Virtex™ Synthesizable Delta-Sigma DAC

XAPP154 September 23, 1999 (Version 1.1)

Application Note by John Logue

Summary

Digital to analog converters (DACs) convert a binary number into a voltage directly proportional to the value of the binary number. A variety of applications use DACs including waveform generators and programmable voltage sources. This application note describes a Delta-Sigma DAC implemented in a Virtex FPGA. The only external circuitry required is a low pass filter comprised of just one resistor and one capacitor. Internal resource requirements are also minimal. For example, a 10-bit DAC uses only three Virtex CLBs. The speed and flexible output structure of the Virtex series FPGAs make them ideal for this application.

Xilinx Family: Virtex Family

Introduction

Figure 1 is a top-level schematic diagram of a typical Virtex DAC implementation. As shown in this diagram, the inputs include reset and clock signals, in addition to the binary number bus.

DACoutDrvr (Virtex output pin) drives an external low-pass filter. V_{OUT} can be set from 0V to V_{CCO}, where V_{CCO} is the supply voltage applied to the FPGA I/O bank driving the resistor-capacitor filter.

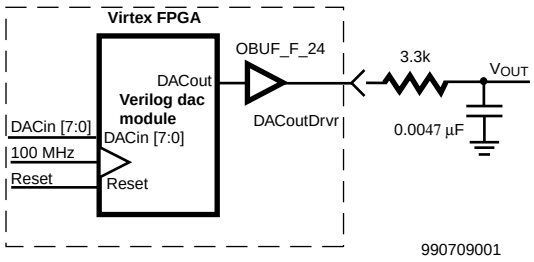


Figure 1: Top-level DAC Implementation

The classic current summing digital to analog converter uses matched resistors to convert a binary number to a corresponding voltage level. This technique works well for high-speed DACs when the binary number is up to ten bits wide. However, it is difficult to maintain accuracy over a range of temperatures as the number of bits increases.

Delta-Sigma Architecture

A Delta-Sigma DAC uses digital techniques. Consequently, it is impervious to temperature change, and may be implemented in programmable logic. Delta-Sigma DACs are actually high-speed single-bit DACs. Using digital feedback, a string of pulses is generated. The average duty cycle of the pulse string is proportional to the value of the binary input. The analog signal is created by passing the pulse string through an analog low-pass filter. While an in-depth discussion of Delta-Sigma conversion is beyond the scope of this application note, the basic architecture, implementation, and trade-offs are covered.

Delta-Sigma DACs are used extensively in audio applications. They are suited for low frequency applications that require relatively high accuracy.

As is standard practice, the DAC binary input in this implementation is an unsigned number with zero representing the lowest voltage level. The analog voltage output is also positive only. A zero on the input produces zero volts at the output. All ones on the input cause the output to nearly reach V_{CCO}. For AC signals, the positive bias on the analog signal can be removed with capacitive coupling to the load. Though the low pass filter can be driven with any of the Virtex SelectIO™ output standards that both sink and source current, this application note emphasizes the LVTTL standard.

Figure 2 is a block diagram of a Delta-Sigma DAC. The width of the binary input in the implementation described below is configurable. For simplicity, the block diagram depicts a DAC with an 8-bit binary input.

The term “Delta-Sigma” refers to the arithmetic difference and sum, respectively. In this implementation, binary adders are used to create both the difference and the sum. Although the inputs to the Delta adder are unsigned, the outputs of both adders are considered signed numbers. The Delta Adder calculates the difference between the DAC input and the current DAC output, represented as a binary number. Since the DAC output is a single bit, it is “all or nothing”; i.e., either all zeroes or all ones. As shown in Figure 2, the difference will result when adding the input to a value created by concatenating two copies of the most significant bit of the Sigma Latch with all zeros. This also compensates for the fact that DACin is unsigned. The Sigma Adder sums its previous output, held in Sigma Latch, with the current output of the Delta Adder.

In most cases, the Delta adder is optimized out when the high level design is synthesized. This is because all bits on either the A or B inputs are zero, so A and B are simply merged, rather than added.

As noted below, the DAC input can be widened by one bit to allow the full analog range of 0V to V_{CCO}. In this case, the Delta adder is needed.

The interface to Verilog module **dac** in Figure 1 includes one output and three input signals as defined in Table 1. All signals are active high.

Table 1: DAC Interface Signals

Signal	Direction	Description
DACout	Output	Pulse string that drives the external low pass filter (via an output driver such as OBUF_F_24).
DACin	Input	Digital input bus. Value must be setup to the positive edge of CLK. For high-speed operation, DACin should be sourced from a pipeline register that is clocked with CLK. For full resolution, each DACin value must be averaged over $2^{(MSBI+1)}$ clocks, so DACin should change only on intervals of $2^{(MSBI+1)}$ clock cycles.
CLK	Input	Positive edge clock for the SigmaLatch and the output D flip-flop.
Reset	Input	Reset initializes the SigmaLatch and the output D flip-flop. In this implementation, SigmaLatch is initialized to a value that corresponds to 0V in and 0V out. If DACin starts at zero, there is no discontinuity.

Implementation

The DAC can be implemented in a single Verilog module, as shown below. The width of the input bus is configurable with defined constant MSBI .

```
‘timescale 100 ps / 10 ps
‘define MSBI 7                // Most significant Bit of DAC input

//This is a Delta-Sigma Digital to Analog Converter

module dac(DACout, DACin, Clk, Reset);
output DACout;                // This is the average output that feeds low pass filter
reg DACout;                   // for optimum performance, ensure that this ff is in IOB
input [‘MSBI:0] DACin;        // DAC input (excess 2**MSBI)
input Clk;
input Reset;

reg [‘MSBI+2:0] DeltaAdder;   // Output of Delta adder
reg [‘MSBI+2:0] SigmaAdder;   // Output of Sigma adder
reg [‘MSBI+2:0] SigmaLatch;   // Latches output of Sigma adder
reg [‘MSBI+2:0] DeltaB;       // B input of Delta adder

always @(SigmaLatch) DeltaB = {SigmaLatch[‘MSBI+2], SigmaLatch[‘MSBI+2]} << (‘MSBI+1);

always @(DACin or DeltaB) DeltaAdder = DACin + DeltaB;

always @(DeltaAdder or SigmaLatch) SigmaAdder = DeltaAdder + SigmaLatch;

always @(posedge Clk or posedge Reset)
begin
     if(Reset)
     begin
          SigmaLatch <= #1 1’bl << (‘MSBI+1);
          DACout <= #1 1‘b0;
     end
     else
     begin
          SigmaLatch <== #1 SigmaAdder;
          DACout <= #1 SigmaLatch[‘MSBI+2];
     end
end
endmodule
```

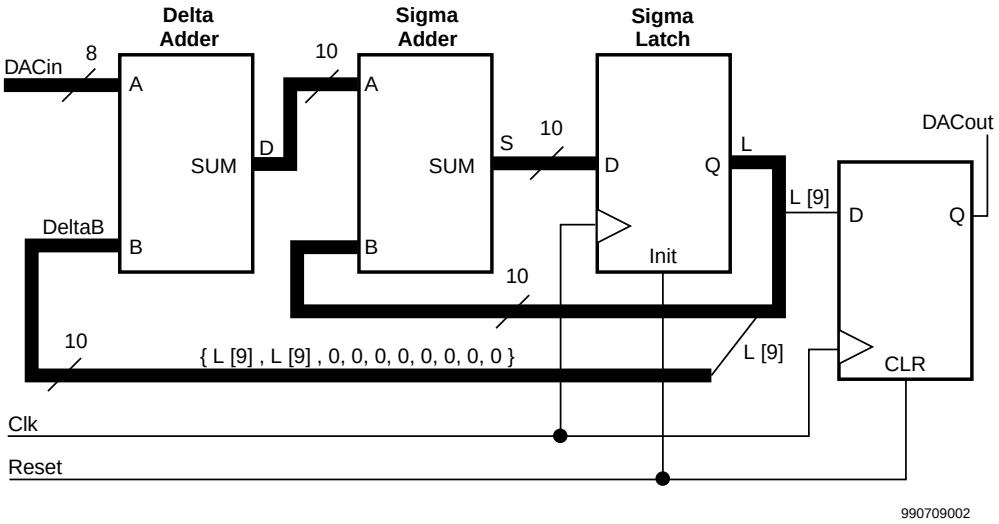


Figure 2: Delta-Sigma DAC Internal Block Diagram

For the implementation in Figure 1, the output voltage (V_{OUT}) as a function of the DAC input may be expressed as follows:

$V_{OUT} = (DACin/(2^{(MSBI+1)})) \times V_{CCO}$ Volts

For example, for an 8-bit DAC (MSBI = 7) the lowest V_{OUT} is 0V when DACin is 0. The highest V_{OUT} is 255/256 V_{CCO} volts when DACin is FF_{16}.

For some applications, it may be important for V_{OUT} to swing through the entire voltage range: 0V to V_{CCO} (rail-to-rail). This is accomplished by increasing the DACin bus width by one bit and leaving all other bus widths the same. For an 8-bit DAC with an input value of 256, $V_{OUT} = V_{CCO}$. Note that all DACin values greater than 256 are illegal and should not be used.

As outlined in this application note, it is often advantageous to use a high-frequency clock. The desired clock may be faster than that which can be practically sourced externally. One or two Virtex Delay Locked Loops (DLLs) can be used to double or quadruple the frequency of an external clock source. See Application note XAPP132 - *Using the Virtex Delay-Locked Loop*.

Low-Pass Filter

The resistor/capacitor low-pass filter shown in Figure 1 is adequate for most applications. A 24 mA LVTTL output buffer is used to provide maximum current drive.

There are three primary considerations in choosing values for the resistor and capacitor:

- **Output Source and Sink Current**. Unlike normal digital applications, it is important that signal DACoutDrvr always switch the entire voltage range from 0 V to V_{CCO} (rail-to-rail). If the value of R is too low and signal DACoutDrvr can not switch rail-to-rail, the analog output is non-linear; i.e., the absolute output voltage change resulting from incrementing or decrementing DACin is not constant. The worst-case output impedance of the 24 mA LVTTL buffer is about 25 Ω. R must be 2.5 KΩ or greater to ensure rail-to-rail switching, with an error of 1% or less.
- **Load Impedance.** Keep the value of R low relative to the impedance of the load so that the current change through the capacitor due to loading becomes negligible.
- **Time Constant.** The filter time constant (τ = RC) must be high enough to greatly attenuate the individual pulses in the pulse string. On the other hand, a high time constant may also attenuate the desired low-frequency output signal. These potentially conflicting requirements are analyzed separately.

Pulse String Filtering

In the midrange voltages, signal DACoutDrvr is switching rapidly making it relatively easy to filter. When the DAC input is at 1 or the highest possible value, the signal DACoutDrvr is at the same level for all but one CLK cycle for each sample period. These are the most difficult output strings to filter; i.e., they are the "worst-case".

Although the filter noise may be calculated as an absolute peak-to-peak voltage, it is more useful to consider it as a fraction of the step voltage. The step voltage (V_S) is defined as the absolute change in V_{OUT} when the DAC input is incremented or decremented. For an 8-bit DAC, V_S equals (1/256) x V_{CCO}.

The worst-case peak-to-peak filter noise for an 8-bit DAC can be expressed as follows:

$PPN_{FS} = (1-e^{-(1/f\tau)}) \times ((1-e^{-(255/f\tau)})/(1-e^{-(256/f\tau)})) \times 256$

where:

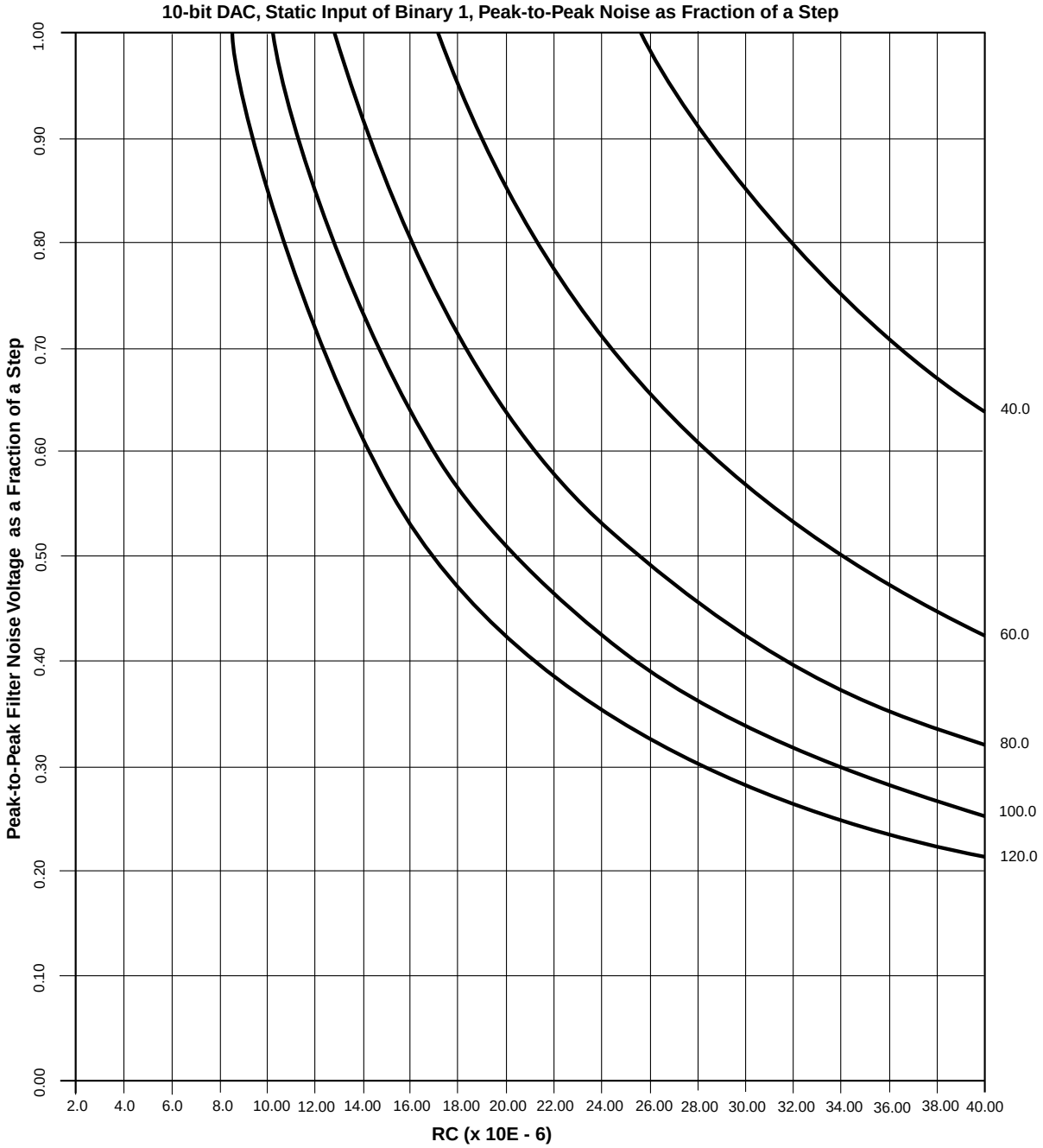
PPN_{FS} - peak-to-peak noise expressed as a fraction of step voltage

f is the DAC clock frequency

τ is the filter time constant, RC.

For simplicity, we did not generalize this equation to handle any width DAC. For other widths, change the constants 256 and 255 to the appropriate power of 2, and (power of 2) minus 1, respectively.

This equation was used to create Figure 3, Figure 4, and Figure 5. These charts may be used to determine the value of RC for the desired worst-case noise voltage and operating frequency. For example, for an 8-bit DAC with a clock frequency of 80 MHz, the user might choose an RC value of 13.0×10^{-6}, corresponding to a peak-to-peak noise voltage of about $0.25V_S$. This leaves $0.75V_S$ noise margin between steps. Part of this noise margin is needed to handle other noise sources such as noise on V_{CCO}.



Note: The number at the end of each curve is the DAC clock frequency in MHz.

99063002

Figure 3: Peak-to-Peak Filter Noise as a Function of RC and Frequency (10-bit DAC)

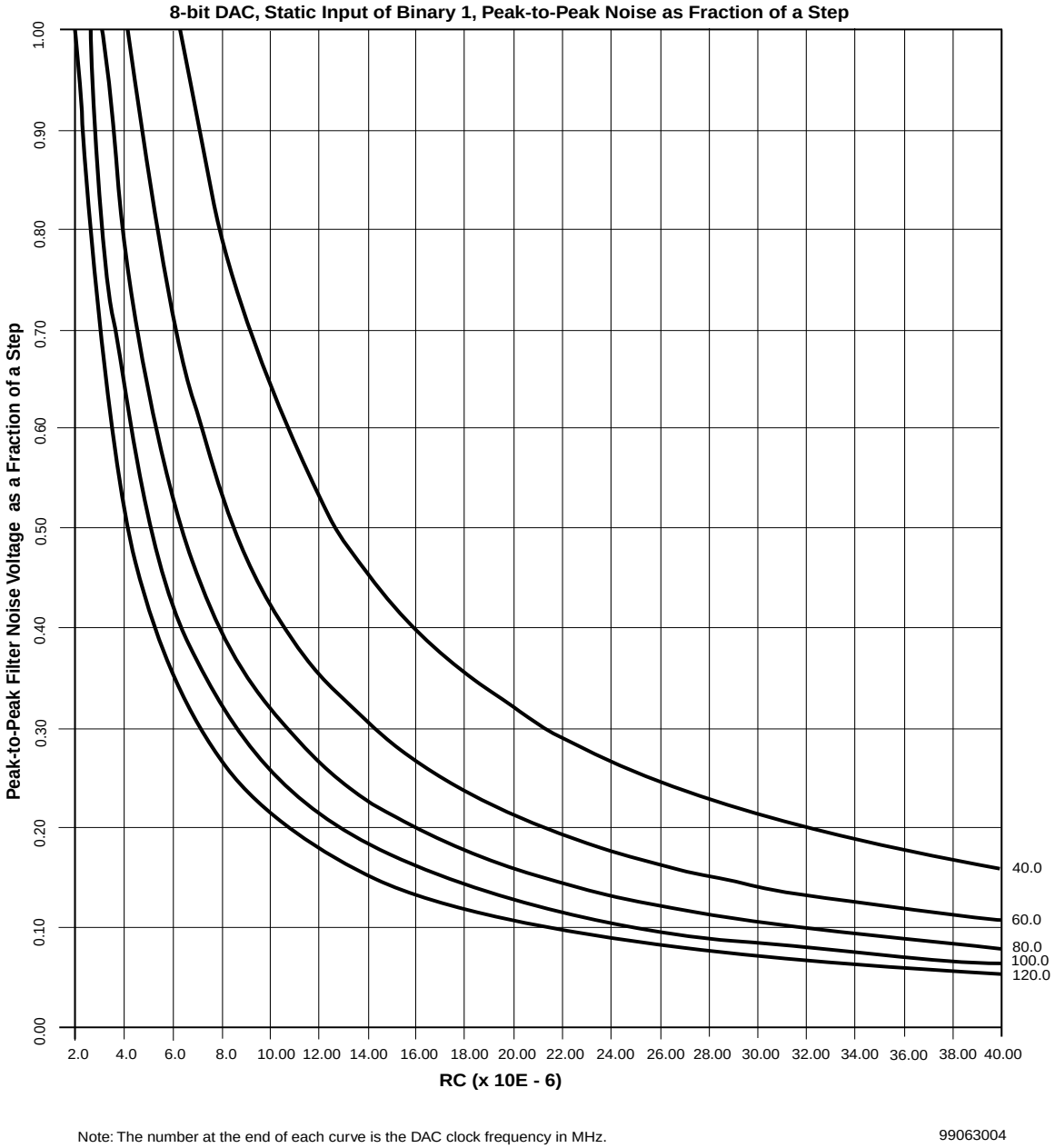
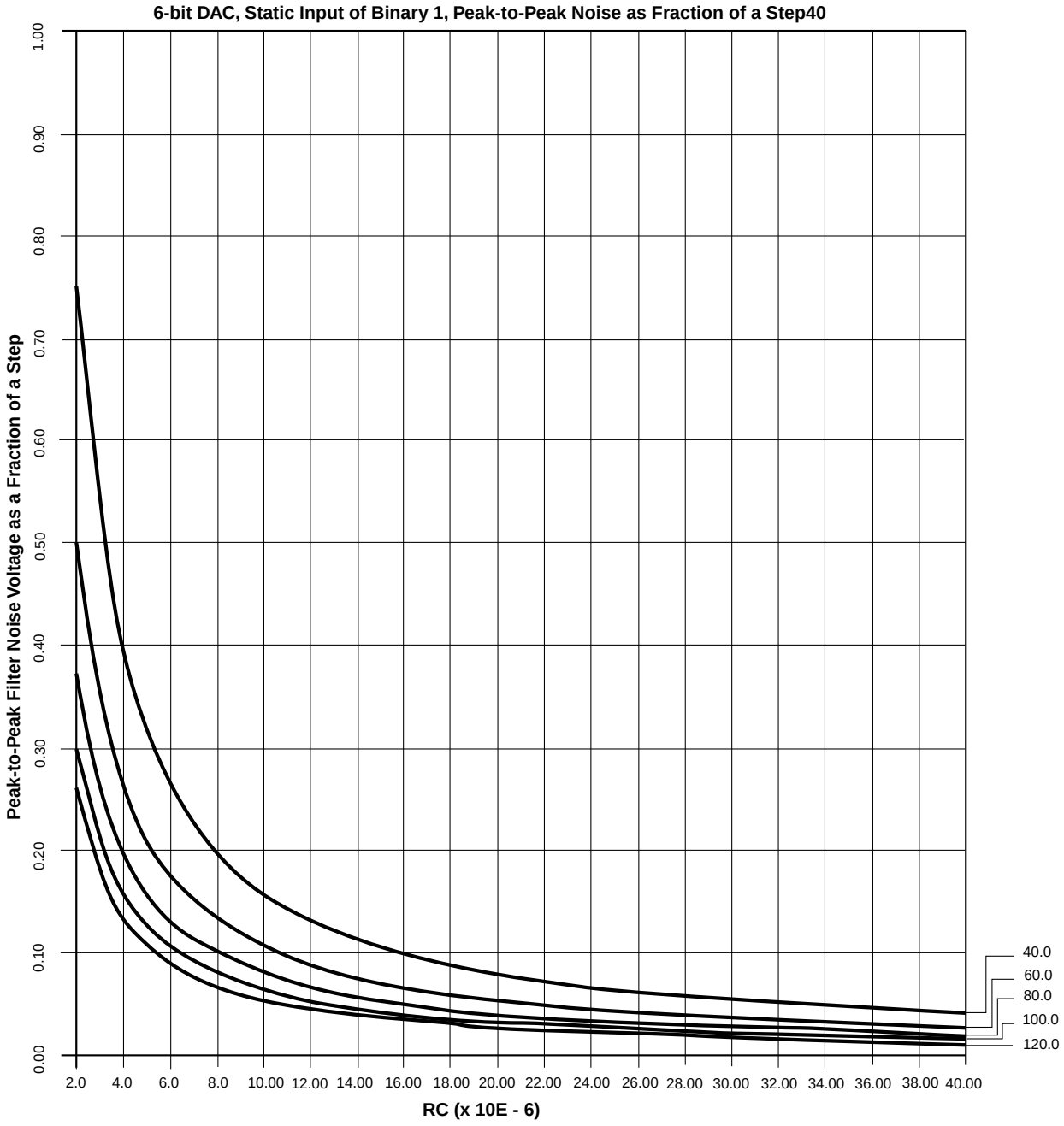


Figure 4: Peak-to-Peak Filter Noise as a Function of RC and Frequency (8-bit DAC)



Note: The number at the end of each curve is the DAC clock frequency in MHz.

99063003

Figure 5: Peak-to-Peak Filter Noise as a Function of RC and Frequency (6-bit DAC)

Output Attenuation

By convention, the cutoff frequency of a low pass filter is defined as the half-power point. The cutoff frequency of the simple RC filter may be expressed as:

$$f_C = 1/(2\pi\tau)$$

where:

f_C is the filter cutoff frequency

τ is the filter time constant, RC.

The above equation was used to create Figure 6.

Figure 6 may be used in conjunction with Figure 3, Figure 4, or Figure 5 to choose the RC time constant that is optimum for a particular application. All figures cover the same RC range.

Figure 4 shows an RC value of 13.0 x 10^{-6} results in a peak-to-peak noise voltage of 0.25V when a DAC clock frequency of 80 MHz is used on a 8-bit DAC. From Figure 6, it can be determined that the filter cutoff frequency for this RC value is about 12 KHz. If the expected output is essentially a DC level, e.g., a programmable voltage generator, then RC may be increased to reduce the clock noise. On the other hand, if the fundamental frequency of the analog output is high, or it has sharp edges, then a lower RC may be needed. When determining the actual component values, remember that R should be at least 2500 Ω.

The user may implement a more sophisticated filter if the simple RC filter has inadequate cutoff or drive characteristics for the application.

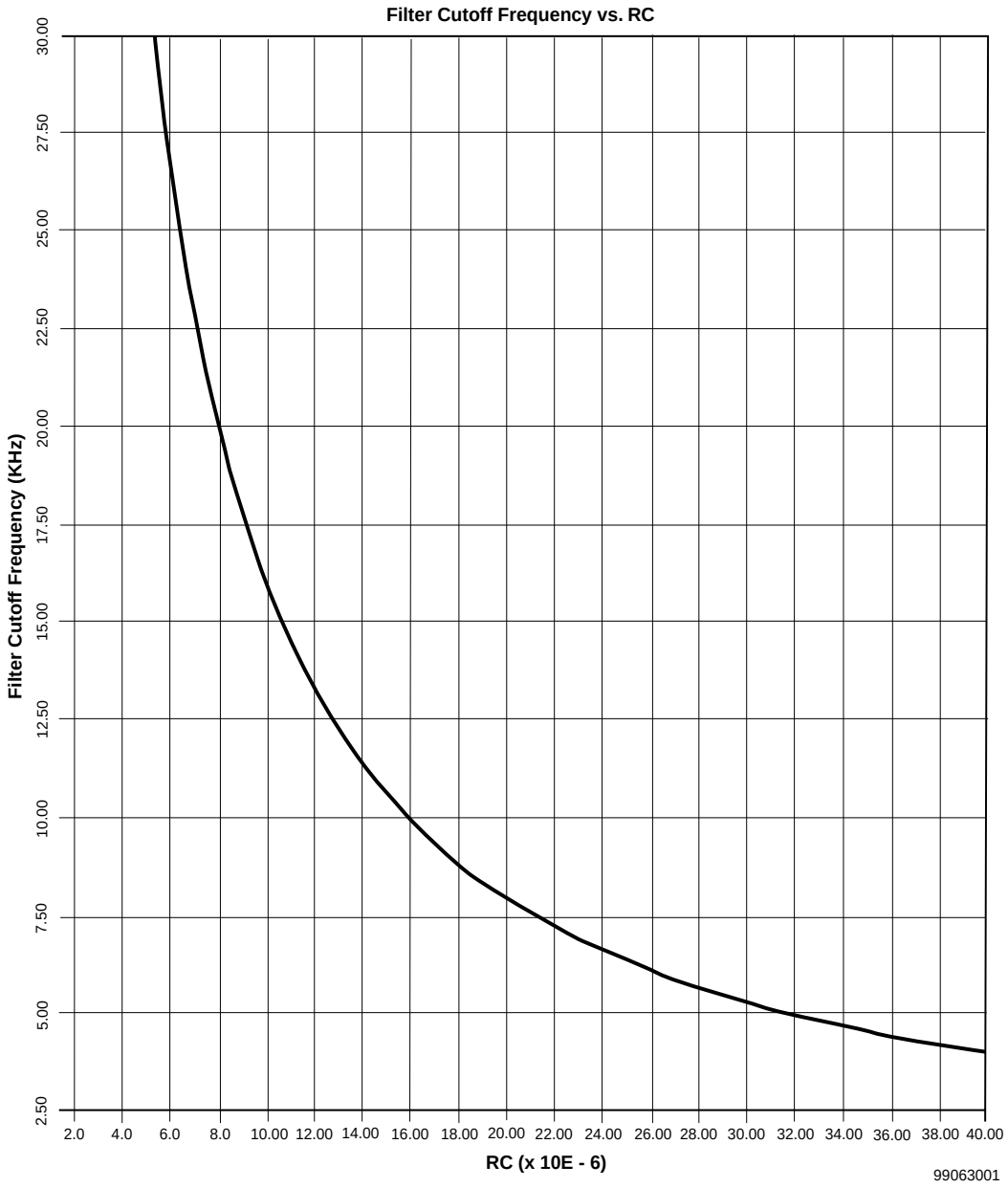


Figure 6: Filter Cutoff Frequency as a Function of RC

Sampling Rate

To resolve each DACin sample to the full precision of a Delta-Sigma DAC, the sample rate, i.e., the rate that DACin changes, must be less than or equal to $1/(2^{(MSBI+1)})$ of the CLK frequency. In some applications, such as a programmable voltage source, this is not an issue.

As DAC width and the desired sample frequency increases, it may not be possible to meet the above criterion. In practice, the sample rate sometimes exceeds $1/(2^{(MSBI+1)})$ of the CLK frequency. Though this compromises precision at higher frequencies, it is often possible to get satisfactory results. For example, the 16-bit audio DACs in a CD system would require a clock frequency of 2.9 GHz for full resolution of the highest frequencies. In practice, a much lower clock frequency is used. One reason this is acceptable is because the sensitivity of the human ear to noise becomes lower as the frequency increases.

This section lists some of the ways this DAC can be used in real-world applications.

- **Programmable Voltage Generator.** A variable voltage between 0V and V_{CCO} can be generated with a granularity determined by the bus width of DACin. In these applications, the voltage typically does not change quickly, so RC may be large to minimize noise.
- **Virtex V_{REF} Generator.** This is a specific application of a Programmable Voltage Generator. For some Virtex SelectIO™ receivers standards, a reference voltage is required for each bank of receivers. If a DAC is used to generate this voltage, V_{REF} can be dynamically changed to verify operating margins when conducting system tests. See XAPP133 for more information on SelectIO.
- **Waveform Generator.** Various analog waveforms, such as sine, sawtooth, triangle, etc., can be created by sequentially feeding the proper values to DACin. The values are normally pre-stored in

SRAM. Virtex Block SelectRAM+ is ideal for this purpose. See XAPP130 for more information on Block SelectRAM+.

- **Sound Generator.** Delta-Sigma DACs are widely used in sound reproduction, speech synthesis, etc. Since the analog output is changing rapidly, RC must be chosen with an acceptable trade-off between noise and frequency response.
- **RGB Color Generator.** Although Delta-Sigma DACs are too slow to directly generate Red-Green-Blue signals for a raster display, they are applicable in some color generation systems that do not operate in real time.
- **Analog to Digital Conversion.** This DAC may be used as a voltage reference in an ADC. See XAPP155 for a complete discussion of this application.

Since each DAC requires only a few CLBs and one output pin, many DACs may be implemented in even the smallest Virtex FPGA.

When the same V_{CCO} source is used for multiple DACs on the same FPGA, the analog outputs will track each other very accurately. This is important, for example, when three DACs are used to generate Red-Green-Blue color signals.

Table 2 provides some typical resistor and capacitor values for 6-, 8-, and 10-bit DACs. This table also lists the approximate maximum DAC clock frequency for a Virtex XCV300-6. The parts cost per DAC is based on the following:

- Virtex Slice $0.03
- Resistor $0.015
- Capacitor $0.07

Table 2: Typical DAC Implementations

No. bits	No. slices	Max. Freq.	Typical R	Typical C	Cost/DAC
6	4	223 MHz	3.3 KΩ	0.0047 µF	$0.205
8	5	219 MHz	3.3 KΩ	0.0047 µF	$0.235
10	6	215 MHz	6.8 KΩ	0.0047 µF	$0.265

Conclusion

The Delta-Sigma DAC is an example of how high speed FPGAs may be used in mixed-signal systems to minimize the number of components. The speed and density of the Virtex family of FPGAs makes them ideal for a wide range of analog signal generating and processing applications.

Bibliography

1. “Analog Devices Data Converter Reference Manual, Volume I”, 1992

2. "High Performance Stereo Bit-Stream DAC with Digital Filter", R. Finck, IEEE Transactions on Consumer Electronics, Vol. 35, No. 4, Nov. 1989.

Revision History

Date	Revision #	Activity
7/9/99	Version 1.0	Initial release
9/23/99	Version 1.1	Updated for Virtex-E designs