Application Note AN2109

The Direct Digital Synthesis Generator

By: Victor Kremin
Associated Project: Yes
Associated Part Family: CY8C25xxx, CY8C26xxx

Summary
The low-frequency programmable signal generator is proposed. The generator employs the direct digital synthesis (DDS) technique to create an output signal with high-frequency resolution. The PSoC device produces the sinusoidal signal, but other output waveforms can easily be obtained. This design can be embedded in various test and measurement equipment, or be part of other high-frequency generators. The design variances and possible modifications are considered as well.

Introduction
There are a number of applications that need precision signal sources. Examples include impedance meters, network analyzers, various sources for test equipment, and even low-cost hobbyist functional generators. The development of semiconductor technology opened new possibilities to using digital techniques for solving this task. The various methods can be employed to create a digitally demanded waveform.

One of the most common approaches incorporates the fixed-frequency reference generator (**GEN**), programmable divider (**1/N**) to clock the lookup (**ROM**) or RAM address counter (**AC**). The waveform memory data output is connected with a digital-to-analog converter (**DAC**) for signal forming. The low-pass filter (**LPF**) can follow the **DAC** to filter the output signal.

Figure 1 illustrates this description. The main drawbacks to this approach are the variable frequency step (which is in inverse proportion to divider coefficient) and the **DAC** frequency sampling inconsistently that requires the difficult-to-implement anti-aliasing **LPF** with variable cut-off frequency.

To get the fixed-frequency step, the phase locked loop (**PLL**) system can be used as a lookup table memory-address counter clock source. Figure 2 illustrates the simplest, single loop **PLL** system.

The voltage-controlled generator (**VCO**) is controlled by a phase detector (**PD**) loop filter (**LF**) output signal. In this system the frequency step is equal to the reference signal frequency. But the frequency switching time in a **PLL**-based system can be too long and **DAC** sampling frequencies vary.

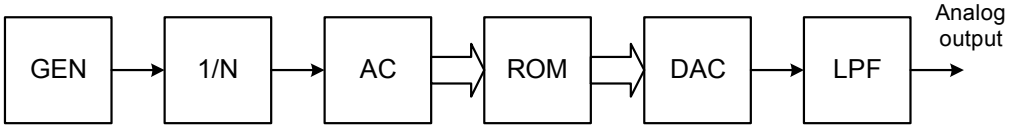

Figure 1: The Simplest Way to Build a Digitally-Controlled Signal Generator

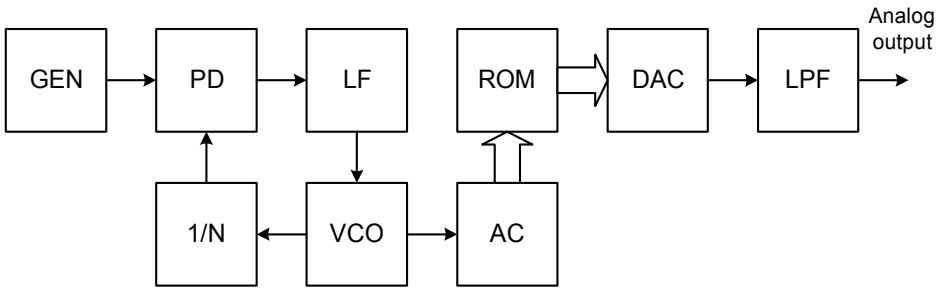

Figure 2: The PLL-Based Digitally-Controlled Signal Generator

The direct digital-frequency synthesis (**DDS**) approach is free from these drawbacks and opens new possibilities in the harmonic and arbitrary design of waveform signal generators. Suppose the sinusoidal signal must be generated:

$$y(t) = A\sin(\varphi_0 + \omega t) \quad (1)$$

Where A is signal amplitude, φ_0 is the initial phase and ω is angular frequency. In this equation, the sin() serves as the function which maps the monotonically growing phase $\varphi_0 + \omega t$ value to the output signal. The sin() function period is equal 2π, so the generated signal will be periodic with period T:

$$T = \frac{2\pi}{\omega} \quad (2)$$

Alternatively, because the sin() is the periodic function, instead of forming a constantly growing functional argument, the phase generator, which is forming the periodic ramp waveform with period T and range $[0..2\pi]$, can be used to produce periodic harmonic signals. This principle can be transformed in the discrete domain. Suppose a phase accumulator exists which performs the following function:

$$P_i = P_{i-1} + W \quad (3)$$

Where P_i is phase output at sample i and W is the phase increment. This device is a digital integrator and produces a linear ramp for slope W. There is also a limitation on the maximum phase value.

When the digital integrator output reaches the maximum allowable value, P_{max}, the integrator resets and starts to integrate from zero. One of the simplest ways is to use an N-bit binary counter-type adder as the phase accumulator.

This accumulator can accumulate the 2^N different phase values; the maximum accumulator value is $2^N - 1$ and minimum is zero. These 2^N accumulator states correspond to the 2π phase span of the sin() argument generator. So, the generator output phase for sample i can be evaluated by the following equation (the accumulator starts counting from zero):

$$\varphi_i = \frac{2\pi}{2^N} W \cdot i \quad (4)$$

The output frequency can be estimated as a different phase expression, $\omega = \Delta\varphi / \Delta t$ or

$$F_{out} = \frac{\omega}{2\pi} = \frac{1}{2\pi}\frac{2\pi}{2^N} W \frac{\Delta i}{\Delta t} = F_{clk}\frac{W}{2^N} \quad (5)$$

Where F_{clk} is the accumulator update rate. Because the binary phase accumulator accepts only integer values W, the frequency step and the minimal possible generated frequency can be estimated by the following equation:

$$\Delta F = F_{\min} = \frac{F_{clk}}{2^N} \quad (6)$$

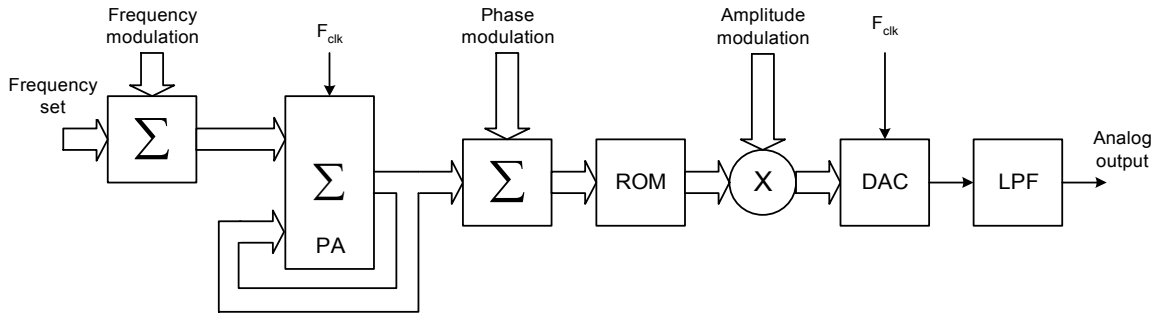

Figure 3: DDS Implementation

For example, if F_{clk} = 100 MHz and N=32, then the frequency step is equal to 0.023 Hz and is independent from the generated frequency. The direct digital synthesis technique can be easily implemented in hardware. More over, the various modulation types can be supported at the same time. Figure 3 depicts one possible implementation.

The DDS generator consists of the phase accumulator (**PA**), which continuously updates the phase; arbitrary waveform memory **ROM**, a signal **DAC** and anti-aliasing **LPF**. The two additional adders and multipliers are intended to provide frequency, phase and amplitude modulation possibilities at the same time. This approach is widely used in measurement devices (signal generators, spectrum analyzers) and communication equipment (cell phone-base stations, transceivers) to create radio-frequency signals with high-frequency resolution. But DDS solutions are complex and expensive.

The unique mixed-signal architecture of PSoC allows you to build a low frequency, high-frequency resolution sine wave or arbitrary waveform generator at minimal cost. The additional service functions, such as user interface and communications, can be implemented without additional cost. Table 1 depicts the technical characteristics of the main generator.

Table 1: Main Generator Characteristics

Item	Item Value
Frequency Range	0.001 Hz-20 kHz
Frequency Step	0.001Hz
Harmonics Level, RMS Value	0.03%-0.05%
Frequency Accuracy: Internal Clock External 32 kHz Resonator	 2.5% 1%
Outputs Types	Sine Wave, Arbitrary Periodic
Output Voltage Level, Amplitude Value	80 mV-2.1V
Communication Interface	RS232
PC Communication Speed	19200 Bauds

The Generator Hardware

Because most of the generator units are implemented using PSoC reconfigurable blocks, the generator schematic is very simple. Figure 4 illustrates both the schematic and PSoC configuration internals.

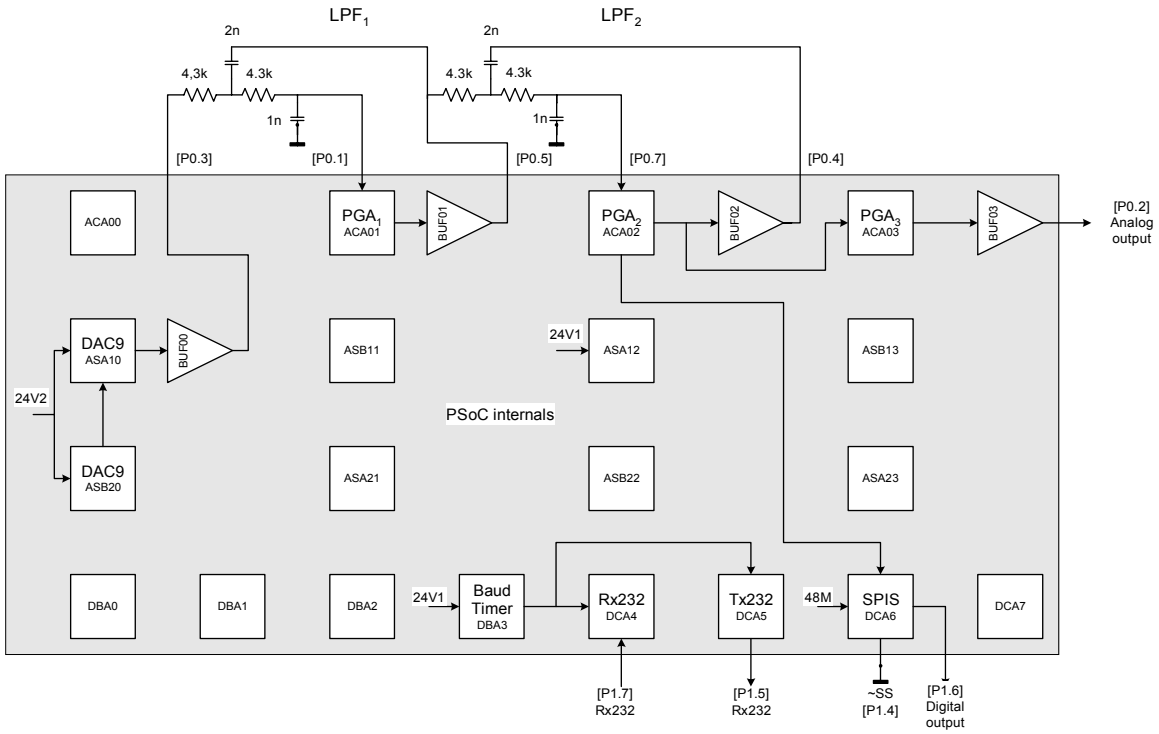

Figure 4: Generator Schematic and PSoC Internals

The 9-bit, digital-to-analog converter (DAC9) converts the waveform data into an analog signal. The DAC sampling frequency is set to 62.5 kHz, but the other sampling rates can be obtained by varying the 24V2 frequency.

The anti-aliasing, low-pass filter is formed using two continuous second-order filters built around programmable gain amplifiers, PGA_1 and PGA_2. The filters form the four-order LPF filter, based on Sallen-Key structures with cut-off frequency near 21 kHz. When compared with a switched capacitor-based filter, the continuous filter is characterized by better signal fidelity and absence of switching frequency products in the filter output spectrum. But this approach needs several external passive components.

The generator has two outputs; analog and digital. The PGA_3 has been used as a programmable gain amplifier to control the analog output signal level. Because the DAC resolution is limited to 9 bits, the best way to change the value of the generator output signal level lies in changing PGA_3 gain. To form the digital output, the PGA_2 comparator bus signal is used. The detection output signal zero-crossing is better than directly using the ROM digital stream.

This is because it allows the process to reach equal on-off time even for high-frequency signals when the ratio of sampling-to-output frequency is low.

There are two pitfalls in routing this signal to an external processor pin. First, there is an undocumented synchronization problem, which lies in the analog CT block comparator bus signal latching with the column frequency independently from the Analog CT Block Control Register values. Secondly, the current PSoC routing scheme does not allow users to directly route the comparator bus signal to Global Output Bus. To decrease the influence of the first problem, the column frequency is set to 24V1 or 4 MHz, which is higher than the maximum output signal frequency. The comparator bus signal can be routed to an external pin via the SPIS User Module, which contains the 8-bit shift register and produces an additional time delay. Because the register is clocked by a 48 MHz signal, the propagation delay influence is negligible in comparison with the period at the maximum output frequency.

The generator uses the UART for remote control. The UART is comprised of a serial transmitter and receiver, and the baud-rate timer. The default communication speed is set to 19200 baud.

The Generator Firmware

The generator firmware serves two functions; to form the DAC data stream and provide serial communications for remote control of the generated frequency and output level. The PSoC hardware register that updates synchronization was used to provide the required DAC sample rate. It was set to 62.5 kHz but easily can be adjusted by modifying the DAC9 sample frequency in PSoC Designer software.

The program was implemented by using the interrupt-main loop technique. The main program loop is responsible for waveform generation and setting the generated frequency and output signal level. The serial communication (both receiver and transmitter) is completely interrupt driven to eliminate UART polling.

The following code fragment illustrates the main program loop:

```
#define SB(var, num) *(((BYTE*)&var)+num)
//used for accessing singles byte in WORD or LONG variables, translates in one
//MOV command only
void main()
{
  DAC_Start(DAC_FULLPOWER);
  PGA_FIL1_Start(PGA_FIL1_HIGHPOWER);
  PGA_FIL2_Start(PGA_FIL2_HIGHPOWER);
  PGA_OUT_Start(PGA_OUT_HIGHPOWER);
  SPIS_Start(SPIS_LSB_FIRST);

  PGA_FIL2_GAIN_CR1 |= 0x40;
  PGA_FIL2_GAIN_CR2 &= ~0x40;
  //enable comparator bus and disable the output latch
  //but comparator bus is still column clock synchronized

  OpenExchange();
  M8C_EnableGInt;

  ExchangeSend("The DDS generator is ready.\n\r");

  while(1)
  {
    msb = strlen(rx_buffer)-2;
    switch (rx_buffer[msb])
    {
      case 'f':
      case 'F':
        rx_buffer[msb] = 0;
        dphase = FREQ_MULT*atof(rx_buffer);
        ExchangeSend("The frequency was set well.\n\r");
      break;

      case 'g':
      case 'G':
        rx_buffer[msb] = 0;
        lsb = atoi(rx_buffer);
        if (lsb < GAIN_LEVELS)
        {
          PGA_OUT_SetGain(gain_table[lsb]);
          ExchangeSend("The gain was set well.\n\r");
         }
      break;

      default:
      break;
    }

    ready = TRUE;

    while(ready)
    {
      accum += dphase;
      SB(adr,0) = SB(accum,0);
      SB(adr,1) = SB(accum,1);
```

```
        if (SB(accum,0) & QUATER_BIT) adr ^= 0xFFFF;
        adr &= TABLE_MASK;

        msb = lsb = sin_table[adr];

        msb >>= 3;
        msb |= 0x80;

        lsb &= 0x07;
        lsb <<= 2;
        lsb  |= 0x80;

        if (SB(accum,0) & SIGN_BIT)
          msb |= 0x20;
        else
          lsb |= 0x20;

      M8C_Stall;
      DAC_MSB_CR0 = msb;
      M8C_Unstall;
      DAC_LSB_CR0 = lsb;
      //instead, the fixed version of the DAC_WriteStall2B(lsb, msb) can be used
      //but you have less time reserve in this case
     }
}
}
```

The peripherals are initialized first. Later, the PGA_2 comparator output is connected to the comparator bus. When hardware is completely initialized the availability of serial receiver data is checked. The generator accepts the two commands so it can set the output frequency and gain. The set-frequency command format is a floating-point number followed by 'F' or 'f' and new line symbols. Valid examples are "1000.50f" or "500F." The set output-signal-level command is the PGA gain level number followed by 'g' or 'G' together with new line symbol, e.g., "20g." The valid numbers are 0..21; 0 corresponds to the minimal output signal level, 21 – maximum. When you send commands to the generator do not forget to add the new line symbol at the end.

When the command is interpreted, the program jumps to the signal generation loop. The serial receiver Interrupt Service Routine can break this loop when a new command is well received by clearing the *ready* variable. Note that the ready-variable checking was implemented without disabling interrupts because the single-byte comparison is an atomic operation.

The signal generation loop consists of the phase software accumulator implementation, lookup table-based sine wave calculation, DAC value calculation, and synchronous DAC updating. The 28-bit phase accumulator is used. The sine wave is characterized by the symmetry properties, which allow for storing only one-quarter waveform in the lookup table to save Flash memory. The most significant bit (27^{th}) determines the waveform signal sign. The next bit (26^{th}) determines the sine wave quarter. The value of this bit is used to calculate lookup table addresses. The table address is inverted when this bit is equal to '1'.

The memory address-bus width was selected according to the recommendations [1]. If the signal generation for the DAC has M-bits resolution, the memory address-bus width must be larger than the DAC resolution by 2 bits. Increasing the lookup memory space will not significantly improve the quality of the sine wave signal. This design incorporates the 2^{10} or 1024-point entry lookup table, because the ninth bit of the DAC is a sign bit, which is calculated directly from the phase accumulator without using the lookup table. This table consumes 1 kilobyte of Flash memory and is presented in the *table.h* header file. Figure 5 depicts the phase accumulator bit assignment.

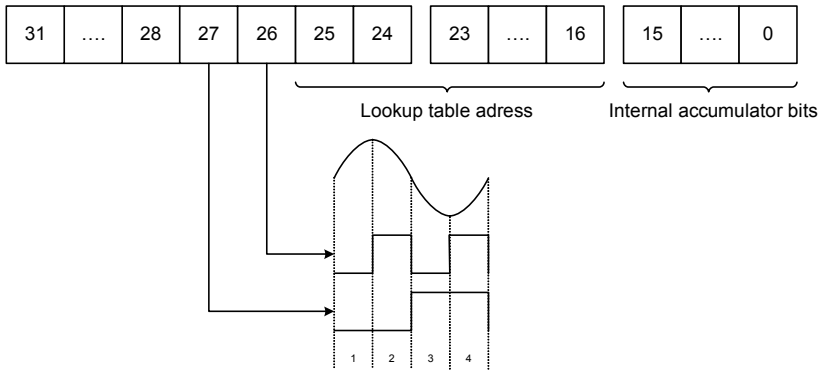

Figure 5: Phase Accumulator Bit Assignment

When the waveform value is calculated, it must be immediately sent to the DAC. In the latest version of PSoC Designer, 3.20, the DAC9 User Module library suggests several functions for synchronous updating. The `DAC_WriteStall(INT/WORD value)` is slow because it needs two-byte parameter decomposition and switched-capacitor block register-value calculations. To reduce loop-execution time, the DAC register values are calculated directly in the main loop and updated at the same time using `M8C_Stall` and `M8C_Unstall` macros.

Design Modifications

This design can be updated to generate any waveform signal by replacing lookup-table content. But, because the waveform can differ from sine wave signal to signal, the table memory must be increased according to the desired waveform. The 4 Kbytes of memory provide enough resolution for most applications. The capability of PSoC Flash memory programming allows users to upload and store arbitrary waveforms remotely via a suitable communication interface.

Secondly, this design can be updated easily to provide quadrature or any phase-shifted output signals. Some applications, such as impedance meters, RLC, and power meters need quadrature signals to measure real and imaginary parts of the complex value. To implement quadrature generation, an additional DAC must be placed and data values for this DAC will be calculated similarly.

Thirdly, if you need higher accuracy for the output signal frequency, you must provide a high-stability external 32 kHz clock. The PSoC internal crystal oscillator guarantees 1% accuracy. When external frequency reference is used, please note that reference signal phase noise is multiplied by the internal PLL and can additionally degrade generator output spectrum.

References/Resources

1. Bar-Giora Goldberg, Digital Frequency Synthesis Demystified, LLH Technology Publishing, 1999, 355 p.

The following book-collection servers can be used to download scanned or electronic books and proved an invaluable resource for this author:

- http://www.elektroda.pl/eboard/index.php, ebooks section
- http://www.mcu.cz/atm/index.php

About the Author

Name: Victor Kremin
Title: Associate Professor
Background: Viktor had earned radiophysics diploma in 1996 from Ivan Franko National Lviv University, PhD degree in Computer Aided Design systems in 2000, and is presently working as Associate Professor at National University "Lvivska Politekhnika" (Ukraine).

His interests involve the full cycle of embedded systems design including various processors, operation systems and target applications.

Contact: You may reach him at vkremin@polynet.lviv.ua

Cypress MicroSystems, Inc.
22027 17th Avenue S.E. Suite 201
Bothell, WA 98021
Phone: 877.751.6100
Fax: 425.939.0999
http://www.cypressmicro.com/ / http://www.cypress.com/aboutus/sales_locations.cfm / support@cypressmicro.com