switched capacitor integrators driven by digital pulse-width modulators. This generates these waveforms without adding to CPU overhead. This application note provides the following:

- A brief explanation of waveform generation with integrators.
- An example of the SCBlock configured as an integrator.
- A quick overview of the PSoC project.

Signal Generation with Integrators

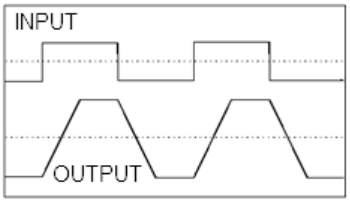
Inputting a constant voltage to an integrator results in a linear ramp of its output voltage; the slope is proportional to the input voltage. Inverting the polarity of the input voltage produces a linear ramp with the opposite slope. Driving the integrator with a symmetrical square wave around the integrator reference results in a triangle wave (see Figure 1). Maintain the duty cycle of this square wave at 50 percent to avoid a slow drift of the integrator voltage toward one of the supply rails.

Figure 1. Integrating a Square Wave



If the integrator reaches either supply rail, it saturates and stops integrating. This results in output voltage remaining at a constant value near the rail (see Figure 2). This can be used if the goal is to produce a trapezoid waveform with controlled edge speeds. Trapezoid waveforms do not have the 50 percent duty-cycle requirement because they are intended to spend time at the rails.

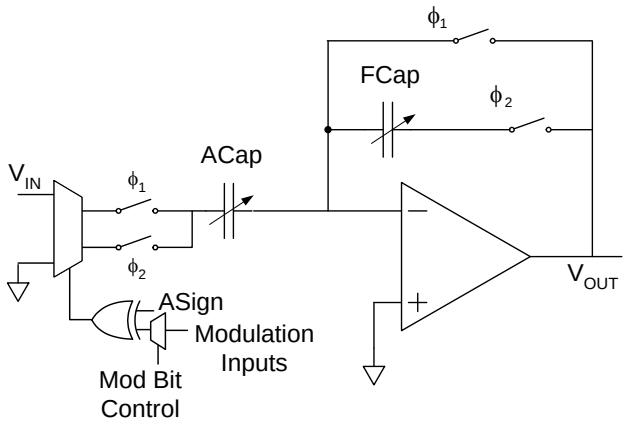
Figure 2. Saturating Integrator



Configuring an SCBlock Integrator

The key settings to configure the SCBlock User Module to be an integrator are FSW1 = On, FSW0 = Off, and AUTOZERO = On. The ratio between capacitors FCap and Acap determines the size of the steps in the integration ramp in proportion to the input voltage. For ramp waveform generation, capacitors Bcap and Ccap are zero so they are not shown in Figure 3.

Figure 3. SCBlock Integrator diagram



Example Project

The associated project demonstrates a variety of waveform generation methods and shows how the blocks can be connected and configured. There is no attempt to provide a user interface or respond to external inputs. It provides access to signals by bringing them to pins where they can be viewed. See Table 1 for the pin mapping, Figure 4 for the user module settings, Figure 5 for the user module placement, and Figure 6 for the Global Resources.

Figure 4. SCBlock Integrator Parameter Settings

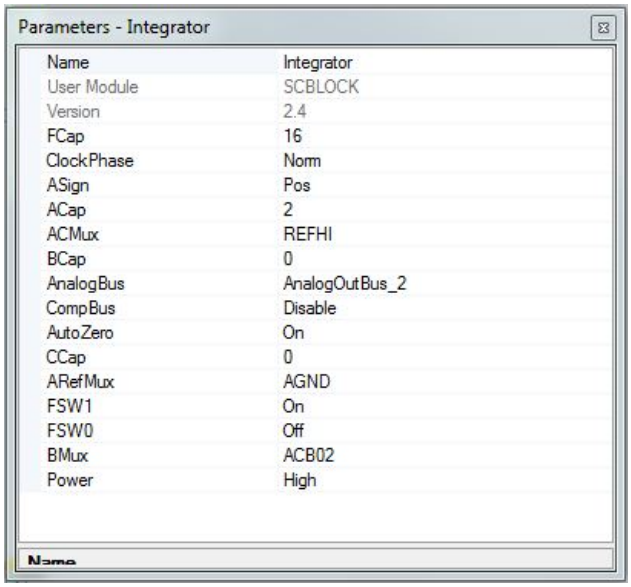


Table 1. Signal Pin Mapping for Example Project

Signal	Port Pin
1 kHz Trapezoid (Modulated DAC)	P0[3]
1 kHz Triangle (Modulated DAC)	P0[5]
10 kHz Triangle (Modulated Integrator)	P0[4]
1 kHz Square (modulation signal)	P2[1]
10 kHz Square (modulation signal)	P2[0]

Figure 5. Example Project – User Module Placement

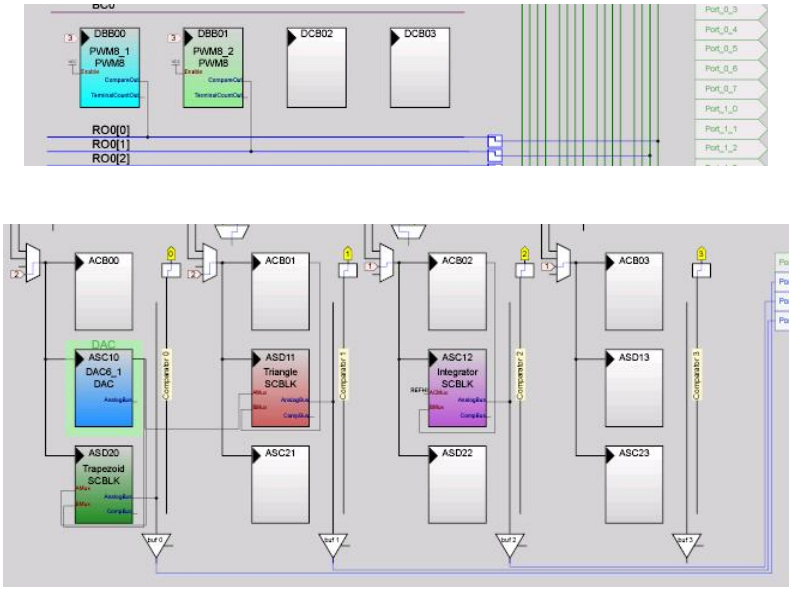
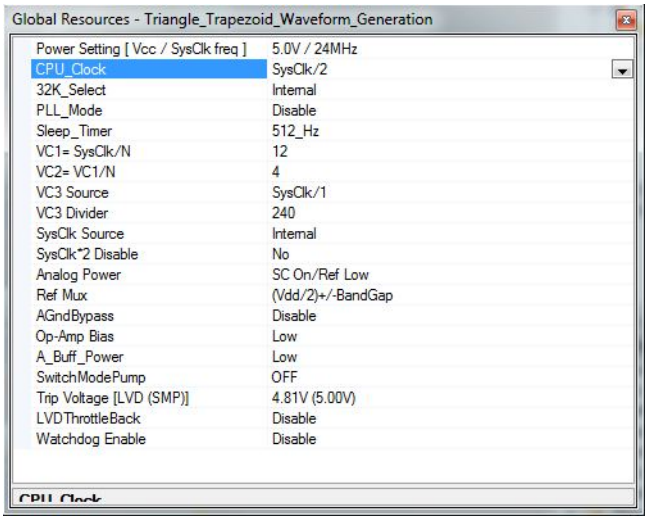


Figure 6. Example project – Global Resources



Modulating the Input to PSoC Integrators

All type C switched capacitor blocks in the PSoC have analog modulation hardware. This modulation hardware is implemented with the 'XOR' gate shown in Figure 3. Placing the SCBlock User Module integrator in a type C block and enabling the analog modulators inside these blocks can modulate the input to the integrator. In this case, the ACMux input is set to REFHI. This REFHI input then has its polarity flipped as the modulation input into the 'XOR' gate changes between HIGH and LOW.

Alternatively, a DAC can be placed in a type C switched capacitor block so that its analog modulator hardware can flip the polarity of its output. The output of the DAC is input into integrators located in any connected block. This method enables generating multiple synchronous waveforms.

The hardware modulation signal is selected by writing to Analog Modulator Control Register 0 or 1 (AMD_CR0 or AMD_CR1). The source code in Appendix A shows how this is done. There are eight possible modulation sources.

- No modulation (off)
- Modulate using Global Output Bus, even bus bit 0 (GOE[0])
- Modulate using Global Output Bus, even bus bit 1 (GOE[1])
- Modulate using Row 0 Broadcast Bus

- Modulate using Analog Column Comparator 0
- Modulate using Analog Column Comparator 1
- Modulate using Analog Column Comparator 2
- Modulate using Analog Column Comparator 3

Direct Modulation of an Integrator Block

The SCBlock User Module placed in ASC12 is configured as an integrator with the ACMux input connected to the REFHI signal. The analog modulator in that block causes the integrator to flip between a positive REFHI and a negative REFHI. The slope is adjusted with the A/F ratio. The PWM8_1 User Module in DBB00 has its output connected to GOE[0]. This signal is connected to the analog modulator in ASC12. The output of this integrator is on P0[4].

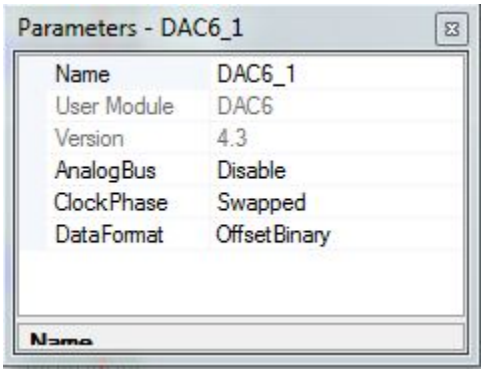
Driving Integrators with a Modulated DAC

By placing a 6-bit DAC into the ASC10 block, it is possible to modulate its output polarity with the analog modulator. Because it drives another switched capacitor block, its ClockPhase setting must be set to Swapped. See Figure 7 for the DAC settings.

The PWM8_2 user module in DBB01 is configured to drive GOE[1] so that it can be connected to the analog modulator in block ASC10. GOE[1] is also visible outside the package on P2[1]. Both ASD20 and ASD11 have routing paths from ASC10, so they can be configured as integrators driven by the modulated output of the DAC.

The A/F ratio for the block in ASD20 is set to 4:16 (Acap = 4, Fcap = 16) to generate a trapezoid within the adjustment range of the DAC. The value written to the DAC controls how fast the trapezoid edges rise and fall. Lower DAC values produce slower edge speeds. If the DAC value is decreased enough, the trapezoid becomes a triangle wave. The trapezoid waveform from this block is seen on P0[3].

Figure 7. DAC6_1 Parameter Settings



The integrator in block ASD11 has an A/F ratio of 1:32 (Acap = 1, Fcap = 32) and ramp more slowly to produce a triangle waveform. The value written to the DAC provides fine control of the amplitude of the triangle generated on P0[5]. Smaller values cause low amplitudes.

A/F Cap Ratio for Slope Control

Use the ratio between the A and F capacitors to adjust the slope of the integrator ramp. When the ratio is 1:1, each sample clock increments the output by the same voltage as the input signal. If Acap = 1 and Fcap = 16, it takes the output 16 sample-clocks to change by the same voltage as the integrator input.

Digital Blocks Driving Integrators

For a triangle waveform, the period must be a multiple of the analog column clock. This multiple can be computed by the formula:

$$M = 4 \times 2 \times STEPS \qquad \text{Equation 1}$$

In Equation 1 $STEPS$ represents the number of steps in each edge of the waveform. When using a PWM block with its CompareType parameter set to Less Than, the PulseWidth parameter is loaded with a value of $M/2$.

Summary

Switched capacitor blocks, configured as integrators, combined with the analog modulator simplify generation of ramped waveforms. These are sampled waveforms. For some applications, continuous time filters are needed to smooth out the steps. For more information on switched capacitor blocks and the PSoC analog modulator, see the application note AN2041. See Appendix A for the example project source code.

Appendix A

Example Project Source Code

```
//--------------------------------------------------------------------------------
// C main line
//--------------------------------------------------------------------------------

#include <m8c.h>        // part specific constants and macros
#include “PSoCAPI.h”    // PSoC API definitions for all User Modules

void main()
{
    // initialize the digital user modules
   PWM8_1_Start();
   PWM8_2_Start();

    // initialize the analog user modules
   DAC6_1_Start(DAC6_1_HIGHPOWER);
   DAC6_1_WriteStall(0);                        //Use DAC to modulate at the rails with
value of 0
   Integrator_Start(Integrator_HIGHPOWER);
   Trapezoid_Start(Trapezoid_HIGHPOWER);
   Triangle_Start(Triangle_HIGHPOWER);

    // enable analog modulators
    // ASC10 modulated with GOE[1], ASC12 modulated with GOE[0]
   AMD_CR0 |= 0x21;
    while(1);    // Infinite loop
}
```

Document History

Document Title: Generate Triangle and Trapezoid Waveforms with a Switched Capacitor Integrator – AN2115

Document Number: 001-36021

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1514403	OGNE	09/27/07	New publication of existing application note
*A	2641908	OGNE	01/21/09	Added parts CY8C24/27/29xxx, CY8CLED04/08/16 Updated software version to PSoC Designer™ 5.0 Updated in new template.
*B	3281121	QVS	06/15/11	No technical updates.
*C	3744325	ARVI	09/12/12	Updated project version.
*D	4413783	RICA	06/20/2014	Updated in new template. Completing Sunset Review.
*E	4622507	ASRI	01/13/2015	Updated Software Version as “PSoC® Designer™ 5.4” in page 1. Updated attached associated project to PSoC Designer™ 5.4.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

PSoC is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2007-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.
This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.
Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.
Use may be limited by and subject to the applicable Cypress software license agreement.