A Runtime-Reconfigurable Operand Masking Technique for Energy-Efficient Approximate Processor Architectures

Moritz Weißbrich*, Alberto García-Ortiz†, Guillermo Payá-Vayá*
*Institute of Microelectronic Systems, Leibniz Universität Hannover, Germany
Email: {weissbrich, guipava}@ims.uni-hannover.de
†Institute of Electrodynamics and Microelectronics, Universität Bremen, Germany
Email: agarcia@item.uni-bremen.de

Abstract—**In this paper, an operand masking approach is proposed to achieve lower energy consumption using approximate computing techniques in programmable high-performance processors, in this case horizontal and vertical SIMD vector processors for embedded computer vision applications. Contrary to state-of-the-art dedicated approximate arithmetic circuits, this mechanism enables programmable fine-grained accuracy control and switching energy reduction at runtime. An evaluation for a 45 nm ASIC technology shows a total effective energy reduction of up to 4.5% for a horizontal SIMD vector processor architecture executing approximate SIFT image feature extraction for an error-resilient egomotion estimation algorithm.**

Index Terms—**Approximate Computing, Processor Architectures, Energy Efficiency, Feature Extraction, Error-Resilient Applications**

I. Introduction

For embedded automotive computer vision applications like real-time motion tracking and scene reconstruction [1], high image processing performance is required within a heavily constrained energy budget. In most cases, programming flexibility for software updates and algorithmic extensions is demanded. This limits the use of fully dedicated hardware accelerators and leads to processor-based systems with a considerably higher energy consumption.

In the last decade, approximate computing hardware mechanisms have emerged as a possible solution for energy-efficient computations in error-resilient algorithms [2]. These mechanisms are aimed at reducing computational accuracy in favor of energy savings. For commonly-used adder and multiplier circuits, numerous approximate hardware architectures have been proposed and compared [3], most of them being parameterizable during the design phase to adapt accuracy and energy consumption to the requirements of a specific application. However, state-of-the-art approximate arithmetic hardware does not offer the necessary runtime reconfigurability to cope with changing accuracy requirements in programmable processor architectures, which may occur due to later software extensions, different accuracy requirements in sub-algorithms running on the same processor datapath, or the execution of different separate programs. If necessary, precise operations must be supported next to approximate computations without the need of implementing separate precise and approximate functional units, causing silicon area and power overhead.

In this paper, an approximate *operand masking* approach is proposed and applied to high-performance horizontal and vertical SIMD vector processors to reduce the power consumption. By using conventional precise arithmetic circuits with runtime-reconfigurable approximated operands, a fine-grained accuracy selection at bit-level is achieved. The required energy per arithmetic operation is lowered by decreasing the switching activity in the circuit. The effect is comparable to a reduced datapath width, however, the hardware implementation is not application-dependent and the accuracy-energy design space can be reconfigured deliberately by the programmer at runtime. Moreover, application code modifications are not required, apart from using special instructions to set the accuracy level, which allows effortless integration into existing software.

The paper is structured as follows: Section II introduces a reference application from related work, which will be used to evaluate the proposed masking mechanism. In Section III, the hardware implementation of the approximate vector processors is presented. Evaluation results and the discussion are provided in Section IV. Section V concludes the paper.

II. Related Work: Egomotion Estimation

One of the key problems in automotive computer vision applications is image interpretation and scene understanding. For this paper, *egomotion estimation* [1] is used as a reference application from related work for evaluating the proposed approximation technique. Egomotion estimation provides a method to reconstruct the 3D scene and measure the movement of a vehicle from a sequence of on-board stereo camera images. This algorithm is based on matching images features, as for example provided by the widely-known Scale-Invariant Feature Transform (SIFT) algorithm [4], by using a circular matching approach to trace image features. The basic idea of egomotion estimation is to match features across left and right stereo images to obtain a 3D position in the surrounding environment by using the horizontal disparity. The features are

This work was partly funded by the German Research Council (DFG) under project number PA 2762/1-1.

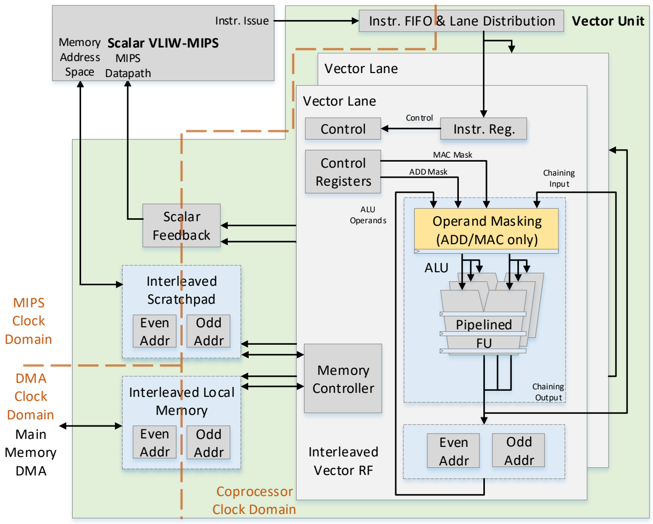

Fig. 1. Simplified block diagram of the vertical SIMD vector unit [5]. The position of the operand masking mechanism is highlighted.

also tracked over time by matching the positions in subsequent stereo image frames to obtain 3D movement of distinct feature points in the surrounding scene. By assuming that the feature points belong to static objects, the proper motion or *egomotion* of a moving vehicle relative to its environment is obtained only by evaluating on-board camera images. A detailed description of the geometric transformation from 2D feature points to 3D camera motion is out of scope of this paper and is addressed in [1]. The image feature extraction process involved in this algorithm is computational intensive and requires a high-performance, yet low-power processing platform to cope with real-time and energy constraints in vehicles.

III. Implementation of High-Performance Approximate Vector Processors

In the following, the datapaths of two vector processor architectures used to evaluate the proposed approximate masking mechanism are presented. The architectures employ either vertical or horizontal SIMD data-level parallelism and are designed as accelerating coprocessors for computational intensive image processing tasks. A 32-bit scalar main CPU, in this case a two-issue Very Long Instruction Word MIPS (VLIW-MIPS) architecture, is in charge of controlling the application flow and to issue vector instructions to the coprocessor. A detailed system description as well as performance and efficiency profiling have been published in previous work [5].

A. *Vertical and Horizontal Datapath Architectures*

Fig. 1 shows the simplified block diagram of the vertical SIMD vector unit. Vertical vectorization refers to the traditional vector processing approach of sequentially applying the same operation, coded in a single instruction, on multiple data elements by using a single ALU per datapath over multiple clock cycles. The datapath itself is called vector lane, which contains a vector ALU composed out of multiple independently pipelined functional units (FU, 1 to 5 pipeline stages), a register file (RF) for results and operands, as well as a memory controller to interface the local memory and processor scratchpad resources in the vector unit. One vector unit contains two vector lanes. The ALU result of one vector lane may be directly forwarded to the second one by using a chaining mechanism, which allows parallel execution of data-dependent vector instructions to increase the vector instruction throughput. To achieve maximum performance, the local memory, scratchpad and RF SRAM blocks are implemented as two separate memories with interleaved and overlapping even/odd address accesses. By this, the pipelined vector lane architecture is able to operate at 1667 MHz for a 45 nm ASIC technology without being constrained by slower SRAM blocks, which are used at a relaxed half-clock timing constraint of 833 MHz [5]. The proposed operand masking mechanism is implemented for the adder and multiply-accumulate (MAC) FUs to enable approximate additions, subtractions and multiplications, and is placed prior to the first ALU pipeline register stage to not adversely affect the timing. Control registers and instructions are used to control the mechanism at runtime.

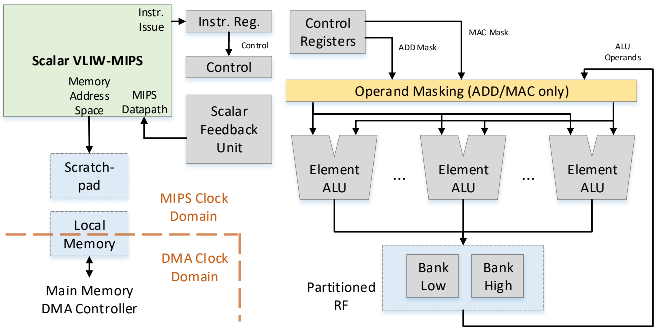

Fig. 2. Simplified block diagram of the horizontal SIMD vector datapath [5]. The position of the operand masking mechanism is highlighted.

In Fig. 2, the simplified block diagram of the horizontal SIMD datapath is depicted. Horizontal vectorization uses multiple ALUs in the datapath to process multiple data elements per clock cycle. All ALUs, the global vector RF, local memory and processor scratchpad resources are combined in a hierarchically flat architecture. The ALU supports a single-cycle MAC operation, requiring an additional accumulator operand from the register file. Instead of utilizing a less efficient multi-port monolithic RF, a two-bank partitioned vector RF is used [6], which is used like an ordinary two-read one-write port RF for most instructions as well as a four-read two-write port RF with bank access restrictions for the MAC operation. The horizontal SIMD datapath computes parallel results in one clock cycle. Due to the datapath simplicity, minimal two-stage pipelining with overlapping instruction decoding and operation execution is sufficient to achieve the VLIW-MIPS CPU clock frequency of 510 MHz for a 45 nm ASIC technology without specific timing optimizations [5]. Since the VLIW-MIPS can only issue a single vector instruction per cycle, maximum instruction throughput and computational

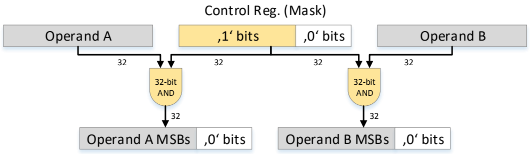

Fig. 3. Mode of operation of the proposed operand masking mechanism.

performance of the coprocessor is ensured. Therefore, the proposed masking mechanism can be combinationally placed at the adder and MAC FU inputs without timing implications.

For both vertical and horizontal SIMD architectures, the SIFT image feature extraction algorithm has been implemented for evaluating the proposed approximate vector processors. The algorithmic theory of SIFT is out of scope of this paper and is addressed in detail in [4]. Implementation details and an extensive application profiling for the given vector architectures have been published in [5].

B. Bitmasking-Based Approximate Mechanism

To enable fine-grained runtime-reconfigurable approximation, the operand masking technique illustrated in Fig. 3 is proposed. The 32-bit input operands for adder and MAC FUs of the ALU are modified by an inserted stage of AND gates, which is controlled by a mask register at bit level. If some LSBs of the control register are set to zero, the operand LSBs presented to the FUs will also be masked to zero, which reduces the switching activity and thus the power consumption in the following ALU circuit at the expense of reduced computational accuracy. For the SIFT implementation, 4 computational intensive algorithmic parts which may utilize approximation are identified, i.e., *scale-space pyramids construction (Gaussian and DoG)*, *keypoint orientation assignment*, *descriptor histogram binning*, and *descriptor normalization*. For each of these 4 parts, the masking control registers are set to defined parameters at runtime. Since there is independent masking of adder and MAC operands by using two control registers, the approximation behavior is described by a total of 8 mask parameters.

IV. Evaluation Results

In this section, the environment and metrics of evaluating the effectiveness of the operand masking technique will be presented first. Afterwards, the approximate vector processors will be evaluated in terms of masking circuit implementation and area overhead, approximate egomotion estimation accuracy as well as the overall energy reduction due to masking.

A. Evaluation Environment and Metrics

For evaluation, a vertical SIMD vector processor configuration with 8 vector units and a horizontal SIMD vector processor with 8 parallel 32-bit ALUs (256 bit vector width) are selected. The SIFT algorithm executed on these processors utilizes a 13.19 fixed-point format for all calculations, so the proposed operand masking reduces the effective number of fractional bits. For computational accuracy evaluation, both processors are emulated on the Xilinx ML605 (Virtex-6) FPGA evaluation platform [7] at 20 MHz. The emulated approximate processors running SIFT are embedded into the egomotion estimation application flow by receiving image frames from a connected host PC and passing back the extracted image features and descriptor vectors. The circular matching of keypoints, the motion estimation itself and generation of error metrics are implemented on the host PC. As an image sequence, the automotive scene *2011_09_26_drive_0005* (160 stereo image frames) from the KITTI dataset [8] is selected. The processors are implemented as ASIC netlists using a synthesis flow for a 45 nm standard cell technology. Switching activity simulations are performed for selected operand masking parameter sets. Due to the long runtime of these simulations, only a single image frame is processed to estimate the overall energy reduction.

To evaluate the quality of approximate egomotion estimation, the *mean relative error of translation velocity* ($\mathrm{MRE}_{\mathrm{trans}}$) and the *mean absolute error of camera pose change* ($\mathrm{MAE}_{\mathrm{pose}}$) metrics are introduced:

$$\mathrm{MAE}_{\mathrm{pose}} = \frac{1}{N}\sum_{i=1}^{N} |\vec{r}_{\mathrm{ref}}(i) - \vec{r}(i)| \tag{1}$$

$$\mathrm{MRE}_{\mathrm{trans}} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\vec{v}_{\mathrm{ref}}(i) - \vec{v}(i)|}{|\vec{v}_{\mathrm{ref}}(i)|} \tag{2}$$

N is the total number of frames of the evaluated video sequence, and $\vec{r}$ and $\vec{v}$ are the camera pose change in rad/Frame and the vehicle translation velocity in $\mathrm{m\,s^{-1}}$ for each sequence frame i, respectively. The golden reference values $\vec{r}_{\mathrm{ref}}$ and $\vec{v}_{\mathrm{ref}}$ are calculated from GPS data provided with the KITTI dataset. These metrics do not compare trajectory endpoints of the estimated motion, but averaged errors per frame. The reason is that endpoint errors are dependent on the frame where an error occurred, which may distort the error results.

B. Evaluation of ASIC Implementation

Fig. 4 shows the circuit area overhead of both evaluated approximate processors. Due to the additional operand masking mechanism, the total cell area is increased by 2% and 0.5% compared to the accurate reference implementation [5] for the vertical and horizontal SIMD processor, respectively. The AND gates, which are located in the *ALU* fraction, have no significant influence on the total area. However, the required mask control registers and the integration of control logic to configure these registers increases the necessary hardware amount and is accounted to the *Other* fraction.

Timing results after synthesis show an unchanged maximum clock frequency of 1667 MHz for the vertical SIMD vector unit. This is because the placement of the masking mechanism prior to the first ALU pipeline stage does not affect the critical path, which is located within unmodified intermediate pipeline stages of the MAC FU. The horizontal SIMD processor implementation, on the other hand, provides enough timing slack to accommodate to the maximum VLIW-MIPS CPU frequency of 510 MHz even with the additional

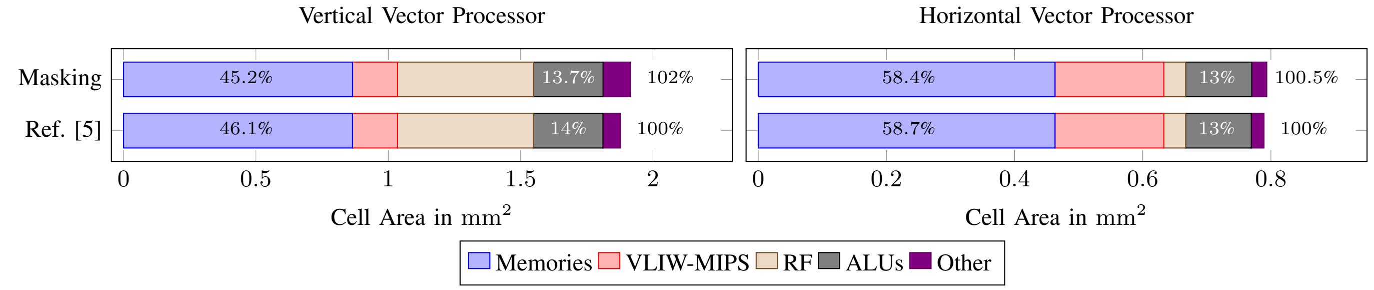

Fig. 4. Cell area increase due to implementing operand masking in a 45 nm general-purpose standard cell and SRAM technology. For comparison, the fractions of memories and ALUs on the total area are added to the corresponding bar segments. The values on the right of each bar plot denote the cell area ratio compared to the accurate reference implementation [5].

masking mechanism in the combinational path through the ALU. Therefore, the proposed operand masking mechanism does not decrease the computational performance compared to the accurate reference architectures [5].

C. Evaluation of Egomotion Estimation Accuracy

To evaluate the trade-off behavior between the operand masking approximation errors and the overall energy reduction, an iterative linear search algorithm for the 8 SIFT mask parameters is employed. Starting from the accurate mask setting (no LSBs masked) in the first iteration, 8 test masks are generated, where each of the parameters is independently increased by one LSB. The egomotion estimation application is run and the number of average circular keypoint matches is obtained for each test mask, resulting in 8 application runs. Due to the FPGA emulation latency of up to 20 s for processing one SIFT image frame on the emulated processors, only the first 4 frames of the KITTI sequence are considered per run. From these 8 runs, the mask parameter setting with the highest average number of circular matches is selected and the iteration index is increased. This procedure is repeated until no more matches are found due to excessive approximation and egomotion estimation fails.

In a finalization step, egomotion estimation is executed on the complete KITTI sequence for each selected mask parameter from the linear search. The error metrics $\mathrm{MRE}_{\mathrm{trans}}$ and $\mathrm{MAE}_{\mathrm{pose}}$ are computed as errors to reference GPS tracking data provided with the KITTI dataset. Fig. 5 shows the error metrics as a function of the linear search iteration index. Iteration 1 corresponds to accurate computation with values of 4.4% and 0.80×10^{-3} rad/Frame for $\mathrm{MRE}_{\mathrm{trans}}$ and $\mathrm{MAE}_{\mathrm{pose}}$, respectively. It is shown that $\mathrm{MRE}_{\mathrm{trans}}$ and $\mathrm{MAE}_{\mathrm{pose}}$ are kept below 5.3% and 1.00×10^{-3} rad/Frame, respectively, for a large range of 101 iterations for the vertical and 85 iterations for the horizontal SIMD vector processor. At this point, at least 4 out of 8 mask parameters have reached 16 masked LSBs of the 32-bit operands. Beyond this point, the amount and quality of matched SIFT feature points quickly drops within 5 iterations, causing the maximum error values to be reached. It can be concluded that the addition and multiplication operations are robust to approximations. Stable keypoints are generated with a reduced operand accuracy of up to 16 LSBs in some application parts in the later linear search iterations, which implies that the fixed-point operands could be resized to only 16 bits in these parts to save memory space and achieve higher performance by SIMD. However, this is specific behavior of the particular SIFT implementation and requires manual code revision and internal number format conversions, which is out of scope of this paper. The focus of this paper is a mechanism which allows energy reduction of processor implementations without major modifications to existing architectures and application code.

It is observed that even though the amount and descriptor vectors of detected SIFT keypoints vary with increased approximation, the variations are consistent between stereo images and following frames, so circular matching in the egomotion estimation application is still operational. However, after a certain approximation level, the consistency is immediately lost and egomotion estimation fails to match keypoints and generate useful results. This gives rise to adjust the approximation level within a wide range to reduce switching and required energy in the processors, because the accuracy in terms of $\mathrm{MRE}_{\mathrm{trans}}$ and $\mathrm{MAE}_{\mathrm{pose}}$ is not significantly changed. Some linear search iterations with locally minimal error values highlighted in Fig. 5 are selected for switching activity simulation and the following energy reduction evaluation.

D. Evaluation of Energy Reduction

Fig. 6 shows the processor energy consumption determined via switching activity simulation for extracting SIFT features. The parameters for approximate operand masking have been selected from the given linear search iterations. For both vertical and horizontal SIMD processor, the leftmost bar denotes the reference energy consumption from [5], where no operand masking mechanism has been implemented. For the reference, the largest fractions of 49.7% and 73.4% of the total energy consumption are accounted to local and scratchpad memory blocks of vertical and horizontal SIMD processor, respectively. The absolute memory energy consumption is higher on the horizontal SIMD processor, even though the absolute memory cell area is higher on the vertical one (refer to Fig. 4). This is related to the clocking scheme. The interleaved memories on the vertical SIMD processor operate at half the clock freqency of the vector pipeline using a clock divider, which is activated

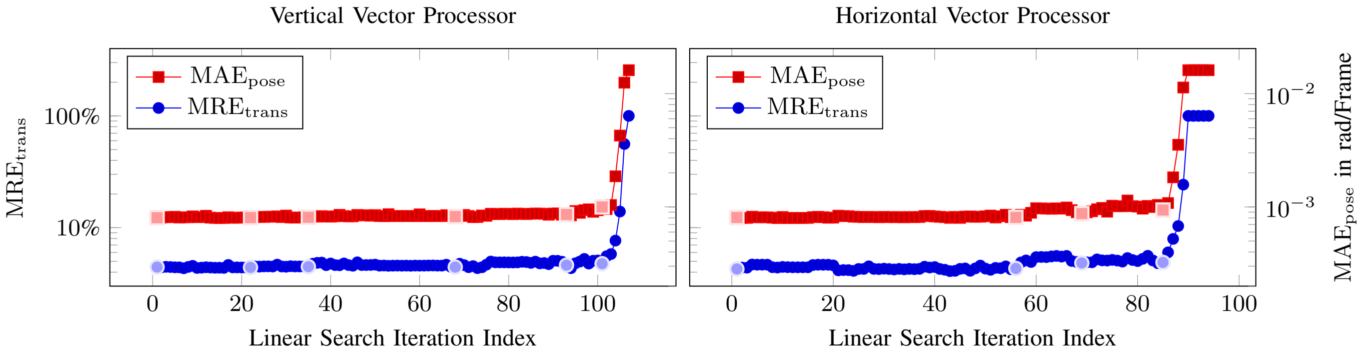

Fig. 5. Mean Relative Error of translation velocity ($\mathrm{MRE}_{\mathrm{trans}}$) and Mean Absolute Error of camera pose ($\mathrm{MAE}_{\mathrm{pose}}$) for approximate egomotion estimation after each search iteration. Left plot: Approximate vertical vector processor (iterations 1, 22, 35, 68, 93, 101 highlighted). Right plot: Approximate horizontal vector processor (iterations 1, 56, 69, 85 highlighted). Highlighted iterations are selected for energy consumption analysis in Section IV-D.

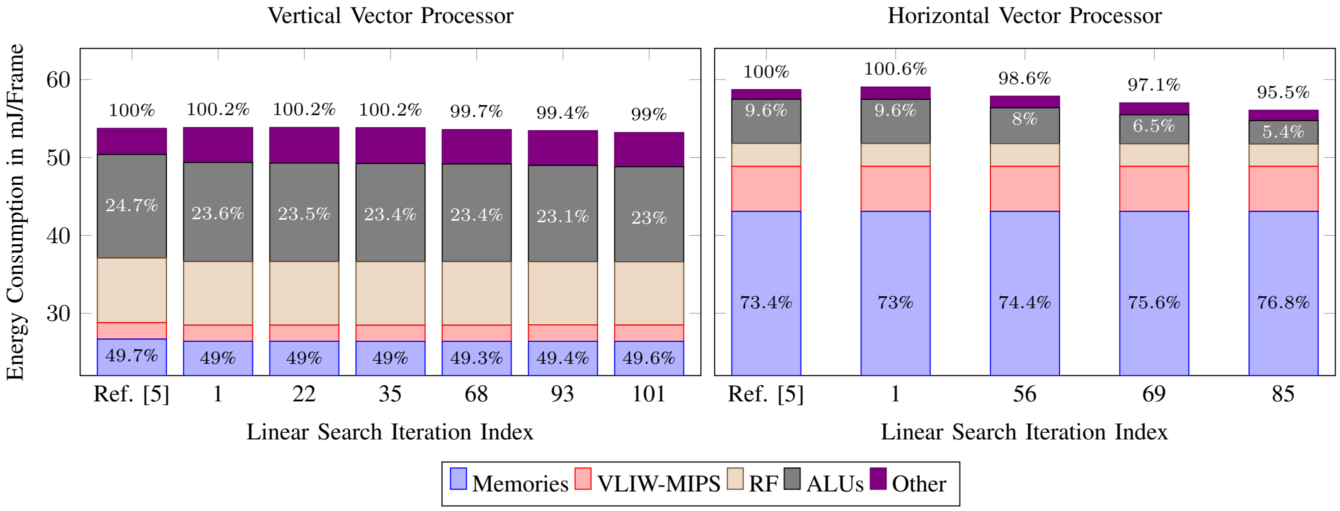

Fig. 6. Energy consumption for extracting SIFT features for selected operand mask iterations. For comparison, the fractions of memories and ALUs on the total energy consumption are added to the corresponding bar segments. The vertical axis is scaled to show only partial memory energy in order to increase the visibility of the other fractions. The values above each bar denote the energy consumption ratio compared to the accurate reference implementation [5].

only at memory accesses. On the horizontal SIMD processor, the memory clocking is simpler and directly uses the CPU clock, resulting in continuous clock transitions at the memory interface which increase clock-dependent energy consumption. Clock gating techniques for energy reduction are not used because of performance being the main optimization goal and in favor of a simpler architecture design.

With the implementation of the operand masking mechanism, the energy consumption of the vertical and horizontal SIMD processors is increased to 100.2% and 100.6% compared to [5], respectively, when using no approximation (linear search iteration 1). This is due to the additional mask control register logic contained in the *Other* energy fraction. By increasing the level of operand masking and thus reducing switching activity, the overall energy consumption is reduced down to 99% and 95.5% of the reference implementation for vertical and horizontal SIMD processor, respectively, saving up to 4.5% of energy. Therefore, operand masking proves to be a considerable technique for saving energy on application-specific, yet fully programmable approximate processors for error-resilient applications. In this particular application scenario, the saved energy even comes without quality implications, since the overall egomotion estimation error metrics $\mathrm{MRE}_{\mathrm{trans}}$ and $\mathrm{MAE}_{\mathrm{pose}}$ are not significantly increased.

However, the energy reduction effect of the proposed operand masking mechanism is limited to combinational ALU standard cells only, which is why it is more effective on the horizontal SIMD processor. To illustrate this, Fig. 7 depicts the switching-related dynamic power distribution per standard cell type for the reference [5] and for the linear search iterations with the highest energy reduction from Fig. 6. For both vertical and horizontal SIMD processors, memory blocks contribute most to the dynamic power. Apart from that, the dynamic power for the vertical SIMD processor is dominated by clocking sequential cells, which is mostly caused by the 5 pipeline stages of the MAC unit. Combinational logic only accounts for 3.3% of the dynamic power and is reduced to 2.5% with operand masking, while the other power components remain

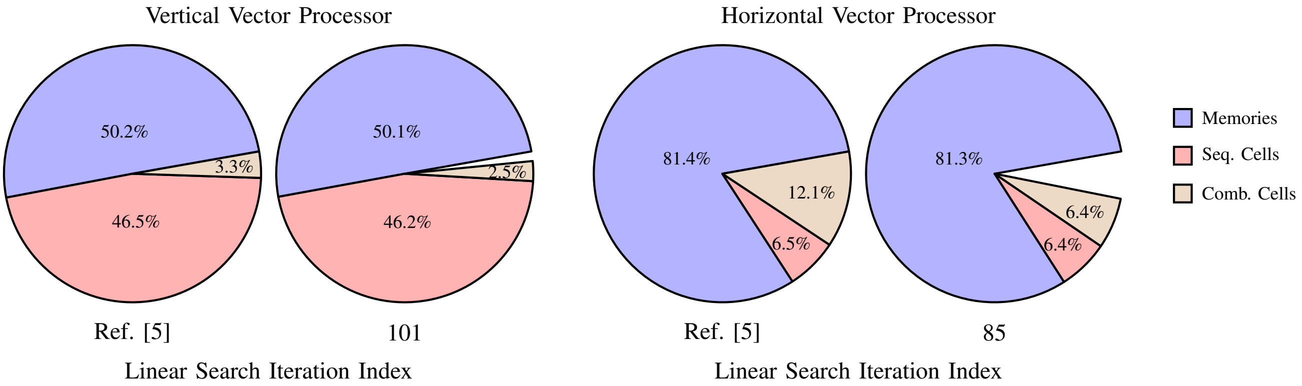

Fig. 7. Dynamic power distribution per standard cell type without and with operand masking. The percentages of memory blocks, sequential and combinational cells are related to the reference implementations of the vertical and horizontal vector processor [5].

nearly unchanged. For the horizontal SIMD processor with only minimal pipelining, on the other hand, combinational logic contributes to a larger extent of 12.1% to the total dynamic power, which is reduced to 6.4% with operand masking. This explains the results from Fig. 6: For the vertical SIMD processor, the energy consumption of the ALUs is only reduced from 24.7% to 23.0%, because it mainly consists of the energy required for clocking the pipeline register stages. For the horizontal SIMD processor, the ALU energy consumption is nearly halved from 9.6% to 5.4% for the horizontal SIMD processor, because it is mainly combinational. It becomes clear that the designer of deeply pipelined approximate processors may need to combine the proposed operand masking technique with other mechanisms like clock gating to also reduce clock-related switching activity in the dominating sequential parts of the circuit. This, however, can lead to design complications during ASIC synthesis and may cause severe performance implications, whereas the implementation cost of the proposed masking mechanism is negligible.

V. Conclusion

In this paper, an operand masking approach is proposed to achieve lower energy consumption by approximate computing techniques in programmable high-performance processors for computer vision applications, in this case vertical and horizontal SIMD vector processors. Contrary to dedicated approximate circuits, the mechanism enables the programmer to deliberately reconfigure operand accuracy at runtime in order to reduce the ALU switching activity and, consequently, the energy consumption. The effectiveness is evaluated with an error-resilient egomotion estimation application, which executes feature matching on approximate SIFT features generated by the aforementioned approximate processors. For the horizontal SIMD processor, total system energy savings of up to 4.5% for a 45 nm ASIC technology are achieved without a significant decrease in quality of the application result.

To obtain masking parameter sets for the mechanism, a linear search algorithm is used to find locally optimal trade-off points in the accuracy-energy design space for the application. However, due to missing global feedback, pareto-optimal parameters may not be found. To improve the parameter selection without infeasible brute-force approaches, global search heuristics like simulated annealing or genetic algorithms can be used to overcome the limitations of the linear search.

Evaluation of the dynamic power distribution in an ASIC implementation shows that the proposed technique is only effective in reducing the switching power of combinational ALU cells. Therefore, operand masking has a larger positive effect in the simple, combinational horizontal SIMD ALUs. The vertical SIMD datapath, however, is deeply pipelined and operand masking can reduce the energy consumption by only 1%, because 96.7% of the dynamic power are consumed by clocked pipeline registers and memory blocks. To further decrease the power, the proposed mechanism has to be combined with other methods, e.g., clock gating, to reduce switching in sequential cells. However, this may complicate meeting the design constraints for high-performance processors, whereas combinational operand masking is easily integrated.

References

[1] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 963–968.

[2] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*. IEEE, 2013, pp. 1–6.

[3] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits," *ACM J. Emerging Technol. Comput. (JETC)*, vol. 13, no. 4, pp. 1–34, 2017.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] M. Weißbrich, A. García-Ortiz, and G. Payá-Vayá, "Comparing vertical and horizontal SIMD vector processor architectures for accelerated image feature extraction," *J. Syst. Archit.*, vol. 100, p. 101647, 2019.

[6] G. Payá-Vayá, J. Martín-Langerwerf, and P. Pirsch, "A multi-shared register file structure for vliw processors," *Journal of Signal Processing Systems*, vol. 58, no. 2, pp. 215–231, 2010.

[7] *ML605 Hardware User Guide*, Xilinx, version 1.8, 02.10.2012.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.