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.



THIS SPEC IS OBSOLETE

Spec No: 001-60220

Spec Title: PSOC(R) 3 / PSOC 5LP MULTIPLEXED COMPARATOR - AN60220

Sunset Owner: Rajiv Vasanth Badiger (RJVB)

Replaced by: 001-95018

PSoC® 3 / PSoC 5LP Multiplexed Comparator

Author: Rajiv Vasanth Badiger
Associated Project: Yes
Associated Part Family: All PSoC 3 and PSoC 5LP parts
Software Version: PSoC Creator™3.2
For a complete list of the application notes, Related Application Notes.
If you have a question, or need help with this application note, contact the author at rjvb@cypress.com

AN60220 describes a way to multiplex a single comparator to monitor multiple signals in PSoC® 3 and PSoC 5LP. The design allows a configurable threshold to be applied to each channel, and it runs with no CPU use during the scanning process.

Introduction

PSoC 3 and PSoC 5LP devices include a maximum of four comparators in their analog arsenals that can be used for analog signal comparisons. However, if there is a requirement for multiple (more than four) signals to be compared with a reference signal, you can meet it in several ways:

- **Opamps:** PSoC 3 and PSoC 5LP devices include up to four opamps. Because they can be operated in open loop, they can be used as comparators. However, this requires two extra pins if the output needs to be routed back to the digital section.
- **Special I/Os (SIOs):** PSoC 3 and PSoC 5LP devices contain SIOs that can be used for comparator applications. Either four or eight SIOs are available, depending on the specific device. However, this solution suffers from high input offset voltage. For more information, see AN60580 – SIO Tips and Tricks in PSoC 3 / PSoC 5LP. For SIO comparator specifications, refer to the PSoC 3 and PSoC 5LP device datasheets.
- **Multiplexed comparator:** Comparators can be multiplexed similar to ADCs, where channels are scanned one at a time. This solution is useful for low-frequency signal applications.

This application note describes a technique for multiplexing the comparator for multiple signals with the ability to uniquely configure thresholds for each channel. This technique is a complete hardware solution that uses no CPU time to scan the channels.

This document assumes that you are familiar with PSoC Creator™ and the PSoC 3 or PSoC 5LP architecture. If you are new to PSoC, read AN54181 – Getting Started with PSoC 3 and AN77759 – Getting Started with PSoC 5LP. If you are new to PSoC Creator, visit the PSoC Creator home page.

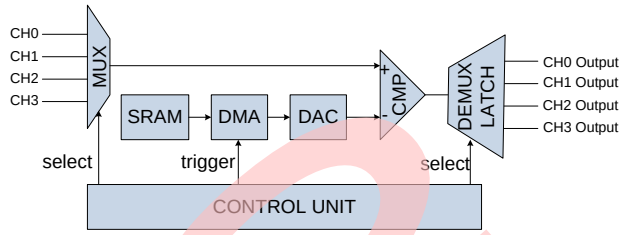
Multiplexed Comparator Design

This design implements a four-channel multiplexed comparator, as shown in the block diagram in Figure 1. It is easily scalable for a different number of channels. The channels are selected one at a time and compared with the DAC value. There is a unique DAC value for each channel, so you can set a unique threshold value.

The output of the comparator is demultiplexed to individual outputs. The demultiplexer is synchronized to the multiplexer. The DAC sets the threshold voltage. The DAC value is updated by direct memory access (DMA) hardware from SRAM. You need to update this value in SRAM to alter the threshold voltage for a particular channel (thresholds are stored in SRAM).

OBSOLETE

Figure 1. Block Diagram of Multiplexed Comparator



Features

- Unique threshold voltage for each channel
- Flexibility in adjusting the number of channels
- Nonoverlapping channel switching (break before make switch)
- No CPU involvement in channel multiplexing
- Configurable scanning rate
- Dedicated latch for each channel
- Option of CPU reading the channel status
- Option of triggering an interrupt on each channel output

Implementing Multiplexed Comparator in PSoC 3 and PSoC 5LP

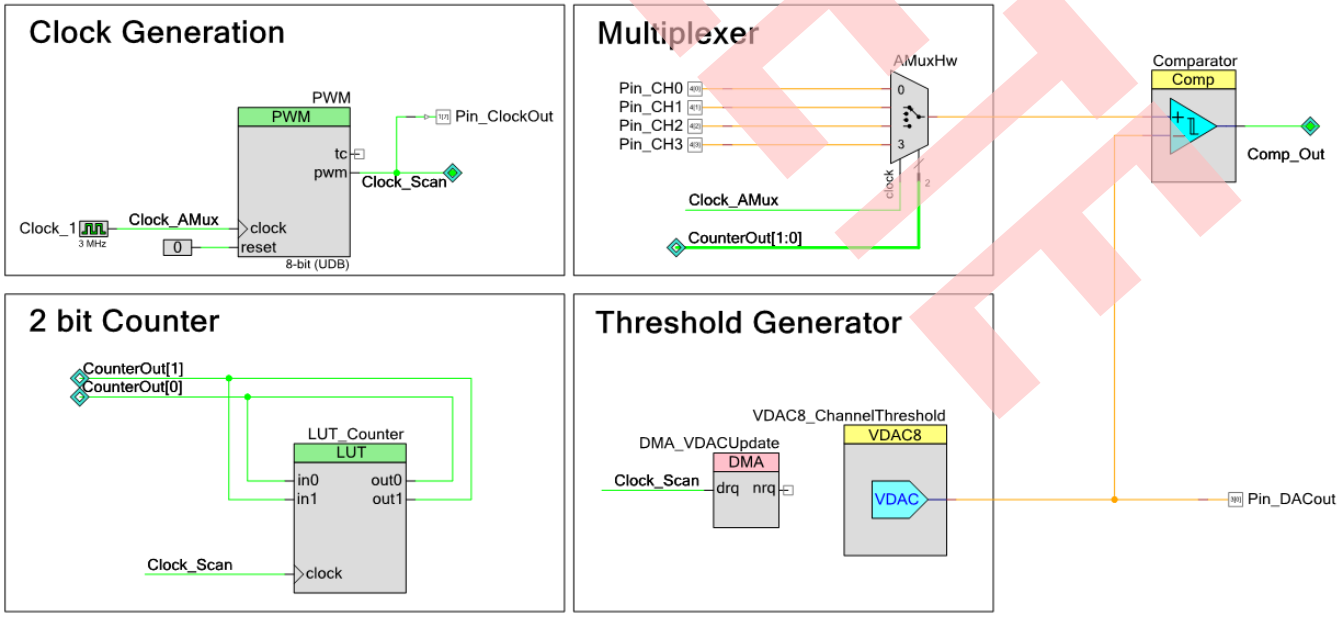
As shown in Figure 1, the following resources are required to build a complete solution:

- Control unit
- Multiplexer for analog signals
- Voltage DAC
- (DMA
- Comparator
- Digital demultiplexer and latch

PSoC 3 and PSoC 5LP provide the necessary hardware to implement all these functions. PSoC Creator simplifies the design task with customizable Components.

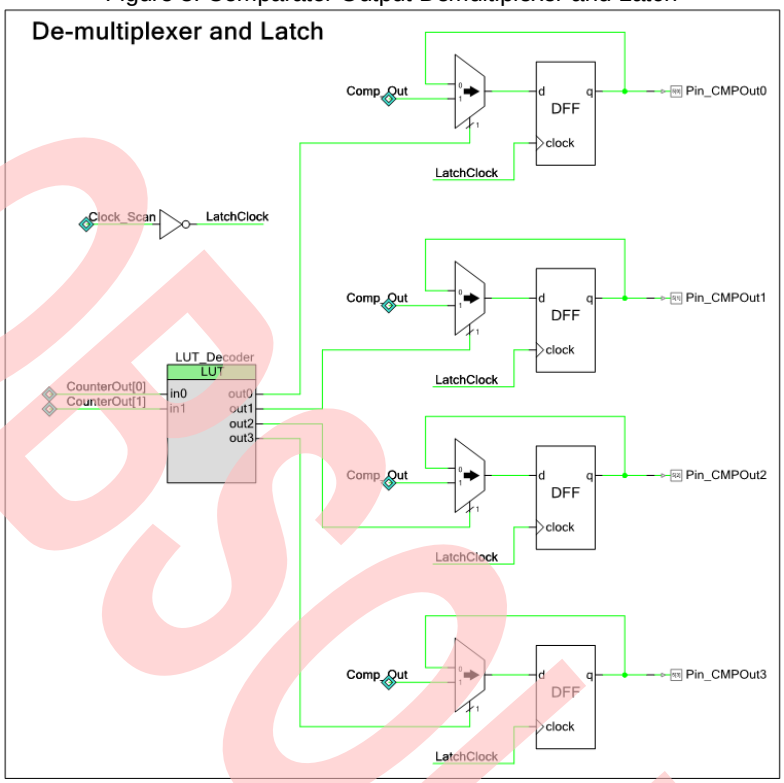
Figure 2**Error! Reference source not found.** shows the multiplexed comparator design and Figure 3 shows the comparator output demultiplexer with latch in PSoC Creator.

Figure 2. Multiplexed Comparator Design



OBSOLETE

Figure 3. Comparator Output Demultiplexer and Latch



Control Unit

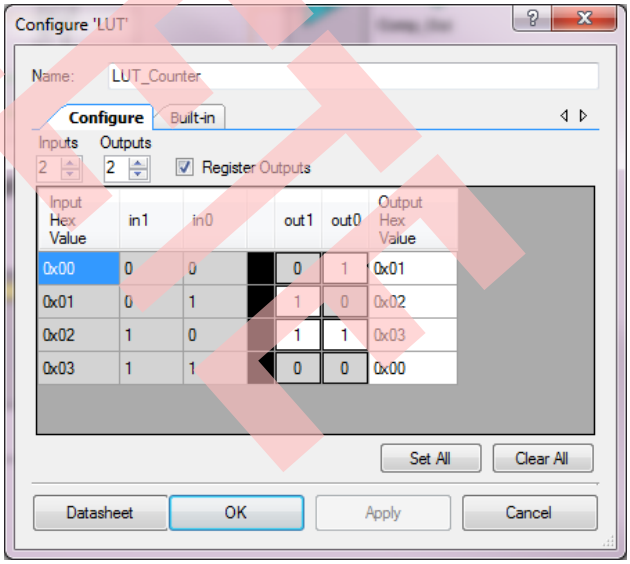
This section consists of a 2-bit counter using a LUT Component, as shown in Figure 4, and clock generation using a PWM Component.

- **2-bit counter:** Selects the channel in the analog multiplexer and the digital demultiplexer. The LUT Component is configured with two inputs and two outputs, resulting in four possible states, as Figure 4 shows. It is clocked by the PWM output, causing the counter to increment on a positive (rising) edge. To learn more about the LUT Component and its applications, see application note AN62510.
- **Clock generation:** Uses a PWM Component with an input clock of 3 MHz. The output PWM frequency sets the scanning speed of the input channels, as given by the following equation:

$$Scanning\ rate/channel = \frac{PWM\ Output\ Frequency}{Number\ of\ channels}$$

In this project, the PWM output frequency is set to 100 kHz, and four channels are used. Therefore, the scanning rate per channel becomes 25 kHz. The duty cycle setting gives a configurable time for the DAC voltage and input channel voltage to settle on the internal analog bus. This is because the comparator output is latched on the negative edge of the PWM output. The duty cycle of the PWM output is set to 50 percent.

Figure 4. 2-Bit Counter - LUT Configuration

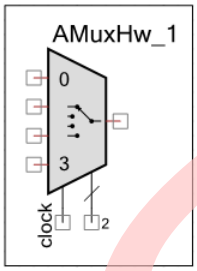


Multiplexer for Analog Signals

The analog hardware multiplexer (AMuxHw) Component is used for the multiplexing function, as shown in Figure 5.

OBSOLETE

Figure 5. AMuxHw Component

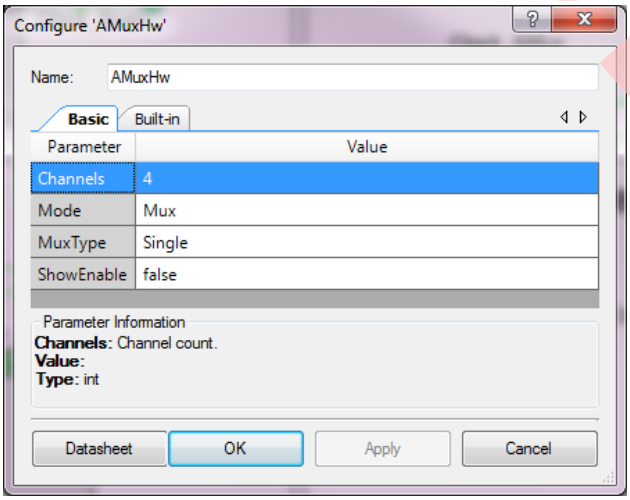


It uses the Analog Global bus (AG) and Analog Mux (AMUX) bus switches, which route the analog signal in and out through the ports. These switches are controlled by digital signals. To synchronize channel switching with the other digital logic, a clock terminal is provided.

In this design, the synchronizing clock is the same as the PWM clock input. It is important to select a much higher frequency clock relative to the channel scanning frequency because the multiplexer takes two clock cycles to change the channel: one to break the previous connection and one to make the next connection.

Figure 6 shows the AMuxHw Component configuration.

Figure 6. AMuxHw Component Configuration



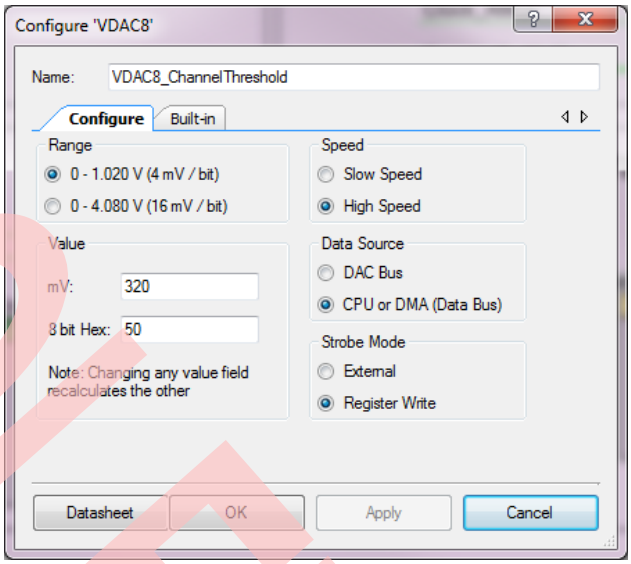
Four channels are configured. The **Mode** of the Component is set to “Mux,” so that only one channel can be connected to the output at a time. Setting it to “Switch” enables multiple input channels to be connected at a time. If the input analog signals are single ended (referenced to PSoC ground), select “Single” for **MuxType**. For differential signals, select “Differential” for **MuxType**. This project uses single-ended input signals.

For more information about the AMuxHw Component, see the component datasheet.

Voltage DAC

An 8-bit voltage DAC (VDAC) converter that is updated by DMA provides the threshold voltage for the comparator. DMA takes the threshold values set by the user from SRAM. Because data resides in memory, threshold values can be changed during run time. Figure 7 shows the VDAC configuration.

Figure 7. VDAC Configuration



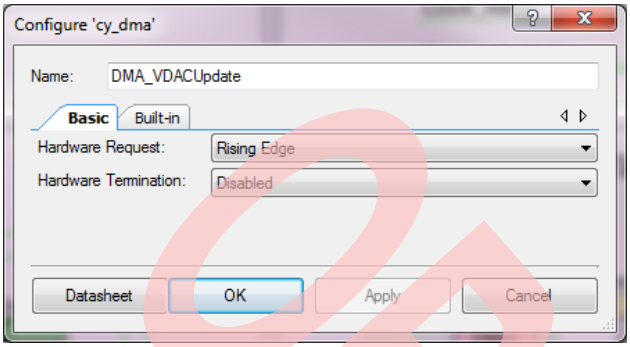
The VDAC output **Range** is set to “0 - 1.020 V.” **Speed** is set to “High Speed” because the VDAC update time affects the maximum scanning rate that can be achieved. Because DMA is used to update the DAC, **Data Source** is set to the “CPU or DMA (Data Bus)” option.

DMA Configuration

The DMA trigger terminal DRQ is enabled. A rising edge signal at this pin initiates a data transfer. The terminal is driven by the PWM output clock, which is used to keep the VDAC voltage (threshold to the comparator) in sync with the channel selected by the 2-bit counter. Figure 8 shows the DMA configuration.

OBSOLETE

Figure 8. DMA Configuration



One transaction descriptor (TD) is defined to carry out transfers from memory to VDAC. Because four channels are set in this application, four 8-bit transfers are required. Four threshold values are stored in memory. A single TD can perform this transfer by using the auto source address increment feature, where the source address can be automatically incremented for the defined number of bytes. This requires the threshold values to be stored in contiguous locations in memory (using an array).

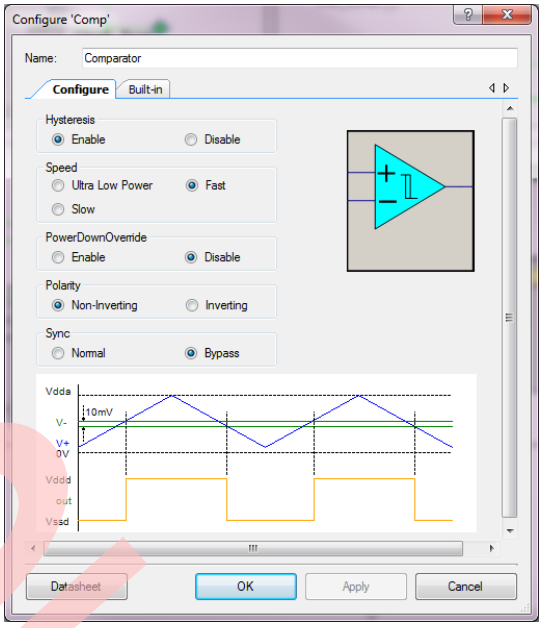
The DRQ trigger signal and TD configuration enables the DMA to transfer 8-bit data from memory to VDAC on receiving each trigger. After every trigger, the source address, which points to threshold values in memory, is incremented to keep it in sync with the selected channel. After transferring 4 bytes (from channel 0 through to channel 3), the source pointer returns to channel 0.

There is an offset in synchronization, however, because of the initial conditions of the 2-bit counter and the DAC value. To begin, the 2-bit counter keeps channel 0 selected, but the DMA waits for the trigger to move the threshold value. Note that the trigger signal for DMA and the clock input to the 2-bit counter is the same PWM output signal. Thus, when DMA receives the trigger, the 2-bit counter also moves to select the second channel. Thus, the DMA state is off by one clock compared to the 2-bit counter state. To compensate, in the firmware, the array declared for the threshold value should hold the channel 1 threshold value at a '0' index, the channel 2 threshold at a '1' index, the channel 3 threshold at a '2' index, and the channel 0 threshold at a '3' index.

Comparator Configuration

Figure 9 shows the Comparator configuration.

Figure 9. Comparator Configuration

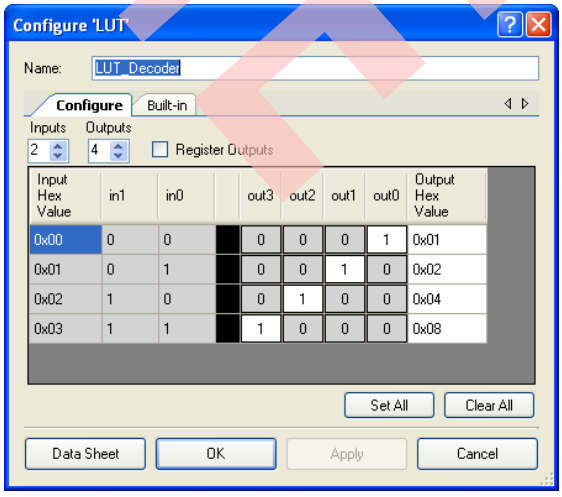


Hysteresis (approximately 10 mV) is enabled, and **Speed** is set to "Fast" to support a low comparator response time. See the Comparator component datasheet for the response time in different modes.

Demultiplexer and Latch

Figure 3 shows the demultiplexer and the latch unit. LUT_decoder implements a 2:4 demultiplexer. It splits the comparator output to enable individual outputs for each channel. Figure 10 shows the LUT configuration.

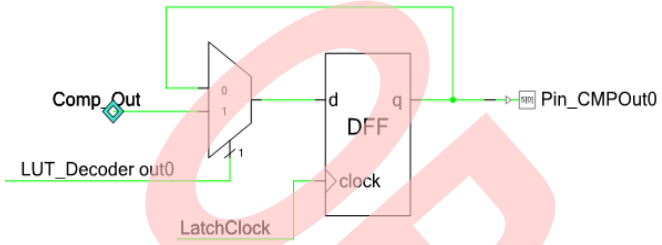
Figure 10. LUT Configuration



OBSOLETE

The output of this decoder is used to select a line for latching the comparator output. Figure 11 shows the logic for channel 0.

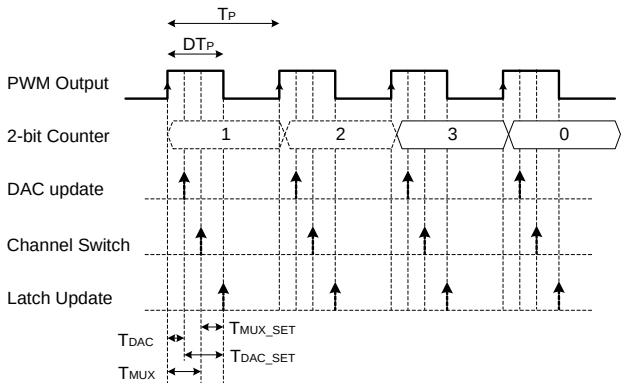
Figure 11. Latch



LatchClock is the inverted version of the PWM output. So, the comparator output gets latched at the negative edge of the PWM output, which allows approximately one-half the PWM cycle time for the signal to settle on the bus after connection. If the LUT decoder output (LUT_Decoder_out0 for channel 0) is 0, then the previous value is maintained at the output. When channel 0 is selected, the LUT decoder output becomes 1, and the comparator output is routed to the input of the D flip-flop. On the negative edge, the D flip-flop will store the comparator status. This logic is repeated for the remaining three channels.

Figure 12 shows the alignment of the functions relative to one another.

Figure 12. Timing Diagram



The 2-bit counter output is incremented for every clock cycle. Notice the channel change, DAC update, and output latch enable instants.

T_{MUX} is the delay created in channel connections from using the AMuxHw Component. A delay of two clock (Clock_AMux) cycles occurs in connecting the channel; one clock is used to disconnect the previous channel and another is used to connect the new channel. In this project, Clock_AMux is set to 3 MHz. This means that T_{MUX} is equal to (2 / 3 MHz= 0.66 µs).

The comparator output is latched at the negative edge of the clock when the channel is selected. The time (T_{MUX_SET}) given for the signal to settle down on the bus can be calculated as follows:

$T_{MUX_SET} = DT_P - T_{MUX}$

D is the duty cycle, and T_P is the time of PWM output. Note that if the source that is driving the channel has higher impedance, more time should be allowed for the signal to settle down. Two options are available to increase the allowed time for settling:

- Increase the PWM duty cycle D: Delays the negative edge at which the comparator output is sampled by the output latch. The scanning rate remains the same.
- Reduce the PWM frequency: Reduces the scanning rate.

Select one of these options, depending on the source impedance and scanning rate requirement.

T_{DAC} is the delay caused in the VDAC update by the DMA. In this application, in which one DMA channel is used to transfer 8-bit data from memory to VDAC, it takes 7 bus cycles to update the DAC.

Therefore, T_{DAC} = (7 / bus clock frequency). In the project, bus clock frequency is set to 24 MHz; T_{DAC} is approximately 0.3 µs. For more information on DMA data transfer timing, see the PSoC 3 and PSoC 5LP technical reference manuals.

If the PWM output frequency that sets the scanning rate is less than 100 kHz, then the T_{DAC} value will be small for most use cases compared to the period of T_P and thus can be neglected.

The time difference between the VDAC update and the comparator output latch is:

$T_{DAC_SET} = DT_P – T_{DAC}$

VDAC should settle to the set value within T_{DAC_SET} time. The maximum update of VDAC for the 1-V range is 1 MSPS. Therefore, T_{DAC_SET} should always be greater than 1 µs. There are two options to ensure it:

- Increase the duty cycle (D): This will push the negative edge at which the comparator output is sampled, thus widening time gap.
- Decrease the PWM output frequency (T_P will increase).

In this application, T_{MUX} = 0.66 µs, T_{DAC} ~ 0.3 µs, D = 0.5. The VDAC range is set to 1 V. With this data, the maximum PWM frequency that allows the maximum scanning rate can be calculated as:

$T_{DAC_SET} = DT_P - T_{DAC}$

PROFIT

T_{DAC_SET} will be minimum when the PWM frequency is maximum, and it is equal to the maximum VDAC settling time: 1 µs in this case.

So, 1 µs = 0.5 T_P – 0.3 µs; therefore, T_P = 2.6 µs.

The maximum PWM frequency is therefore 1/T_P ~ 384 kHz.

Table 1 summarizes the numbers for different VDAC settings (duty set to 50 percent, T_{DAC} ~ 0.3 µs).

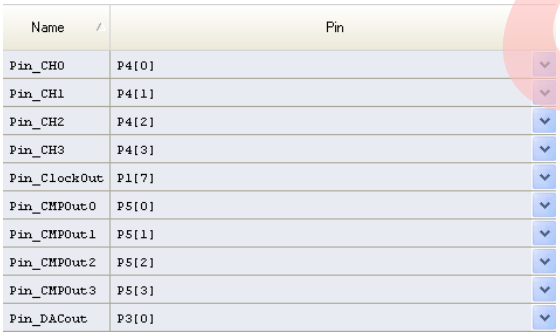
Table 1. Maximum Scan Rate

VDAC range	Maximum update rate	Maximum PWM frequency	Maximum scanning rate / channel (number of channels = 4)
1 V	1 MSPS	384 kHz	96 kHz
4 V	250 KSPS	116 kHz	29 kHz

Pin Selection

Figure 13 shows the pins selected in the application.

Figure 13. Pin Selection



Testing

Test the project by applying a triangular waveform signal at the channel and observing the output.

Oscilloscope CH1- Threshold Levels	Oscilloscope CH3- Output
Oscilloscope CH2- Input Signal	Oscilloscope CH4- PWM output

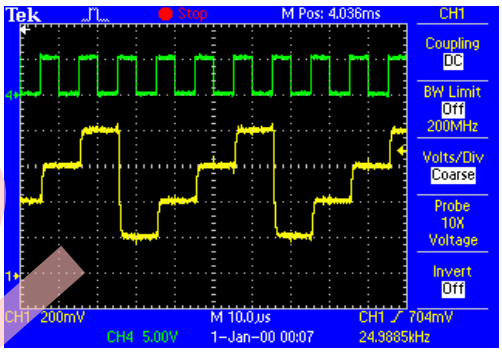
Table 2 shows the VDAC threshold levels set in the project.

Table 2. VDAC Threshold Levels

Channel	VDAC counts	VDAC voltage
CH0	200	800 mV
CH1	50	200 mV
CH2	100	400 mV
CH3	150	600 mV

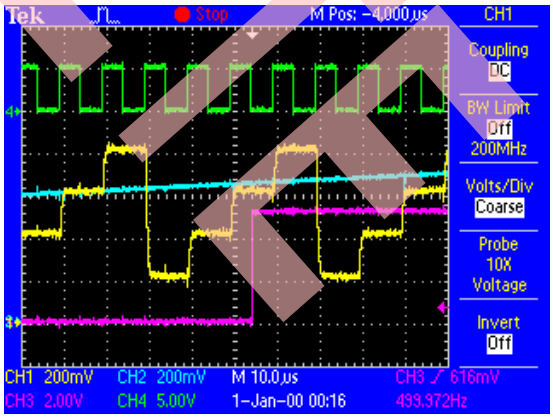
The DAC threshold stays at one value for one PWM output cycle. It changes at the positive edge of the clock, as shown in Figure 14.

Figure 14. DAC Threshold and Scan Clock



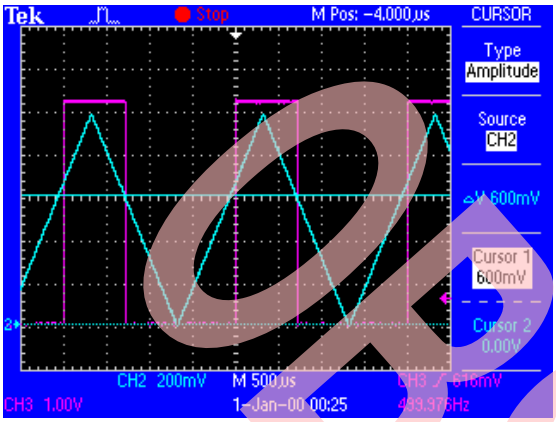
Provide a ramp signal of 1 V p-p and 500 Hz at the CH3 input P4[3], and observe the output at P5[3]. As shown in Figure 15 and Figure 16, the output goes high at the negative edge of the PWM when the input signal exceeds the CH0 threshold of 600 mV. You can repeat this step for other channels as well.

Figure 15. Comparator Output Switching



OBSOLETE

Figure 16. Comparator Output Switching for Triangular Input



There is a slight offset in time when the output switches owing to the finite scanning speed and comparator hysteresis.

Available Options

- **Checking the status of the comparator using the CPU:** If you need to check the comparator status for each channel using the CPU, then a Status Register Component can be connected at the output of the latch. The CPU can then read the status register to check the status of each channel.
- **Providing an interrupt when the signal crosses the threshold level:** The Interrupt Component can be connected at the output of the latch.
- **Increasing or decreasing the number of channels:** The counter used in the control unit should be such that:
  - Modulus of the counter = number of channels.
  - The LUT is designed accordingly.
  - The number of D flip-flops in the output latch and analog hardware multiplexer dimension changes.
- **Changing the scanning rate:** Changing the PWM output frequency, which drives the counter, changes the scanning rate.

Summary

This application note demonstrated multiplexed comparator implementation. The design is structured in such a way that you can easily modify it for your application, depending on how many channels are needed, the scanning speed, and other variables.

Related Application Notes

AN54181 – Getting Started with PSoC 3

AN77759 – Getting Started with PSoC 5LP

AN52705 – PSoC and PSoC 5LP – Getting Started with DMA

AN81623 – PSoC 3, PSoC 4, and PSoC 5LP Digital Design Best Practices

About the Author

Name:	Rajiv Vasanth Badiger
Title:	Applications Engineer Staff
Background:	B.E. Electronics and Communication

OBSOLETE

Document History

Document Title: PSoC® 3 / PSoC 5LP Multiplexed Comparator – AN60220

Document Number: 001-60220

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2893274	RJVB	03/16/2010	New application note
*A	3194626	RJVB	03/12/2011	Updated the design with analog hardware multiplexer. Updated timing diagram and maximum scanning rate calculations.
*B	3442273	RJVB	11/30/2011	Updated template and project files. Component customizer screenshots updated.
*C	3556272	NIDH	03/20/2012	Added contact email ID.
*D	3809219	NIDH	11/12/2012	Updated for PSoC 5LP.
*E	4733606	RJVB	04/21/2015	Obsolete application note.

OBSOLETE

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
Optical Navigation Sensors	cypress.com/go/ons
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 5

Cypress Developer Community

Community | Forums | Blogs | Video | Training

OBSOLETE

PSoC is a registered trademark of Cypress Semiconductor Corp. PSoC Creator is a 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, 2010-2015. 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.