A CMOS Implementation of a Spike Event Coding Scheme for Analog Arrays

Luiz Carlos Gouveia, Thomas Jacob Koickal and Alister Hamilton
Institute for Micro and Nano Systems - School of Engineering and Joint Research Institute for Integrated Systems
University of Edinburgh, EH9 3JL, U.K.
Email: L.Gouveia@ed.ac.uk, {Thomas.Koickal, Alister.Hamilton}@ee.ed.ac.uk

***Abstract*— This paper presents a CMOS circuit implementation of a spike event coding/decoding scheme for transmission of analog signals in a programmable analog array. This scheme uses spikes for a time representation of analog signals. No spikes are transmitted using this scheme when signals are constant, leading to low power dissipation and traffic reduction in a shared channel. A proof-of-concept chip was designed in a 0.35μm process and experimental results are presented.**

I. Introduction

Configurable Analog Blocks (CABs) are the basic processing units of Field Programmable Analog Arrays (FPAAs) and are configured to perform different types of analog functions [1]. The communication between CABs is an important issue because traditional techniques like crossbars or switch matrices [2] degrade the transmitted signals. These techniques cause distortion of the signal due to voltage drops, parasitic capacitances along wires and the switches and through signal interference, limiting the size of analog arrays.

Pulse modulations are an alternative to these techniques. They map the amplitude of analog signals onto the timing domain. The use of these discrete-amplitude continuous-time modulations allows greater system scalability than analog routing methods. Synchronous versions were used in analog arrays [3], but as they require a global clock signal, these modulations suffer from clock skew and high power consumption.

Asynchronous modulations are used in some biological inspired systems [5] [6]. Recently, a *spike event coding scheme* was proposed by the authors to transmit analog signals between CABs [4]. It presents advantages over other pulse modulations, such as transmission of information on demand and, therefore, reduction in communication traffic, and low energy dissipation, freedom from clock skew and low crosstalk due to asynchronous coding.

In this paper we present a CMOS implementation of the spike event coding scheme with programmable resolution. First we review the working principles of the scheme. Later, we describe the design parameters and the circuits used to realize the coder. Finally, we present results from a tested chip.

II. Spike Event Coding Scheme

The spike event coding scheme [4] is shown in Fig. 1(a). The spike event coder operates by generating a signal similar to the input signal. In other words, a feedback signal $z(t)$ is forced to track the input signal $x(t)$ by bounding the error $e(t)$ between them:

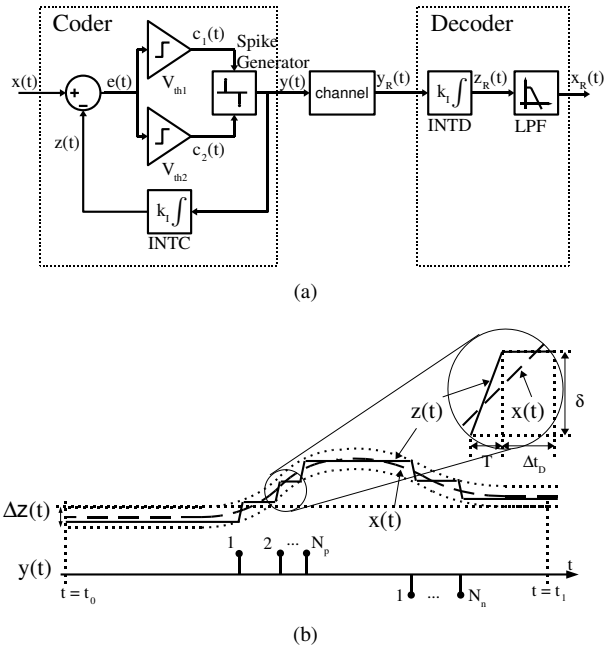


Fig. 1. (a) Block diagram of the spike event coding scheme and (b) an example of the behavior of the main signals. δ is the tracking step, T is the spike width and Δt_D is the time interval between successive spikes.

$$e(t) = x(t) - z(t) = x(t) - \int y(t)dt \tag{1}$$

where $y(t)$ is the coder output. This output is represented either by positive or negative pulses with a short and fixed duration (spikes). These spikes are produced by the spike generator and transmitted both to the communication channel and to the input of the feedback integrator (INTC).

Each positive or negative spike results in an incremental or decremental change δ (tracking step) at the output of the feedback integrator:

$$\Delta z(t) = \delta\ (Np - Nn) \tag{2}$$

where Np (Nn) is the number of previous positive (negative) spikes since t_0 as shown in Fig. 1(b).

Considering an ideal channel, the spikes are also transmitted to the decoder integrator INTD, which presents the same

gain K_I of the coder feedback integrator INTC. The decoder output $x_R(t)$ is given by:

$$x_R(t) \approx \overline{\int y(t)dt} = \overline{z(t)} = \overline{x(t) - e(t)} \quad (3)$$

with the decoder Low Pass Filter (LPF) removing high frequency harmonics and averaging the signal $z_R(t)$. Therefore, the maximum difference between the decoder output $x_R(t)$ and coder input $x(t)$ is $|e(t)|_{max}$, which is defined by the specification of the spike event scheme resolution:

$$|e(t)|_{max} = \frac{\Delta x(t)_{max}}{2^{N_B} - 1} \quad (4)$$

where $\Delta x(t)_{max}$ is the input dynamic range and N_B is the desired resolution in bits.

III. Analog VLSI Implementation

The spike event coding scheme was implemented in a proof-of-concept chip for validation of the scheme. The layout dimensions of the spike event coder are 240μm x 120μm using AMS 0.35μm CMOS process.

In this section we describe the spike event coder and decoder circuits implemented on chip: comparators, spike generator and integrators. Both coder and decoder integrators were implemented using the same design. The decoder LPF was implemented off-chip, using an offline digital filter.

A. Comparators

In the block diagram in Fig. 1(a), the error $e(t)$ is limited by two comparators with different thresholds (V_{th1} and V_{th2}):

$$|e| \leq \Delta V_{th} = |V_{th1} - V_{th2}| \quad (5)$$

From (4) and (5), the resolution of the spike event coding scheme is a function of the thresholds of both comparators.

The design of the comparators can be implemented using a preamplifier (PA) followed by a decision circuit (DC) and an output buffer (OB) [7]. To provide the required ΔV_{th}, capacitive or resistive dividers can be used at the comparator input nodes. However, these dividers compromise the input impedance of the circuit.

Another method to provide ΔV_{th} is to implement offset comparators. Composite transistors can be used to provide the offset [8], however this topology suffers from low dynamic range. A programmable offset can also be generated by another preamplifier which provide a respective ΔI_{off} on the decision circuit input [9]. Both impedance dividers and offset comparators allow continuous resolution values to be used.

In this implementation, both outputs $c_1(t)$ and $c_2(t)$ are generated by a compound comparator in Fig. 2. Instead of using four preamplifiers, with two sensing the inputs $x(t)$ and $z(t)$ and two providing different offsets, we use only two preamplifiers. The preamplifier PAA outputs a differential current ΔI_{xz} as the result of the comparison between $x(t)$ and $z(t)$. Thus, the capacitive loads of these nodes are reduced by using only one preamplifier. The other preamplifier (PAB) provides a differential current $\Delta I_{off}/2$ according to $\Delta V_{th}/2$

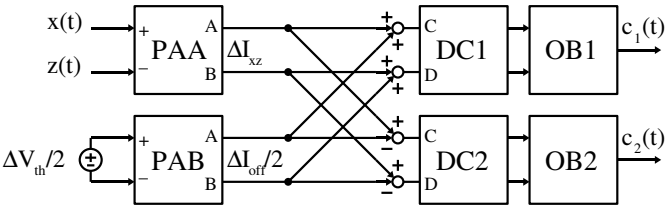


Fig. 2. Block diagram of the compound comparator. Preamplilfier PAA outputs a current ΔI_{xz} according to the inputs $x(t)$ and $z(t)$, while preamplifier PAB generate a fixed offset current $\Delta I_{off}/2$. PAB outputs are added or subtracted from PAA outputs and the results are applied to the respective decision circuits (DC) and output buffers (OB).

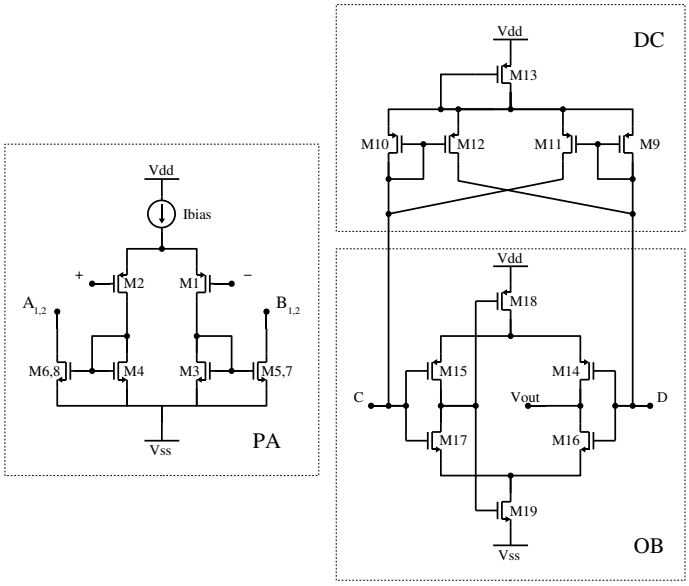


Fig. 3. Circuit schematic of each of the compound comparator blocks. Transconductance preamplifier (PA,left), decision circuit (DC,top right) and output stage (OB,bottom right).

voltage on the inputs. The results of adding ($\Delta I_{xz}+\Delta I_{off}/2$) and subtracting ($\Delta I_{xz}-\Delta I_{off}/2$) these currents are forwarded to the decision circuits to speed up the comparison result. Finally, output buffers generate the digital outputs.

Fig. 3 presents the preamplifier, the decision and the output buffer schematic circuits implemented on chip. The preamplifier is a transconductance amplifier with two identical differential output currents at nodes (A_1, B_1) and (A_2, B_2). The decision circuit is a positive feedback circuit and the output buffer is a self-biased amplifier [10].

B. Spike Generator

The spike generator block can provide either a positive or a negative spike according to the output state of the comparator. When the error $e > \Delta V_{th}/2$, a negative spike is transmitted. Similarly, a positive spike is generated when $e < -\Delta V_{th}/2$. Otherwise, no spikes are transmitted. The control circuitry for this logic was implemented using a technique for the design of asynchronous digital circuits [11]. These spikes can be transmitted using switch matrices or any asynchronous Medium Access Channel (MAC) protocol, like AER [12].

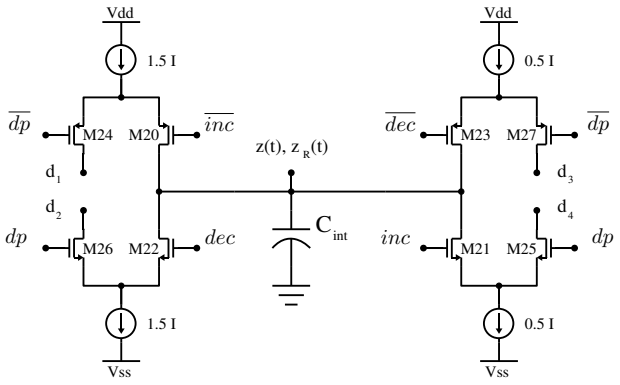


Fig. 4. Schematic of the integrator based on a charge pump technique. When a positive spike arrives, M20 and M21 turn on and $1.5I$ and $0.5I$ charges and discharges the capacitor C_{int}, respectively, resulting in a voltage increment of $\delta_d = T \times I/C_{int}$. For negative spikes, the complementary process decreases the capacitor voltage. In the absence of spikes, currents are drawn to low impedance nodes (d1-d4). The schematic of the delay generators is not shown.

C. Integrators

In this implementation, the spike width T is used in the integrator block to increment or decrement $z(t)$ by δ. A minimum interspike interval $\Delta t_{Dmin} = kT$ is also introduced in the design to avoid overload on the communication channel ("refractory period"). The coder output spike frequency is a function of the magnitude of the input derivative and the coder integration step:

$$f = \frac{\left|\frac{dx(t)}{dt}\right|}{\delta} = \frac{1}{T + \Delta t_D} \tag{6}$$

The first conclusion from (6) is that no spikes are transmitted when the input derivative is zero, i.e. the input signal is constant. This is true only after the feedback signal $z(t)$ has tracked the input signal $x(t)$, i.e. $e(t) \leq \Delta V_{th}$.

We also conclude from (6) that the maximum output frequency occurs when the input signal presents its maximum absolute input derivative. From the specification of this derivative, Δt_{Dmin} is defined and pulse width T is:

$$T = \frac{\delta}{(k+1)\left|\frac{dx(t)}{dt}\right|_{max}} \tag{7}$$

The integrator block includes programmable delay circuits for the generation of T and Δt_{Dmin} time intervals.

For an optimum performance, the tracking step δ is equal to the difference between the thresholds ΔV_{th} of the comparators. Setting $\delta < \Delta V_{th}$, more output spikes will be needed for the feedback signal $z(t)$ track the input signal $x(t)$. Conversely, $z(t)$ will oscillate for $\delta > \Delta V_{th}$. However, a design margin is required because of the random offsets present in the comparator due to process variations. Therefore, the *designed* tracking step is:

$$\delta_d \leq \Delta V_{th} - 6\sigma(V_{os}) \tag{8}$$

where $\sigma(V_{os})$ is the comparator offset standard deviation.

We implemented the integrator block using a charge pump integrator as shown in Fig. 4. A unipolar version of this type of circuit driving resistors is used in steering current cells of some Digital-to-Analog Converters (DACs).

When a negative spike arrives at the integrator input, the dec signal turns high during an interval T, allowing currents $1.5I$ and $0.5I$ flowing in transistors M22 and M23, respectively. Similarly, for the case of a positive spike arrival, transistors M20 and M21 provide symmetrical operation with inc and $\overline{inc}$ signals. The resulting current I that discharges or charges the integrating capacitor C_{int} is given by $I = C_{int} \times \delta_d/T$. Therefore, the integration gain is given by $K_I = \delta_d/T$. When there are no spikes, currents are driven to low impedance nodes (d_1-d_4) through M24-M27 by setting the signal dp high.

The use of two different branches (M22 and M23 or M20 and M21) to both charge and discharge the capacitor reduces charge injection on the integration node [13] at the cost of doubling the power consumption required. If switches M20 and M21 (M22 and M23) have the same dimensions, the charges injected from the gate to drain capacitance of the complimentary switches cancel each other.

IV. Chip Results

We used two different input signals to test the chip: a speech signal and a sine wave. The speech signal was sampled at 44.1 kSps with 8 bit resolution. The coder was designed to provide a resolution of 6 bits. The coder input signal $x(t)$, the decoder integrator output $z_R(t)$ and the coder output spikes $y(t)$ are shown in Fig. 5(a). The same signals are presented in detail in Fig. 5(b) to show the absence of coder output spikes when the signal is constant or when its change is smaller than ΔV_{th}.

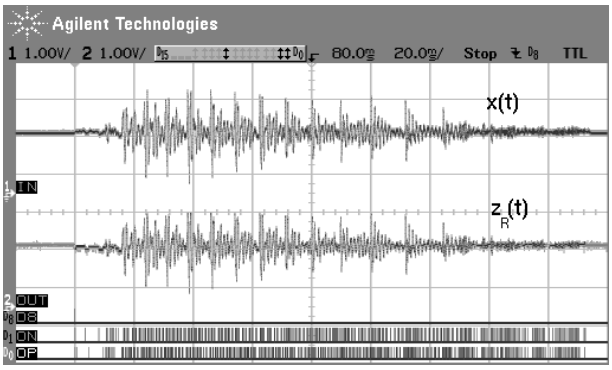
The resolution of the system was measured using a sine wave input signal. The sine wave presents 1.0 Vpp amplitude and 4.4 kHz frequency (f_{in}) sampled at 555 kSps and the coder was designed to provide a 4 bit resolution. A snapshot of the input and output signals is presented in Fig. 6(a). Offline filtering results are shown in the Fig. 6(b) for a digital LPF with a cutoff frequency of 4.4 kHz, the same frequency as the input signal.

The measured resolution of the pre-filter signal is 3.83 bits. The resolution increases to 6.97 and 5.29 bits for post-filter signals, using filter cut-off frequencies equals to f_{in} and $10f_{in}$, respectively. However, this resolution improvement causes attenuation and phase shift of the signal. These results are similar to the simulation values presented in [4]. The measured power consumption of the coder is 0.4 mW, with approximately 90% of if used by the comparator.

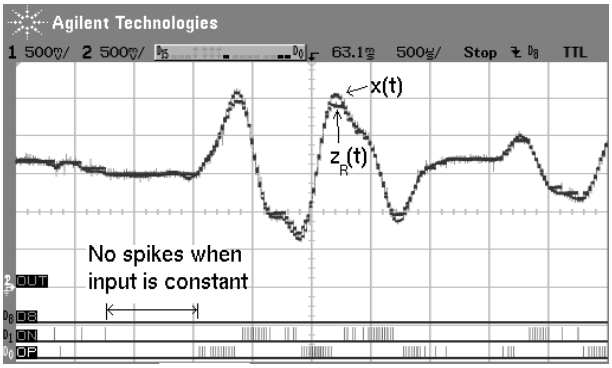
V. Conclusions

In this paper we described a CMOS implementation of a spike event coding scheme. This scheme is intended to be used to transmit analog signals inside a FPAA and/or between different FPAAs using asynchronous events (spikes).

This scheme presents an efficient utilization of channel resources and lower power consumption because no output activity is present when signals are constant. A chip was designed

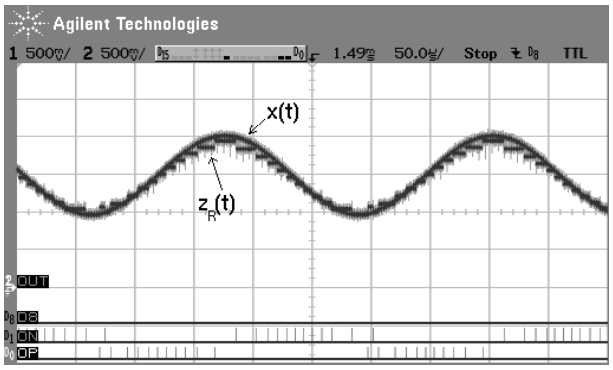


(a)

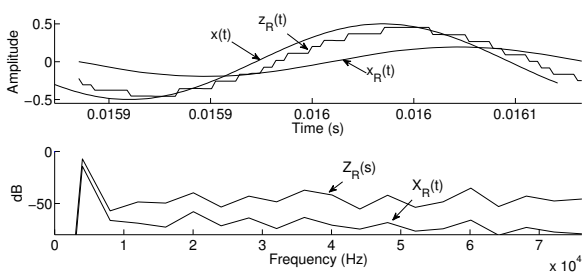


(b)

Fig. 5. (a) Coder input $x(t)$ and decoder integrator output $z_R(t)$ for a 2.5Vpp speech signal. Negative (ON) and positive (OP) output coder spikes are also shown at the bottom of the figure. (b) A detailed view of the same waveforms to show the absence of spikes during the periods when the input signal is constant.



(a)



(b)

Fig. 6. (a) Snapshot of 1.0Vpp sine wave input $x(t)$ and decoder integrator output $z_R(t)$ and negative (ON) and positive (OP) spikes from coder output (bottom). (b) $x(t)$, $z_R(t)$ and the low pass filter output $x_R(t)$ (top). The low pass filter cut-off frequency is the same as the input signal. The frequency spectrum $Z_R(s)$ and $X_R(s)$ of the filter input and output (bottom).

in a 0.35μm process and the experimental results validate the circuit design. This scheme is being implemented within a small array of CABs developed by the authors [14] [15].

ACKNOWLEDGMENT

The authors would like to thank the supports from Engineering and Physical Sciences Research Council and the Scottish Funding Council for the Joint Research Institute with the Heriot-Watt University, part of the Edinburgh Research Partnership in Engineering and Mathematics (ERPem).

REFERENCES

[1] Lee, E.K. and Gulak, P.G., "A CMOS field-programmable analog array". *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 1860-1867, December 1991.

[2] D'Mello, D. R. and Gulak, P. G., "Design approaches to field-programmable analog integrated circuits". *Special Issue on Programmable Analog Systems, Analog Integrated Circuits and Signal Processing*, Hingham: Kluwer Academic Publishers, vol. 17, no. 1-2, pp. 7-34, September 1998.

[3] Papathanasiou, K., Brandtner, T. and Hamilton, A., "Palmo: pulse-based signal processing for programmable analog VLSI". *IEEE Transactions on Circuits and Systems-II, Analog and Digital Signal Processing*, vol. 49, no. 6, pp. 379-389, June 2002.

[4] Gouveia, L., Koickal, T. and Hamilton, A., "An asynchronous spike event coding scheme for programmable analog arrays", in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2008, pp.1364-1367.

[5] Lazar, A.A. and Toth, L.T., "Perfect recovery and sensitivity analysis of time encoded bandlimited signals", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol.51, no.10, pp. 2060-2073, October 2004.

[6] Wei, D. and Harris, J.G., "Signal reconstruction from spiking neuron models". *Proceedings of the International Symposium on Circuits and Systems*, vol. 5, pp. 353-356, May 2004.

[7] Baker, R.J., Li, H.W. and Boyce, D.E., *CMOS: Circuit Design, Layout and Simulation*. New York:IEEE Press, 1998.

[8] Fayed, A.A. and Ismail, M., "A High Speed, Low Voltage CMOS Offset Comparator", *Analog Integrated Circuits and Signal Processing*, vol. 36, no. 3, pp. 267-272, September 2003.

[9] Yaklin, D., "Offset Comparator with Common Mode Voltage Stability", U. S. Patent 5517134, May 1997.

[10] Bazes, M., "Two novel fully complementary self-biased CMOS differential amplifiers", *IEEE Journal of Soild-State Circuits*, vol. 26, pp. 165-168, February 1991.

[11] Martin, A., "Translating Concurrent Programs into VLSI Chips", in *Proceedings of the 4th International PARLE Conference on Parallel Architectures and Languages EuropeI*, London:Springer-Verlag, 1992, pp. 515-532.

[12] Boahen, K., "Point-to-point connectivity between neuromorphic chips using address events", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 416-434, May 2000.

[13] Pan, H., "Method and system for a glitch-free differential current steering switch circuit for high speed, high resolution digital-to-analog conversion", US Patent US20060092062, May 2006.

[14] Koickal, T., Hamilton, A. and Gouveia, L., "Programmable analog VLSI architecture based upon event coding", *Second NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 554-562, August 2007.

[15] Koickal, T., Hamilton, A. and Gouveia, L., "Bio-inspired Event Coded Configurable Analog Circuit Block", *Evolvable Systems: From Biology to Hardware*, LNCS 5216, Berlin:Springer-Verlag, 2008, pp. 285-295.