techXclusives

AnswerDatabase

Search Reset

Advanced Search
Answer Browser
Software Manuals
Feedback
Agents
Forums
Problem Solvers
WebCase
Site Map

Digitally Removing a DC Offset
(or "DSP Without Math?") - Part 2

By Ken Chapman
Staff Engineer, Core Applications - Xilinx UK

Introduction

Welcome to the second part of this TechXclusive! Were you were able to spot the first optimisations that I am going to make to the circuit?

After the optimisations are made to the algorithm, I will focus on how to make a smaller and more efficient version for audio telecom applications. This is pure hardware engineering, and I don't feel that any design is complete until it has an SRL16E in it! I suggest you read my previous TechXclusive articles about the SRL16E before you read the last section of this one.

Removing the Multiplier Logic

Potentially, the largest part of the circuit so far is the multiplier. Although you may have some dedicated multipliers to spare in future Virtex-II designs, it is likely that the multiplier is costing slices in most cases. For really cost-sensitive designs with Spartan-II, we should be doing everything possible to reduce size and stay in the smallest device.

It is actually very easy to remove the multiplier from this circuit, and hopefully you have already seen how it can be achieved. I left a rather big clue on the two response plots in the last article by specifying the coefficient values as k=1/32 and k=1/256. In both cases, these values are represented by numbers in which only one bit is active. I am going to adopt the second of these values as I consider that the low ripple is much more desirable than the response time -- especially as even 100ms is relatively short.

```
  77                    01001101
× 0.0039              ×       0.00000001
  0.3003                      0.01001101
```

Since the multiplication process only requires that the variable input be multiplied by "1," the output product is the same as the input. Clearly, there is no need for a real multiplier; the output product is the same value, and the bit width is the same as the variable input. All that is required is to apply the variable input value with the binary point reassigned to the correct position. Our complete DC offset removal circuit is then reduced to the following logic...

techXclusives

Visit the ***techXclusives*** home page for the complete listing of techX articles!

Forums

Got questions or comments about Ken Chapman's article? Be heard at the XCLUSIVES FORUM!

To join in on other Xilinx forums, visit our general forum area.

Related Info

Get an overview of the CORE Generator™.

Get more info on the Spartan-II™ FPGA.

Upcoming *techXclusives*

Peter Alfke presents his thoughts on "**Moving Data Across Asynchronous Clock Boundaries**." Coming on July 10!

Previous *techXclusives*

PETER ALFKE:
"Choices, Choices and Options"

"Printed Circuit Board Considerations"

KEN CHAPMAN:
"Cost Savings Using the SRL16E":
Part 1
Part 2
Part 3

"8X12 Does NOT Equal 12x8"

12x8

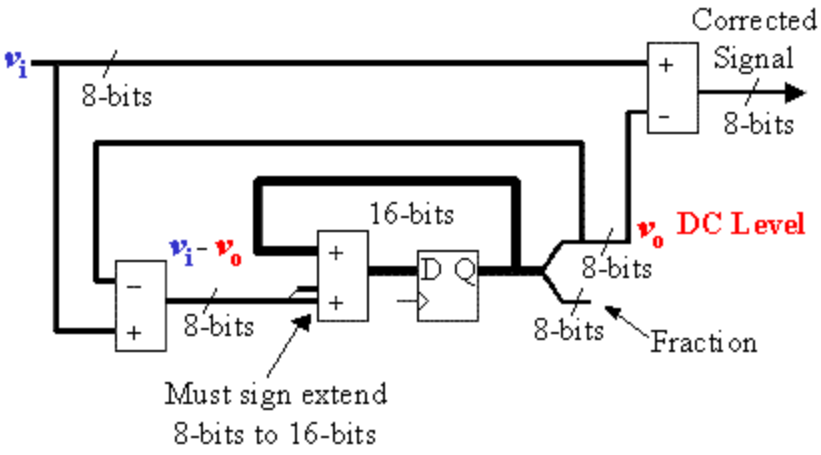
AUSTIN LESEA:
"Signal Integrity Tips and Tricks"

"Jitter"

ANDY MILLER:
"Colour Space Conversion"
Part 1
Part 2

AMIT DHIR:
"FPGAs Driving Voice-Data Convergence"
Part 1
Part 2

techXclusive home page



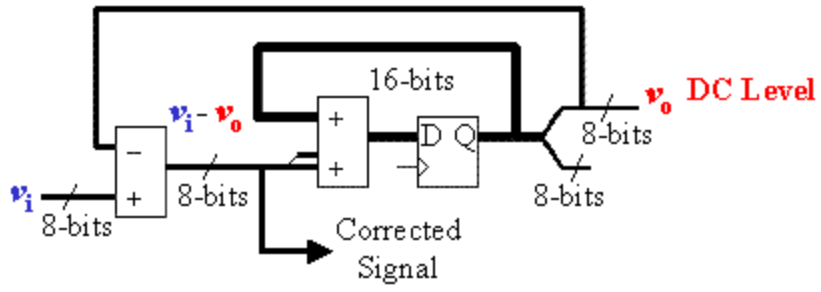
The circuit now consists only of an accumulator and two subtractors. Care is required when connecting the variable difference signal (vi - vo) to the accumulator input. To represent that the 8 bits are all fractional (to the right of the binary point), they are applied to the least significant byte of the 16-bit input. However, the upper byte must also be defined, and this must be achieved using sign extension (replicate the MSB of the 8-bit value a further 8 times to form either "00" or "FF" hexadecimal); this is so that the twos complement logic of the accumulator will correctly add both positive and negative values.

With this very small "k" value, it is obvious why the accumulation of the fractional parts (as well as the integer parts) must be performed.

Removing a subtractor

Having drawn out the DC level detector and the DC removing subtractor on the same diagram for the first time, it may appear more obvious that one of the subtractors is redundant. The corrected signal is the original signal with the DC level subtracted from it. This means that the output is the value vi - vo, which is the same as the difference signal being created by the subtractor within the DC detection circuit.

This means that the complete DC offset removal circuit can be reduced to just one accumulator and one subtractor...



Using the simple but accurate rule that a 2-bit add or subtract function fits into a "slice," then this circuit now only requires 12 "slices." For each additional bit of sample width, the subtractor and accumulator will each increase by 1-bit and, accordingly, increase the total size by 1 "slice." Therefore, with 16-bit input samples, the size would increase to 20 "slices." As a parallel circuit, this can also support a sample rate well in excess of 100MHz.

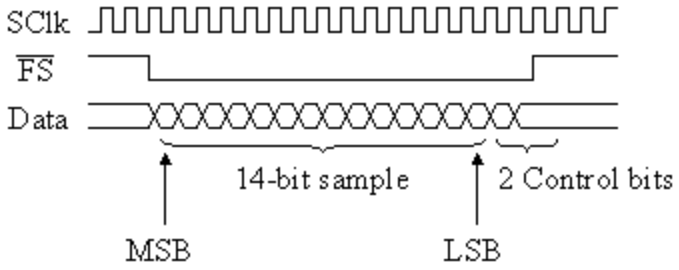
Low Sample Rate Applications

Although 12 and 20 "slices" may not sound like very much, it is still 6% to

10% of the smallest XC2S15 device, and this DC offset removal may well be seen as just a pre-process to the main function.

In a typical application, this DC offset removal may be required in a telephone conferencing facility. Each of the input lines (represented by digital samples) will ultimately be summed together within the system, and the contribution of multiple small DC offsets may have an adverse effect on the overall dynamics. The requirement for a DC offset removal circuit on each line input would soon see all those 6-10% units of a device mounting up to something significant.

Also typical of audio telecommunications, data samples are transmitted serially between units. These may be as packets within data frames, or even directly from the A/D converter. The Texas Instruments TLC320AC01C analogue Interface circuit (AIC) device uses a serial communications protocol with 14-bit A/D samples being transmitted with the most significant bit first as part of each 16-bit transfer...



To use the parallel implementation of the DC offset removal circuit, such serial data samples would need to be applied to a 14-bit shift register in order to read the sample in parallel. This requires an additional 7 "slices."