



Direct Digital Synthesis 101

Need a refresher on direct digital synthesis? Robert brings you up to speed by covering DDS theory, a few chip-based solutions, and some firmware implementations.

Welcome to the Darker Side. If you are a regular reader, you probably remember my December 2007 article about using a phase-locked loop (PLL) to generate precise and stable frequencies ("Are You Locked?: A PLL Primer," *Circuit Cellar* 209). I also briefly introduced another interesting concept: the direct digital synthesizer, or DDS for short. At the time, I promised to dig into DDS techniques in a future column. This month, I'll make good on my promise.

You may also remember that I already presented a DDS project back in 2001, but this time I will go further than just describing a project ("DDS-GEN," *Circuit Cellar* 129 and 130). My aim is to help you understand how DDS techniques can help you in future projects. To do so, I will describe the pros and cons of using them. So, come with me on a journey to DDS world.

DDS BASICS

The simplest form of a digital waveform synthesizer is a table look-up generator (see Figure 1). Just program a period of the desired waveform in a digital memory (Why not an EPROM for old timers?), connect a binary counter to the address lines of the memory, connect a DAC to the memory data lines, keep the memory in Read mode, clock the counter with a fixed-frequency oscillator F_{CLOCK} , and voilà, you've got a waveform on the

DAC output. Don't forget to add a low-pass filter to clean the output signal, with, as you know, a cut-off frequency a little less than $F_{\text{CLOCK}}/2$ to please Mr. Nyquist. This design works, but it is not too flexible. If you want to change the output frequency, you need to change the clock frequency, which is not easy to do, especially if you need a fine resolution.

The DDS architecture is an improvement on this original design (see Figure 2). Rather than add one to the table look-up address counter at each clock pulse like the counter did in the previous example, a DDS uses an N-bit long-phase register and adds a fixed-phase increment (W) at each clock pulse to this register. N can be quite high (e.g., 32 or 48 bits), so only the most significant bits of the phase

register are used to select a value from the phase-to-amplitude look-up table, which is usually nothing more than a ROM preprogrammed with a sine waveform. Assume that you are using the P most significant bits as an address. Then the output of the look-up table is routed to a DAC. And, of course, the analog signal finally goes through a low-pass filter, which is called a "reconstruction filter." You will understand why in a minute.

How does it work? If the phase increment W is set to one, you will need 2^N clock pulses to go through all of the values of the look-up table. One sine period will be generated on the F_{OUT} output each 2^N clock pulses, exactly like the aforementioned counter-based architecture. If W is 2, it will be twice as fast and the output

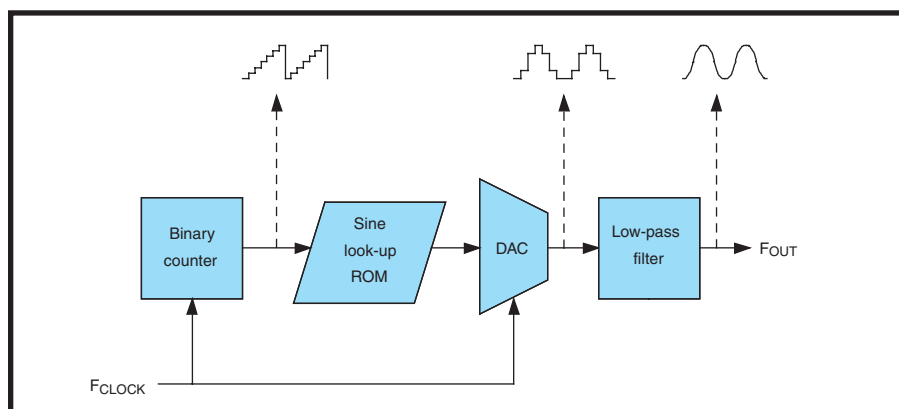


Figure 1—The most basic digital signal generator is built with a simple binary counter. Its output sequentially addresses the rows of a memory, which holds the successive points of the output signal. It is then converted to an analog signal and filtered.

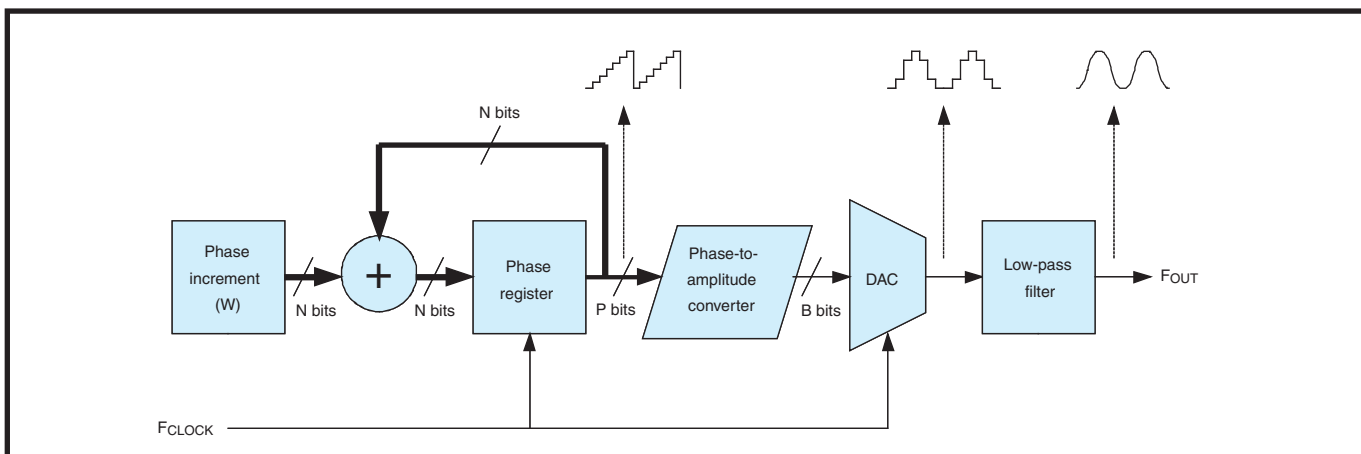


Figure 2—The basic architecture of a DDS is a variant of the counter-based digital generator, but it allows a fine frequency resolution thanks to a phase register and a binary adder. The key point is that the increment is not necessarily a divider of the phase register maximum value.

frequency will be doubled. As you know, you need a little more than two samples per period to be able to reconstruct a sine signal, so the maximum value of W is $2^{N-1} - 1$. The formula giving the output frequency based on the phase increment is then:

$$F_{OUT} = W \times \frac{F_{CLOCK}}{2^N}$$

Don't be confused. It is not a simple

programmable divider because the phase register doesn't loop back to the same value after each generated period. The table in Figure 3 may help you understand it. What make a DDS a fantastic building block are the numeric examples. Just take a standard, low-performance DDS with a phase register of $N = 32$ bits and a reference clock $F_{CLOCK} = 20$ MHz. Your DDS can then generate any frequency from DC to nearly 10 MHz

with a resolution of the following:

$$1 \times \frac{20 \text{ MHz}}{2^{32}} = 0.0046 \text{ Hz}$$

Not bad. In fact, the maximum frequency will be a little lower due to constraints on the low-pass filter, as you will see later.

DDS FLEXIBILITY

Another great advantage of a DDS generator is that you can use it for any

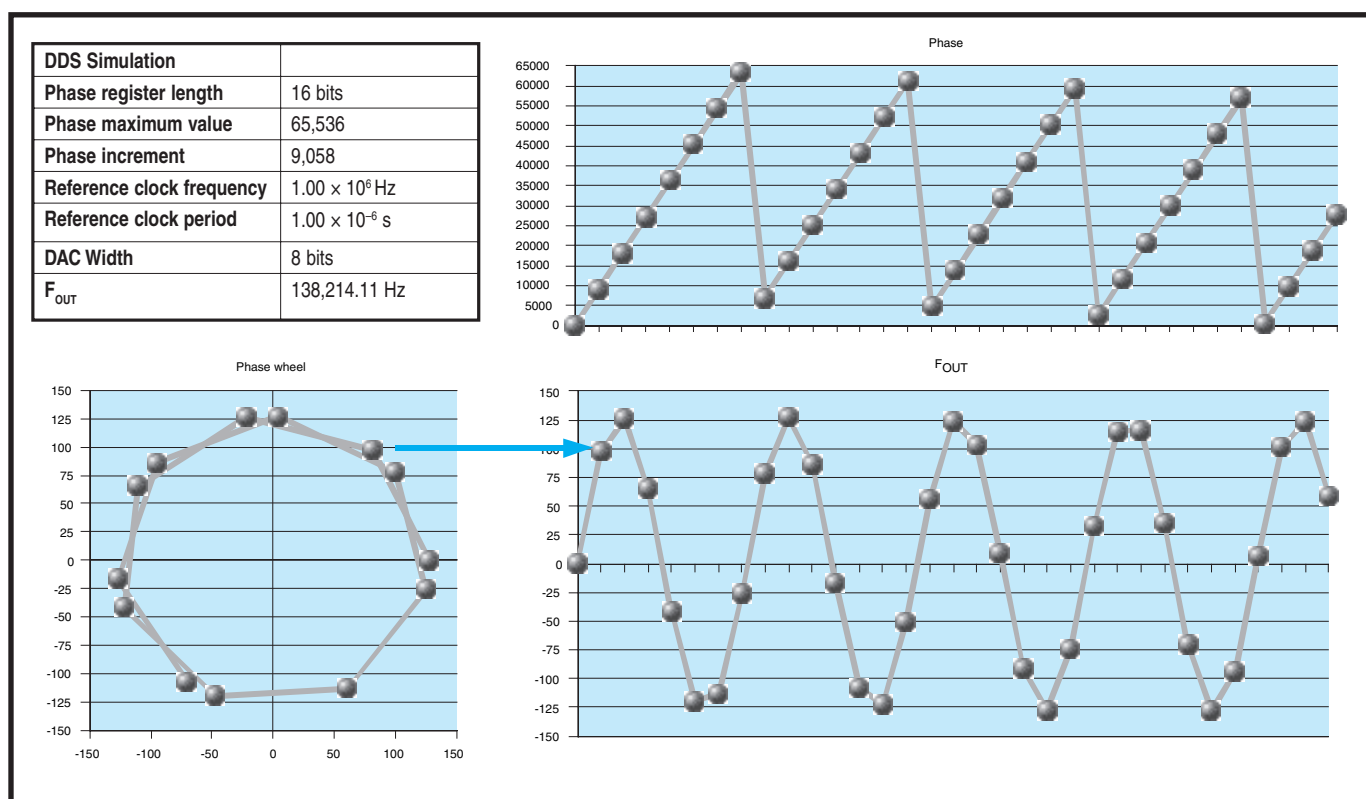


Figure 3—This spreadsheet simulation shows the “phase wheel” concept. A fixed angle is added to the phase register at each clock pulse. Note that each period of the output signal is not identical to the previous ones because the phase doesn't go back to the same value after a full turn.

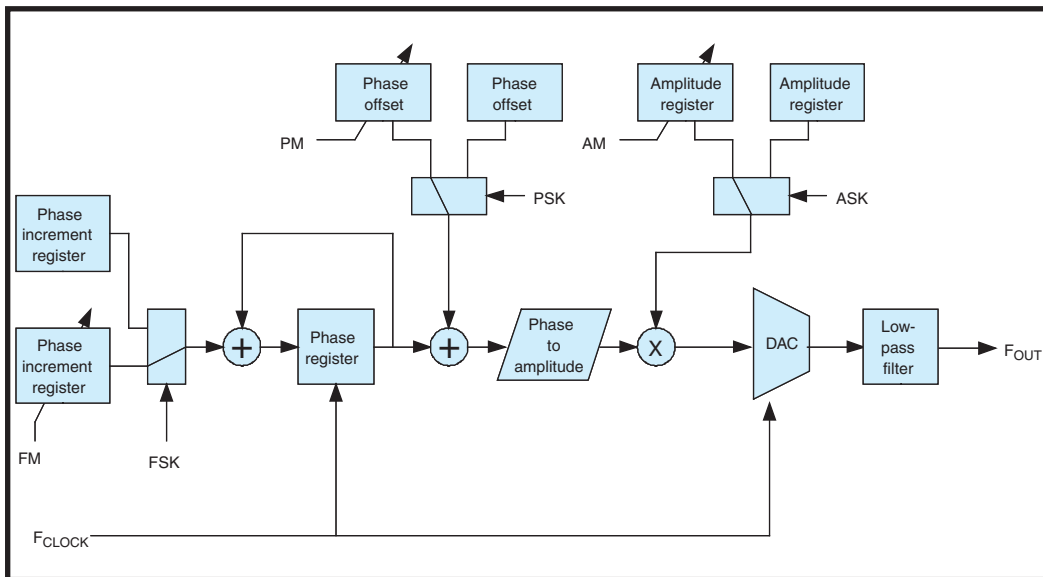


Figure 4—A DDS generator can be easily improved to add full digital modulation features, including either frequency, phase, or amplitude modulations.

kind of modulation, still fully in the digital domain. Refer to Figure 4, which shows a little enhanced DDS architecture. With a DDS, you can easily change the output frequency on the fly without any delay or phase shift just by loading a new value in the phase register or switching between different phase registers for FSK-like transmissions. You can also add a fixed value to the phase register independently from the DDS itself, which is ideal for phase modulation or PSK. You can add a digital multiplier before the DAC to implement software-controlled amplitude or AM modulation.

You can generate waveforms other than sine just by loading a period of your designed signal in the look-up table. But in that case, be careful, because you will be drastically limited in terms of maximum frequency. Due to the mandatory output low-pass filter, all harmonics above the Nyquist limit of $F_{\text{CLOCK}}/2$ will be filtered out. So, for non-sine signals, you

will be limited to output frequencies low enough to ensure that all harmonics required for a good generation of your signal are significantly

There are a lot of other possibilities thanks to the digital structure of a DDS, and silicon makers are imaginative in these areas. You will see

below this limit, which usually means frequencies not above $F_{\text{CLOCK}}/20$ to $F_{\text{CLOCK}}/100$. For example, look at the datasheet for a lab-class arbitrary signal generator like the Agilent Technologies 33220A, which is 50 Mps. It states maximum sine frequency = 20 MHz and maximum triangle frequency = 200 kHz.^[1] Now you know why. If you need to generate a square signal, you will not have these limitations because you can generate a sine and add a simple comparator to extract a square signal with the same frequency.

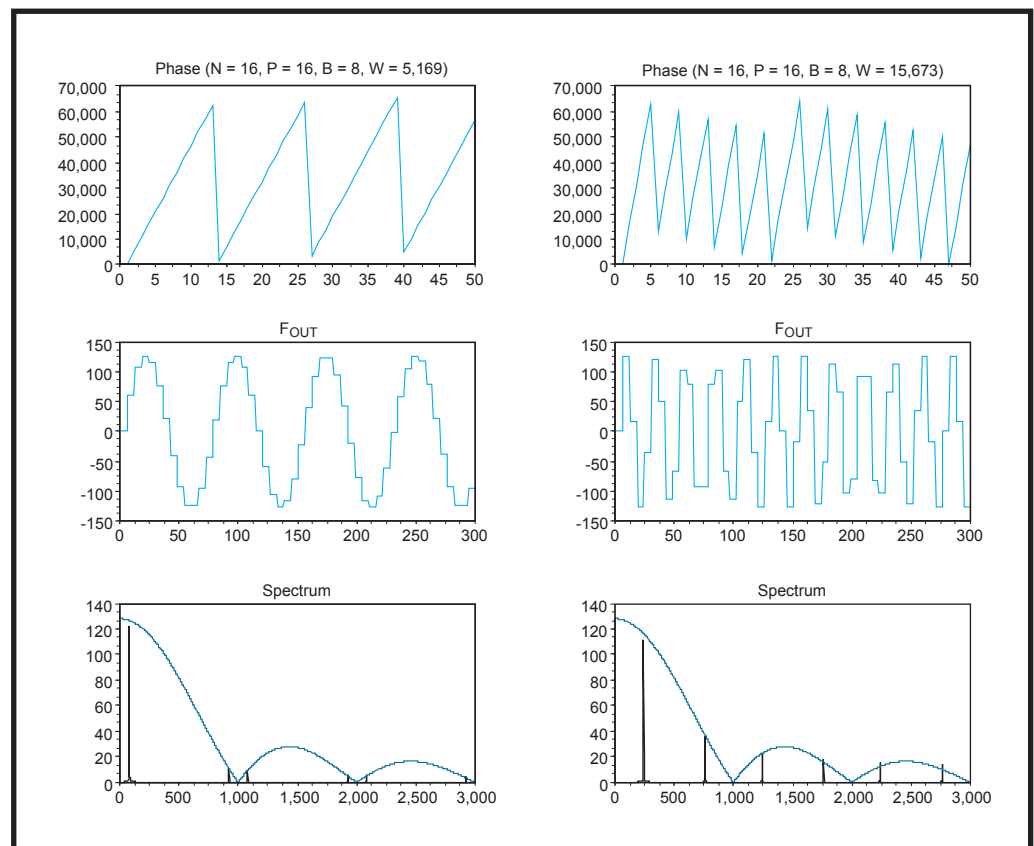


Figure 5—This Scilab simulation shows the phase, output signal, and output spectrum of a 16-bit DDS clocked at 1 GHz, with two different tuning words ($F_{\text{OUT}} = 78$ MHz on the left and 239 MHz on the right). The amplitude of the fundamental frequency gets lower when the frequency increases—following a $\sin(x)/x$ curve (dark blue)—and image frequencies get more powerful and unfortunately closer to the desired frequency.

some examples later on.

WHO SAID $\sin(x)/x$?

You have now discovered all of the key advantages of a DDS architecture, but what are the difficulties that you may encounter? First, you have to look at the low-pass reconstruction filter again. Why do you need it?

Because the output of the DAC is not a sine signal but a succession of steps that match a sine curve only at the clock-edge events, even if you assume that there are no other sources of error elsewhere. In the frequency domain, this means that the spectrum of the output signal will not be a simple fundamental F_{OUT} , but a more complex signal. I used SciLab, a Matlab-like open-source tool, for a simulation (see Figure 5).

There are image frequencies in the output. You get not only the frequency F_{OUT} , but also $F_{CLOCK} - F_{OUT}$ and $F_{CLOCK} + F_{OUT}$, and even $2 F_{CLOCK} - F_{OUT}$ and $2 F_{CLOCK} + F_{OUT}$, and more. The respective amplitudes of these image frequencies follow a curve mathematically defined as $\sin(x)/x$, which happens to be the Fourier transform of a single step of width $1/F_{CLOCK}$.

But there is another problem. When your output frequency goes higher and higher, the power of the image frequencies gets higher too. Power needs to be found somewhere. This implies that the power of your desired F_{OUT} signal becomes lower and follows the same $\sin(x)/x$ curve shown in Figure 5.

This leads to two problems. One, you need to know (and compensate for if necessary) the reduction of signal amplitude when the frequency goes closer and closer to the Nyquist limit, at which point the theoretical power reduction is 3.92 dB. Two, when you come close to this limit, the first image frequency, which you need to cancel out with the low-pass filter, comes closer to your desired frequency and, worse, at a similar amplitude. Because the required low-pass filter would be impossible to build, you can't actually generate a signal arbitrarily close to the $F_{CLOCK}/2$ limit (see Figure 6). The usual reasonable limit is around 40% of F_{CLOCK} even with sharp filters.

However, nothing prevents you from using one of these image frequencies instead of the fundamental. Just replace the low-pass filter with a band-pass filter and you can use a DDS to generate a frequency higher than the Nyquist limit, far in the UHF area. The amplitude will be lower, but it will work as long as your filter is well designed.

ANY OTHER PROBLEM?

Once you have managed to filter out any image frequencies, will you get a

perfectly clean sine signal? You will, but only if you have a perfect DDS with an infinite number of bits and infinite precision everywhere. Unfortunately, you are not that rich. One of your enemies will be DAC resolution. Because the resolution B of the DAC is not so high, there will be a quantization error, which will translate into quantization noise in the output spectrum. Once again, I have a small SciLab simulation with two different DAC resolutions (see Figure 7). The

PROTEUS

The Complete Design Solution

Save time & money on your next embedded design!



- Design your schematic
- Simulate your design
- Debug in 1/3 the time
- Make your PCB

***FREE downloadable demo**

"I am very pleased with the simulator. I got my project designed and working in no time, without having to develop a breadboard. Well worth every penny we paid for it!!"

-Boeing Satellite Development Center, El Segundo, CA

R4SYSTEMS INC.
Toll Free 866.499.8184
info@r4systems.com

Other Products:
PIC & AVR Compilers
PIC & AVR Programmers
Discovering PIC Course

Proteus Starter Kit \$299 * Complete Systems from \$589

-Works with popular compilers and assemblers: CCS, Keil, IAR, Boost C & Hi-Tech
-Available for PIC, 8051, AVR, HC11, ARM/LPC2000 and Basic Stamp

www.r4systems.com

theory says that the signal to total quantization noise power ratio is $1.76 + 6.02B$ dB, with B as the resolution in bits of the DAC. For example, with an 8-bit DAC, you can expect a 50-dB (i.e., $1.76 + 6.02 \times 8$) signal-to-noise power ratio. But that's just an average. However, there is a trick if the quantization noise is a problem. Because the noise is somehow spread from DC to the Nyquist limit, you can limit it just with a band-pass filter around your frequency of interest. If you reject all frequencies except a 10% pass-band around F_{OUT} , then the quantization noise will be divided by 10. Another solution is oversampling. If you increase F_{CLOCK} without increasing the low-pass filter corner frequency, the quantization noise will be lower in the filter passband too.

DDS also has another issue that's often more crucial than quantization errors: phase accumulator truncation.

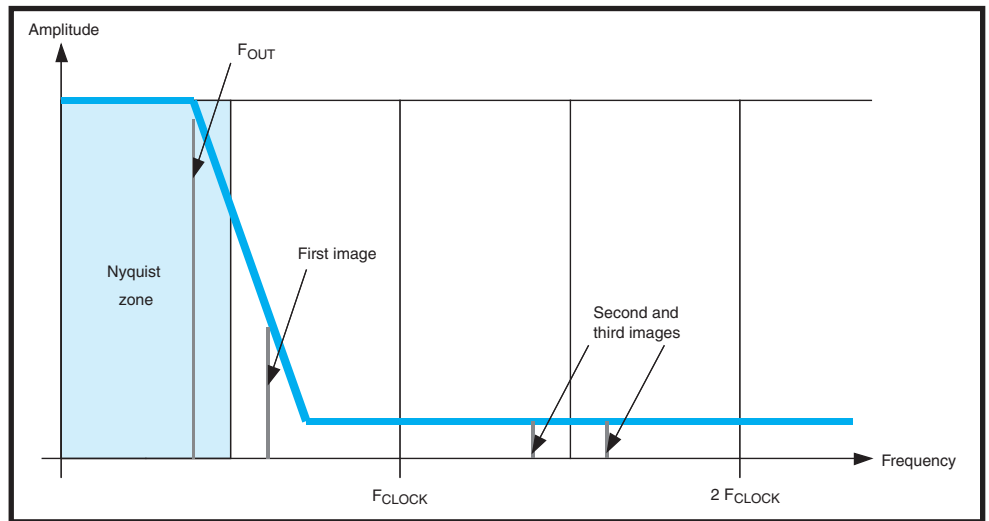


Figure 6—The role of the low-pass filter is fundamental. It must keep the fundamental frequency nearly untouched but provide a high attenuation on all image frequencies. That's why straight filters are usually required. It is also why the maximum frequency is usually 40% and not 50% of F_{CLOCK} .

The number of bits in the phase accumulator register is not infinite; furthermore, the number P of input bits in the look-up table is not infinite. This will give another error on the output. Contrary to DAC quantization, this error will not generate

broadband noise but discrete spurious frequencies on the output spectrum. You may think of it as a miniature unwanted DDS generator working on the unused bits and unfortunately added to the output. Once again, the theory helps. It says that the relative

Cellular and GPS capable Data Mover



- Flash file System
- 4 Chan 12-Bit A/D
- 1MB SRAM
- 512KB FLASH
- 4 Isolated Inputs
- 4 Hi Current Outputs
- 2 External RS-232
- 2 Internal RS-232
- Bat Backed Clk/Cal
- Cell Modem Option
- Internal GPS Option
- Metal Case Option
- 1ma Standby Option

It's easy and cost-effective to do mobile or solar-powered data collection and asset monitoring with the JK micro's **Data Mover**.

With the ability to integrate a Cellular Modem, GPS and DOS-based embedded controller in a single rugged enclosure, you can capture and transmit your data quickly, easily and at low cost. Inexpensive development kits including Borland C/C++ and PowerBasic are available now. Call or email us for more details.

Call **530-297-6073** Email **sales@jkmicro.com**
On the web at **www.jkmicro.com**

JK microsystems

WIRELESS RS-232

WCSC (Willies Computer Software Co)



- * Extremely easy to install and use
- * Distances up to 100 meters (329 feet), 500 meters with extended antenna (1640 feet)
- * Bluetooth Serial Profile
- * One to One virtual cable or selectively connect one to many.

- * Communicates with Bluetooth enabled PDAs, Laptops, Smartphones & Computer
- * No special software required!

Other Products

- Professional serial communication libraries & development tools.
- PCI, PCMCIA, Universal PCI, & ISA multiport RS232, Rs422, & Rs485 cards
- Bluetooth/USB Dongles
- USB/Serial Adapters

http://CircuitCellar.wcscnet.com
sales@wcscnet.com 281.360.4232

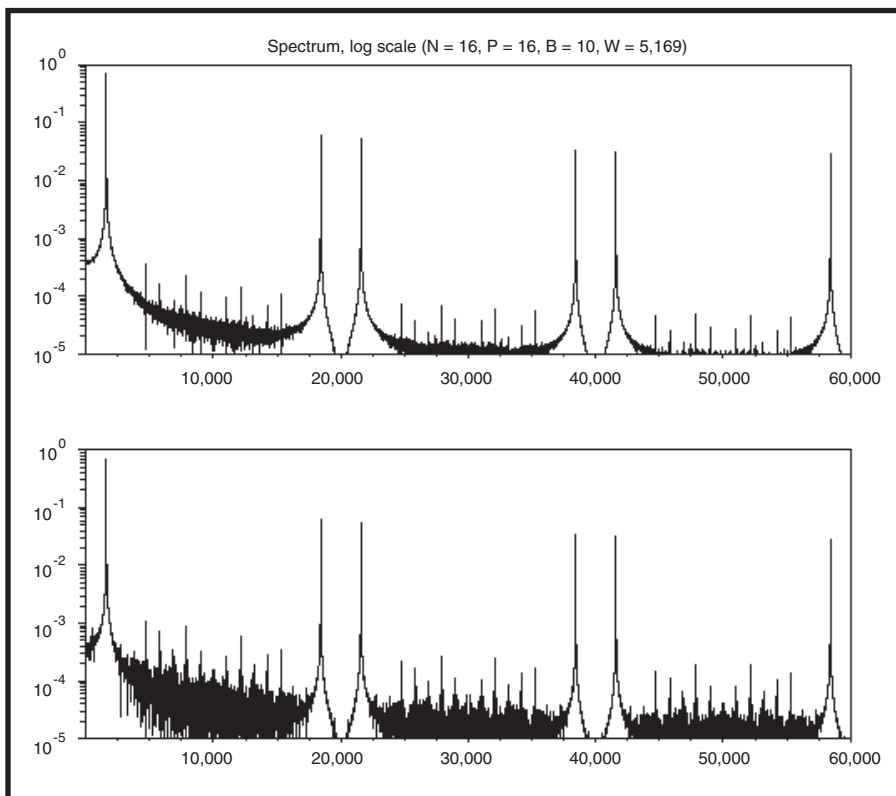


Figure 7—This simulation, still done with Scilab, shows DAC quantization effect. These are simulated output spectra of a 1-GHz 16-bit DDS tuned to provide a 78-MHz output. The top curve is with a 10-bit DAC. The bottom curve is with an 8-bit DAC. The vertical scale is logarithmic.

power of the largest spur is around $-6.02 P$ dBc, with P being the number of phase bits after truncation. The difficulty is that the count, frequencies, and amplitudes of these spurious signals are dependent on both P and the phase increment value W . If you change the selected DDS frequency a little, the spurious frequencies will be drastically different (see Figure 8, p. 67). This behavior makes life for a DDS designer a little more complicated, but it is also a potential friend. If you have some flexibility in the DDS parameters, for example, and if you can have a slightly different F_{CLOCK} or F_{OUT} , then you may find another combination that gives fewer spurious frequencies (or at least less fewer spurs in a given frequency band). The good news is that the behavior of the DDS is predictable and some good simulation tools are available from chip manufacturers.

There are other sources of noise in DDS (clock jitter, DAC nonlinearity, clock feed-through, and more), but image frequencies, DAC quantization,

USBees DX Oscilloscope and Logic Analyzer - 32 Channels/200 MHz

Includes the PacketPresenter™ Protocol Analyzer

Logic Analyzer
Oscilloscope
Protocol Analyzer
Signal Generator

Powerful Debugging
Small and Portable

USBees.com

Actual Size

and phase-register truncation are usually the main contributors. However, don't conclude that DDS generates only noisy signals. These problems exist, but a good, well-designed DDS can have signal-to-noise ratios well above 70 dBc, large enough for the vast majority of applications. By the way, you will find two different figures in the specifications: the signal-to-noise ratio and the spurious-free dynamic range. They are correlated but not equivalent. The former is the ratio of signal power to the sum of all noises. The latter is the ratio of signal power to the strongest spurious frequency.

SOFTWARE IMPLEMENTATION

Enough theory. It's time to demonstrate how to build an actual DDS generator. There are some impressive dedicated integrated circuits around. I will examine them later. For now, let's start with a firmware-based implementation because this month's theme is Embedded Development.

Programming a DDS in a general-

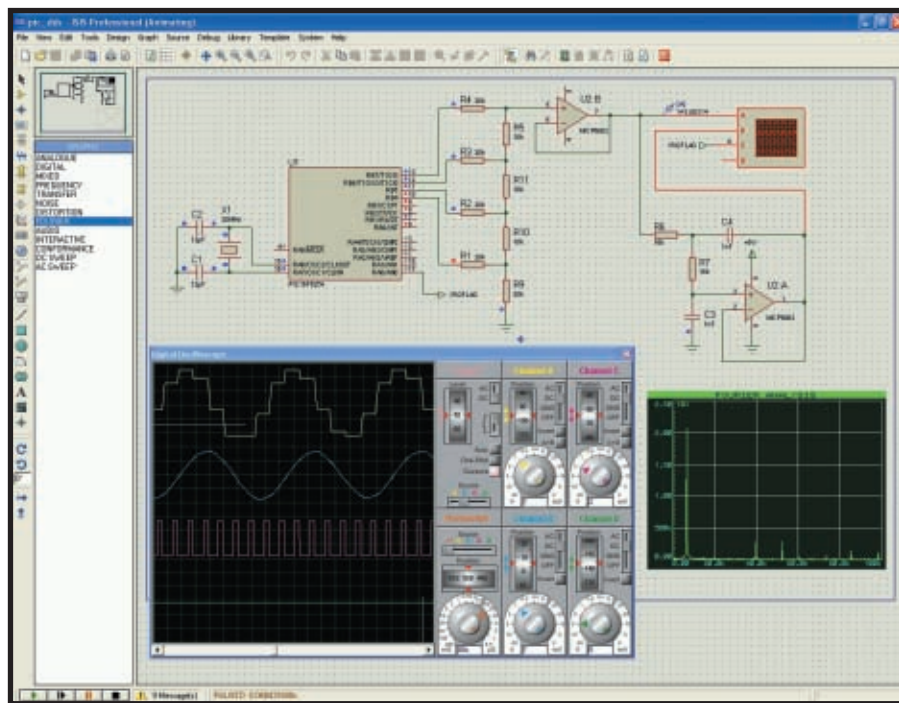


Photo 1—This is the Proteus VSM in action. The schematic shows you that I have used a PIC microcontroller, a small R/2R 4-bit DAC, and an active filter. Just virtually load the associated firmware in the microcontroller, connect a virtual oscilloscope or spectrum analyzer to the output, click "run," and you have the simulated output on the display. Impressive.

purpose microcontroller or DSP is often an effective solution for signal

generation. Imagine that you are using a small microcontroller (i.e., a

Easy Embedded Linux

\$169
Qty 1

16MB FLASH / 32MB RAM

200Mhz Arm9 CPU

16 Digital I/O

Watchdog

10/100 Ethernet

Battery backed Clock/Calendar

Audio In/Out

2 USB

2 Serial Ports

We brought you the world's easiest to use DOS controllers and now we've done it again with Linux. The **OmniFlash** controller comes preloaded with Linux and our development kit includes all the tools you need to get your project up and running fast.

Out-of-the-box kernel support for USB mass storage and 802.11b wireless, along with a fully integrated Clock/Calendar puts the **OmniFlash** ahead of the competition.

Call **530-297-6073** Email sales@jkmicro.com

On the web at www.jkmicro.com

JK microsystems

CIRCUIT CELLAR®

back issues available as

Searchable Archives on CD-ROM

CD-ROM #12 2007 Issues 198-209

CD-ROM #11 2006 Issues 186-197

CD-ROM #10 2005 Issues 174-185

CD-ROM #9 2004 Issues 162-173

Order Online:
www.circuitcellar.com
 or call 860.875.2199

Microchip Technology PIC16F629A clocked at 20 MHz) and you need to generate a 7,117-Hz signal, either sine or square. This is just an example, of course, but real-life applications can include DTMF generation, data rate generation, and similar problems.

The first idea that you, a firmware developer, will have will be to use the on-chip timer. Just configure a timer to count processor cycles (5 MHz maximum on this PIC variant) and toggle the output each N cycles. Calculate N for the output to be as close as possible to the required 7,117 Hz. Here you can have either 7,122 Hz (i.e., 5 MHz/702) or 7,112 Hz (i.e., 5 MHz/703). That isn't too bad, but it's quite far from the target, and you can't program a "fractional" count on a timer.

This is where DDS helps. Imagine another approach: configure the on-chip timer for an interrupt at any frequency but significantly above $2 \times 7,117$ Hz

(e.g., 50 kHz). At each interrupt, add a fixed amount W to a 16-bit phase register, convert it to a sine using an 8-bit ROM-based look-up table, and send the value to a DAC. Then, filter it with a 10-kHz low-pass filter. Refer to the schematic in Photo 1, in which I have just used a simple 4-bit passive R-2R network as a DAC and a pair of Microchip MCP6002 op-amps as a buffer and low-pass filter. If you need a square signal, you can simply route the filtered signal back to the comparator available inside the PIC. The associated source code, fully coded in C using the free Hi-Tech Software PICC-Lite compiler, is available on the *Circuit Cellar* FTP site and is no longer than one page. You have built an actual DDS, and you can generate any frequency calculated as:

$$W \times \frac{50 \text{ kHz}}{65,536}$$

Thus, any frequency from 0.76 Hz to

close to 20 kHz with a frequency step of 0.76 Hz! For example, just choose $W = 9,328$ and you get a frequency of 7,116.69 Hz. That's far closer to the 7,117-Hz target, isn't it? The magical trick comes from the fact that a DDS allows drastically finer frequency steps because the phase increment is not necessarily a sub-multiple of the period.

At this point, I can't resist telling you about a great simulation tool for mixed-signal designs. Labcenter Electronics's Proteus tool suite includes tools for schematic entry, Spice simulation, and PCB design. It also provides an impressive simulator named virtual system modeling (VSM) as an option. With VSM, you can simulate the code running on a microcontroller, like any firmware simulator, and the electronic circuits, like any Spice-like simulator, but you can simulate both simultaneously. Take another look at Photo 1. A virtual scope enabled me to verify the DDS signals

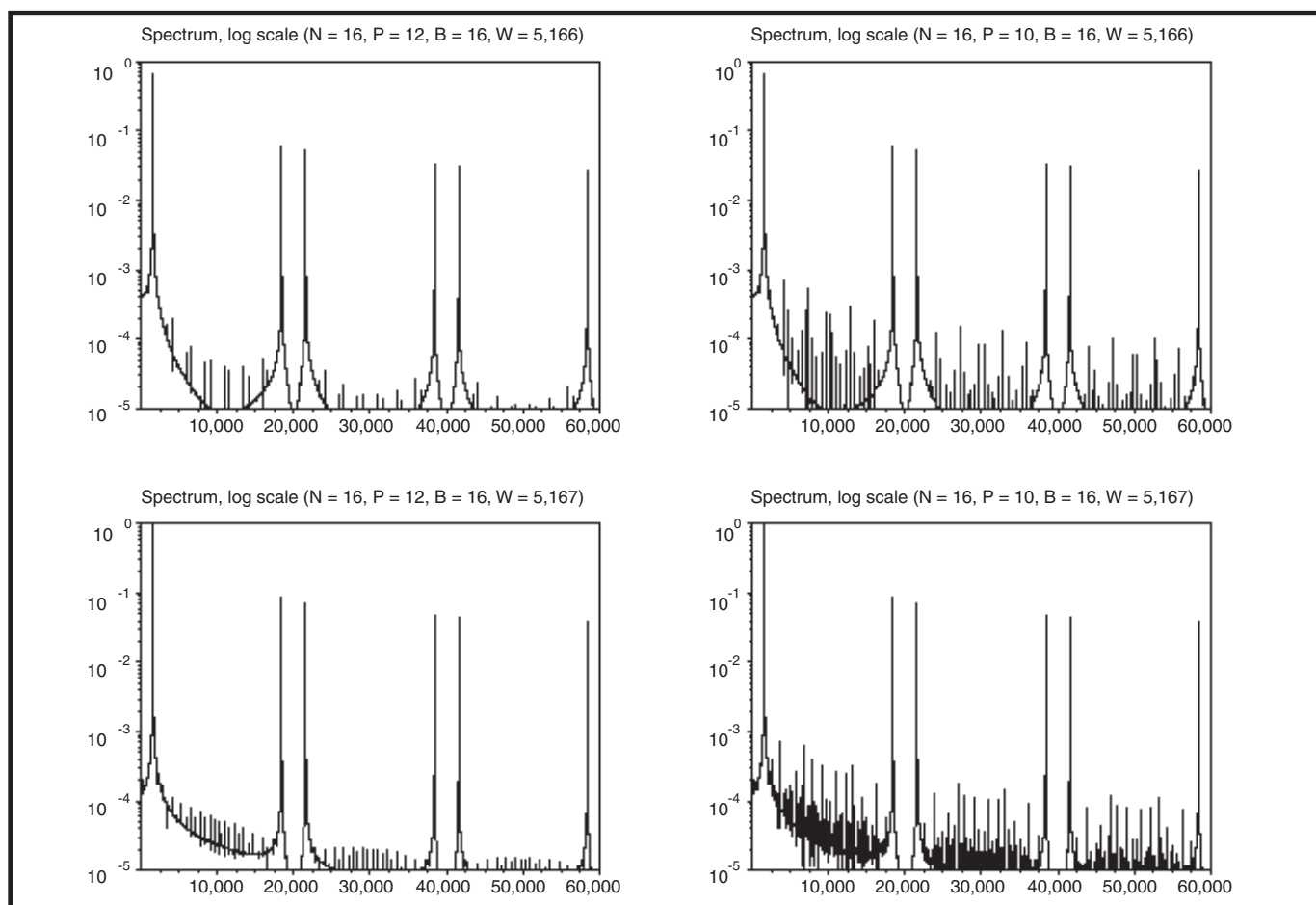


Figure 8—Here I'm illustrating spurious phase truncation. The left column is a DDS with 12 bits effectively used as a look-up table address. The right is with only 10 bits. The spurious spectrum is far more numerous and powerful in the latter. Finally, the bottom line shows what happens in the same condition with just a small change in the tuning word value (5,167 vs. 5,166). The spectrum of spurs is different.

generated by the PIC and filtered by the MCP6002, all without having to switch on the soldering iron! With the advanced simulation option, Proteus VSM can even calculate the spectrum of the filtered signal, as shown. Close to expectations, isn't it?

SILICON SOLUTIONS

Even if software-based DDS is possible, there are plenty of impressive silicon versions, particularly from Analog Devices. An example is an easy-to-use chip such as the AD9833, a \$4 (qty. 1,000) low-power chip fitted in a 10-pin 3 mm × 3 mm MSOP package that enables you to generate frequencies up to nearly 12.5 MHz with 0.1-Hz resolution. The chip is driven by a standard SPI port, which enables you to connect it to any microcontroller. Since energy consumption is such an important consideration, note that this chip eats

no more than 5.5 mA at 3 V, which is impressive for a 25-MHz chip. Technically speaking, it has a 28-bit phase register, a 12-bit look-up table, and a 10-bit DAC, which provides around 60 dB of signal-to-noise ratio. Now you will easily understand such a datasheet.

Now let's focus on a current, top-of-the-line DDS chip. The Analog Devices AD9910 depicted in Figure 9 is nearly 10 times more expensive than the AD9833. It costs around \$35 (qty. 1,000) as I write this. But what a piece of silicon! First, its clock can be as high as 1,000 MHz, providing a useful output range up to 400 MHz. Providing a 1-GHz clock may be difficult, but these guys had the good idea to include an on-chip PLL to allow more reasonable external clock sources. Its 32-bit phase accumulator provides sub-hertz resolution, and it is equipped with a high-speed 14-bit DAC, enabling a spurious-free dynamic

range up to 65 to 70 dBc, and still around -55 dBc at 400 MHz. But that was for the DDS core alone, and this chip has plenty of other blocks.

First, it has an auxiliary DAC to define the full-range amplitude without compromising the quantization noise. It can also automatically compensate for the $\sin(x)/x$ amplitude rolloff I discussed earlier, with a digital filter that has an inverse $\sin(x)/x$ response placed between the look-up table and the DAC. You can program eight different settings for frequency, phase, and amplitude, and then switch among them in nanoseconds via three external pins. If necessary, it can also automatically manage linear frequency, phase, or amplitude sweeps. In addition, it has a built-in 1,024 × 32 RAM that enables you to predefine custom frequency/phase/amplitude profiles and execute them at high speeds, which is perfect for generating complex

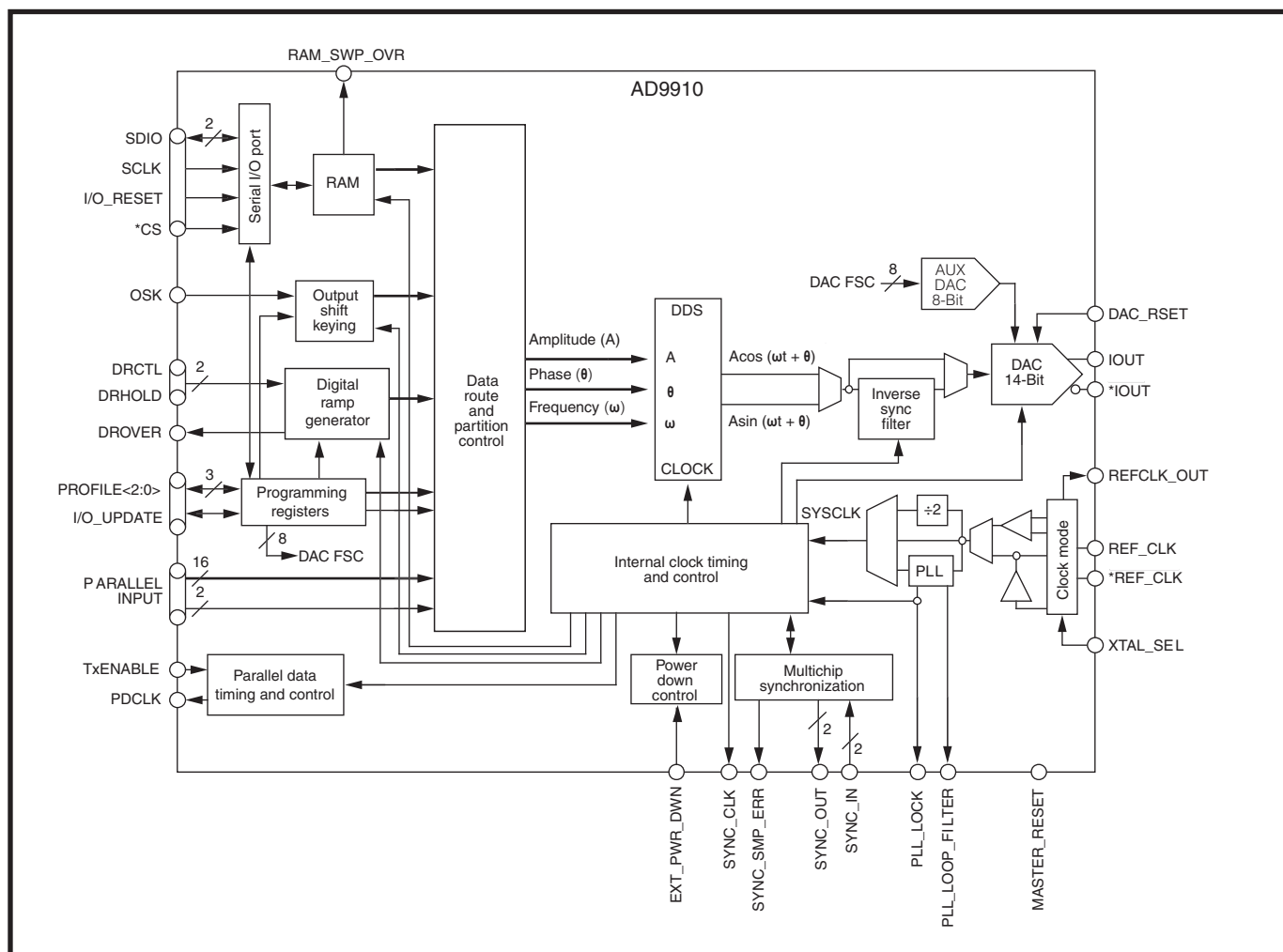


Figure 9—This is the internal architecture of the AD9910 high-end DDS chip (courtesy of Analog Devices). As you see, the DDS core is just a small part of the chip. It is surrounded by a zillion advanced high-speed digital modulation and control blocks, as well as a 14-bit Gsps DAC and a reference clock PLL multiplier.

modulated waveforms.

What else? Oh yes, it can be synchronized with other chips if its features are not enough for your application. It is managed through a SPI, but it also has a high-speed parallel bus for time-critical applications. OK, it is not a low-power chip (800 mW, 1.8 V, and 3.3 V), and you will have to solder its 100 pins and read its 64-page datasheet if you decide to use it, but that's the price you pay for such a list of features.

Another interesting chip is the Analog Devices AD9912. It is a 1-Gsps chip and has less modulation options, but it provides a 48-bit phase register and a 4- μ Hz resolution up to 400 MHz. (I'm not sure if such a resolution is useful. You should double-check the stability of your reference oscillator too.) Note that the AD9912 has an interesting new feature: SpurKiller channels. Theoretically, this feature will enable you to cancel any given pair of spurious signals. It is based on nothing less than two independent mini-DDS generators that can be tuned to generate a signal at the same frequency as the spurious one you want to kill, but in opposition of phase. The circuit then adds these signals on the output, all entirely in the digital domain, prior to the DAC. This feature seems to require delicate tuning. But a typical spurious reduction of around 6 to 8 dB is announced, with specific configurations providing up to 30-dB attenuation. Something to be bench-tested someday for sure!

WRAPPING UP

Here you are. You should now have a better idea about the pros and cons of direct digital synthesis. But let me summarize for good measure.

A direct digital synthesizer (DDS) will provide you with a marvelous sub-hertz frequency resolution, immediate frequency hopping, and efficient full-digital modulation features. However, its frequency range will be limited to around 40% of the clock source, except if you try to use image frequencies, and you may suffer from some nasty spurious signals on the output spectrum.

It is interesting to compare these characteristics with a synthesizer

based on a PLL with programmable dividers. A integer PLL with its single divider can't have simultaneously fast tuning and a fine frequency resolution, which are always in opposition. Even with fractional PLLs that have two dividers, you will usually get only kilohertz-range frequency steps, and tuning to a new frequency will take tens or hundreds of microseconds. However, a PLL can be used to generate an output signal far above its reference frequency, and its output is usually clean, except for when it's close to the center output frequency or its harmonics. There aren't any "digital spurious frequencies" like with a DDS.

Based on this comparison, you will deduce that a PLL/VCO combination is usually more suited to local oscillators, where high frequencies and clean signals are a must. A DDS finds its key applications as a modulation source where agility is most important. However, DDS chips have gotten cleaner and cleaner over the years, and nobody would have imagined seeing a 1-GHz DDS chip for tens of dollars a couple of years ago.

For the best of both worlds, there are chips with both PLL and DDS cores. So, stay tuned, things can change quickly!

Now it's your turn. You should be ready to put a DDS in your next design, either as a piece of silicon or some lines of firmware. DDS is no longer on the darker side for you! ☒

Author's note: I want to thank Labcenter Electronics and its U.S. and French distributors, R4 Systems and Multipower, respectively, who were kind enough to provide me with a Proteus VSM license for this project. Also, thank you to Jeff Keip and Eric Benoist at Analog Devices who were more than helpful (as usual)!

Robert Lacoste lives near Paris, France. He has 18 years of experience working on embedded systems, analog designs, and wireless telecommunications. He has won prizes in more than 15 international design contests. In 2003, Robert started a consulting company, ALCIOM, to share

his passion for innovative mixed-signal designs. You can reach him at rlacoste@alciom.com. Don't forget to write "Darker Side" in the subject line to bypass his spam filters.

PROJECT FILES

To download code and additional files, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2008/217.

REFERENCE

- [1] Agilent Technologies, Inc., "Agilent 33220A 20 MHz Function/Arbitrary Waveform Generator Data Sheet," 2005.

RESOURCES

Analog Devices, Inc., "A Technical Tutorial on Digital Signal Synthesis," 1999, www.analog.com/UploadedFiles/Tutorials/450968421DDS_Tutorial_rev12-2-99.pdf.

K. Gentile, D. Brandon, and T. Harris, "DDS Primer," Analog Devices, Inc., 2003, www.ieee.li/pdf/viewgraphs_dds.pdf.

SOURCES

33220A DDS-Based lab generator
Agilent Technologies, Inc.
www.agilent.com

AD9833 Waveform generator, AD9910 direct digital synthesizer, and AD9912 direct digital synthesizer
Analog Devices, Inc.
www.analog.com

PICC-Lite Compiler
Hi-Tech Software
www.htsoft.com

Proteus VSM Mixed-signal simulator
Labcenter Electronics
www.labcenter-electronics.com

MCP6002 op-amp and PIC16F629A microcontroller
Microchip Technology, Inc.
www.microchip.com

Scilab Simulator
Scilab
www.scilab.org