A Novel Speculative Pseudo-Parallel \Delta\Sigma Modulator

32rd NORCHIP 2014

Citation for the published paper:
Johansson, J. ; Svensson, L. (2014) "A Novel Speculative Pseudo-Parallel \Delta\Sigma Modulator". 32rd NORCHIP 2014 pp. Article number 7004712.

http://dx.doi.org/10.1109/NORCHIP.2014.7004712

Downloaded from: http://publications.lib.chalmers.se/publication/204262

A Novel Speculative Pseudo-Parallel $\Delta\Sigma$ Modulator

Jesper Johansson and Lars Svensson
Dept. of Computer Science and Engineering, Chalmers University of Technology, Sweden
jjesper@student.chalmers.se, larssv@chalmers.se

***Abstract*—We present a novel speculative pseudo-parallel $\Delta\Sigma$ modulator structure, which almost halves the logic depth of the critical path in the pseudo-parallel Hatami structure. Following Hatami, our modulator calculates a block of n consecutive output bits in parallel, and then employs a parallel-serial interface to output the bits at n times the modulator clock frequency. We circumvent the block-to-block dependence, which limits the clock speed of the Hatami structure, by speculatively calculating the outputs based on all possible output values of the previous block, and then selecting the correct one. We present cost and performance estimates for an initial implementation of the modulator, synthesized towards an FPGA and an ASIC technology.**

I. Introduction

Delta-sigma ($\Delta\Sigma$) modulators are highly useful for improving the SNDR of an N-bit quantization operation beyond the standard level of $(6.02 \cdot N + 1.76)$ dB for a full-range sinewave input. In a $\Delta\Sigma$ modulator, a feedback-loop filter creates separate transfer functions for the converted signal and the quantization noise added by the core quantizer, reducing the relative noise level in the band of interest. The noise-shaping feedback loop allows the designer to keep the core quantizer resolution low at the cost of a higher sample rate and processing speed. In particular, a core quantizer resolution of only one bit eliminates many linearity concerns [1].

As illustrated in Figure 1, a $\Delta\Sigma$ modulator works in conjunction with a frequency-selective noise-suppression filter. For example, a baseband, or lowpass, $\Delta\Sigma$ modulator pushes quantization noise to high frequencies, where it may be removed with a lowpass filter chosen to only marginally affect the signal of interest.

The noise-shaping principle thus relies on faster-than-Nyquist sampling. The ratio of the sample rate to the Nyquist sample rate (the *oversample ratio*, or OSR), is a major design parameter for $\Delta\Sigma$ modulators. A high OSR value facilitates a high SNDR value; but a high OSR value with a large signal bandwidth requires higher sampling and processing rates, which are expensive or in extreme cases even unreachable. The OSR requirements may be somewhat reduced through careful design of a higher-order $\Delta\Sigma$ modulator loop filter, but such higher-order modulators have stability issues [2].

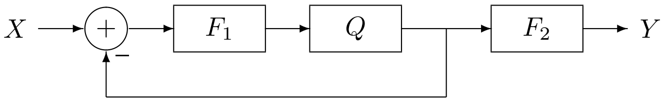

Figure 1: A schematic view of a $\Delta\Sigma$ modulator system, showing the input signal X, the loop filter F_1, the core quantizer Q, the noise-suppression filter F_2, and the output signal Y.

The $\Delta\Sigma$ principles are useful in ADC design as well as in all-digital sample-stream generation. The work presented here mainly concerns the generation of high-speed, low-resolution digital signals where the noise suppression is carried out in the analog domain.

II. Parallel modulator approaches

As the OSR requirements may make a certain performance level unreachable for a given core quantizer resolution and signal bandwidth, considerable effort has been spent on approaches to reduce these requirements, and thereby the clock frequency requirements, through different variations of parallel processing.

A summary and overview of several methods for parallel $\Delta\Sigma$ AD conversion is given by Eshraghi [3]. A conceptually simple example is frequency-band decomposition [4]. Here, N bandpass $\Delta\Sigma$ modulators with abutting frequency bands each handle $1/N$ of the entire band of interest; ideally, the OSR requirement will then apply to the sub-band width rather than to the entire band, for a gain of a factor of N. The outputs of the sub-band modulators must then be combined digitally after noise-suppression filtering.

Other criteria than pure frequency may also be used to split the signal information across several paths [5]. For example, in time interleaving [3], several $\Delta\Sigma$ modulator channels produce alternate output samples. Such methods require carefully selected and matched noise-suppression filters for the separate paths and are not immediately applicable to an analog-filtering case.

Recent work by Hatami [6] proposes to unroll the $\Delta\Sigma$ feedback loop in time, in order to compute several output samples during each loop iteration. In contrast to the methods mentioned above, the Hatami method does not truly parallelize the problem, but rather exposes the sequential operations to rearrangement and optimization across computations of several samples. As there is still only one modulator and one sample stream, the complications of combining signal paths and matching noise-suppression filters do not arise.

Figure 2 shows a block diagram of a Hatami "pseudo-parallel" modulator (henceforth: P$\Delta\Sigma$M) unrolled by a factor of four. Each processing element (PE$_i$) in Figure 2 carries out one iteration of the operations of a loop filter, given the input value $x[n]$ and the filter state variables. The filter state

978-1-4799-6890-9/14/$31.00 ©2014 IEEE

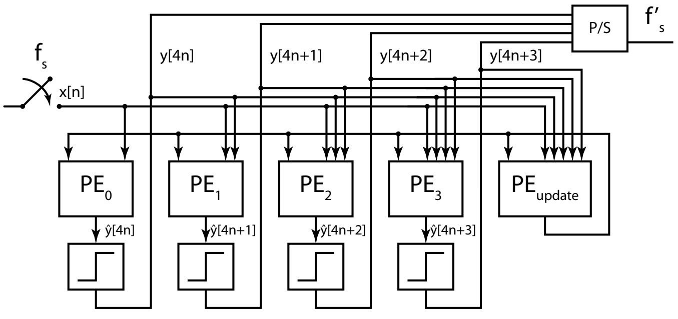

Figure 2: Block diagram of a four-way-unrolled $\Delta\Sigma$ modulator (P$\Delta\Sigma$M) as proposed by Hatami [6].

variables are updated by the final processing element and fed back to all the PE$_i$s to prepare for the next cycle.

As multiple output samples are produced in a single cycle, the effective OSR of the P$\Delta\Sigma$M is extended by the *unrolling factor UF* (four, in the case shown in the figure). However, Figure 2 also indicates that the critical path of the structure passes from the $x[n]$ input through all PE$_i$s and quantizers in turn, before ending at the parallel/serial converter at the output.

The critical path determines the maximum reachable clock frequency; therefore, for a given signal bandwidth and unrolling factor, the achievable OSR; and therefore, for a given core quantizer resolution, the achievable SNDR. Thus, a critical-path reduction may extend the bandwidth and/or SNDR for the P$\Delta\Sigma$M approach.

III. Speculative modulator architecture

In the work presented here, we use *speculation* to alleviate the connection between the modulator critical path and the clock frequency. We thus eliminate waiting for the result of the previous iteration, at the expense of having to duplicate the modulator for each possible set of output bits calculated in the previous iteration of the modulator. Thus, our speculative modulator architecture allows increased clock speed and OSR at a cost of a significant increase in hardware.

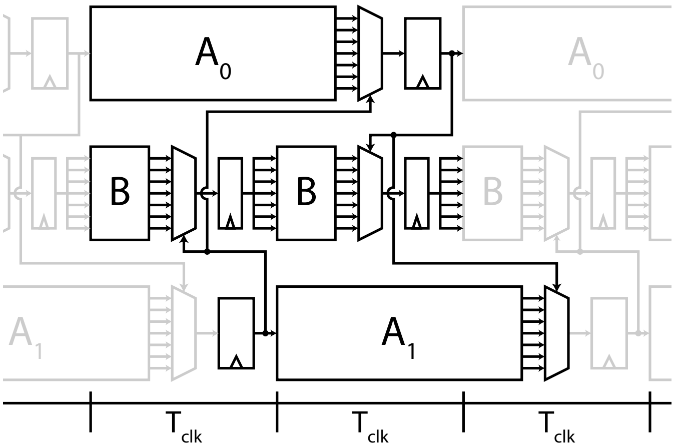

Figure 3: Graphical representation of the different blocks in the speculative modulator array and their interaction in time. The A$_0$ and A$_1$ block arrays calculate the output bitstream; the B block array updates the internal states. There is only one physical instantiation each of A$_0$, A$_1$, and B.

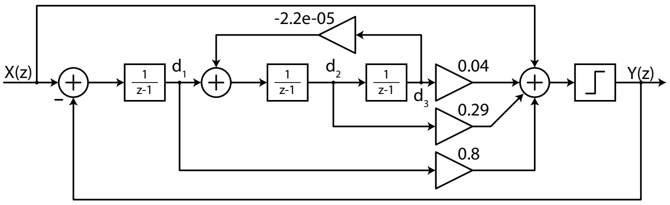

Figure 4: A third-order LP $\Delta\Sigma$ modulator used as a design example.

In the following description, we assume an unrolling factor of four, as illustrated in Figure 2; other factors are clearly possible.

The speculative version of the P$\Delta\Sigma$M (henceforth: SP$\Delta\Sigma$M) comprises two different block types. Block type A is tasked with calculating four new output bits using the internal states of the modulator $d_1[n], d_2[n], d_3[n]$ (see Figure 4), an input value $x[n]$, and a set of speculative previous output bits $y_s[n]$ (see Figures 2 and 3). Sixteen such blocks (2^{UF}) are needed to account for all possible values of $y_s[n]$; these are shown as a block array in Figure 3.

Block type B is responsible for updating the internal states of the modulator from one iteration to the next, using only the previous internal states $d_1[n-1], d_2[n-1], d_3[n-1]$ and the previous input value $x[n-1]$. Again, sixteen blocks are needed to cover all combinations of previous output bits; in the P$\Delta\Sigma$M case shown in Figure 2, the output bits enter into the calculations for the updated states, whereas in the SP$\Delta\Sigma$M case, the bit values are known at design time (although different for each block in the array).

The full modulator structure thus consists of three block arrays: two arrays, each consisting of sixteen type A blocks, operating in leapfrog fashion; and one array of sixteen type B blocks.

The modulator operates as follows. As one modulator array calculates the valid output bit $y[n]$ of an input value $x[n]$, the other modulator array begins calculation of the next output bit $y[n+1]$ based on the next input value $x[n+1]$. Since each array consists of 16 modulators, all possible results of $y[n]$ can be assumed in the array that calculates $y[n+1]$. When the first array finishes its calculation of $y[n]$, the modulator which will produce the correct $y[n+1]$ becomes known as well. This modulator will in turn select the correct modulator in the first array when it has finished its calculation of $y[n+1]$, etc.

The block-B array operates at f_{clk}. Importantly, its critical path determines the possible speed of the entire SP$\Delta\Sigma$M, since speculation allows each of the block-A arrays to operate during two clock cycles, and thus effectively at $f_{clk}/2$ (see Figure 3).

IV. Experimental implementation

Our initial validation of the SP$\Delta\Sigma$M principle is based on a third-order modulator also used by Hatami, with $UF = 4$.

Figure 4 shows this modulator before unrolling. The three integrators provide a third-order high-pass noise transfer function and thus a high SNR at low frequencies. In our case, the driving system example is a purely digital RF signal generator for wireless communications. In order to generate a high-pass signal, the output of the original LP modulator was mirrored in the frequency domain by inverting every other bit of the output signal. The resulting signal corresponds to the baseband signal fed into the modulator, transposed and mirrored at the Nyquist frequency. Figure 5 shows MATLAB simulation results for a single-frequency test signal, before and after a time-continuous reconstruction filter.

Several minor implementation optimizations were applied to both modulators. First, the constant multiplications shown in Figure 4 were implemented as single-addition shift-and-add operations, at a small pole/zero-placement accuracy cost. Second, for each PE, the possible outputs were pre-calculated, and the correct output result was selected on the arrival of the result bit of the previous PE.

Careful inspection of Figure 4 reveals that computation of the integrator input signals (implemented in blocks of type B in SP$\Delta\Sigma$M) involve fewer operations than computation of the input for the quantizer (implemented in blocks of type A), and that the quantizer input computation has a higher adder depth. (The actual blocks aggregate computations for UF iterations of the original loop.) This observation provides an intuitive explanation to the difference in size and delay of the block types, and indicates what kind of $\Delta\Sigma$ loop structures would tend to benefit from the SP$\Delta\Sigma$M treatment.

V. Results

A. FPGA implementation

We synthesized our SP$\Delta\Sigma$M design as well as a Hatami-style P$\Delta\Sigma$M design for a Xilinx Virtex 6 LX550-T with a speed grade of –2 [7], for which a development board was available. This FPGA includes high-speed parallel-in-serial-out (PISO) transceivers which make it possible to generate bitstreams at speeds significantly higher than the fabric speed.

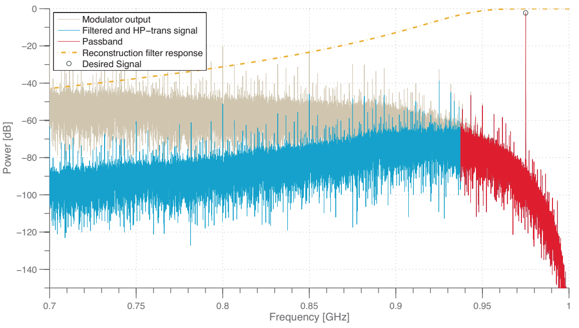

Figure 5: Simulated spectra of the chosen third-order LP $\Delta\Sigma$ modulator, working at four times the Nyquist rate and with $UF = 4$, for an effective OSR of 16. The sampling frequency is 2 GHz. The output signal is mirrored in the frequency domain by inverting every other bit.

	delay	delay	1/delay	LUTs	LUTs
	(ns)	(rel)	(MHz)		(rel)
P$\Delta\Sigma$M	6.6	1.00	152	476	1.00
SP$\Delta\Sigma$M	6.9	1.05	145	6746	14.2
– Block A	6.0	0.91	167	–	–
– Block B	3.6	0.55	278	–	–

Table I: Synthesis results for a Xilinx Virtex-6 FPGA for the P$\Delta\Sigma$M and SP$\Delta\Sigma$M designs, respectively. Delay values for the constituent blocks of the SP$\Delta\Sigma$M design are added for comparison.

	delay	delay	1/delay	area	area	power	power
	(ps)	(rel)	(GHz)	(μm^2)	(rel)	(μW)	(rel)
P$\Delta\Sigma$M	1171	1.00	0.854	8160	1.00	10980	1.00
SP$\Delta\Sigma$M	893	0.76	1.12	90461	11.1	87546	7.97
– Block A	771	0.66	1.30	–	–	–	–
– Block B	614	0.52	1.63	–	–	–	–

Table II: Synthesis results for a 65-nm ASIC flow for the P$\Delta\Sigma$M and SP$\Delta\Sigma$M designs, respectively. Delay values for the constituent blocks of the SP$\Delta\Sigma$M design are added for comparison.

The PISO principle also fits well with the Hatami principle of generating the output bitstream several bits at a time. The arithmetic operations in our implementation used a wordlength of $8 + 3$ bits.

We used Xilinx ISE v14.3 [8] to perform the synthesis, optimizing for speed. All delays were evaluated using Xilinx PlanAhead v14.3 [8]. Placement and routing were carried out twice, optimizing for area and for speed, and the fastest of the two results was chosen. The synthesis results are shown in Table I.

The performance of blocks A and B in isolation correspond to the expectations, in that block B was almost a factor of 2 faster than block A, indicating that the half-rate processing illustrated in Figure 3 is a promising approach. Block B was also roughly a factor of 2 faster than the P$\Delta\Sigma$M design, indicating that overall performance gains may be possible. In the full design, however, the extra routing delay across the much larger area was enough to nullify the gains; in fact, the SP$\Delta\Sigma$M was somewhat slower than the P$\Delta\Sigma$M. An FPGA with a faster switching fabric might be a better choice for these designs. Here, 68% of the overall delay for the SP$\Delta\Sigma$M critical path was due to routing.

The number of LUTs used for the SP$\Delta\Sigma$M design is, as expected, much higher than for the P$\Delta\Sigma$M counterpart; but it is still less than 1.9% of the number of LUTs available in the targeted FPGA.

B. ASIC implementation

As a second implementation experiment, we synthesized the P$\Delta\Sigma$M and SP$\Delta\Sigma$M designs for a commercially available 65-nm bulk-CMOS process, using Cadence Encounter RTL Compiler [9] with commercial libraries, and optimizing for speed. No PISO transceiver was included in either design.

The synthesis results are shown in Table II. No floorplanning, placement, or routing was performed; delays are as estimated by RTL Compiler based on its own area estimates. Power estimates are also as reported by RTL Compiler and are based on an average input switching probability of 50%.

In the ASIC case, the best critical-path value reached for the full SP$\Delta\Sigma$M offered an improvement over the P$\Delta\Sigma$M, at 76% of the best P$\Delta\Sigma$M value. Again, the critical path of the B-block array is much shorter, at only 55% of the P$\Delta\Sigma$M value. As much of the delay still depends on interconnect, we surmise that some floorplanning effort might further improve SP$\Delta\Sigma$M performance.

The area and power overheads of the speculation design are once again large—the factors are 11 and 8, respectively—though not as large as the speculation factor $2^{UF} = 16$. Further investigation may show ways to reduce the cost of the P$\Delta\Sigma$M. We are more comfortable with the speed numbers, as they broadly correspond to architectural expectations.

VI. Conclusion

Our work indicates that the block-wise critical path in the Hatami modulator can be reduced further by making the modulator speculate on the results of the previous iteration. We have validated our speculative architecture with a synthesizable implementation, which we used to evaluate the reachable performance and the hardware cost for two different implementation technologies. In both technologies, the signal delays across the larger size of the SP$\Delta\Sigma$M design eat into the gains—in the FPGA, enough so that the overall gains were negative. We did register a net performance improvement in the ASIC case. More implementation effort should refine the estimates significantly, especially for the ASIC case.

The hardware cost of speculation is significant in relative terms, as the amount of hardware grows exponentially with the Hatami unrolling factor *UF*. In absolute terms, the cost may still be acceptable in some applications, as even the speculative modulator occupies only a small fraction of a current FPGA or of a DSP ASIC.

The P$\Delta\Sigma$M unrolling and the SP$\Delta\Sigma$M speculation both seek to address the same problem: the high clock frequencies needed for large OSR values in $\Delta\Sigma$ modulators for large signal bandwidths. Future work should include a further investigation of the parameter spaces for unrolling factor, speculation, and possibly other approaches to parallel processing, in order to map out their implications for performance and cost.

References

[1] R. Schreier and G. Temes, *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, 2005.

[2] S. Norsworthy, R. Schreier, and G. Temes, *Delta-Sigma data converters: Theory, Design, and Simulation*. Wiley-IEEE Press, 1997.

[3] A. Eshraghi and T. Fiez, “A comparative analysis of parallel delta-sigma ADC architectures,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 3, pp. 450–458, 2004.

[4] J. Cormier, R.F., T. Sculley, and R. Bamberger, “Combining subband decomposition and sigma delta modulation for wideband A/D conversion,” in *Circuits and Systems, 1994. ISCAS ’94., 1994 IEEE International Symposium on*, vol. 5, May 1994, pp. 357–360 vol.5.

[5] E. King, A. Eshraghi, I. Galton, and T. Fiez, “A Nyquist-rate delta-sigma A/D converter,” *Solid-State Circuits, IEEE Journal of*, vol. 33, no. 1, pp. 45–52, 1998.

[6] S. Hatami, M. Helaoui, F. Ghannouchi, and M. Pedram, “Single-bit pseudoparallel processing low-oversampling delta-sigma modulator suitable for SDR wireless transmitters,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 922–931, April 2014.

[7] *Virtex 6 Family Overview*, Xilinx Inc., accessed on Aug 13 2014. [Online]. Available: www.xilinx.com

[8] *ISE Design Suite 14: release notes, installation, and licensing*, Xilinx Inc., accessed on Aug 13 2014. [Online]. Available: www.xilinx.com

[9] *Encounter RTL Compiler*, Cadence Design Systems, accessed on Aug 13 2014. [Online]. Available: www.cadence.com