Categories

Antennas (2)
Electronics (10)
Linux (5)
Microcontrollers (8)
Programming (7)
Radio (10)
Test Equipment (2)
Everything! (20)

Sections

GitHub
Datasheets
Electronic Modules
Reference

Simple Direct Digital Synthesis Tutorial

By Stephen Stebbing 2013-11-28 06:02:27 1 comment

1. Introduction

I recently gave a talk at HADARC on a AD9850 based dds vfo project that I had been working on.

Contents

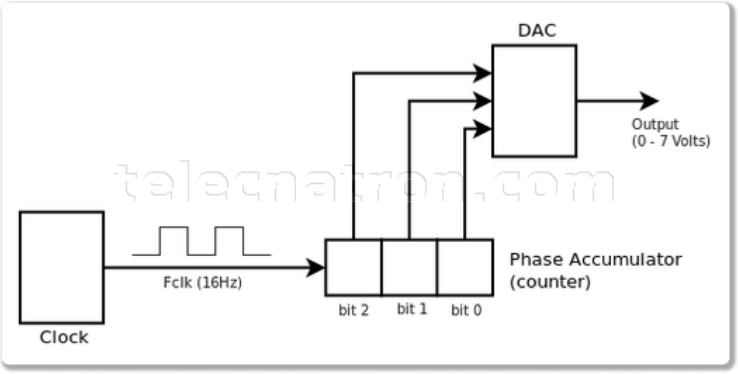
Speaking to some of the members prior to preparing the talk, it became apparent that although everyone had heard of DDS, very few had any idea of how it worked, and so an introductory tutorial was going to be in order.

Searching the web for tutorial material that I could use, I found only articles that jump straight in to hard-core mathematics and digital signal processing theory. It doesn't have to be so complicated in my opinion, so I came up with my own and share it here in the hope that it might be of use to a wider audience.

2. Step 1: Fixed Frequency

2.1. Hardware

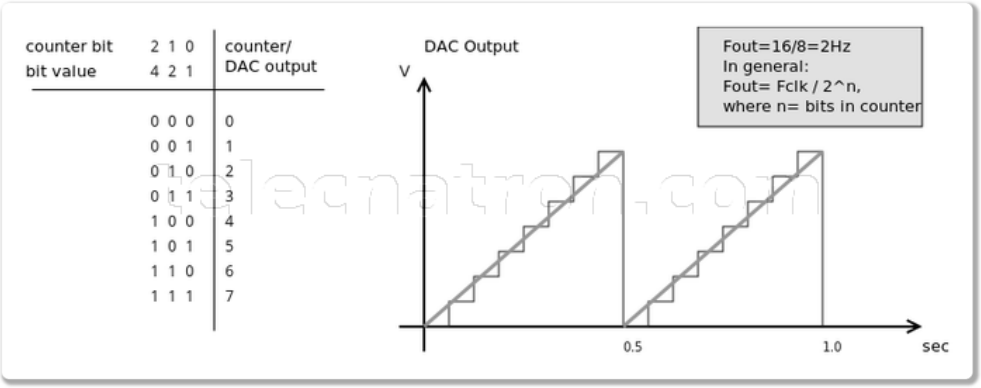
To generate a fixed frequency DDS style, we start out with a clock source which just generates a pulse-train of fixed-frequency square waves which clock the binary counter whose output is connected to the digital to analogue converter (DAC).



To keep things simple, we use a 16Hz clock frequency, a 3-bit counter and an DAC whose output voltage is equal to the counter's output.

In DDS terms, the counter is known as the *phase accumulator.*

2.2. Waveform



As you can see in the truth table at left, on each clock pulse the counter counts up by one until it reaches it maximum value, when it resets and starts again from 0 on the next pulse.

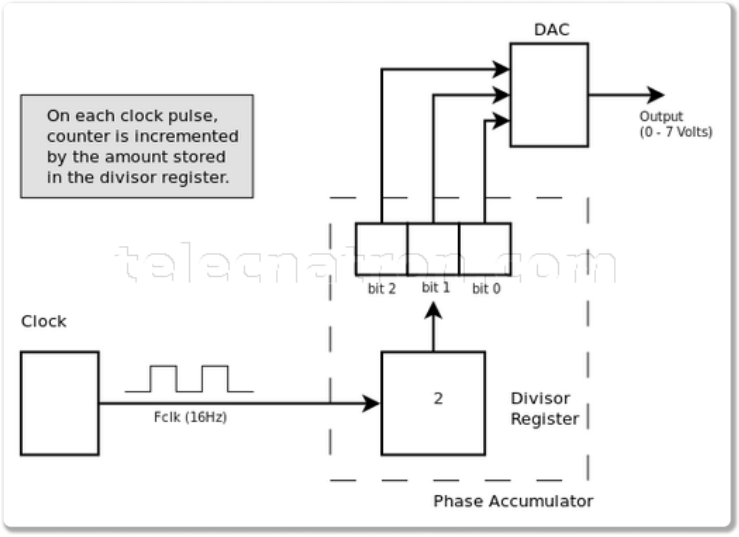
The DAC output then is shown in blue on the graph, and in this example, the counter has 8 discrete output values and there are 16 clock pulses per second, and hence two complete wave cycles for a frequency of 2Hz. The output frequency is determined by dividing the clock frequency by two raised to the power of the number of bits in the counter.

A filter on the DAC output could be used to convert the staircase waveform to a triangle waveform as shown in green.

3. Step 2: Variable Frequency

3.1. Hardware

To be able to generate other frequencies besides Fclk/2 a component called the *divisor register* is added to the phase accumulator. Now, on each clock pulse, the counter is incremented by the value in the divisor register.

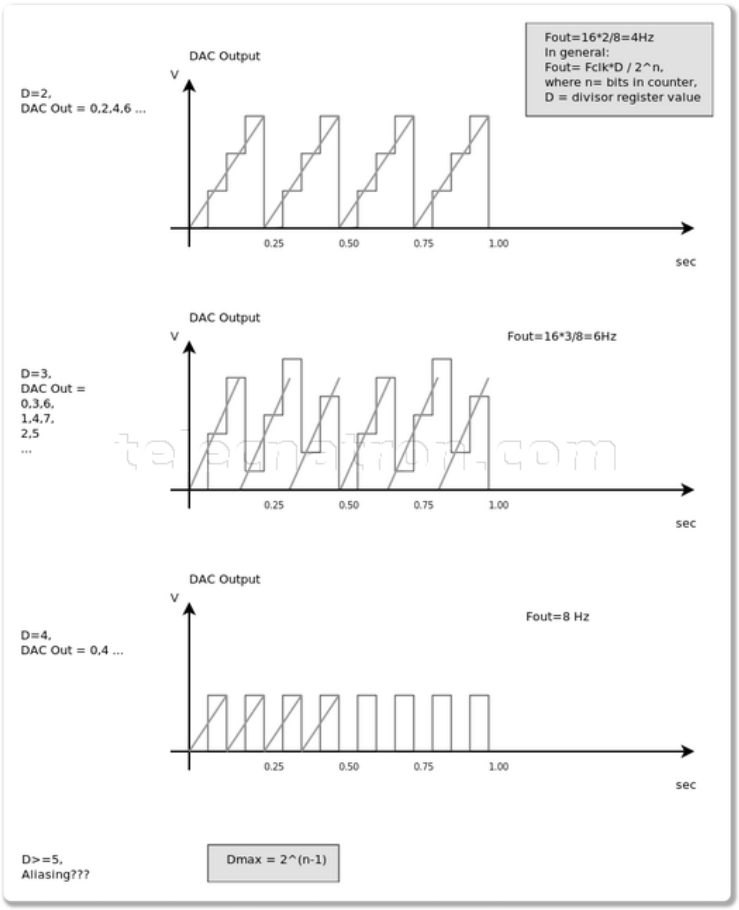


With the divisor register having a value of D=2, on each clock pulse the counter is incremented by two.

The counter now outputs values of 0,2,4,6 before repeating, giving a frequency of twice that in the previous step.

3.2. Waveforms

With D=2, we now have 4Hz. The output frequency is determined by mulitplying the clock frequency by the divisor register value and dividing by two raised to the power of the number of bits in the counter.



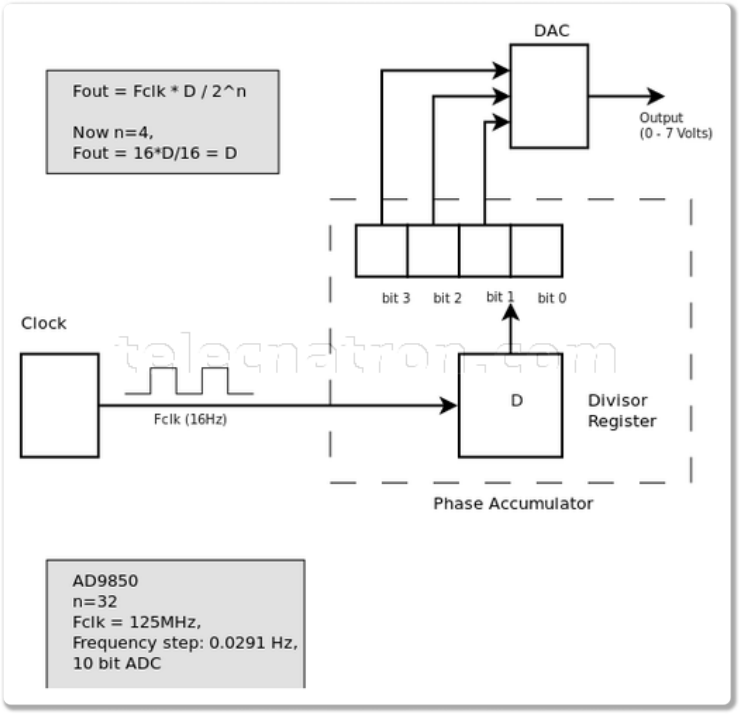
With D=3 for 6Hz, the clock frequency is no longer exactly divisible by the divisor register value and it now takes three cycles through the counter's count range before the pattern repeats. Note how the pattern of (2,5) only includes two steps whereas the others have three, but overall we still have six cycles in the one second period. In this case there is more reliance on the DAC output filter to produce a nice sawtooth wavefrom.

For D=4 there is only one step per cycle; you can see how the output waveform becomes more coarse as the frequency increases.

D=5 doesn't work because **aliasing** occurs, and results in many other lower frequencies being generated. To avoid aliasing, the output frequency cannot be more than half of the clock frequency.

4. Step 3: Variable Fractional Frequencies

In the previous step, we were able to generate any frequency between zero and half of the clock frequency with a resolution of the clock frequency divided by two raised to the power of the number of bits in the counter, which, in the examples was 2Hz.



To double the resolution to 1Hz, all that needs be done is to add a bit to the counter as shown in the diagram.

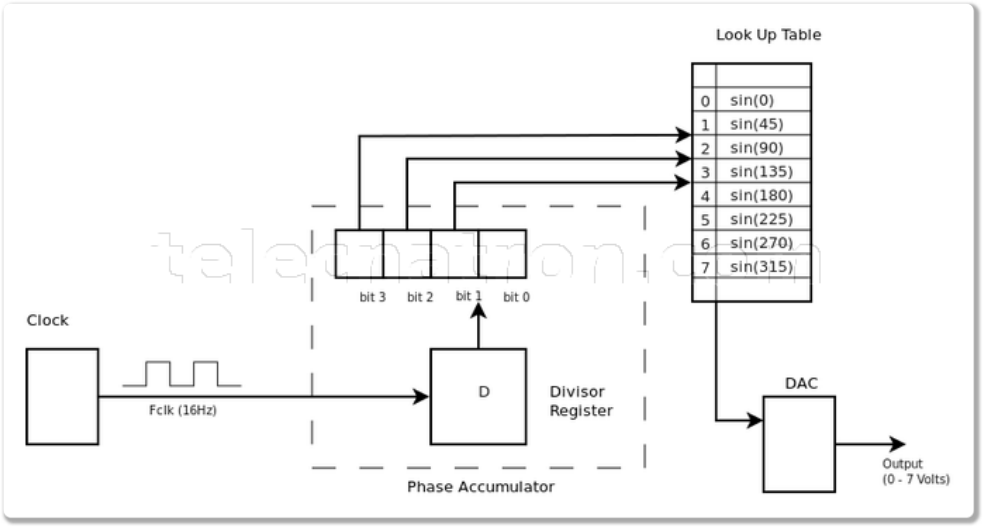
The formula for calculating the output frequency remains the same but now n=4.

For each bit that is added to the counter, the resoluton is doubled (or should that be halved?).

A real world device such as the **AD9850** has a 32 bit counter, which at its maximum clock frequency of 125MHz, has a resolution of 0.0291Hz(!) and with it's 10 bit ADC, can generate 1024 discrete output steps.

5. Step 4: Sinewave Synthesis

The final step is generate a sine wave from the DAC's sawtooth output, this is done by means of the *look up table.*



Here, the lookup table takes (LUT) takes the counter ouput and outputs the corresponding sine value. For example, with a count of 0 the LUT outpus a value of sin(0) = 1 which the DAC converts to 7 volts. With a count of 1, the LUT outpus sin(45)=0.707 and the DAC outputs 0.707 * 7 = 4.95 volts, etc and hence, after filtering, giving a close approximation to a sine wave.

Note that the look up table can contain values for any waveform, and not just a sine wave and could be used as an **arbitary waveform generator.**

6. Conclusion

Hopefully this tutorial has shed some light into how DDS works in an easy to understand manner.

I would like to hear what you think, so please use the comments link below if you have any questions, suggestions or comments.

1 comment