A FLEXIBLE GLOBAL READOUT ARCHITECTURE FOR AN ANALOGUE SIMD VISION CHIP

Piotr Dudek

Department of Electrical Engineering and Electronics
University of Manchester Institute of Science and Technology (UMIST)
PO Box 88, Manchester M60 1QD, United Kingdom
p.dudek@umist.ac.uk

Abstract

A new vision chip, SCAMP-2, has been developed in a 0.35µm CMOS technology. In this paper, the design of the chip is presented, with particular emphasis on its readout architecture. A combination of the addressing flexibility provided by the novel readout scheme and the global operation capability of the analogue processors results in the increased functionality of the smart sensor device. Example applications of the proposed architecture are discussed. In addition to low-level image processing algorithms, such as filtering or edge detection, which are efficiently executed on a nearest-neighbour connected fine-grain massively parallel SIMD array, it is also possible to implement global algorithms, such as histogramming. The proposed architecture enables increased control over wave-propagation-type algorithms and simplifies the design of winner-take-all algorithms. Significant speed-up can be achieved in applications such as visual tracking.

1. Introduction

The idea of tightly integrating CMOS image sensors with a processor array, so that the images are input to the processors in parallel and processed directly on the focal-plane, has received significant research attention in recent years. These systems, called “artificial retinas”, “vision chips” or “smart sensors” exploit inherent fine-grain parallelism of the computationally intensive low-level image processing tasks. Some of the vision chips are application-specific and contain circuits implementing particular algorithms in hardware. More versatile are software-programmable, general-purpose vision chips, capable of executing a variety of algorithms. While several approaches to the design of a programmable processor array have been proposed, their general architecture is that of an SIMD (Single Instruction Multiple Data) parallel computer. The processing elements (PEs) can be implemented in digital technology using a bit-serial approach [1,2], or using analogue techniques [3,4], the latter offering a particularly efficient solution in terms of performance, cost and power dissipation of the system.

Due to the physical constraints (silicon area available to a PE and power dissipation requirements) the design of the PE is critical to the success of the fine-grain SIMD processor approach. Ideally we would like to construct pixel-per-processor arrays capable of dealing with high-resolution images. The massively parallel execution of low-level image processing algorithms, such as filtering, is then rather straightforward and the execution time can be far shorter than the time required to read-out the output image. Overcoming the output bottleneck becomes a critical issue. Indeed, an important aspect of the vision chip approach is its ability to reduce the amount of data that is transmitted to the higher-level processor. Simple examples are provided by adaptive thresholding, or calculation of an edge map, where original gray-scale images are reduced to 1-bit images. Even greater data reductions are possible by replacing the 2D “bitmap” images with a set of image descriptors, extracting some global measures from the image such as a histogram or the coordinates of the centres of objects present in the image. Very high-speed operation is potentially possible if these data can be extracted from the image inside the vision chip. Recently, several vision chips have been proposed, which achieve this goal by introducing dedicated circuitry to implement a particular global operation, such as winner-take-all [5,6] or moment extraction [1]. Our motivation, however, is to provide a general-purpose solution. This not only provides the system user with a freedom to implement a variety of algorithms, but also results in the efficient use of hardware resources. In this paper we present the architecture of a new vision chip based on the SCAMP approach [4,7]. A flexible global readout scheme designed for this chip attempts to address several issues of global communication and data extraction, making the chip suitable for high-speed visual tracking and similar applications.

2. SCAMP-2 Chip

The SCAMP-2 vision chip is a fine-grain 2D SIMD processor array, in which each PE corresponds to a single pixel in the image. A single controller issues instructions, which are broadcast to each PE in the array, and the PEs execute instructions in parallel, each operating on their local data. While the architecture and the functionality of the PEs is similar to that of digital processors, they represent data using analogue sampled signals and process data using analogue circuits, and are thus called Analogue Processing Elelments (APEs) [8]. The overall block diagram of the system is shown in Fig.1.

The controller is a purely digital microprocessor and its primary function is to deliver a stream of Instruction Code Words (ICWs) to the processor array. It can also communicate with the APEs in the array via A/D and D/A converters. The controller can address individual APEs, as well as groups of APEs and the program execution flow in the controller can be conditioned by the status of the array.

The design of the APE in the SCAMP-2 chip is based on switched-current techniques and is similar to the one reported in [4], although several circuit improvements have been introduced to reduce power dissipation, improve matching and reduce errors associated with analogue signal processing. Each APE contains an integration-mode photodetector, eight general-purpose registers, a local communication register, an output port for external data transfers, an input port (typically used to generate an immediate argument of an instruction) and a local-autonomy activity-flag register.

0-7803-7762-1/03/$17.00 ©2003 IEEE

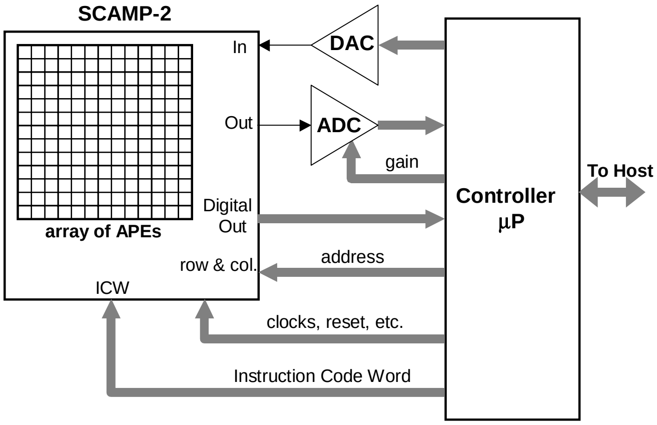


Fig.1. Overview of the SCAMP-2 system

The registers are capable of storing a variable, e.g. a pixel value in a grey-level image, with a resolution/accuracy approaching 8-bits (the stored data is continuous, but in practice limited by sampling errors and noise). The instruction set of the APE supports register transfers, basic arithmetic operations, parallel input and output operations, local communication with nearest-neighbour APEs, conditional instructions and some special control instructions (e.g. reset of photodetectors). The arithmetic operations of addition, inversion, and division are executed exploiting computational capabilities inherent in the design of switched-current registers and no additional hardware is required for this purpose [8], which is one of the reasons for high efficiency and small silicon area of the "analogue microprocessor" approach. It should be noted that the previous SCAMP implementation contained an explicit current multiplier. On the SCAMP-2 chip multiplication is performed by accumulated additions while division is performed directly in registers by current-bisections. This approach reduces the size of the processor, while providing adequate speed/accuracy for typical values of coefficients used in early vision algorithms. Logic operations can be simulated by arithmetic operations. A current comparator is included, for conditional operations. The activity-flag register is set or reset, depending on the result of the conditional operation. Only those APEs in the array that have their activity-flag registers set are active (i.e. change the state of their registers as a result of a broadcast instruction), which enables local data-dependent control of the program flow.

The details of the APE circuitry design and operation will be reported at a later date, in the remaining sections of this paper we will concentrate on the novel architectural features of the SCAMP-2 chip.

3. Flexible Global Readout Architecture

At the end of the image processing operation some registers may contain pixel values that correspond to a grey-level output image. This output image can be read-out via the output port. For this purpose, the SCAMP-2 array can be treated as a random-access memory. When the row and column address of the APE is asserted, the analogue bus of the selected APE is connected to the serial analogue output port. The output image from the entire array, or its portion, can thus be read-out.

3.1. Global Summation

A salient feature of the flexible global readout architecture is the possibility of addressing several APEs at the same time. One, many, or all APEs can be addressed for readout. Since the APE is a current-mode processor, when several APEs are connected to the output port their output currents are added together. Thus the very useful global summation operation is performed in the analogue system with no extra hardware requirements and in a single instruction cycle (this is in stark contrast to the digital system, where the global summation usually requires the controller to read out the PEs serially and accumulate the output values).

3.2. Digital Output

The SCAMP-2 chip also supports high-speed digital output. An 8-bit column-parallel port can be used to read out binary images, which are often the results of the execution of low-level image processing algorithms (i.e. thresholded image, edge map, etc.). The status of the activity-flag register is read out in this case. There is also a serial output digital port, which is used for global logic operations. A logic OR is performed on the activity-flag registers of all addressed APEs. This useful feature is, again, implemented with little area overhead, using a "wired" OR gate.

3.3. Control Feedback

The global analogue output is digitised by an 8-bit A/D converter with a programmable gain and its value is available to the controller. Digital outputs are also available to the controller. The program execution in the controller (conditional jumps) can depend on the values read out from these ports and so the sequence of ICWs delivered to the array can depend on the status of the array. This control feedback provides a facility to achieve global control over wave-propagation-type algorithms, as well as general data-dependent program-flow, and is extensively used in the design of algorithms exploiting the flexible global readout architecture.

3.4. Addressing Modes

The simultaneous selection of multiple APEs is facilitated by enabling *don't care* values in the row and column address words. In practice this is achieved by using two address words to provide binary addresses for rows and columns in the array and introducing two additional address words to provide the row and column addresses with the *don't care* attribute. In the present implementation all four address words are multiplexed onto a single address port; this introduces little time overhead during usual readout procedures while significantly reducing the number of I/O pins.

Examples of multiple APE addressing using *don't cares* are given in Fig. 2. The addressing scheme allows great addressing flexibility while being relatively easy to implement in hardware, with no additional silicon area required in the design of the row/column address decoders.

Although any combinations of '0' '1' and 'X' (*don't care*) are allowed in the address words there are four cases of particular importance, listed below:

Mode 0 (Single APE selection). When no *don't cares* are used in the address then a single APE is selected.

Mode 1 (Entire Array selection). When row and column address words consist entirely of *don't cares* then all APEs are selected. This is useful for global parameter extraction and control feedback.

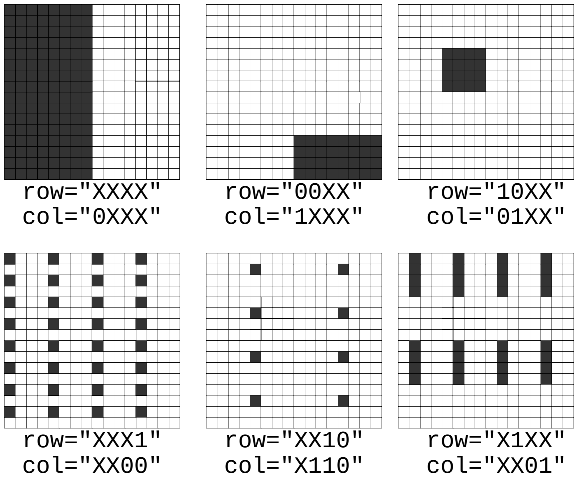


Fig.2. Addressing examples. For illustration a 16×16 array is shown with addressed APEs marked dark. The address (0,0) points to the bottom left corner of the array.

Mode 2 (Block selection). When *don't cares* are used at the *n* least significant bits of row address and the *m* least significant bits of column address, this correspond to the simultaneous selection of a block of 2^n rows and 2^m columns of APEs. This mode can be used, for example, to provide multiresolution output with pixel binning. It can also be very effectively used for determining (x,y) object locations in a visual target tracking algorithm.

Mode 3 (Periodic selection). When *don't cares* are used at the *n* most significant bits of the *N*-bit row address and the *m* most significant bits of the *M*-bit column address, this corresponds to the selection of multiple APEs, at locations every 2^{N-n} rows and 2^{M-m} columns of the array. This mode is typically used for processor clustering. The processor array may be configured as an SIMD array of "virtual PEs", each comprising a cluster of APEs. This feature can be used for multiresolution image processing, or to increase the number of registers available in each PE, however the detailed discussion of processor clustering is outside the scope of this paper.

3.5. Selected Input

The flexible addressing system can be also used in conjunction with a global input signal. While some of the APEs are addressed, their local analogue bus is connected to the global analogue bus. This feature can be used to selectively enable the addressed processors or to replace a local value in selected processors with a global value.

4. Application example

Basic low-level image processing algorithms, such as convolutions with 3×3 kernels, median filtering, Sobel edge detection, etc. are easy to implement and efficient to execute on a pixel-per-processor SIMD array [7]. Here we will present an example of a simple algorithm that highlights some of the features of the flexible global readout architecture. The aim is to extract the (x,y) coordinates of a target object. The algorithm we use is clearly not optimal, but our intention here is to illustrate the functionality of our system rather than to provide a viable solution to the target tracking problem. The intermediate results at various steps in the algorithm are shown in Fig.3. These results were obtained from the simulation of the SCAMP architecture, with 128×128 array size. The SCAMP simulation and software development tool has been written in C and includes behavioural modeling of analogue processing errors associated with switched-current APEs. An indication of the execution time is given in the description below, assuming the processors operate at 2MHz.

Step 1. An image is obtained from the photodetector. Automatic exposure control is provided by adjusting the integration time depending on the global sum of all pixel values. A software-macro implementing correlated double sampling is used to reduce the fixed pattern noise. An example input image is shown in Fig.3a.

Step 2. In order to threshold the image, an optimum threshold level can be selected by the host processor based on the histogram of the image. A histogram is easily obtained using the global sum capability of the chip. For each histogram bin, grey-level pixel values are compared with the bin boundary values. These APEs which have pixel values within the bin range (e.g. pixels with intensities within the 15th bin of a 16-bin histogram are shown in Fig.3b) have a register loaded with a constant, which acts as a "counting unit". A global sum is read-out from the array and it corresponds to the pixel-count for this particular histogram bin. The execution time for the full image histogram with 16-bins is 102μs. Thresholding is then performed in a few instruction cycles.

Step 3. In this simple example we aim to track a single object present in the image. We use a wave-propagation-type algorithm to find a pixel within the object, located furthest away from the object boundary (the coordinates of this pixel will be considered to represent the location of the object). A binary morphology operation of erosion is applied iteratively, stripping off the outside pixels, while global control feedback ensures that this operation stops just before all the pixels belonging to the object are removed (see Fig.3d and 3e). At the end of this step, the object is reduced to

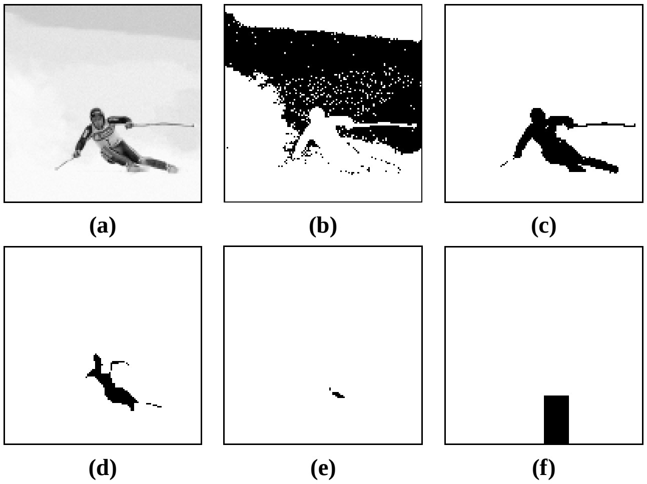
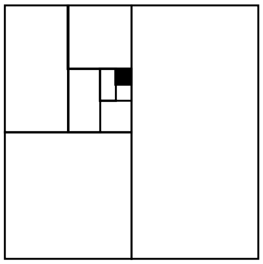


Fig.3. Intermediate results of the tracking algorithm. (a) the input image, (b) map of active APEs during the histogram bin count (pixel intensity between 14/16 and 15/16 of max value), (c) the thresholded image, (d) the image after two iterations of erosion in Step 2, (e) the object reduced to a few pixels at the end of Step 2, (f) selected APEs during the 5–th step of the successive approximation.



(a)

step	row	col	OR
1	XXXX	0XXX	1
2	0XXX	0XXX	0
3	1XXX	00XX	0
4	10XX	01XX	1
5	10XX	010X	0
6	100X	011X	0
7	101X	0110	0
8	1010	0111	0
result:	**1011**	**0111**	

(b)

Fig.4. The “search path” in a successive approximations of a single-pixel location for a 16×16 image: (a) diagram illustrating array bisections, (b) table showing address words and global OR result in each step.

a single pixel (or a number of pixels located at an equal, maximum distance from the object boundary)

Each wave-propagation step takes 7μs and the number of iterations depends on the size of the object. The total execution time of this algorithm, for the image in Fig.3, is equal to 45μs.

Step 4. The (x,y) coordinates of the pixel are extracted from the binary image using an algorithm based on a search by bisection exploiting the flexible addressing scheme. This algorithm is illustrated in Fig.4, assuming that only one pixel in the image has value ‘1’, other pixels are equal to ‘0’. Initially the search area is set to a half of the image, by setting all row and column address bits to ‘X’ (don’t care), except for the most significant bit of the column address which is set to ‘0’. Then, the global OR operation is performed, and if its result is equal to ‘1’ (i.e. a pixel was found inside the search area), this means that the most significant bit of the column address of the pixel must be equal to ‘0’. Otherwise, if the result of the global OR operation is equal to ‘0’ (i.e. no pixel was found inside the search area), this means that the most significant bit of the column address of the pixel must be equal to ‘1’. The search is then repeated over a quarter of the image, by setting the most significant bit of the row address to ‘0’, and inverting it if no pixel is found as a result of performing the global OR operation. This “spatial successive approximations” are repeated for the consecutive bits of the column and row addresses, until the result converges on the pixel. If several pixels are present in the image, as is the case in Fig. 3e, then this algorithm converges on one of the pixels. (For multiple target tracking the first found target could be then removed from the image and the entire process repeated). The entire process, for 7-bit row and column addresses, is completed within 21μs.

It is worth noting that the only data that has to be read-out from the vision chip to the host processor are the 16 bytes representing the histogram values (in Step 2) followed by the 14 bits that specify the row and column location of the object (in Step 4). Potentially, this algorithm, with object size as in Fig.3, could be executed at over 5000 frames/sec (in practice the time required for image acquisition and the host processing time in Step 2 should also be taken into account).

5. Implementation & Conclusions

A prototype SCAMP-2 chip has been designed in a 3-metal 1-poly 0.35μm CMOS technology and submitted for fabrication. Each APE occupies less than 50μm×50μm silicon area. The 10mm^2 chip integrates an array of 49×39 APEs and the readout circuits, while the controller microprocessor, together with A/D and D/A and converters will be implemented off-chip. A subsequent fabrication of a larger array is planned. The SCAMP approach is computationally very efficient, in terms of performance/area and performance/power dissipation [8]. The performance figures of merit for this implementation are expected to surpass the previous implementation and will be reported at a later date (the experimental results should be available by May 2003).

The SCAMP-2 vision chip not only facilitates rapid execution of massively parallel low-level image processing algorithms, but also addresses the issue of global information extraction and data reduction, eliminating the need to output full-size images to the higher-level host processor. The flexible addressing scheme, together with the global summation and global logic operation capability, enable new algorithms with a potential for very high speed of operation, while at the same time requiring little silicon area overhead for the hardware implementation. The readout scheme implemented on SCAMP-2 can be readily used in other systems, as long as global operations are supported. It can be employed on analogue processor arrays with global summation capability as well as digital processor arrays with global ‘OR’ capability. Some of the features of the proposed readout architecture might even be applicable to special-purpose vision chips, although its full potential can only be exploited in a general-purpose, software-programmable system such as the SIMD processor array.

References

[1] M.Ishikawa, K.Ogawa, T.Komuro, I.Ishii, “A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing”, Proc. ISSCC’99, TP 12.2, 1999.

[2] F. Paillet, D. Mercier, and T.M.Bernard, “Making the most of 15kλ^2 silicon area for a digital retina”, Proc. SPIE, Vol. 3410, AFPAEC’98, 1998

[3] G. Liñán-Cembrano et al. “A Processing Element Architecture for High-Density Focal-Plane Analog Programmable Array Processors”, Proc. International Symposium on Circuits and Systems, ISCAS 2002, May 2002, vol.III, pp.341-344.

[4] P.Dudek and P.J.Hicks, “An Analogue SIMD Focal Plane Processor Array”, Proc. International Symposium on Circuits and Systems, ISCAS 2001, May 2001, vol.IV, pp.490-493.

[5] G.Indiveri, P.Oswald and J.Kramer, “An Adaptive Visual Tracking Sensor with a Hysteretic Winner-Take-All Network”, Proc. ISCAS 2002, May 2002, vol.II, pp.324-327.

[6] A.Fish, D.Turchin and O.Yadid-Pecht, “An APS with 2-D Selection Employing Adaptive Spatial Filtering, Bad Pixel Eleimination and False Alarm Reduction”, Proc. ISCAS 2002, May 2002, vol.II, pp.328-331.

[7] P.Dudek and P.J.Hicks, “A General-Purpose CMOS Vision Chip with a Processor-Per-Pixel SIMD Array”, Proc. European Solid State Circuits Conference ESSCIRC 2001, September 2001, pp.228-231

[8] P.Dudek and P.J.Hicks, “A CMOS general-purpose sampled-data analog processing element”, IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing, vol. 47, no. 5, May 2000, pp. 467-473