1D Cellular Automata for Pulse Width Modulated Compressive Sampling CMOS Image Sensors

Marco Trevisi, Ricardo Carmona-Galán, Ángel Rodríguez-Vázquez
Instituto de Microelectrónica de Sevilla (IMSE-CNM)
CSIC-Universidad de Sevilla, Spain.
E-mail: {trevisi, rcarmona, angel}@imse-cnm.csic.es

Abstract— **Compressive sensing (CS) is an alternative to the Shannon limit when the signal to be acquired is known to be sparse or compressible in some domain. Since compressed samples are non-hierarchical packages of information, this acquisition technique can be employed to overcome channel losses and restricted data rates. The quality of the compressed samples that a sensor can deliver is affected by the measurement matrix used to collect them. Measurement matrices usually employed in CS image sensors are recursive random-like binary matrices obtained using pseudo-random number generators (PRNG). In this paper we analyse the performance of these PRNGs in order to understand how their non-idealities affect the quality of the compressed samples. We present the architecture of a CMOS image sensor that uses class-III elementary cellular automata (ECA) and pixel pulse width modulation (PWM) to generate on-chip a measurement matrix and high the quality compressed samples.**

Keywords—compressive sampling; elementary cellular automaton; restricted isometry property;

I. Introduction

Compressive Sampling (CS) is a data acquisition technique that can be used to represent the content of an image with fewer samples than required by Shannon-Nyquist theorem. Reconstruction algorithms based on convex optimization exploit the sparsity of the original image in order to recover it from these compressed samples by finding a unique solution to the underdetermined linear system [1]:

$$\min\|\mathrm{x}\|_1 \quad \text{s.t.} \quad \mathrm{y} = \Phi\mathrm{x} \tag{1}$$

where $\mathrm{y} \in \mathbb{R}^M$ is the set of compressed samples, $\mathrm{x} \in \mathbb{R}^N$ represents the unknown values of the pixels and $\Phi \in \mathbb{R}^{M\times N}$ $(M \ll N)$ is referred to as measurement matrix. A necessary condition for the exact recovery of an image is the restricted isometry property (RIP) of the measurement matrix used to sample it [1]. A matrix Φ satisfies RIP of order k if there is constant δ_k $(0 < \delta_k < 1)$ so that, for all vectors x with $\|\mathrm{x}\|_0 \leq k$, Φ holds:

$$(1-\delta_k)\|\mathrm{x}\|_2^2 \leq \|\Phi\mathrm{x}\|_2^2 \leq (1+\delta_k)\|\mathrm{x}\|_2^2 \tag{2}$$

A matrix with RIP of order k can be used to sample without error images that can be represented with a maximum of k significant elements in some domain. A possible interpretation of Eq. (2) is that, the higher is k, the less sparse the sampled image needs to be for a reconstruction algorithm to deliver an exact solution. The most common approach used to generate a near optimal measurement matrix in signal processing is to extract its elements from a normal Gaussian distribution. There are two major limitations that prevent the implementation of such matrix at sensor level. First of all, CS differs from standard acquisition-&-compression techniques in that the image must be already sensed in a compressed form in the analog domain. For this reason, the implementation of a particular measurement matrix defines the architecture of the CS CMOS image sensor (CS-CIS): if the coefficients of the measurement matrix Φ were real numbers [2], a CS-CIS would require the implementation of analog multipliers in-pixel. This is not practical in terms of pixel sensitivity and spatial resolution. Furthermore, from Eq. (1) we can see that each compressed sample is a linear combination of the weighted readings of all pixels. The upper bound on the amount of bits, $\mathrm{b_{CS}}$, required to represent them would be:

$$\mathrm{b_{CS}} = \lceil \log_2 N \rceil + (\mathrm{b}_\Phi + \mathrm{b_I}) \tag{3}$$

where $\lceil\cdot\rceil$ denotes the smallest integer greater than the argument and $\mathrm{b_I}$ and b_Φ are the number of bits used to describe the pixel values and the measurement matrix coefficients, respectively. It is virtually impossible to design ADCs with such resolutions in standard technologies. In fact, one of the main drawbacks of most CS-CIS implementations [3] is the lack of dynamic range to properly represent the compressed samples.

The use of binary matrices and block-based compressed sampling (BCS) [4], which divides a pixel array into smaller sub-arrays that are digitized independently, is essential for the design of practical CS-CISs. Moreover, to reduce the amount of on-chip resources dedicated to the implementation of a measurement matrix, CS-CISs use pseudo-random number generators (PRNG) to recursively create them one row at the time. In exchange for this approximation, the resulting reconstruction will present errors and additional samples are required to achieve a prescribed accuracy. We propose a CS-CIS architecture that uses class-III elementary cellular automata (ECA) and pixel pulse width modulation (PWM) for the generation of high quality compressed samples.

II. PRNG-Generated Measurement Matrices

PRNG commonly used to implement measurement matrices into CS-CIS are Linear Feedback Shift Registers (LFSRs) [5]. A LFSR consists of a set of flip-flops connected serially. Some of them, called taps, besides being connected to the flip-flop that they follow, are also aggregated by means of XOR logic gates and connected to the input of the very first one. An appropriate choice of taps forces the evolution of the LFSR into an aperiodic behaviour [5]. When considered individually, the output of each flip-flop evolves in time with a pattern that resembles a symmetric Bernoulli distribution: $\mathrm{P} = \overline{\mathrm{P}} = 0.5$. When these outputs are used as column or row selectors of a pixel array it is possible to recursively create the pseudo-random binary coefficients of the measurement matrix one row after the other.

LFSRs though are not the only example of PRNG that might fit in a CS-CIS design. A valid alternative are ECA [6]: one-dimensional temporally discrete dynamic systems made of identical interconnected cells. The future output (NS) of a cell (Fig. 1) of an ECA depends only on its current output (S) and that of its two immediate neighbours (L and R). There are 256 possible logic configurations to combine these three inputs. Each configuration is known as rule. We will focus on rules that belong to class-III, known for their aperiodic behaviour [7], in particular rule 30 (Table I shows rule 30 truth table).

Table I. Truth table of Rule 30

L	S	R	NS
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

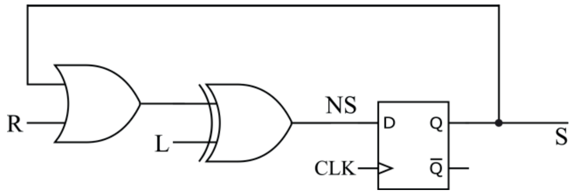

Fig. 1. Implementation of a Rule 30 cell of a cellular automaton

Even if, due to their low impact on area consumption, LFSRs and ECA are both good candidates for the generation of on-chip measurement matrices, ultimately, they still follow deterministic patterns. It is necessary to analyse the matrices that these patterns produce to understand how well they approximate random matrices. To avoid recurring to RIP, a practical approach is to bind it with mutual coherence. From Eq. (2), a matrix Φ that respects RIP with a small δ_k preserves Euclidean distances of the represented elements:

$$\|\mathrm{x}\|_2^2 \approx \|\Phi \mathrm{x}\|_2^2 = \|\mathrm{y}\|_2^2 \qquad (4)$$

which means that, among other properties, Φ must also be nearly orthonormal at least when operating on sparse vectors [8]. Mutual coherence of a matrix $\mu(\Phi)$ is the fined as:

$$\mu(\Phi) = \max_{j \neq k} \left|\varphi_j^H \varphi_k\right| \qquad (5)$$

where φ_k is the k-th normalized column of matrix Φ and φ_j^H is the conjugate transpose of the j-th normalized column of the same matrix. Since mutual coherence represents the maximum absolute value of the cross-correlations among normalized columns of Φ, by definition, a matrix compliant with RIP, must have null mutual coherence.

Given Eq. (4) and (5) it is possible to analyse the patterns generated by LFSRs and ECA in terms of mutual coherence: the lower the better. As previously mentioned, circuit designers choose LFSR because the sequence of their outputs is akin to $\mathrm{P} = \overline{\mathrm{P}} = 0.5$. But, looking at Table I, this holds true for rule-30 ECA as well. For this reason, we can define the number of non-zero elements in a column of Φ as $\mathrm{P} \cdot M$. When we multiply two columns of Φ to obtain their cross-correlation, the probability that their product be different from 0 is equal to the joint probability of the single elements. For this reason, the expected outcome of $\varphi_j^H \varphi_k$ is:

$$E\left(\left\|Diag\left(\varphi_j^H \cdot \varphi_k\right)\right\|_0\right) = \mathrm{P}^2 \cdot M \qquad (6)$$

where $Diag(\cdot)$ is the operator that extract the diagonal vector from the vector multiplication $\varphi_j^H \cdot \varphi_k$ and $\|\cdot\|_0$ is the l_0-norm as described in [9]. All elements of a normalized binary vector are equal to the inverse of the vector Euclidean norm ($1/(\mathrm{P} \cdot M)$). As such, the expected value of the mutual coherence of a random binary measurement matrix derived from a symmetric Bernoulli distribution is:

$$E[\mu(\Phi)] = \mathrm{P} \qquad (7)$$

Eq. (7) gives way to two conclusions. Firstly, since the expected mutual coherence of random binary matrices differs from 0, the RIP has low order and errors during reconstruction are to be expected. Moreover, the limit found on the expected $\mu(\Phi)$ holds only because there is no repeating pattern in the rows of Φ otherwise $E[\mu(\Phi)]$ would increase.

To compare the dynamical behaviours of LFSRs and rule-30 ECA in search of repeating patterns we can use Power Spectral Density (PSD) analysis. This technique has already been applied to these systems [10]. Following their notation, the DFT of PRNGs can be expressed as:

$$\phi_h(f) = \frac{1}{M}\sum_{i=1}^{N} \varphi_h(\tau) e^{-i2\pi tf/_{M}} \quad \text{with} \quad \tau = 1,2,\cdots,M \qquad (8)$$

where $\phi_h(f)$ is the DFT value of the h-th element of the PRNG at frequency f, M is the number of discrete time steps of the PRNG as well as the number of rows in Φ and $\varphi_h(\tau)$ is the state of the h-th element of the PRNG at time τ as well as the τ-th element of the h-th column of Φ. PSD expresses the distribution of the energy of a waveform among its different frequency components. Any peak in a graphic of $\mathrm{PSD}(f)$ over f would represent a strong repeating pattern among the rows of Φ. Given all $\phi_h(f)$, PSD can be computed as:

$$\mathrm{PSD}(f) = \frac{1}{N}\sum_{e=1}^{N} |\phi_h(f)|^2 \qquad (9)$$

where N is the number of elements of the PRNG as well as the number of columns in Φ. The PSD profile of a suitable and efficient PRNG for CS should resemble white noise since its energy should equally spread throughout the entire spectrum (i.e. no repeating pattern). We devised a MATLAB experiment in which we evolved a 64-cells rule-30 ECA (red) and a 64-flip-flops LFSR (blue). Fig. 2 shows their PSD profiles.

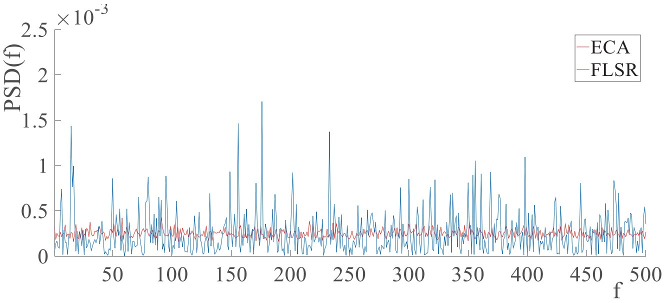

Fig. 2. Power spectral density of rule-30 ECA and LFSR.

These graphics have been produced for $\tau_{max} = 1000$ time steps starting from an initial random binary seed with P = 0.5. Since the PSD of Rule-30 ECA is more flat it appears that they are a better choice over LFSRs to extract quasi-independent coefficients aimed to resemble to a Bernoulli distribution.

III. CS-PWM Sensor Architecture

As shown in Eq. (3), compressed samples require a large number of bits to be described and, as such, standard AD conversion cannot digitize them with enough precision. One way to overcome this problem is to modulate the pixels outputs in order to combine them in a different domain. One of the most common CMOS pixel modulation techniques is Pulse Width Modulation (PWM). It encodes light intensity into time. It uses an integrating photodiode that discharges its cathode at a rate determined by the photocurrent and an in-pixel comparator that triggers an event when the cathode voltage drops below a given reference. The comparators used in our pixels are digital inverters with offset cancellation, the lower (higher) the light intensity on the diode is, the longer (shorter) it takes for the inverter to switch [11]. This event enables the content of a global counter, located outside the array, to be stored into an in-pixel memory. In standard imaging, when high precisions (i.e. many bits) are required, the area occupied by the memory and that used for routing become too large for practical purposes. Since compressive samples do not require acquiring each pixel value separately, in CS there is no need for in-pixel memories: it is possible to directly send the events triggered by the pixels outside the array.

The floor plan of the CS-CIS that we designed is presented in Fig. 3. Its central core is a squared 64 × 64 PWM pixel array surrounded by a rule-30 ECA whose cells deliver their outputs to a whole column or row of the array. The outputs of the cells are used as inputs for in-pixel XOR gates that create the coefficients of the measurement matrix. Since an XOR gate gives a true output only if the number of true inputs is odd, the coefficients that they generate present a ratio of ones and zeros similar to the outcome of a Bernoulli distribution. The pixels of the same column are connected to a common bus that is used to deliver their events, in the form of pulses, to a 20-bit sum-&-accumulate element (S&A). At the top of each column there is an event control unit that controls the duration of these pulses. Each pulse encodes a pixel value in the period of time that has passed between the photodiode reset and the pulse arrival to the S&A. A method to translate these pulses into digital codes is to use them to activate the sampling of the global 8-bit counter. Each time a event arrives, the counter is sampled and its value is added to the contributions of other pixels already stored within the S&A. At the end of the integration period the S&A are connected sequentially. After some delay due to propagation, the rightmost S&A will show, at its output, the sum of all of the S&A connected before it without the need of a dedicated memory frame-buffer or extra operations.

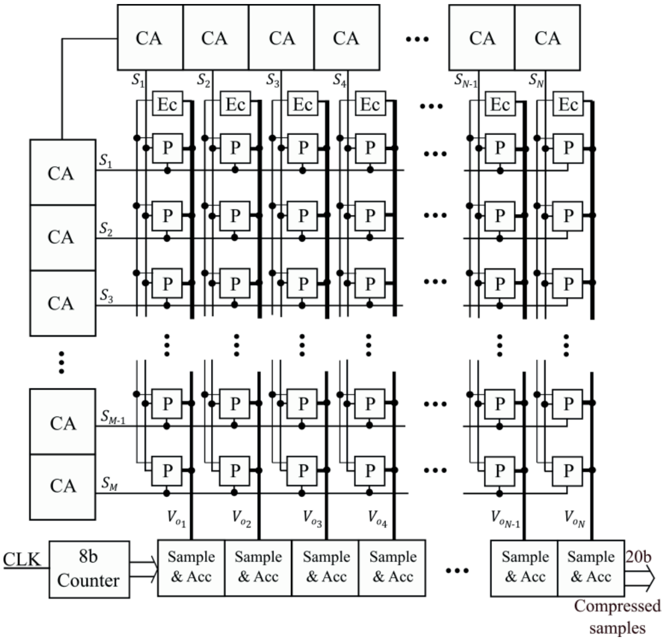

Fig. 3. Conceptual floorplan of the sensor chip

Another consequence of Eq. (3) is that there is an amount of compressed samples beyond which the uncompressed image is smaller than its compressed counterpart. In our case, as pixel values are encoded by 8 bits and compressed samples by 20 bits, the compression ratio, i. e. the number of samples delivered divided by the total number of pixels in the image, needs to be below 0.4. In addition, compressed samples are generated sequentially. If we considered a frame rate of 30 fps, it is necessary to operate the imager at a frequency of ≈ 50 kHz maximum. This leaves a minimum time of 20μs to collect a single compressed sample.

The 8-bit counter clock needs to tick 256 times in this 20μs. The relationship between photo-generated current, I_{ph}, and time that passes between the photodiode reset and the pulse arrival to the sum-&-accumulate element, t, is hyperbolic. This can be derived by modelling the voltage drop at the cathode of the photo-diode, ΔV_{ph}, as the discharge of a capacitor:

$$I_{ph}t = C \cdot \Delta V_{ph} \tag{10}$$

This approximation is accurate for small values of ΔV_{ph}. From Eq. (10) we conclude that PWM cannot work with a counter that counts at a fixed frequency. The frequency of the counter must change linearly in time, with an expression like

$$f_c(t) = f_0 - kt \tag{11}$$

being k a positive constant. Joining Eq. (10) and (11) we obtain an expression to derive the frequency of the 8-bit counter clock given an estimate of the photodiode capacitance $C = 60\text{fF}$, of · $\Delta V_{ph} = 0.1\text{V}$ and of $\max(I_{ph}) = 300\text{pA}$:

$$f_c(t) = f_0 - k\frac{C\cdot\Delta V_{ph}}{I_{ph}} \tag{12}$$

Resulting in a range of frequencies of 200MHz< f_c <10MHz.

IV. Chip Prototype

A prototype chip has been designed in a CMOS 0.18μm technology (Fig. **4**). The die size including pads is $3.17 \times 2.23\text{mm}^2$. It has 84 pads, of which one third is dedicated to power supply and ground connections. Table II contains a summary of the features of the prototype that we fabricated.

Table II. Summary of chip features

Technology	CMOS 0.18□m 1P6M
Pixel size	22μm × 22μm
Fill factor	9.2%
Photodiode type	n-well/p-substrate
Power supply	3.3V-1.8V
Predicted power consumption	<100mW
Frame rate	30fps
Max. compressed sample rate	50kHz
Clock Freq.	200MHz-10MHz

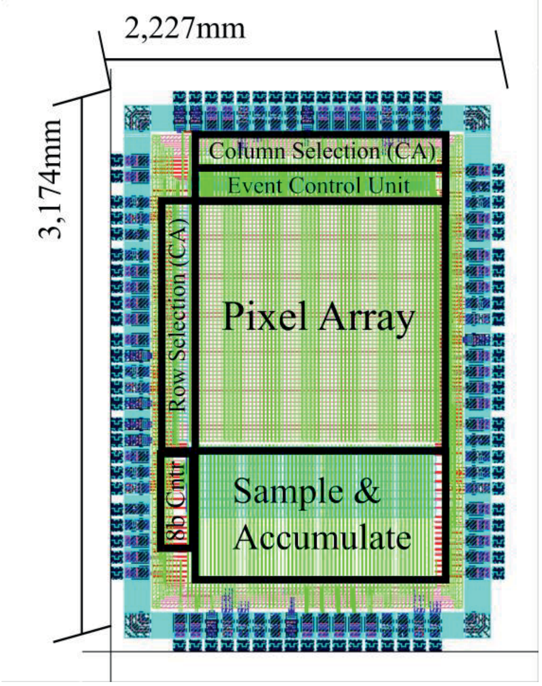

Fig. 4. Conceptual floorplan of the sensor chip

V. Conclusions

We designed a CS-CIS prototype based on class-III ECA for the generation of on-chip measurement matrix and PWM to generate 20-bit compressed samples. This design improves the trade-off between feasibility and accuracy of reconstruction common to this type of sensors. BCS imagers usually employ blocks of 8 × 8 pixels and standard ADC. The small size of these blocks increases the asymmetry of the sensor worsening reconstruction beyond the limit imposed by the low order RIP typically found in pseudo-random binary measurement matrices. Experimental characterization of the prototype will allow verifying the advantages of this implementation with respect to other CS applications.

Acknowledgment

This work has been funded by the Spanish Government through projects TEC2015-66878-C3-1-R MINECO (European Region Development Fund, ERDF/FEDER), by Junta de Andalucía through project TIC 2338-2013 CEICE and by the Office of Naval Research (USA) through grant N000141410355 and CONACYT (Mexico) through grant MZO-2017-291062.

References

[1] Candès. “Compressive sampling”. *Int. Congress of Mathematics*, pp. 1433-1452. Madrid, Spain. August, 2006

[2] J. A. Tropp, J. S. Wright. “Computational Methods for Sparse Solution of Linear Inverse Problems”. *Proc. of the IEEE*, Vol. 98, No. 6, pp. 948-958, Jun 2010..

[3] Y. Oike, A. El Gamal. “CMOS Image Sensor With Per-Column ΣΔ ADC and Programmable Compressed Sensing”. *IEEE Journal of Solid-State Circuits*, Vol. 48, No. 1, pp. 318 - 328, January. 2013.

[4] Lu Gan "Block Compressed Sensing of Natural Images", *15th International Conference on Digital Signal Processing*, pp. 403-406. Singapore, Republic of Singapore. July, 2007.

[5] G. Boeing. “Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction”. *Systems*, Vol. 4, No. 37. Nov, 2016.

[6] E. Jen. “Aperiodicity in one-dimensional cellular automata”. *Physica D: Nonlinear Phenomena*, Vol. 45 No. 1-3, pp 3-18. September, 1990.

[7] S. Wolfram. “Statistical Mechanics of Cellular Automata”. *Review of Modern Physics*, Vol. 5, No. 3, pp. 601-644. July, 1983.

[8] R. Obermeier, J.A: Martínez-Lorenzo. “Sensing Matrix Design via Mutual Coherence Minimization for Electromagnetic Compressive Imaging Applications”. *IEEE Transactions on Computational Imaging*, Vol. 1 No. 2, pp 217-229. Jun, 2017.

[9] D. Donoho. “Compressed Sensing”. *IEEE Trans. Inform. Theory*, Vol. 52, No. 4, pp. 1289-1306. April, 2006.

[10] S. Ninagawa. “Power Spectral Analysis of Elementary Cellular Automata”. Complex Systems Publications, Inc., Vol. 17, No. 4, pp. 399–411. Dec, 2008.

[11] M. Trevisi, H. C. Bandala, J. Fernández-Berni, R. Carmona-Galán and Á. Rodríguez-Vázquez. "Concurrent focal-plane generation of compressed samples from time-encoded pixel values". *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1316-1320, Dreseden, Germany, Mar. 2018.