DelSig_16Channel Example Project
1.00

Features

- 12-bit Delta-Sigma ADC with 16 multiplexed single-ended inputs.
- The Analog Mux and ADC are controlled in hardware and ADC results stored in memory using DMA – No firmware intervention.
- Switch can be used to cycle through readings displayed on LCD.

General Description

This example project is also a PSoC Creator starter design for PSoC 3/5LP devices. It uses the Delta Sigma ADC, DMA, and other hardware to sequence, sample, and store 16-channels of input data in memory without CPU intervention.

Development Kit Configuration

The following configuration instructions provide a guideline to test this design. For simplicity, the instructions describe the stepwise process to follow when testing this design with the PSoC Development Kit (CY8CKIT-001) board, but can be generalized for the PSoC 3 Development Kit (CY8CKIT-030) as well.

1. Set LCD power jumper J12 to ON position, assign SW3 to the 5V position, and leave the rest of the board at default configuration.
2. Observe the mux channel select signals P3[3:0], the ADC SOC P3[6] and the ADC EOC P3[7] on an oscilloscope.
3. Connect P6[1] to SW2 on the board, and ensure that a character LCD is connected to P2[6:0].
4. If testing the project with Analog inputs, connect the first 8 to P4[7:0] and the other 8 to P0[7:0]. Ensure that these voltages are between 0 and 2.048V – the input range set for the ADC.
5. Build the DelSig_16Channel project and then program the hex file to the PSoC device using the MiniProg3. After programming is complete, disconnect the MiniProg3.
6. Reset the PSoC device.

Project Configuration

The top design schematic is shown in Figure 1. The 16 analog inputs are fed into the Analog hardware mux whose output is connected to the single-ended 12-bit ADC. The ADC is configured (Figure 2) in Single Sample mode – the Start-of-Conversion (SOC) input to the ADC needs to be triggered for each sample. This SOC signal is generated by delaying the ADC End-of-Conversion (EOC) output by 1.5 µs using the PWM. This delay allows the mux inputs to settle when switching between inputs. The EOC is thus used as a trigger for the PWM in One-shot-multi-trigger mode (Figure 4). It also serves as the DRQ signal for the DMA and clock for the LUT. So after each ADC conversion, the DMA transfers the ADC output to memory, and the LUT updates its output. The Start_Reg control register gates the ADC SOC, and is set to 1 at the start to allow for ADC conversion.

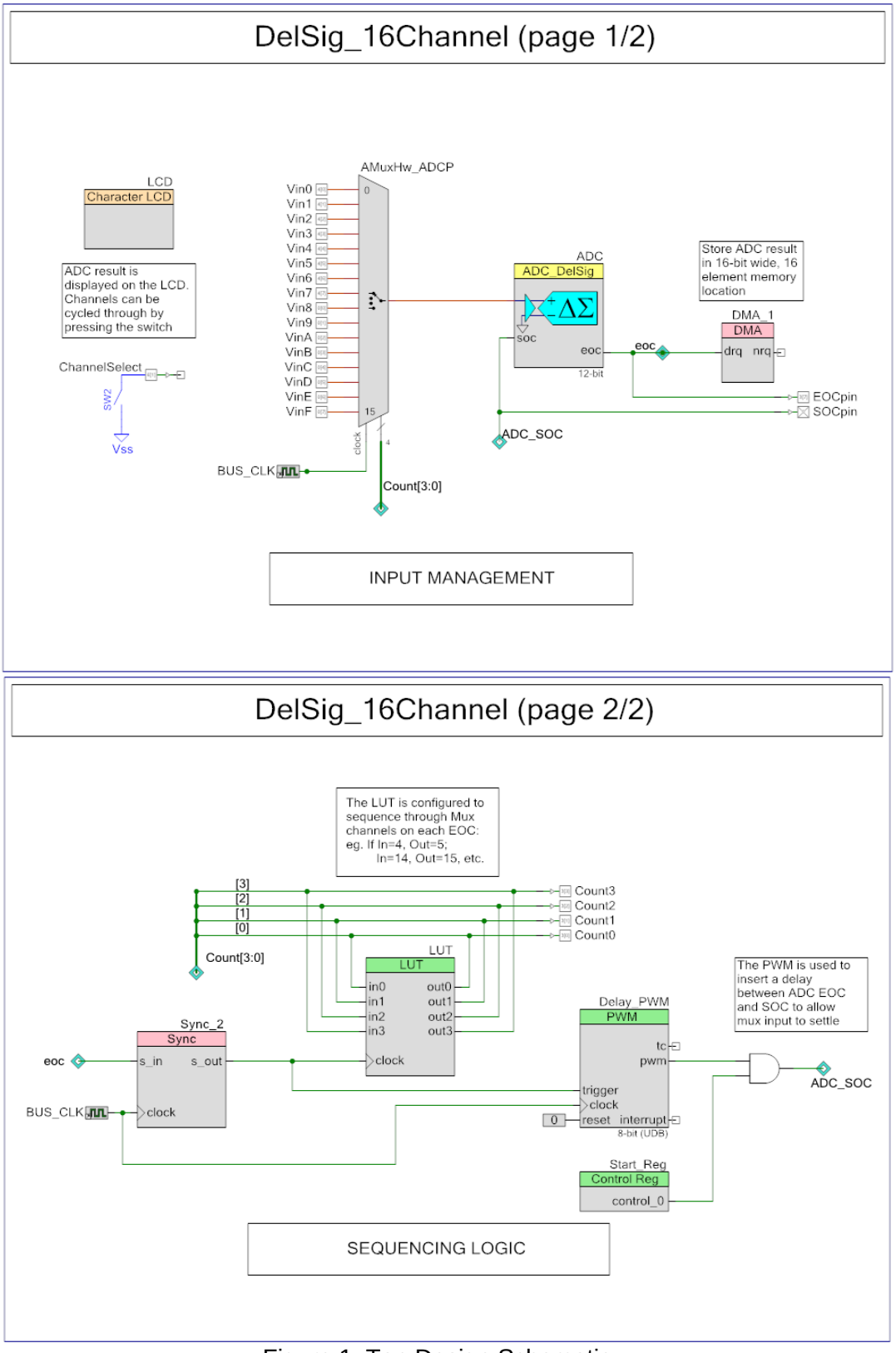

Figure 1. Top Design Schematic

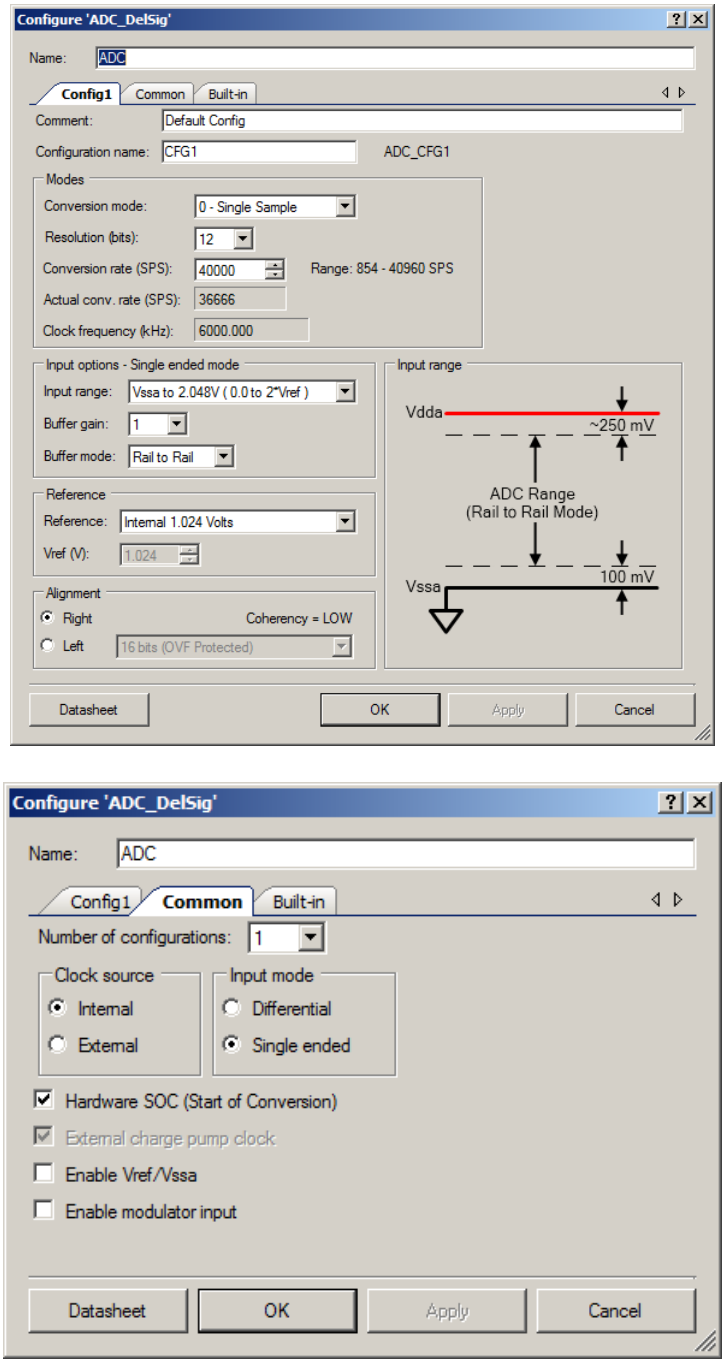

Figure 2. ADC_DelSig Component Configuration

As shown in Figure 2, the input range of the ADC is 0-2.048V, and the input rail-to-rail buffer is enabled. The conversion rate is set to 41 ksps. This is for all 16 channels, so factoring in the delay between EOC and SOC, the expected per-channel conversion rate is roughly 2.4 ksps. However, the actual sampling rate is marginally lower, due to the overhead when starting the ADC and the finite clock divider used to get the conversion rate desired.

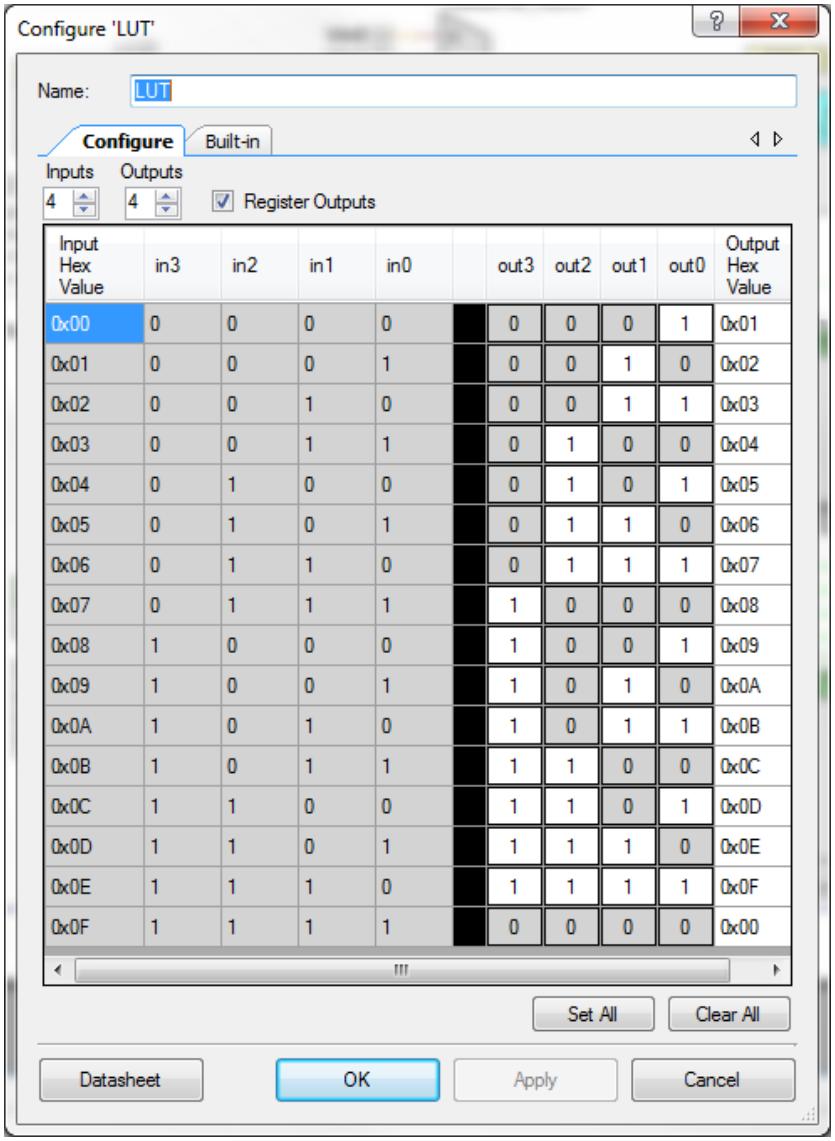

Figure 3. Look-Up Table configuration

The LUT is set up to cycle sequentially through the mux channels 0-15 on each ADC EOC pulse (Figure 3).

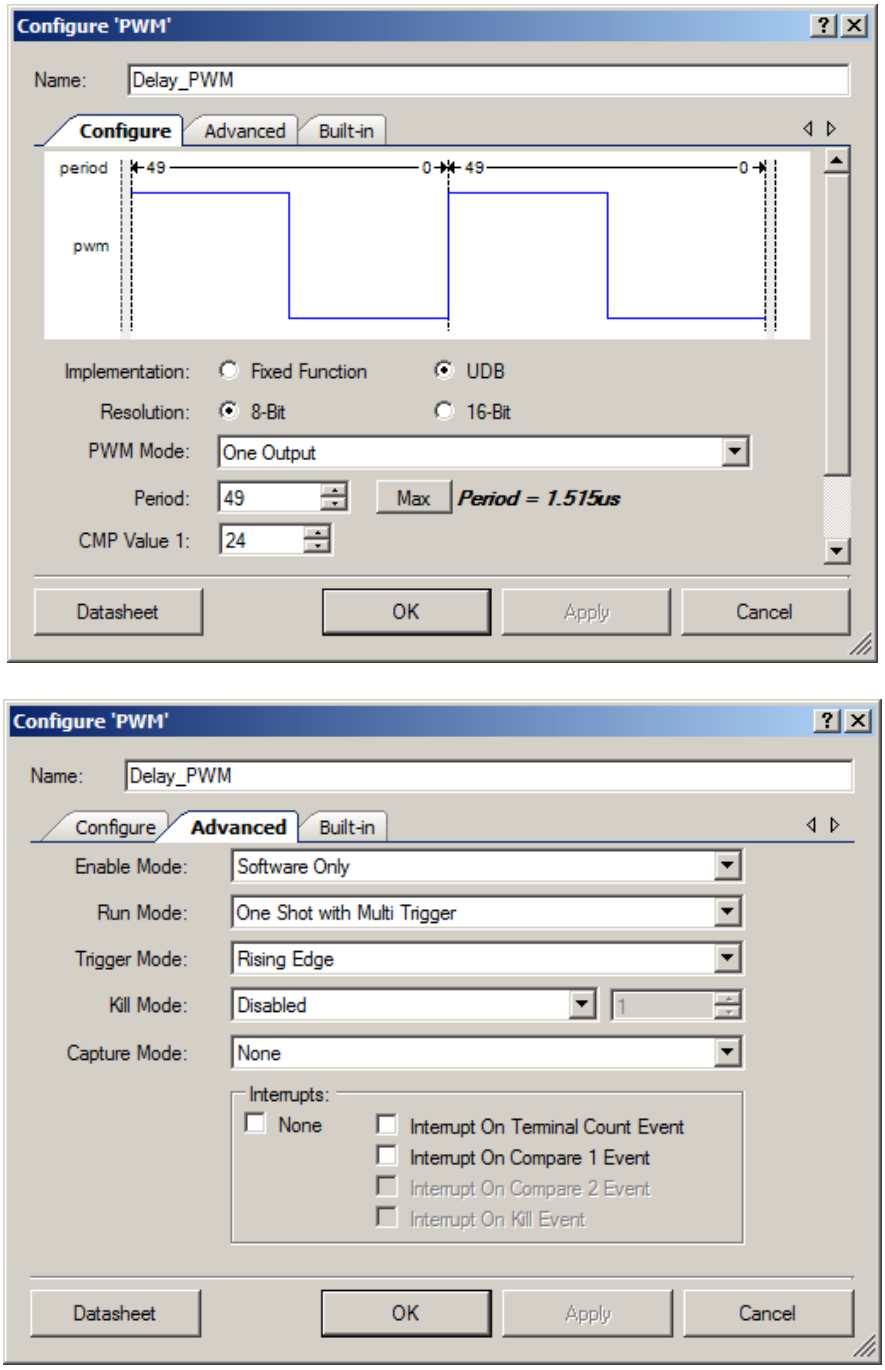

Figure 4. PWM component configuration

As seen in Figure 4, the PWM is set up to have a one-shot multi-trigger run mode. The duty cycle is kept as 50%, but its value actually does not matter. What matters is the compare type which is set to ‘greater than’. Since the PWM counts down from the ‘Period’ value, this ensures that the PWM output generates a rising edge each time the period is reloaded – that is, for every terminal count event. Since the Start_Reg is initially set to 1, this rising edge causes the ADC to start conversion.

Project Description

In the main function all components are started. Also, the DMA is configured to transfer the ADC output to memory.

As explained in the previous section, the ADC conversion is controlled in hardware. In the ‘forever’ loop in main, the ADC output for a single channel is displayed and continually updated on the Character LCD. The various channels of the mux can be cycled through by pressing the SW2.

Expected Results

- The ADC output results as displayed on the LCD should match the analog inputs given (if any).
- The count value P3[3:0] should continuously cycle between 0-15 at a frequency of approximately 17.1 kHz.
- The ADC SOC and EOC signals should be negative and positive pulses respectively, of frequency 34.4 kHz.



Figure 5. Expected output on the LCD

Related Material

Example Projects

- DelSig_I2CM
- DelSig_I2CS
- DelSig_SPIM
- SAR_SPIM_USB
- ADC_DMA_VDAC
- Filter_ADC_VDAC

Application Notes

- AN52705 - PSoC® 3 and PSoC 5 - Getting Started with DMA
- AN61102 - PSoC® 3 and PSoC 5 - ADC Data Buffering Using DMA
- AN57821- PSoC® 3 and PSoC 5 Mixed Signal Circuit Board Layout Considerations

Training

- PSoC 3 Video - Create an ADC LCD Project
- PSoC 3 and PSoC 5 104: Introduction to Analog Peripherals

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, 2012. 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. PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

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.