On the Dynamic Behavior of a Novel Digital-Only Sigma–Delta A/D Converter

Marcel Jacomet, Josef Goette, Venanz Zbinden, and Christian Narvaez
University of Applied Sciences Bern
Interdisciplinary Institute of Integrated Systems
CH-2501 Biel-Bienne, Switzerland

ABSTRACT
Conventional sigma-delta (ΣΔ) analog-to-digital (AD) converters are based on an analog ΣΔ modulator followed by a digital filter. In this paper we propose a new architecture of a first-order ΣΔ modulator that needs no active analog components. We call this ΣΔ modulator “digital-only,” and implement with it AD converters in FPGA’s or directly in the software of microprocessors. We here discuss aspects of the dynamic behavior of the proposed structure.

Categories and Subject Descriptors: B.7.1: Types and Design Styles.
General Terms: Algorithms, Design.
Keywords: A/D converter, FPGA, ΣΔ modulator.

1. INTRODUCTION
The emergence of powerful digital signal processing in CMOS VLSI technology creates the need for high-resolution AD converters that can be integrated in fabrication technologies optimized for digital circuits and systems. However, the same scaling of VLSI technology that makes possible the continuing and dramatic improvements in digital signal processing also severely constrains the dynamic range available for implementing the interfaces between the analog and the digital representation of signals. Oversampled AD converters based on ΣΔ modulation use sampling frequencies well above the Nyquist rate in order to exchange resolution in time for resolution in amplitude.

The ΣΔ conversion technique exists since many years. Technological advances make the technique practical and their use widespread. The converters are used in applications such as communication systems, consumer and professional audio, industrial weight scales, and precision measurement devices. The key feature of the ΣΔ converters is that they are the only low-cost conversion method which provides both high dynamic range and flexibility in converting low-bandwidth input signals [1].

The performance of digital signal processing systems is generally limited by the precision of the digital input signal which is achieved at the interface between the analog and the digital signal representation. ΣΔ modulation based AD conversion technology is a cost effective approach for high-resolution converters which can ultimately be integrated on digital signal-processor ASIC’s. Conventional ΣΔ converters employ ΣΔ modulators built with active analog components. Such analog integrated circuits have high design costs. The resulting oversampling AD converters then achieve a relatively high resolution with more than 12 bits and Nyquist rates above 50 kHz, and the development trends are towards even higher Nyquist rates. There exist, however, also application areas for certain sensor-signal AD conversions where lower resolutions and lower Nyquist rates are sufficient, see Section 5. Therefore, we propose a new architecture for a first-order ΣΔ modulator that uses as the only analog components a discrete resistor and a capacitor to realize the (lossy) analog integrator-part of the modulator. Because we need no active analog components to realize the ΣΔ modulator, we succeed in implementing our “digital-only” ΣΔ AD-converter directly in an FPGA. Alternatively, we may implement our approach in the software of a microprocessor.

The “digital-only” implementation of the proposed ΣΔ AD converter greatly enhances compatibility: Any digital FPGA or microprocessor can be programmed to implement the necessary ΣΔ-based AD conversion directly on chip or in software, respectively, without using any additional active analog components. An important advantage then is that we may design new applications with the high assurance of a first-right implementation, as digital circuit designs provide, thus eliminating the higher design risks of mixed analog-digital designs.

Although our new architecture has limited resolution due to theoretical limitations imposed by the used lossy integrator, we find—as already indicated above—a broad range of application areas, mainly in sensor-data analog-to-digital conversion. We believe that the simplicity of our approach outweighs resolutions limited to something in the range of 8 to 10 bits and Nyquist rates considerably lower than that of conventional ΣΔ AD converters.

2. THE SIGMA–DELTA MODULATOR
We consider the circuit diagram in Figure 1. The D-flip-flop (DFF) represents a threshold input/output port of the involved FPGA or the microprocessor, respectively. The DFF is clocked by the oversampling frequency \( f_0 \) and processes
an analog input according to

\[
q(v_c(nT_0)) = \begin{cases} 
  V_{dd} \text{ if } v_c(nT_0) < V_{dd}/2, \\
  0 \text{ if } v_c(nT_0) \geq V_{dd}/2, 
\end{cases}
\]

where \( T_0 \) is the oversampling interval, \( T_0 = 1/f_0 \). Note that this DFF acts as quantizer in our \( \Sigma \Delta \) converter and that it prepares the negative feedback by a sign change.

![Figure 1: Schematic of the proposed circuit.](image)

The current source \( i_s \) represents the analog input signal to be converted to digital form. In our main application this current signal comes from a photo-diode, but any sensor whose output signal is a current (or can be converted to a current) can be interfaced by our circuit to a digital hardware.

The input current signal \( i_s \) and the feedback current signal \( i_f \) (with logically negative sign) are summed in node \( \oplus \) (we assume that there is no current flow into the DFF). The capacitance \( C \) acts as an integrator for the total current and outputs the voltage signal \( v_c \), which is the input to the clocked quantizer-block DFF.

We may model the operation of the circuit in Figure 1 for left-open right-closed time intervals \((nT_0,(n+1)T_0)\) by

\[
\dot{v}_c = \left( -\frac{1}{RC} \right) v_c + \left( \frac{1}{RC} \right) q(v_c[n]) + v_s, \tag{2}
\]

where we have defined the equivalent input voltage \( v_s \) by

\[
v_s = R \cdot i_s \quad \text{and where } v_c[n] = \text{the capacitor voltage at the end of the previous interval } (n-1)T_0, \text{ and } v_c(nT_0). \]

To obtain a circuit description in the form used by other authors, we find it useful to transform the state variable \( v_c(t) \) in (2) into a new state variable \( u(t) \) according to

\[
u(t) = \frac{2}{V_{dd}} \cdot v_c(t). \]

In this new state the circuit is described by

\[
\dot{u} = \left( -\frac{1}{RC} \right) u + \left( \frac{1}{RC} \right) (x - \text{sgn}(u[n])), \tag{3}
\]

where \( \text{sgn}(\cdot) \) denotes the signum function, and the new input \( x(t) \) is related to \( v_c(t) \) by \( z \doteq -1/(V_{dd}/2) \cdot v_s \). The output of the transformed circuit is \( \text{sgn}(u[n]) \) and is a sequence of numbers “+1” and “-1” which may be symbolically represented by “bits” 1 (+1) and 0 (-1). The output of our proposed circuit, \( q(v_c[n]) \), is a sequence of numbers \( V_{dd} \) and 0 and is related to the output of the transformed circuit by

\[
q(v_c[n]) = (\text{sgn}(u[n]) + 1) \cdot V_{dd}/2.
\]

Because we are only interested into the output sequence \( \text{sgn}(u[n]) \) and the sequence of states at the end of the left-open time intervals \((nT_0,(n+1)T_0)\), \( u(t = (n+1)T_0) = \text{sgn}(u[n+1]), \) we solve (3) for these time intervals \((nT_0,(n+1)T_0)\) and evaluate at \( t = (n+1)T_0 \). Superimposing the free response due to initial conditions \( u(nT_0^+) = \text{sgn}(u[n]) \), and the forced response due to the input \( x(t) = \text{sgn}(u[n]), \) \( t \in (nT_0,(n+1)T_0) \), we obtain, after evaluating at \( t = (n+1)T_0 \), the following first-order difference equation:

\[
u(n+1) = p \cdot u(n) + (1-p) \left( x[n+1] - \text{sgn}(u[n]) \right), \tag{4}
\]

Here the feedback gain \( p \) is given in terms of the parameters of our circuit, \( R \) and \( C \), and the oversampling interval \( T_0 \) by \( p = \exp\left(-T_0/(RC)\right) \), and the input gain \( g \) is, as indicated, related to \( p \) by \( g = 1-p \).

The input \( x[n+1] \) in (4) is related to the true input \( x(t) \) in the differential equation (3) as follows: If \( x(t) \) is a piecewise constant signal, that is, it is a constant inside the intervals \((nT_0,(n+1)T_0)\), \( x[n+1] \) denotes these constant values and the description (4) is an exact discrete equivalent of (3). If \( x(t) \) is not a piecewise constant signal, we may assume in practice that it is slowly time-varying with respect to the oversampling-interval duration \( T_0 \) and that it is bounded. From the mean-value theorem of the integral calculus we know then that \( x[n+1] \) is a sample of \( x(t) \) at some time instant \( \xi \in (nT_0,(n+1)T_0) \). Likewise we may interpret \( x[n+1] \) as an average value of \( x(t) \) in \((nT_0,(n+1)T_0)\) (the \( RC \) combination in our circuit filters the signal \( x(t) \)).

We note that the difference equation (4) is the single-loop lossy \( \Sigma \Delta \) system analyzed by other authors such as [2] and [3].

Compared to the sigma-delta modulator analyzed in [2], our modulator has the additional constraints:

\[ p = \exp\left(-\frac{T_0}{RC}\right), \quad g = 1-p, \]

that is, the loss \( p \) of the discrete-time equivalent modulator in (4) is given by the design of the analog realization in Figure 1—the time constant \( RC \)—and by the selected oversampling frequency \( f_0 = 1/T_0 \); and the input gain \( g \) of the discrete-time equivalent modulator in (4) is fixed once the feedback gain \( p \) is selected. Intuitively, it is clear that designing a feedback gain \( p \) as close to 1 as possible is preferable, because \( p \) approaching 1 means approaching the ideal non-lossy integrator situation; however, we see that in the limit \( p \to 1 \) the input gain \( g \) to our—for \( p = 1 \) ideal—accumulator becomes zero, killing our approach. We next discuss some basic theoretical limitations of our lossy \( \Sigma \Delta \) modulator circuit, that is, we discuss the effects of \( p < 1, \quad g = 1-p \).

Feely and Chua discuss in their paper [2] the lossy sigma-delta modulator (4) for DC input signals: For modulators with ideal integrator and constant input, the output of the

\[1\] We thus see that our architecture with only passive analog components is equivalent to any active analog first-order \( \Sigma \Delta \) modulator; the digital input to the FPGA or microprocessor acts as quantizer and is the hidden active analog component.
system—when averaged over a “long enough” time period—equals the input; for integrators with leak, however, they show that the plot of input versus average-output is no longer linear but has a fractal “staircase” structure, the devil’s staircase. Therefore, the resolution of the modulator is limited by the width and the displacement of these steps.

Figures 7 and 8 in the paper [2] show the devil’s staircase of the constant-input $x$ to average-output map of a lossy sigma-delta modulator with $p = 0.8$ for $-1 \leq x \leq 1$. We have reproduced such a devil’s staircase by the curve “devil’s staircase” in Figure 3 together with the characteristics of a “uniform quantization” of the interval $-1 \leq x \leq 1$ with $b = 4$ bits. We see then that the widest step of the devil’s staircase appears in the vicinity of zero inputs. Formula (6) of [2] gives for the width of this widest step a value of $2(1-p)/(1+p)$. 

If the final goal of our complete sigma-delta converter is to obtain a uniform quantization of the interval $-1 \leq x \leq 1$ with $b$ bits, as Figure 3 shows for $b = 4$ bits, we argue that the widest step-width must be smaller than the aimed-at resolution. We obtain

$$p > \frac{2^b - 3}{2^b - 1} \equiv p_{\text{min}}$$  \hspace{1cm} (5)$$

for lower limits of the allowable loss $p$. Figure 2 shows the minimally needed loss values that are, in the best case, necessary to achieve a uniform quantization with $b$ bits; “best case” means that we use an averaging of the sigma-delta modulator output bit-stream over a “long enough” time period, that is, that we use a sufficiently powerful decoding in the subsequent digital processing stage.

In the example shown in Figure 3 we also see the mentioned displacements of the steps in the devil’s staircase with respect to the “ideal $y = x$” characteristic. We argue that the errors introduced by these displacements can be removed by employing a pre-warping in the decoder, see [4]. So the displacements of the steps from the “ideal $y = x$” positions are the lesser problem, but no decoder can remove the errors due to non-zero step widths in the devil’s staircase—our sigma-delta modulator is just not able to distinguish inputs lying in intervals of same height of the devil’s staircase. The accompanying loss of resolution is a non-linear function of the input, and the greatest loss of resolution occurs in the neighborhood of the rational numbers with lowest denominators, see also the discussion in [2].

As far as we have not yet discussed the influence of the non-unity gain $g$ of our sigma-delta modulator. Reference [2] notes that a non-unity gain $g$ has only an influence on how quickly the modulator’s output $\text{sgn}(u[n])$ enters periodicity: $\text{sgn}(u[n])$ latches onto its periodic cycle as soon as the state $u[n]$ enters the region $[g(x-1), g(x+1)]$. For our $\Sigma\Delta$ modulator with $g = 1 - p$ we now argue as follows: If we like to obtain a better resolution, we must realize a larger $p$ according to Figure 2. Such a larger $p$ consequently gives a smaller $g = 1-p$, and the latching interval $[g(x-1), g(x+1)]$ becomes smaller. Therefore, we expect that it takes more

$\text{Note that we use this low 4 bit resolution to make visible the characteristics of the resulting staircase; as we demonstrate below, our circuit achieves a much higher resolution.}$

$\text{The period index } N \text{ in formula (6) of [2] has to be taken as } N = 2, \text{ because the widest step is that for a limit cycle “-1, 1” having period } N = 2 \text{ and giving an average output of 0.}$

Figures 7 and 8 in the paper [2] show the devil’s staircase of the constant-input $x$ to average-output map of a lossy sigma-delta modulator with $p = 0.8$ for $-1 \leq x \leq 1$. We have reproduced such a devil’s staircase by the curve “devil’s staircase” in Figure 3 together with the characteristics of a “uniform quantization” of the interval $-1 \leq x \leq 1$ with $b = 4$ bits. We see then that the widest step of the devil’s staircase appears in the vicinity of zero inputs. Formula (6) of [2] gives for the width of this widest step a value of $2(1-p)/(1+p)$.

If the final goal of our complete sigma-delta converter is to obtain a uniform quantization of the interval $-1 \leq x \leq 1$ with $b$ bits, as Figure 3 shows for $b = 4$ bits, we argue that the widest step-width must be smaller than the aimed-at resolution. We obtain

$$p > \frac{2^b - 3}{2^b - 1} \equiv p_{\text{min}}$$  \hspace{1cm} (5)$$

for lower limits of the allowable loss $p$. Figure 2 shows the minimally needed loss values that are, in the best case, necessary to achieve a uniform quantization with $b$ bits; “best case” means that we use an averaging of the sigma-delta modulator output bit-stream over a “long enough” time period, that is, that we use a sufficiently powerful decoding in the subsequent digital processing stage.

In the example shown in Figure 3 we also see the mentioned displacements of the steps in the devil’s staircase with respect to the “ideal $y = x$” characteristic. We argue that the errors introduced by these displacements can be removed by employing a pre-warping in the decoder, see [4]. So the displacements of the steps from the “ideal $y = x$” positions are the lesser problem, but no decoder can remove the errors due to non-zero step widths in the devil’s staircase—our sigma-delta modulator is just not able to distinguish inputs lying in intervals of same height of the devil’s staircase. The accompanying loss of resolution is a non-linear function of the input, and the greatest loss of resolution occurs in the neighborhood of the rational numbers with lowest denominators, see also the discussion in [2].

As far as we have not yet discussed the influence of the non-unity gain $g$ of our sigma-delta modulator. Reference [2] notes that a non-unity gain $g$ has only an influence on how quickly the modulator’s output $\text{sgn}(u[n])$ enters periodicity: $\text{sgn}(u[n])$ latches onto its periodic cycle as soon as the state $u[n]$ enters the region $[g(x-1), g(x+1)]$. For our $\Sigma\Delta$ modulator with $g = 1 - p$ we now argue as follows: If we like to obtain a better resolution, we must realize a larger $p$ according to Figure 2. Such a larger $p$ consequently gives a smaller $g = 1-p$, and the latching interval $[g(x-1), g(x+1)]$ becomes smaller. Therefore, we expect that it takes more

$\text{Note that we use this low 4 bit resolution to make visible the characteristics of the resulting staircase; as we demonstrate below, our circuit achieves a much higher resolution.}$

$\text{The period index } N \text{ in formula (6) of [2] has to be taken as } N = 2, \text{ because the widest step is that for a limit cycle “-1, 1” having period } N = 2 \text{ and giving an average output of 0.}$

$\Sigma\Delta$ modulator clock cycles for $\text{sgn}(u[n])$—the $\Sigma\Delta$ modulator output bit-stream—to enter periodicity.

Because we have not yet found a theoretical result specifying the number of sigma-delta modulator clock cycles needed for $u[n]$ to enter into the interval $[g(x-1), g(x+1)]$, we have programmed an exhaustive search over possible $\text{DC}$ input values $x[n+1] = \text{const} \equiv x$ and initial conditions $u[0]$, determining the maximum number of clock cycles needed to enter said interval. In the rectangle ($-1 \leq x \leq 1$) $\times$ ($-1 \leq u[0] \leq 1$) we find for $p = p_{\text{min}}$ according to (5) and $g = 1 - p$ the results compiled in Table 1.

<table>
<thead>
<tr>
<th>$b$</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>10</td>
<td>31</td>
<td>86</td>
<td>218</td>
<td>528</td>
<td>1237</td>
<td>2834</td>
</tr>
</tbody>
</table>

Table 1: The maximum number of clock cycles needed to enter periodicity for target bit resolutions $b$.

Although we have done experiments with various resolutions of the rectangle $(-1 \leq x \leq 1) \times (-1 \leq u[0] \leq 1)$, and we have always obtained the values given in Table 1, these experiments are of course no proof for correctness. Nevertheless, we are convinced that the obtained figures for the number of clock cycles needed for $\text{sgn}(u[n])$ to enter periodicity are, at least, good indications.

An idea to reduce the maximum number of clock cycles needed to enter the latching interval is to use “dither noise.” We add to the DC input value to be converted random numbers, $x \mapsto x + u[n]$ with $u[n]$ a low-variance noise sequence, and hope that the bit sequence $\text{sgn}(u[n])$ sooner enters periodicity. A consecutive question then is whether $\text{sgn}(u[n])$ stays, in the presence of noise, on its periodic cycle. We have found, however, that “dither noise” does not help; on the contrary, it increases the maximum number of needed clock cycles, with larger noise levels leading to stronger increases.
3. SIMULATION RESULTS

To verify that our arguments above are correct for the mixed analog-digital circuit in Figure 1, we have performed simulations with MATLAB/SIMULINK.

Concerning the minimally needed loss-values \( p_{\text{min}} \) for a target resolution, we find by our mixed-signal simulations that our circuit indeed achieves the expected resolutions. As an example, the curve “simulated staircase” in Figure 3 shows the obtained staircase for \( p = p_{\text{min}} \) (4 bits) = 0.8667, \( g = 1 - p \). Simulations for bits \( b \in \{5, 6, 7, 8, 9, 10\} \) with values of \( p_{\text{min}} \) according to Figure 2 yield corresponding results. As the steps in the respective staircases become very small and are hardly visible, we do not present here the respective graphs, see [4].

Concerning the latching onto limit cycles, our simulations of the mixed-analog-digital circuit in Figure 1 re-obtain the worst-case values given in Table 1. Unfortunately, these worst-case values are quite high: For a ten bit resolution, it is true that the notion “needed number of clock cycles to enter periodicity” is not well defined for time-varying signals to be converted. To underpin this claim, we have performed mixed-signal simulations with a sinusoidal input-signal applied to our \( \Sigma \Delta \) modulator, and have traced the sequences of input values and initial conditions. These two sequences then define a locus in the plane ”DC input/initial condition” of Figure 4. Figure 5 shows the corresponding path that would be taken in the landscape of Figure 4. The simulation results indicate that for time-varying input signals the needed number of clock cycles to enter periodicity is in fact very low: \(^5\) For our sinusoidal input-signal simulations we see from Figure 5 that said path stays, after an initial downhill movement due to intentionally used worst-case initial conditions for the complete simulation, always in the bottom of the valley of the landscape in Figure 4.

4. HARDWARE MEASUREMENTS

To test and to characterize \( \Sigma \Delta \) modulators built according to our ideas in hardware, we have built a test environment for oversampling \( \Sigma \Delta \) converters. Figure 6 shows its block diagram, and Figure 7 is a photograph of its components. Whereas this test environment lets the modulator part of oversampling converters run in real time, associated decoding algorithms may be tested and characterized by an off-line post-processing on a host computer. This off-line decoding facility allows easy experimentation with alternative decoding procedures proposed in the literature, but applies to bit-streams from modulators running in real time on true analog hardware components. The \( \Sigma \Delta \) modulator according

\(^5\)It is true that the notion “needed number of clock cycles to enter periodicity” is not well defined for time-varying input signals as it is for DC input signals. We here cross-compare only to gain an idea of what happens in the much more complicated situation of time varying inputs.
to Figure 1 runs, as mentioned, in real time and is driven by a test current-signal source. This current source is controlled by a software running on a real-time dSpace system, [5], allowing us to freely generate signal forms of the analog current-signal to be converted. The bit stream of the \( \Sigma \Delta \) modulator under test is clocked into a Gecko system, [6], where it is converted to 32-bit wide packets that are in turn read-back to the dSpace system. Presently, the dSpace systems run at clock frequencies of up to 70 kHz, and the \( \Sigma \Delta \) modulator can run at 500 kHz. \( \Sigma \Delta \) modulator bit-streams collected on the dSpace system can be read-back to the host computer for later off-line processing, but the host computer can program the dSpace software for test-signal generation as well.

First measurements on our test environment concern DC input signals; they confirm our theoretical and simulation results. The curve measured staircase for \( P_{\text{min}}(4 \text{ bits}) \) in Figure 8 very well compares to the curve theoretical devil’s staircase \( P_{\text{min}}(4 \text{ bits}) \). For comparison we also supply the curve measured staircase \( P_{\text{min}}(10 \text{ bits}) \).

Concerning the dynamic performance of our novel \( \Sigma \Delta \) modulator, we have setup an overall experiment—including the decoding stage—as follows: On the MATLAB level we generate a sinusoidal signal \( y_0[n] \) with frequency \( f_{\text{sin}} \) and sampling frequency \( f_0 = 500 \text{ kHz} \). A resampled version of this signal \( y_0[n] \) is sent to the digital-to-analog converter of the dSpace system running at a sampling frequency of 70 kHz. This dSpace output signal controls our current-source generating the input signal for the \( \Sigma \Delta \) modulator under test. The \( \Sigma \Delta \) modulator is clocked with 500 kHz to generate its bit stream. Our test-environment reads the generated bit-stream and passes it to the off-line decoding stage realized in MATLAB. Here we presently only use a sinc\(^2\) filter. The resulting sample sequence \( y[n] \) is next compared to the original signal \( y_0[n] \), see Figure 9.\(^6\)

For a numerical comparison we compute the error sequence \( y[n] = y_0[n] - y[n] \), and in turn its power \( P_\Delta \), to finally obtain the signal-to-error ratio \( 10 \log_{10}(P_0/P_\Delta) \), where \( P_0 \) is the power of the original signal \( y_0[n] \).\(^7\) We find

\(^6\)Note that we have not yet applied a sinc\(^2\) correction to the output signal \( y[n] \), which explains the reduced amplitude of the output signal as well as its shift-in-time in Figure 9.

\(^7\)We have carried-out the discussed power computations in
a signal-to-error ratio of approximately 45 dB. Computing the signal-to-error ratio with respect to an ideal reference signal means that we assume an ideal output signal from the dSPACE analog output. Since the dSPACE digital-to-analog converter has 14 bits resolution, the output signal is not an ideal sinusoid. The error due to this non-ideality is, by the present experiment, attributed to our ΣΔ converter. Furthermore, non-idealities of the current source in our test environment likewise introduce errors that are attributed to our ΣΔ converter. To attribute all the errors due to the test environment to the device under test is incorrect. Therefore, the obtained results are worst-case-results for our ΣΔ converter. First guesses indicate that our test environment adds about 6 dB to the signal-to-error ratio, leading to the conclusion that the obtained measurements correspond to a resolution of the expected 8 bits. The described experiment has been repeated for various input signal frequencies \( f_{\text{sin}} \), yielding results of comparable performance.

5. CONCLUSIONS

We have proposed a novel architecture for a first-order ΣΔ modulator that needs no active analog components. We have shown here that for DC and sinusoidal input signals our approach achieves resolutions of more than 8 bits. We have obtained our measurement results on a versatile test environment for oversampling converters. A detailed characterization of the errors additionally introduced by the test environment remains to be investigated. An improved test environment for higher oversampling frequencies \( f_0 \) will next be built solely based on the GECKO system-on-chip environment. Furthermore, more enhanced decoding algorithms will be combined with our ΣΔ modulator to find out preferable combinations.

With the presented “digital-only” ΣΔ modulator we implement AD converters in FPGAs and in the software of microprocessors: We have realized our novel “digital-only” ΣΔ modulator in AD converters used in an e-commerce product. This application involves an optical data-channel with four photo-diode receivers. Four oversampling AD converters based on our new “digital-only” ΣΔ modulator run concurrently on one ARM7TDMI microprocessor. Each AD converter has a resolution of 7 bits and a Nyquist rate of 1 kHz.

6. REFERENCES