Novel Image Processing Architecture for 3D Integrated Circuits

Benjamin Pfundt[1], Marc Reichenbach[1], Christopher Söll[2], Dietmar Fey[1]

[1]Chair of Computer Architecture
Department of Computer Science
[2]Institute for Electronics Engineering
Department of Electrical, Electronic and Communication Engineering
Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Germany
{benjamin.pfundt, marc.reichenbach, christopher.soell, dietmar.fey}@fau.de

Abstract: Utilizing highly parallel processors for high speed embedded image processing is a well known approach. However, the question of how to provide a sufficiently fast data rate from image sensor to processing unit is still not solved. As Trough-Silicon-Vias (TSV), a new technology for chip stacking, become available, parallel image transmission from the image sensor to processing unit is enabled. Nevertheless, the usage of a new technology requires architectural changes in the processing units. With this technology at hand, we present a novel image preprocessing architecture suitable for image processing in 3D chips stacks. The architecture was developed in parallel with a customized image sensor to make a real assembly possible. It is fully functionally verified and layouted for a 150 nm process. Our performance estimation shows a processing speed of 770 up to 14.400 fps (frames per second) for 5×5 filters.

1 Introduction

Due to the continuously rising performance requirements in image processing systems, novel approaches in architecture design are desperately needed. One solution to fulfill these requirements is the processing or at least preprocessing of the captured image near to the sensor. For that reason, image sensor and processing unit will be connected together, which is the idea behind *smart cameras*. Due to the fact that image processing algorithms are generally easily parallelizable, a high performance can be achieved in the domain of *smart cameras* with a well designed parallel processing architecture. Nevertheless, a common problem with high speed data acquisition frequently occurs: while capturing and processing of the image can be executed in parallel, the data link in between is mostly designed using serial links. This slows down the processing and limits the possible degree of parallelism and therefore performance in the processing architecture.

To overcome this issue, a paradigm shift from smart cameras to *smart sensors* is needed. This can be achieved by integrating processing structures in or very close to the sensor. A straightforward implementation is to construct a SIMD array of processing elements

(PEs) and assign it to one or more pixel cells. Especially local processing algorithms profit from these fine grained processor arrays because data exchange to and from neighboring elements only requires additional wires in the simplest case. Also, specialized high speed and resource consuming transmission logic for high volume raw sensor data can be dropped. Still, a low latency is achieved as sensor data is directly read by the processing elements. Due to the massively parallel transmission, a high bandwidth is possible if all processing elements are considered while the elements themselves could have a low processing frequency. Furthermore, a large on-chip storage can be omitted as processing elements only operate on few pixels.

Though these apparent advantages, major drawbacks arise at the IC design level. If an array of elements consisting of photo diodes and processing logic is created, only a very low fill factor can be achieved. Due to the extra size of the processing logic and analog to digital converter, the pixel size strongly increases. This results in a low sensor resolution and limits the practical use. Solving this problem by splitting pixel and processing leads to other drawbacks, e.g. a massive increase in wiring complexity or a large footprint.

A promising approach to bypass planar layout problems is vertical chip stacking. Several chips with different functions are stacked upon each other and are connected by a multitude of through silicon vias (TSVs). Figure 1 illustrates an example stack: photo diodes, ADCs, processing logic and memory could be placed on separate layers. The result is a smart image sensor chip stack with a much smaller footprint compared to a planar design, yet offering the possibility to increase the bandwidth between pixel and processing array. The interconnect length decreases while a large number of connections can be implemented as the diameter of TSVs can be as small as $1\mu m$ [Tor13]. Furthermore, chips from heterogeneous technologies can be stacked and troublesome mixed signal designs can be avoided. The possibilities of 3D chip stacking were recognized early on. First concrete ideas for processing schemes [Tan85] and also simple stacked IC designs [NIS$^+$87] are nearly as old as monolithic chip designs of sensor and processing logic.

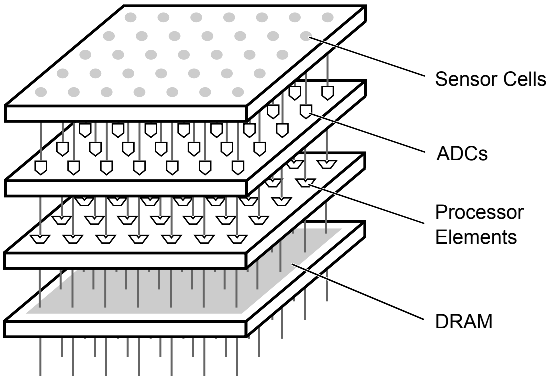

Figure 1: Heterogenous Chip Stack with Massive Parallel Transmission

Although, the benefits of 3D ICs for processing raw image data close to the sensor are at hand, a digital image processing architecture which harnesses this potential is not straightforward. We therefore propose a parallel image preprocessing architecture in this paper which can be connected via TSVs to an image sensor. This image sensor is currently developed in close cooperation with our partners at LTE [SSB+15]. The architecture is functionally verified and completely layouted for productive use.

Over the years, different architectures have been proposed. Hence, we will first provide a short overview about recent designs in the next section. Based on the shortcomings, we will then develop a novel processing scheme for image preprocessing in Section 3. Our goal is to outweigh performance and resource usage to design an efficient low power smart sensor for embedded purposes. The actual realization as ASIC is presented in Section 4 and the results are discussed afterwards. Finally, Section 6 offers a conclusion and addresses further enhancements.

2 Related Work

The topic of integrating photo sensor and processing capabilities into a single chip has been investigated for decades and many designs have been proposed. Among the early monolithic chips are designs based on cellular automata. In [GZD85] an 8×8 array of elementary processors easily extendable to 256×256 elements was proposed. Each elementary processor was assigned to one photodiode and could perform combinatorial operations on its and the neighboring pixel's values. Also in optoelectronic processing close to the sensor has been put forward. In this domain the term *smart pixel* was coined to describe a hybrid design of optical devices, e.g. photo diodes, and electronics for processing [Hin88].

The designs of image processors for vertical integration generally split into three categories. One approach is similar to earlier smart pixels or cellular automata designs and uses pixel parallel processing arrays [RVCGFB+14, LD11]. In addition to this, there exist designs with larger processing units. These units are assigned to a large portion of pixels or can even work on the whole image [DFJAM14, CHF+12]. Additionally, a combination of both approaches has been suggested in [SBP+12].

With respect to architectural complexity, the easiest way to design 3D image processors is to use a SIMD array with a processing element for each pixel. In-pixel ADCs on a different layer than the photo diodes have been successfully manufactured in [GHI+14], however the resolution of 64×64 was rather low. A multiple layer chip stack is proposed in [LD11] offering cellular automata operation for 128×96 pixels. Although the digital processing part has been split on two layers of $25mm^2$ each, the achievable fill factor is still low.

If a large portion of pixels is to be processed by a single or a couple of processing units, one main advantage of processing close to the photo diodes diminishes: the exchange of values between neighboring pixel areas cannot be achieved without storing the whole or portions of the image. In [DFJAM14] a 48×32 sensor array is introduced where a column-wise computation takes place. No data exchange between array elements is possible and the operations are very limited. A much more complex design has been proposed in [CHF$^+$12] where a large multi-core chip of $63mm^2$ accommodates eight RISC processors. The digital layer should be connected to a partitioned layer of ADCs which in turn should be connected to a tiled image sensor with a resolution of 2048×1536. The introduced digital layer has two SDRAM controllers for external memory. A large memory is needed to calculate even simple neighborhood operations like 2D filters or stencil codes. Most probably, only a fraction of the actual computational power can be obtained for bandwidth bound problems.

Both extremes, pixel-parallel and large scale computation, have their disadvantages. Therefore, we pursue an image preprocessing architecture for 3D stacking which provides a balanced mixture of parallel computation and chip utilization as well as resource utilization.

3 Architectural Conception

In this section we introduce a fine grained parallel architecture which provides data exchange between sensor elements at a minimum of additional resources in form of specialized buffer structures.

3.1 Overall Layout

Vertical interconnections between different IC layers influence the coupling and also the architectural layout of each layer. The smart pixel and pixel parallel approaches had one ADC per photo diode. If more photo diodes shared one ADC, a homogeneous distribution would not be possible. For the application domain of cellular automata and image filters, a one-to-one ratio between ADCs and PEs leads to a simple logic layer. For every pixel and its neighbors the respective operation has to be carried out. If more pixel are feed into one PE, the exchange and storage becomes more complex. Although a one-to-one ratio is straightforward, the main drawback is the space consuming ADC which eventually causes low fill factors. Therefore, the goal has to be to reduce the number of ADCs and use a more traditional approach where photo cells are read out row by row and column by column. The number of ADCs can be increased if a couple of pixels per row are read out simultaneously. This can be achieved if the output of the column multiplexer of an off-the-shelf CMOS image sensor is enlarged as Figure 2 illustrates. The fill factor is not changed, as the ADCs are not located inside the pixel cells.

3.2 Partitioning

High sensor resolutions require many ADCs to achieve decent frame rates. For parallel mask operations, pixels of coherent image regions have to be converted. This has two main disadvantages for high resolutions. First of all, neighboring pixel cells have to be converted simultaneously and therefore connected to different ADCs. This increases the

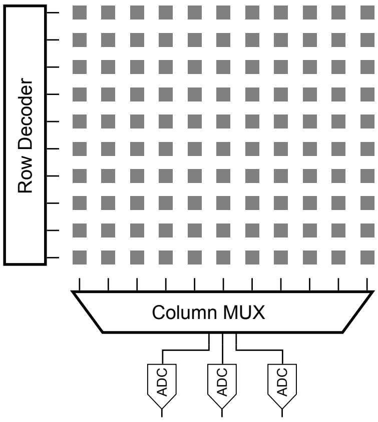

Figure 2: Image Sensor with Parallel ADCs

wiring complexity dramatically for a large number of ADCs. Secondly, the number of PEs has to be adjusted to the number of ADCs. This leads to many stores and loads as previous pixel cells have to be temporarily stored if they should be reused again. To cope with this problem, we propose a partitioning scheme where the sensor is split into rectangular tiles. Each tile has its ADCs and is connected to an array of PEs. The number of ADCs and PEs is decreased with the lower resolution per partition. This greatly relieves the wiring complexity and limits the local memory traffic, while the overall frame rate remains constant.

As 2D filter operations also include neighboring pixels, communication has to take place across partition borders. Due to local masks requiring only a small image region, just a portion of the partition's pixel data has to be held in memory. Therefore, we propose a partitioning sequence which is depicted in Figure 3. The pixel cell read out starts in the middle of the image sensor and proceeds to the opposite end of the partition either meandering or line by line. The starting point could also be at the corners of the image sensor as long as the read out proceeds similarly in each partition. The current pixel values are held inside local buffers. Due to the processing order, it is ensured that PEs at partition border can access data elements from other partition as they are currently held in the buffers of an other PE array. All PE arrays can be directly exchange the appropriate data.

A further advantage results from the partitioning scheme. Besides the configuration possibilities, the partitions can be reused. The system becomes easily scalable if the constraints for a new design are changed. Thus, our architecture is highly configurable and can be exactly adjusted to application and image sensor constraints. The result is a light-weight and balanced system which efficiently employs the resources used.

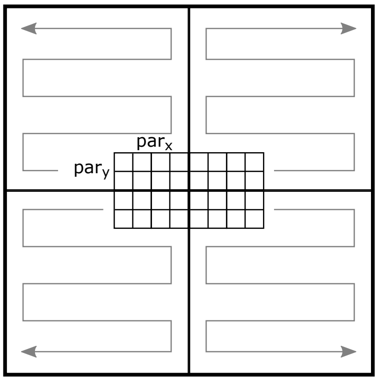

Figure 3: Processing Scheme for Synchronous Data Exchange Between Partitions

3.3 Processing Scheme

As the rows of the sensor are read out successively, pixel values have to be stored to allow operations which need the neighboring pixel values. If only a CPU is available, the whole image is commonly stored in a RAM utilizing double buffering. This results in a high power consumption due to a large RAM and access is slowed down by additional latencies. A more resource efficient approach is to process the image data on the fly while it is streamed out of the image sensor. With a CPU this could be achieved utilizing circular buffer structures. For our target applications, e.g. 2D filters, an even more light-weight custom implementation is possible which will be presented in the next paragraphs.

On-the-fly processing of 2D filters and other mask operators can be efficiently realized utilizing line buffers in a full buffering scheme. A scalable full buffering architecture for FPGAs was presented in [SRF12]. A processing scheme for 3D chip stacking can be devised similarly. The basic structure for 3×3 masks is illustrated in Figure 4. Pixel value transmission from the ADCs goes directly into registers. The array of registers holds all data elements which are needed to carry out the operations by PEs in parallel. Larger storage elements, e.g. SRAM blocks, are used as line buffers to store exactly the number of previous pixel values which are needed for further calculations. After the PEs have finished their calculation, a line buffer behaves like a FIFO. Newly received pixel values will replace older elements which are in turn feed into the PE registers.

Parallelism for full buffering structures can be increased in two ways. On the one hand, the number of PEs can be increased which is is limited by the number of possible ADCs. As solution a demultiplexer could be introduced after the ADCs to serve more PEs. Then, the one-to-one relation between ADCs and PEs had to be modified according to the respective operation frequencies. On the other hand, parallelism can be implemented by building several stages of full buffering structures. The parallel output of one structure will then be feed into the next one. Depending on the application, the number of PEs per stage can vary as long as a synchronization mechanism between the stages is applied.

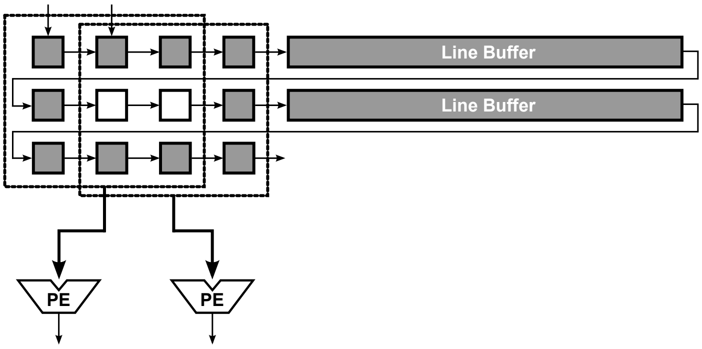

Figure 4: Full Buffering Structure with Dual PE

4 Implementation

The architectural concept has been implemented in VHDL to produce a real chip stack. In cooperation with our university partner, we developed the constraints for a two layered smart sensor. The final sensor chip will consist of a partitioned photo sensor with a resolution of 216×216. Nine pixels per partition can be accessed in parallel and are feed into an analog processing unit for basic 3×3 filter masks [SSB$^+$15]. The uneven sensor resolution is a multiple of the parallel accessible pixels. In the final design either 9 parallel 16 bit ADCs will transform the raw pixel data or only one will convert the calculated value.

The transmission via TSVs to the digital processing part is done bit-parallel. As up to nine pixel values can be transmitted, the same number of PEs has been implemented. For the digital part a window size of 5×5 was implemented. This leads to a minimum of 144 bit of parallel in- and outputs per partition. If available, the output could be stored in a DRAM layer. Otherwise a deserializer is placed at the outputs to limit the number of pins.

4.1 Layout

For VHDL synthesis we used *Design Compiler*® from Synopsys. The IC layout has been generated with *Encounter*® from Cadence. As the design tool support for 3D ICs is still not mature, the tool support for ball grid arrays (BGAs) is used as workaround [Tor13]. The BGAs is placed and assigned in at the place and route step. In a further layout step the actual TSV cells are placed. We passed through the design flow with a 150 *nm* from LFoundry. The diameter of the TSV cells used is $1.2\ \mu m$ at a $10\ \mu m$ pitch. The complete design flow including the TSV assignment has been scripted and can therefore be easily rerun with different parameters.

4.2 Structure

The implemented architectural structure can be seen in Figure 5. Besides the partition unit, the architecture consists of five major building blocks which will be described in the next paragraphs.

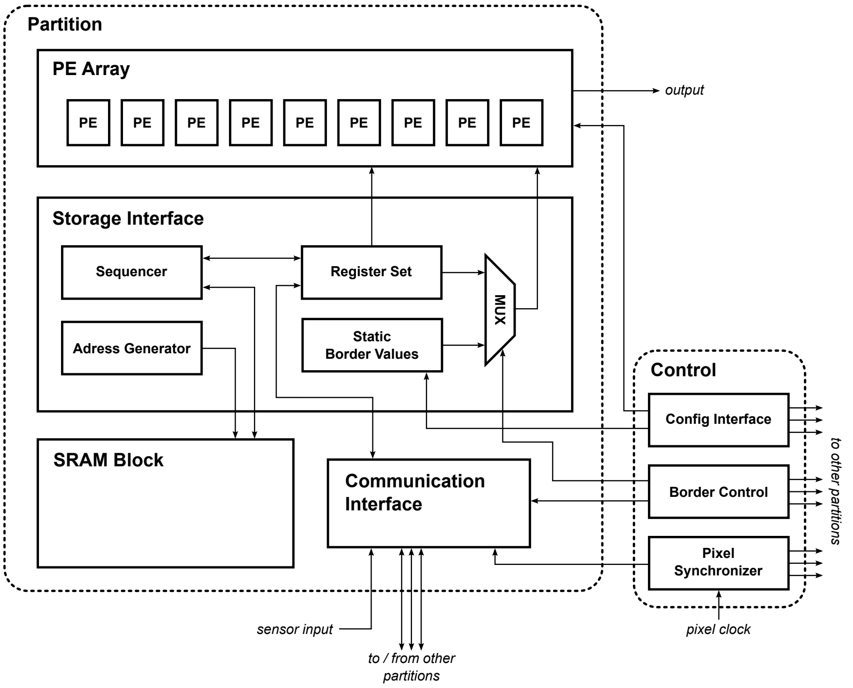

Figure 5: Final Architecture of Digital Processing Layer

Control The partitions have to be configured and controlled to work together, this is done by the separate unit *Control* which is connected to each partition. The component *Config Interface* includes the setting of the static border values needed for mask operations at the image sensor edges. Another task is to configure operation modes for the PEs. The second component *Border Control* uses an internal counter to indicate if a communication with an other partitions has to take place or if the static border values have to be used. Finally, the *Pixel Synchronizer* monitors the *pixel clock* from the sensor to indicate if new pixel values have been converted by the ADCs.

Communication Interface One main unit which finally joins the partitions is the *Communication Interface*. This interface picks up new sensor data as well as reroutes and controls the data flow from and to other partitions.

PE Array The main work is done by an array of PEs. In the current design, a single cycle mean filter is implemented for performance and comparability reasons. Thus, it is possible to compare the analog and digital implementation in size, processing speed and accuracy. Other filter or local operator can be implemented easily, as the *PE Array* interface is rather generic and provides all inputs in parallel. Similar to the pixel clock, a signal can be activated to indicate if a potential multi cycle PE calculation is finished.

SRAM Block For ASIC designs the line buffers have to be placed in SRAM blocks. To save resources we placed all line buffers of a partition in one *SRAM Block* with a single port interface. The storage requirement in bits resulting from the full buffering structure of Figure 4 can be directly calculated with Equation 1. For our design, this results in a minimum size of 6080 bits.

$$\begin{aligned} mem \;\geq\; & (partition\_width - \#PE - mask\_size + 1)\;\ldots \\ & \times (mask\_size - 1) \times resolution \end{aligned} \tag{1}$$

Storage Interface A more complex unit which controls the storage and provides the appropriate data for partition exchange is the *Storage Interface*. The static border values and the current registers are implemented and appropriately connected to the *PE array*. Particular registers have to be substituted, if pixels at partition edges are processed and the mask reaches across the border. A *Sequencer* and *Adress Generator* map the four line buffers of the current design to the single ported SRAM.

5 Results

The final layout of the digital IC in 150 *nm* technology is displayed in Figure 6. The SRAM cells were placed close to the power rings to provide sufficient power supply. The possible positions of the TSVs can be recognized by the uniform grid, though not every position is actually assigned. Approximately 2.5 % of the over 47000 positions were used for data pins. Further chip characteristics are shown in Table 1. The rectangular chip dimensions which does not perfectly fit to a rather quadratic image sensor is owed to the SRAM blocks. As the layer dimensions of a 3D IC do not have to fit exactly, this is no real problem. The size of the digital chip might be enlarged without affecting the functionality.

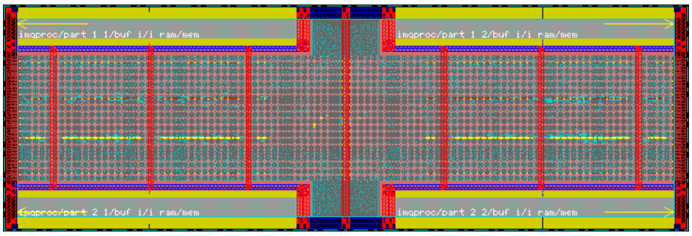
imgproc/part 1 1/buf i/i ram/mem
imgproc/part 1 2/buf i/i ram/mem
imgproc/part 2 1/buf i/i ram/mem
imgproc/part 2 2/buf i/i ram/mem

Figure 6: Layout of Chip Stacking Enabled Digital Processing ASIC

The IC runs at 40 MHz which is nearly at the limit of the SRAM blocks. If the analog part could operate at half the speed to respect an additional cycle for the *Sequencer*, a frame rate of $9\,pixels \times 20\ MHz/(108\ \times\ 108\,pixels/frame) \approx 15.430$ *fps* would be possible. This would result in an overall bandwidth of approximately 1.44 *GByte/s*. But even for a very pessimistic pixel clock of 1 *MHz*, 770 *fps* without analog processing are possible. These values clearly demonstrate the practical advantages of 3D ICs for smart sensor application. With a low power consumption and moderate clock speeds, high frame rates can be achieved.

Table 1: Digital IC Properties

Property	Value
Chip Area	$1.5 \times 5.0\,mm^2$
Density	61%
Voltage	$1.8\,V$
Estimated Core Power	$60\,mW$

6 Conclusion and Outlook

Based on the technological possibilities of TSVs, chip stacking is at hand. Especially image processing architectures can benefit from these new developments to overcome the problem with serial transmissions between image sensor and processing unit. Therefore, we presented in this paper a novel image processing architecture for 3D chip stacks. The proposed design exhibits a high degree of parallelism. Firstly, subsequent pixels are processed in parallel. Secondly, due to a distribution of ADCs at the image sensor, the image is divided in four partitions for parallel processing. To avoid external memory, line buffering is used. Data exchange between partitions is implemented which enables a high flexibility for possible extensions to more partitions.

Although the chip is fully layouted and functionally verified, it still has to be manufactured and field tested. Due to rare 3D design kits, we will create a 3D chip stack prototype together with Lfoundry Srl. for a new generation of image acquisition and processing systems. In the near future, we want to connect our processing chip with the image sensor, which is developed by our colleagues.

Acknowledgements This work was financially supported by the Research Training Group 1773 “Heterogeneous Image Systems” funded by the German Research Foundation (DFG). Additionally, the authors would like to thank Andreas Bänisch for his valuable hints and comments.

References

[CHF$^+$12] C. Cheng, H. Hsieh, T. Fan, W. Tang, C. Liu, and P. Huang. High Resolution and Frame Rate Image Signal Processor Array Design for 3-D Imager. In *Interna-*

tional Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), 2012, pages 735–739, Nov 2012.

[DFJAM14] M. Di Federico, P. Julian, A.G. Andreou, and P.S. Mandolesi. Fully Functional Fine-grain Vertically Integrated 3D Focal Plane Neuromorphic Processor. In *SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S), 2014*, pages 1–2, Oct 2014.

[GHI+14] M. Goto, K. Hagiwara, Y. Iguchi, H. Ohtake, T. Saraya, M. Kobayashi, E. Higurashi, H. Toshiyoshi, and T. Hiramoto. Three-Dimensional Integrated CMOS Image Sensors with Pixel-Parallel A/D Converters Fabricated by Direct Bonding of SOI Layers. In *International Electron Devices Meeting (IEDM), 2014*, pages 4.2.1–4.2.4, Dec 2014.

[GZD85] P. Garda, B. Zavidovique, and F. Devos. Integrated Cellular Array Performing Neighborhood Combinatorial Logic on Binary Pictures. In *11th European Solid-State Circuits Conference, 1985. ESSCIRC '85.*, pages 58–63, Sept 1985.

[Hin88] H.S. Hinton. Architectural Considerations for Photonic Switching Networks. *IEEE Journal on Selected Areas in Communications*, 6(7), Aug 1988.

[LD11] A. Lopich and P. Dudek. Architecture and Design of a Programmable 3D-Integrated Cellular Processor Array for Image Processing. In *19th International Conference onVLSI and System-on-Chip (VLSI-SoC), 2011*, pages 349–353, Oct 2011.

[NIS+87] T. Nishimura, Y. Inoue, K. Sugahara, S. Kusunoki, T. Kumamoto, S. Nakagawa, M. Nakaya, Y. Horiba, and Y. Akasaka. Three Dimensional IC for High Performance Image Signal Processor. In *International Electron Devices Meeting, 1987*, volume 33, pages 111–114, 1987.

[RVCGFB+14] A. Rodriguez-Vazquez, R. Carmona-Galan, J. Fernandez Berni, S. Vargas, J.A. Lenero, M. Suarez, V. Brea, and B. Perez-Verdu. Form Factor Improvement of Smart-Pixels for Vision Sensors through 3-D Vertically-Integrated Technologies. In *Circuits and Systems (LASCAS), 2014 IEEE 5th Latin American Symposium on*, pages 1–4, Feb 2014.

[SBP+12] M. Suarez, V.M. Brea, F. Pardo, R. Carmona-Galan, and A. Rodriguez-Vazquez. A CMOS-3D Reconfigurable Architecture with In-pixel Processing for Feature Detectors. In *International 3D Systems Integration Conference (3DIC), 2011*, pages 1–8, Jan 2012.

[SRF12] M. Schmidt, M. Reichenbach, and D. Fey. A Generic VHDL Template for 2D Stencil Code Applications on FPGAs. In *15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012*, pages 180–187, April 2012.

[SSB+15] L. Shi, C. Soell, R. Baenisch, A. Weigel, J. Seiler, and T. Ussmueller. Concept for a CMOS Image Sensor Suited for Analog Image Pre-Processing. In *DATE Friday Workshop on Heterogeneous Architectures and Design Methods for Embedded Image Systems (HIS), 2015*, pages 16–21, March 2015.

[Tan85] K. Taniguchi. Three Dimensional IC's and an application to High Speed Image Processor. In *7th Symposium on Computer Arithmetic (ARITH), 1985*, pages 216–222, June 1985.

[Tor13] K. Torki. 3D-IC Integration. In *CMP annual users meeting*, Paris, Jan 2013.