Application Note AN2350

Sample and Hold in Variations

Author: Erich Birnbaum
Associated Project: Yes
Associated Part Family: All (Tested with CY8C29466)
Software Version: PSoC Designer 4.2
Associated Application Notes: No

Abstract
The PSoC® device has no ready-to-use user module for a sample-and-hold circuit. Analog switched-capacitor user modules are based on switched capacitor blocks; this Application Note describes how to use one of these blocks as a sample and hold.

Introduction
In signal-processing systems you need several functions, which include amplifiers, filters, ADCs, DACs… In the PSoC, these functions are realized through user modules. In some cases you need a constant analog signal (for example, with a following SAR-ADC), or you need the analog value sampled at a pre-defined time. For this, you would normally use a sample-and-hold circuit. Even though the PSoC is a sampled-data system, there is no ready-to-use UM. The analog switched-capacitor user modules are based on switched capacitor blocks (as is a sample and hold) and this Application Note describes how to use one of these blocks as a sample and hold.

In a PSoC device, the switched-capacitor user modules are driven from an analog clock, which separates the processing into phases:

1. **Phase 1**: Capacitors are (dis-)charged, Ua=0 (AGND).
2. **Phase 2a**: Output capacitor is charged.
3. **Phase 2b**: Ua from the switched-capacitor user module is transferred to the capacitor in the analog bus.

How to Use the Examples
The following examples are only intended to show the waveform of the signals. You must connect an analog signal (approximately 50 Hz sine, this produces about 5 steps/period) to the input and control the analog output with an oscilloscope. The input frequency must be tuned so that the scope shows a stable wave. The given port and pin numbers are for the CY8C29466 device.

Example 1 Using the Hold Phase in the Analog Bus (Phase 2b)
If the analog clock is switched off, the analog voltage is sampled in the analog bus. In this example, a Counter8 and a switched-capacitor user module form a sample and hold. For test purposes, a Counter16 delivers a control signal.

- Analog In = P2[1] (Pin 8)
- Analog Out = P0[3] (Pin 3)

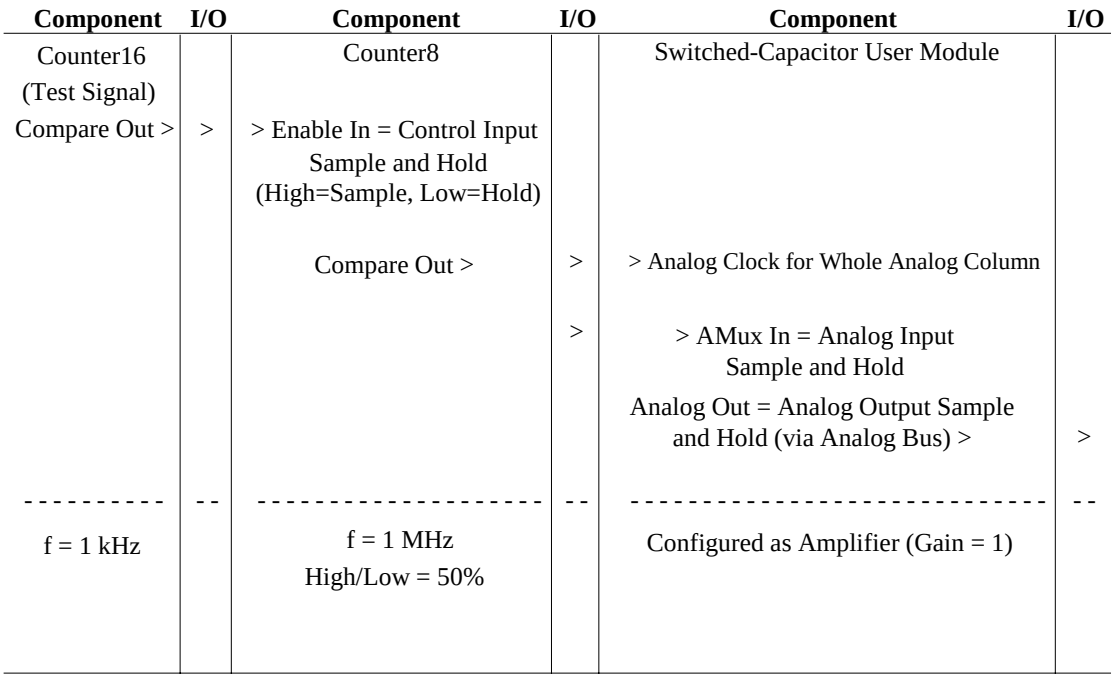

Figure 1.

Example 2 Using the Hold Phase in the Switched-Capacitor User Module (Phase 2a)
In example 1, you use the hold phase of the analog bus, so you are wasting one of the maximum four analog outputs. You can use the hold phase of a switched-capacitor user module for internal signals (without an analog bus), if you stop the analog clock just at the end of Phase 2a. This can be done with a personalized clock that stops once after 3 pulses and then again after multiples of 4 pulses. In this example, a Counter16 generates an interrupt as a test-signal source (1 kHz). The interrupt routine realizes the signal described above.

Phases 2a and 2b show the importance of the timing (2a as described above, with 2b intentionally wrong to show the difference: the analog output is 0V/AGND in the "hold phase").

The DigBuf UM delivers the analog clock. The interrupt routine (from Counter16) creates the analog clock by "bit-banging" the input register.

The PGA (continuous time UM, therefore, no hold phase on the corresponding analog bus) transfers the internal analog output signal (of the switched-capacitor user module) to a pin.

- 2a: In = P2[1] (Pin 8)
  PGA_Out = P0[5] (Pin 2)
- 2b: In = P2[2] (Pin 21)
  PGA_Out = P0[4] (Pin 26)

Example 3 Elegant Variation of Example 2
The connection of a Counter16 UM and a Counter8 UM creates the needed analog clock in a more “elegant” way than described in example 2. Counter16 (1 kHz) delivers a compare high (4 µs) to the enable-input of the Counter8. The period of the Counter8 is first initialized at 2 µs. Soon thereafter, the Counter8 period decreases to 1 µs. The new value is validated upon the first reload event (the first period is 2 µs, the following periods are 1 µs). The terminal count interrupt (Counter16) is the start of the sample-and-hold hold phase. In this example, the sampled analog value gets digitized in a SAR6 ADC and is immediately returned to its analog value with a DAC6.

- Analog In = P2[1] (Pin 8)
- Analog Out = P0[3] (Pin 3)
- DAC Out = P0[5] (Pin 2)

The following is an excerpt from *main.asm*:

```
        mov a,DAC6_1_HIGHPOWER
        call DAC6_1_Start
        mov a,SAR6_1_HIGHPOWER
        call SAR6_1_Start
        mov a,SCBLOCK_1_HIGHPOWER
        call SCBLOCK_1_Start
        Counter16_1_Start_M
        ;period=24000-1=1ms, compare=96=4µs
        Counter8_1_Start_M
        ;period-reg initialized to 48-1 ->
T1=2us
        mov reg[Counter8_1_PERIOD_REG],24-1
        ;Tn=1µs
        Counter16_1_EnableInt_M
        M8C_EnableGInt
.terminate:
    jmp .terminate

;;*****************************************
```

Code 1. main.asm

The following is an excerpt from *counter16_1int.asm* (there is no need for `push/pop A`):

```
        call SAR6_1_cGetSample
        call DAC6_1_WriteStall
        reti
```

Code 2. counter16_1int.asm

Conclusion

You must consider that the minimum analog clock frequency is about 1 kHz (found in the MDAC6/8 User Module Data Sheet), so the maximum hold phase is approximately 4 ms.

The following configuration seems interesting. If you need a fast SAR6 ADC behind a filter, you can route the analog clock on the sample-and-hold block to the filter blocks. Upon ADC startup, if you disable the analog clock to the block, the analog output signal will stay constant, which enables you to use the SAR ADC with higher frequencies.

About the Author

Name: Erich Birnbaum
Title: Research Engineer
Background: Erich earned his electro-mechanical degree in 1983 from the Technical University of Darmstadt (Germany).
Contact: ebirn@gmx.de

Cypress Semiconductor
2700 162nd Street SW, Building D
Lynnwood, WA 98087
Phone: 800.669.0557
Fax: 425.787.4641

http://www.cypress.com/