Characterization of processing errors on analog fully-programmable cellular sensor-processor arrays

Stephen J. Carey\textsuperscript{a}, Ákos Zarándy\textsuperscript{b} and Piotr Dudek\textsuperscript{a}

\textsuperscript{a}School of Electrical and Electronic Engineering
The University of Manchester, Manchester, M13 9PL, United Kingdom
p.dudek@manchester.ac.uk

\textsuperscript{b}Institute of Automation and Control (MTA-SZTAKI) and Pazmany Peter Catholic University, Budapest, Hungary
zarandy.akos@itk.ppke.hu

Abstract— Analog processor arrays, particularly for vision chips, have been in development for a number of years. Based normally on the SIMD computing paradigm, they achieve very high instruction parallelism through use of compact processing elements. A fundamental aspect of processor operation is the ability to copy content of a register to another. This operation carries with it a number of errors. This paper identifies these errors and provides the means by which they can be separated and measured. These techniques can then be applied to analyze any operation upon an analog processor array. The results from such an analysis reflect on the susceptibility of the circuit to process mismatch, and thermal and switch noise performance of a particular analog memory design.

I. INTRODUCTION

Analog processor arrays have been in development for inclusion in vision sensors or image processors for a number of years [1,2]. For lower cost process nodes (e.g. 180nm) they have offered simple integration with light sensors and higher functionality per unit area than is feasible by a digital counterpart in the same process node. Additionally, they have shown high power efficiency and speed [2,3,4,5]. The particular processors that we are concerned with in this paper are discrete-time analog processor arrays usually programmed as SIMD computers (Fig.1). These processors are fed instructions, which they execute in a similar clocked manner as digital processors, with typical operational frequencies of between 1MHz and 10MHz.

While characterization of digital processors is relatively straightforward, analog processing brings with it a spectrum of metrics generally unknown to the digital programmer. Analog processors will always produce an erroneous result to some degree. In this paper, we provide a framework for the characterization of these errors. Through an understanding of the operational attributes of analog processors, we may realize workarounds, choose different algorithms or indeed output the data and process off-chip.

Storage elements commonly used within analog processor arrays are usually switched-current [6] or switched capacitor cells [7], each of which has advantages dependent on the desired computation. SC cells being ideally suited to signal averaging, such as performing diffusion operations with SI cells suited to addition, subtraction operations.

II. ERROR MODEL AND CHARACTERIZATION

The particular operation we characterize in this work is that of "copy". This then can provide a model by which other operations may be characterized. The effect of the copy operation within an SIMD array computer is to replicate a register's contents within every processor element, i.e. for an \( M \times N \) array of elements:

\[
A_{i,j} = B_{i,j}
\]

where \( 1 \leq i \leq M \) and \( 1 \leq j \leq N \). The operation considered in shorthand is simply \( A \leftarrow B \). Generally a means is provided for setting a register with a code \( c \). Through this system, it is possible to set an initial register \( B \) to any value within its dynamic range. Both initial register \( B \) and target register \( A \) may then be readout.

The copy process is accompanied by an error \((E_{i,j})\) specific to each and every processor element; it is this error that we wish to illuminate. \( E_{i,j} \) is the difference between the internal stored values \( A_{i,j} \) and \( B_{i,j} \) and is a function of time, \( t \) and \( c \), the code (or signal level) with which the source register \( B \) is initially loaded. The error can be decomposed into an offset of \( E_{0}(c) + \delta E(c) \), where \( E_{0}(c) \) is the array mean offset and \( \delta E(c) \) is a PE specific or fixed pattern offset around the mean, and a time dependent offset for each transfer of \( E_{i,j} \). For a single transfer, the error is given by:

\[
A_{i,j}(c,t) + B_{i,j}(c,t) = E_{i,j}(c,t) = E_{0}(c) + \delta E(c) + \varepsilon_{i,j}(c,t)
\]
If the internal transfer is repeated $R$ times, with repeats numbered $1 \leq r \leq R$ then

$$E_{ij}(c, R, t) = R(E_0(c) + \delta_{ij}(c)) + \sum_{r=1}^{R} \xi_{ij}(c, t_r)$$  \hspace{1cm} (3)

Equation (3) shows the correlated noise of the offset increasing by a factor of $R$. Random noise ($\xi$) is uncorrelated and is assumed white gaussian in nature; thus (right-hand-side of (3))

$$\sum_{r=1}^{R} \xi_{ij}(c, t_r) = \sqrt{R} \xi_{ij}(c, t)$$  \hspace{1cm} (4)

When array memories $A$ and $B$ are read-out (equivalent to reading out two frames), additional noise is added constituting a read-out noise $N_a$ and $N_b$ for each frame, with additionally a fixed pattern noise (FPN), $F$, for the second frame ($A'$). While normally inconsequential, this FPN arises from the finite degradation that has occurred to register $A$ during the course of register $B$'s readout; this is caused by register $A'$s extended storage time (relative to $B$) and readout switching effects. If the entire repeat copy process is repeated $Q$ times (in sequence where $1 \leq q \leq Q$ ) then the difference between two observable frame measurands (for $A$ and $B$) for the $q$th frame-pair is:

$$\Delta_{ij}(c, R, t_q) = A'_{ij}(c, R, t_q) - B'_{ij}(c, t_q)$$  \hspace{1cm} (5)

with the relations between read-out ($A$' and $B$') and internal values being:

$$A'_{ij}(c, R, t_q) = A_{ij}(c, R, t) + N_a(t_q) + F_{ij}(c)$$  \hspace{1cm} (6)

$$B'_{ij}(c, t_q) = B_{ij}(c, t) + N_b(t_q)$$  \hspace{1cm} (7)

From (2,5-7):

$$\Delta_{ij}(c, R, t_q) = E_{ij}(c, R, t_q) + N_a(t_q) - N_b(t_q) + F_{ij}(c)$$  \hspace{1cm} (8)

By careful processing of the read signals ($\Delta$) each individual error source (within $E_{ij}(c)$) can be extracted. We start the process with determining the scalar average offset error $E_0(c)$.

### A. Average offset error

If the Q frame-pairs are averaged (denoted by overbars); this reduces the magnitude of the temporal noise sources by a factor of $\sqrt{Q}$. Taking the pixel-wise array average of $Q$ frame-pairs of (8) with substitution from (3-4) gives us:

$$\overline{\Delta_{ij}(c, R, t)} = R(E_0(c) + \delta_{ij}(c)) + \overline{\sum_{r=1}^{R} \xi_{ij}(c, t_r)}$$  \hspace{1cm} (9)

Once the array average of (9) is taken, the uncorrelated noise terms ($\epsilon$, $N_a$, $N_b$) become (even more) insignificant and since the mean of array FPN ($\overline{\delta}$) is by definition zero, the sampled array average of read-out error differences is:

$$\frac{\overline{\Delta(c, R)}}{R} = \frac{\sum_{r=1}^{R} \sum_{i,j=1}^{N} \Delta_{ij}(c, R, t)}{NM} = R \ E_0(c) + F(c)$$  \hspace{1cm} (10)

Taking two data-sets with $R$ copy operations and $R=1$, allows $F(c)$ to be removed to determine $E_0(c)$:

$$E_0(c) = \frac{\overline{\Delta(c, R)}}{R}$$  \hspace{1cm} (11)

The deviation around this error, across the array, is the FPN ($\overline{\delta}$) which we now determine.

### B. Fixed pattern noise

The FPN of a register copy is the standard deviation of the offset error. Using the principles of propagation of error[8], and assuming that $N_a(t_q) = N_b(t_q)$ and that they are uncorrelated, the variance ($\sigma^2$) of the pixel difference of the two frames at read-out is

$$\sigma^2_{\delta(c,R)} = \left(\frac{\partial \delta}{\partial t} \right)^2 \sigma^2_{\delta_0} + \left(\frac{\partial F}{\partial t} \right)^2 \sigma^2_F + \left(\frac{\partial \xi}{\partial t} \right)^2 \sigma^2_{\xi_0} + \left(\frac{\partial \xi}{\partial t} \right)^2 \sigma^2_{\xi}$$  \hspace{1cm} (12)

Applying this principle to (9) over all pixels (all $i,j$), the variance of frame error is:

$$\sum_{i,j=1}^{N} \sum_{r=1}^{R} \overline{(\delta_{ij})^2}$$

With typical values of $R = 5$ and $Q = 20$, it is usually unnecessary to consider the last two terms of this equation; this can be confirmed by later calculation of random and readout errors.

From (13), the code-dependent standard deviation of offset (i.e. fixed pattern noise) of register copy can be found:

$$\overline{\delta_{ij}(c)} = \sqrt{\overline{(\delta_{ij}(c))^2}}$$  \hspace{1cm} (14)

with:

$$\overline{\delta_{ij}(c)}^2 = \frac{\sum_{i,j=1}^{N} \sum_{r=1}^{R} (\delta_{ij}(c,R,t_r) - \delta_{ij}(c))^2}{NM}$$  \hspace{1cm} (15)

Equation (15) provides an array-global fixed pattern noise metric for register copy. It can be useful to view the FPN graphically in addition. By a similar derivation from (9), each pixel's offset is given by:

$$\delta_{ij}(c) = \Delta_{ij}(c) - E_0(c)$$  \hspace{1cm} (16)

Many image processing algorithms are based on spatially localised operations (e.g. convolution kernels etc). In many such cases, global low-frequency spatial variation that contributes to the FPN calculated across the entire array is much less of a problem (especially if local average can be easily obtained and compensated for). However variation seen in a ‘local’ zone is not correctable, hence it is useful to determine a figure for local FPN. Taking a 5x5 grid of pixels, the local variance is defined as:

$$\overline{\delta_{\text{local}ij}(c,R)} = \sum_{i,j=1}^{N} \sum_{r=1}^{R} \left( \delta_{ij}(c,R,t_r) - E_{\text{local}}(c) \right)^2$$  \hspace{1cm} (17)

with local average $E_{\text{local}}(c)$ given by:

$$E_{\text{local}}(c) = \sum_{i,j=1}^{N} \sum_{r=1}^{R} \left( \delta_{ij}(c,R,t_r) \right) / \left( N \cdot M - 5 \right)$$  \hspace{1cm} (18)

To afford a single measure of the local FPN, the array average of the local FPN figures is taken:

$$\overline{\delta_{\text{local}ij}(c,R)} = \sum_{i,j=1}^{N} \sum_{r=1}^{R} \overline{(\delta_{ij}(c,R,t_r))^2}$$  \hspace{1cm} (19)

### C. Random Noise

If PEs are considered as individual entities, then errors relative to other pixel's of the array can be excluded (i.e. $E_{\text{local}}$, $\delta$ and $F$); a PEs output error will vary only with those variables that are time dependent, i.e.$N_a$, $N_b$ and random noise of register copy, $\xi$, which we determine in this section. Applying error propagation to a single pixel's differences
\( (\Delta_i) \) in the same manner as in \((12)\), and without the benefit of averaging \(Q\) samples, yields a function of its variance:
\[
\sigma_{\Delta_i}(C,R) = R\sigma_{\Sigma}(C) + 2\sigma_{\delta^2}^2  \tag{20}
\]
The variance of a pixel's error at read-out is determinable from measurements and is given by:
\[
\sigma_{\Delta_i}(C,R) = \frac{\sum_{q=1}^{Q} \Delta_i(c,xt_q) - \Delta_i(c,R)}{Q} \tag{21}
\]
with mean \(\Delta_i(c,R)\) given by the average of \(Q\) samples:
\[
\Delta_i(c,R) = \frac{\sum_{q=1}^{Q} \Delta_i(c,xt_q)}{Q}  \tag{22}
\]
Determining random noise for copy per pixel \(\sigma_{\delta}(c)\) is then feasible, but since there is only a small increase in noise due to increased repeats (relative to the magnitude of readout noise), improved results are obtained by assuming that random noise of register copy is equal across the array. In this case, the array average of the variance of read-out error can be determined:
\[
\sigma_{\Delta,AVG}(C,R) = \frac{\sum_{j=1}^{N} \sum_{i=1}^{M} \sigma_{\Delta_i}(C,R)}{NM} \tag{23}
\]
From \((20)\) and by taking two parallel data sets with \(R=1\) and \(R\) copies, we can then find the random error of register copy for the average pixel:
\[
\sigma_{\delta}(C) = \sqrt{\frac{\sigma_{\Delta,AVG}(C,R)^2 - \sigma_{\Delta,AVG}(C)^2}{R-1}} \tag{24}
\]
If necessary, this data can be fed back into equation \((13)\) to improve the accuracy of the determination of \(\sigma_{\delta}(C)\); however, this is not normally necessary.

III. RESULTS

Two different analog processor arrays were acquired, programmed and tested at room temperature; we will refer to these systems as System \(\alpha\) and System \(\beta\). We do not identify the systems since our intention in this work is to advocate the testing methods used rather than promote a particular system's merits/demerits.

The measurement process used \(20\) repeats of each measurement (i.e. \(Q=20\)) with the number of internal copy repeats \((R)\) taking values of \(1,2,3,4,5\) and \(6\). Codes of \(0\) to \(255\) in approximate steps of \(16\) were input to the devices (17 values). With two frames per measurement, a total of \(4080\) frames were available for processing from each system.

The pseudo-code for this acquisition is:

```
for q=1 to 255 step 16 [last step increments 15]
    load X, #c
    OUTPUT X [Frame B]
    for r = 1 to R
        Y ← X
        X ← Y
        next r
    OUTPUT X [Frame A]
    next q
next c
```

Note that for the purposes of this work, a single copy iteration is composed of two transfers in order to facilitate the repeat process. During the analysis, these are treated as a single transfer. For uncorrelated random noise, individual transfers will be \(1/\sqrt{2}\) of the noise values recorded. For FPN, the net effect of the two transfers is reported; additional work is required if the effect of single transfers is to be determined assuredly.

Shown in Figures 2 and 3 are the four measurements of offset \((11)\), global fixed pattern noise \((14)\), local fixed pattern noise \((19)\) and random noise \((24)\) of register copy for System \(\alpha\) and System \(\beta\) respectively (applicable equation in brackets). All measurements are taken for \(R=6\) (and hence make use of the dataset for \(R=1\) in addition).

Figure 4 shows an example of a full data-set for different \(R\) values for offset error for System \(\beta\) (from equation \((11)\)). This shows a clear gain error of copy in System \(\beta\) leading to a significant change in code \(c\) during the course of the multiple copy operation - hence the x-axis points used for this plot (and in addition Figures 2 and 3) is the average of the input and output codes. The fact that the plots lie in the same location for multiple measurements taken with different values of \(R\) indicates the model for this system is correct - indicating the importance of collecting data for multiple values of copy repeat. We have found this an invaluable tool when debugging the measurements and models. Other specific aspects of the data are outlined below.
D. indicating that local transistor matching is generally good.

Transfer is different in character from the global noise figure, i.e. (15) becomes:

\[ \alpha \]  

\( \sum G^2 \) for determining this metric.

\[ \beta \]  

\( \sum G^2 \) for determining this metric.

\[ \gamma \]  

\( \sum G^2 \) for determining this metric.

\[ \delta \]  

\( \sum G^2 \) for determining this metric.

\[ \epsilon \]  

\( \sum G^2 \) for determining this metric.

\[ \zeta \]  

\( \sum G^2 \) for determining this metric.

\[ \kappa \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \chi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.

\[ \omega \]  

\( \sum G^2 \) for determining this metric.

\[ \theta \]  

\( \sum G^2 \) for determining this metric.

\[ \varphi \]  

\( \sum G^2 \) for determining this metric.

\[ \psi \]  

\( \sum G^2 \) for determining this metric.