Algorithm Level Error Detection in Low Voltage Systolic Array

Mehdi Safarpour, *Student Member, IEEE*, Reza Inanlou, *Student Member, IEEE*,
Olli Silvén, *Senior Member, IEEE*

Abstract—In this brief an approach is proposed to achieve energy savings from reduced voltage operation. The solution detects timing-errors by integrating Algorithm Based Fault Tolerance (ABFT) into a digital architecture. The approach has been studied with a systolic array matrix multiplier operating at reduced voltage, detecting errors on-the-fly avoiding energy demanding memory round-trips. The analysis of the solution has been done using analog-digital co-simulation to extract the transient behavior under different voltages and clock frequencies. HSPICE simulations using 90nm CMOS transistor models, and experiments by reducing operation voltage of a FPGA device were carried out. HSPICE simulations, showed possibility of 10x increase in energy-efficiency by approaching near-threshold region.

Index Terms—energy efficiency, low power, matrix multiplier, FPGA, accelerator, near threshold

I. INTRODUCTION

WITH the ultra-densification of wireless infrastructure through 5G technologies, with aim already at 6G solutions, much of the computing is expected to take place in the "edge", at short latency from the sensor and actuator nodes. This development is fueled by the increasing reliance on machine learning applications. With disappearing data communications bandwidth and latency constraints "intelligence" can be implemented and coordinated in the edge computing resources. Neural inference and massive mu-MIMO (multi-user Multiple-Input-Multiple-Output) radios are among the technological enablers [1] that both are computationally demanding and a challenge for energy-efficient digital design.

Power consumption of digital circuits has a quadratic relation to the supply voltage (Eq. 1). Therefore, a straightforward approach to gain in efficiency is to reduce the voltage [2].

$$P = \alpha f C V^2 + I_{leak} V \quad (1)$$

The first proposal to operate digital logic near and below the threshold voltage of the transistors was in 1972 [3]. In theory, these operating regions offer potential to improve the energy efficiency by 10x to 20x [4], while in experiments improvements up to 8x have been demonstrated by adopting Near-Threshold voltage regimes [4], [5], [6]. Although the nominal

operating voltages have been reducing, recent research, e.g. [2], has shown that the vendor specifications for off-the-shelf components such as Field-Programmable-Gate-Arrays (FPGA) and Graphical-Processing-Units (GPU) are pessimistic [7] to accommodate for process variations, such as inconsistencies in transistor geometry, oxide thickness and doping. It has been demonstrated that the supply voltage can be scaled down by around 12%, 20%, and even 30% in case of CPUs, GPUs, and FPGAs, respectively [2]. In most cases the voltage margin to where errors start appearing is large; up to 60% energy savings has been reported with FPGAs without observing one single fault [7] and without performance loss.

Unfortunately, the impact of variations can not be modeled deterministically in reduced voltage settings [8]. This uncertainty discourages the manufacturers from adopting aggressive voltage reduction schemes to utilize near-threshold and sub-threshold regions. In reduced voltage settings the effects of process variations are exacerbated, even up to 100x differences between Fast-Fast (FF) and Slow-Slow (SS) process corners [9], [10]. A challenge is that lowering the supply voltage without setting the clock frequency to the optimum, results in either loss of performance or loss of reliability. Too slow clocking results in the loss of energy efficiency and performance, while too fast clocking results in timing errors, hence loss of reliability. Reliable low-voltage implementations often require significant investments into development time and may reduce the fabrication yield [9], [?].

In this brief, we propose an algorithm-dependent technique for the detection of errors from aggressive voltage reductions. Similar approaches, e.g., Error-Correction-Code (ECC) algorithms are practical in detection and correction of memory errors. However they are not applicable for error detection in data and control paths [11]. Traditional fault-tolerance approaches such as Triple-Module-Redundancy (TMR) or Double-Module-Redundancy (DMR) and their different flavors, while provide error-resiliency, increase the gate-count and gate-activity substantially. This translates into fault-resiliency at the cost of energy-efficiency [12], while our approach is to achieve energy-efficiency through leveraging fault-tolerance. The proposed Algorithm Based Fault Tolerance (ABFT) approach is demonstrated through transistor and system level co-simulation and FPGA implementation of a systolic matrix multiplier¹. The solution enables utilizing reduced voltage operating regions at minor development

This manuscript was submitted on Jan. 15, 2020 to IEEE Transaction Circuits and Systems for review.

This work was supported by 6G Flagship research programme under Academy of Finland Grant 318927.

Mehdi Safarpour and Olli Silvén are with Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland (e-mail: first-name.lastname@oulu.fi).

Reza Inanlou is with School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran (e-mail: reza.inanlou@ut.ac.ir).

¹Simulation models are available from <https://github.com/NeuroFan/SystolicArray>

overheads. The targets are low-power high-performance applications that allow for error correction by recalculation after voltage/frequency adjustment.

II. PRIOR APPROACHES

A simple technique to optimize the operation voltage is to embed the design with a delay chain, e.g., a sequence of inverters that mimics the timing behavior of the longest delay path in the circuit. While this enables taking into account global variations, supply voltage drops, and temperature fluctuations, local variations and cross-coupled noise can not be effectively modeled [9], [6]. In practice, the longest delay path is stimulated less frequently than the shorter ones, rendering delay chain based voltage tuning either too optimistic or too pessimistic [6]. This translates to loss of reliability or reduced efficiency gains, respectively.

Another state-of-the-art solution that is commercially used [5] is to arm the critical paths with Timing Error Detection (TED) circuits [9]. The TED method was originally proposed to mitigate susceptibility to ambient and internal variations to increase manufacturing yield, while in reduced voltage cases TED circuits aid in setting an improved operation point to maximise energy efficiency [9]. Based on detections by TED circuits the voltage and clock frequency can be adjusted in an adaptive manner. Nonetheless, TED schemes are not readily applicable with off-the-shelf platforms and require significant development investment when adopted to custom ASIC designs [13]. Being clock synchronized digital circuits, TEDs also add non-trivial power consumption overheads [6].

Fig. 1. Simplified concept of Timing-Error-Detection circuit.

Due to increased impacts of process and ambient variations at lower voltages, deterministic operation may be possible only with performance or energy losses, that is, by lowering the clock frequency, or using error correction. Solutions such as shortening the critical paths or TED circuits are either costly in silicon real-estate or overly conservative [14].

Borrowing concepts from the supercomputing community, we propose a simple, yet an effective, algorithm level solution. It enables low-voltage integrated circuit design, manifesting sub-linear overhead from error detection in terms of power and circuit complexity.

III. PROPOSED SOLUTION

We propose adjusting the voltage and frequency according to errors detected at the data output of the computing logic. For detections an ABFT scheme is integrated into the design. The objective is to enable pushing the supply voltage down, while maintaining reliability with minimal overheads.

The restriction of the approach is that ABFT methods are algorithm specific. In our study they are for matrix operations that are a worthwhile target, representing the most energy hungry computations in many applications. For instance, more than 98% of computations in a neural network inference episode [15], and majority of the computations of a 5G radio [16] involve matrix operations.

One of the most energy efficient and high performance architectures for matrix operations is systolic array introduced by Kung [17] in 1980s. Systolic designs have recently regained attention, being utilized in Google Tensor-Processor-Units (TPU) for acceleration of neural network computations [18].

Our contribution is demonstrating an ABFT solution for error detection in matrix multiplication targeted systolic array and verifying its performance. The objective is to support sustained operation at reduced voltage compensating, e.g., for temperature dependency of the threshold voltage.

A. Algorithm Based Fault Tolerance

The idea of ABFT was originally conceived by Huang and Abraham [19]. They initially described a low-overhead technique to detect and correct computational errors striking multiplication operations. Subsequently, ABFT was extended to other linear algebra operations, including transposition, QR decomposition, FFT, and 2-D convolutions [12].

The fundamental idea of ABFT for matrix operations is to augment input matrices with *checksum property*. This enables detecting computational errors by inspecting the final result, while correction can be done in limited cases provided that hardware support has been included in the design. In our proposed approach correction is done by recalculating after clock frequency or voltage change.

Assuming A is an $N \times N$ matrix, then a *row checksum* matrix A^r is defined as a $N \times (N + 1)$ matrix, as below [20]:

$$A^r = \begin{bmatrix} A & Ae^T \end{bmatrix} \quad (2)$$

where e is column vector $e_N = [1, 1, \dots, 1]$, hence the n th element of the column vector Ae contains the sum of elements of the corresponding row of matrix A , i.e.,

$$a_{n,(N+1)}^r = \sum_i a_{n,i} \quad (3)$$

Similarly *column checksum*, A^c , and *full checksum*, A^f matrices are defined as

$$A^c = \begin{bmatrix} A \\ eA \end{bmatrix} \quad \text{and} \quad A^f = \begin{bmatrix} A & Ae^T \\ eA & eAe^T \end{bmatrix} \quad (4)$$

respectively. The row vector $e^T A$ denotes the sums of the elements in the columns of matrix A , and $e^T Ae$ is the sum of all elements in matrix A , a scalar value. The checksums can be used to detect errors instead by multiplying matrices $A_{N \times N}$ and $B_{N \times N}$

$$C = A \times B \quad (5)$$

we multiply the column checksum matrix A^c and row checksum matrix B^c to obtain full checksum matrix C^f

$$C^f = A^c \times B^r \quad (6)$$

$$= \begin{bmatrix} A \\ eA \end{bmatrix} \begin{bmatrix} B & Be^T \end{bmatrix} = \begin{bmatrix} AB & ABe^T \\ eAB & eABe^T \end{bmatrix}$$

This outcome is depicted in Fig. 2. Provided that there are no errors in the checksum calculations, the location of a single row or column error in the result matrix C can be detected.

In the current study the interests are in controlling the operating voltage and frequency, and understanding the energy costs of the checksum logic. We assume that the occasionally added latencies from recalculations, and the silent error rate fit the application constraints.

Fig. 2. Checksum row and column help to detect the errors.

B. Systolic Array for Matrix Multiplication with ABFT

We limit our treatment to 2-D structured systolic array shown in Fig. 3 for matrix multiplication, equipped with ABFT logic. The array consists of a grid of identical Processing Elements (PE) that each perform multiply-accumulate (MAC) operation on data received from the adjacent top and left PEs and pass the result or the input data to the next neighbouring PEs on the right and below. Since the row checksum of the output is inspected by the circuitry, only matrix B needs to be augmented. This is done on-the-fly when the elements of matrix B enter in the array as depicted in Fig. 3.

In the current design ABFT is merged with a $N \times N$ systolic array structure by adding a column of N PEs for checksums and another column consisting of N digital integrators and comparators for error detection. We recognize that by not including column checksum logic, the silent error rate increases, however, it stays low as we show later. In a similar manner we could check just a fraction of the bits in the checksums.

Except for a two clock cycle delay from checksum calculation and inspection, ABFT augmentation does not change the design [17]. An advantage of the scheme is that errors are detected on-the-fly as the result matrix is clocked-out from the array. No memory round-trips are needed. The checksums can be disposed after inspection. The added overhead from columns for checksum and error checking is only $O(1/N)$, targeting transient-errors in reduced voltage settings rather than tackling permanent faults [21].

IV. CO-SIMULATION AND EXPERIMENTATION

Simulation models for the design were created for MATLAB and HSPICE to investigate the functionality and implementation characteristics. The analog-digital co-simulation setting is depicted in Fig. 4. As no standard cell libraries exists that are characterized for reduced voltage operations, the main focus was on HSPICE analyses of transient behavior under different operation voltages, while the MATLAB model

Fig. 3. The proposed systolic array structure integrating ABFT

was used for error detection. As a partial confirmation of the scheme in the real-world, reduced voltage experiments were carried out on an FPGA.

Fig. 4. Co-simulation paradigm of reduced voltage processing platform.

In the HSPICE model of the processing elements 8-bit Wallace-tree multiplier is followed by 24-bit ripple-carry adder. The structure is shown in Fig. 5. While the HSPICE simulations are time consuming, detailed results were desired rather than probabilistic models [22] to estimate the energy impacts. The digital behavior was modeled using MATLAB, closing the loop from analog simulations.

Fig. 5. A processing element of the systolic array. k and M are data length and number of columns, respectively

HSPICE was used to model variations in W/L , oxide thickness, temperature etc. Signal transitions were obtained from system level simulations with random data inputs to compile sets of test vectors.

Overhead analysis: The number of arithmetic operations of N by N matrix multiplication is $(2N^3 - N^2)$. Extending the input matrices with checksums according to Fig. 2, the operation count increases to $(2N^3 + 3N^2)$. Detection of errors through only the row checksum vector requires additional N^2 summation operations plus N comparisons. The outcome is

Fig. 6. An energy efficient point can be achieved using error detections.

$(2N^3 + 4N^2 + 3N)$ operations. Compared to similar methods such as Result Checking (RC) [12], ABFT has overhead rate of $O(1/N)$ compared to the $O(N)$. Moreover, ABFT can be adapted to operate on-the-fly.

V. RESULTS

The co-simulation model was used to investigate the potential of ABFT in operating a systolic array at reduced voltages. The scheme in our simulations was to keep the clock frequency and reduce the operating voltage until errors start appearing. Then, the clock frequency was reduced by predetermined steps until the errors disappear.

Matrix size 32-by-32 was selected due to its application relevance [16]. The dots in the upper part of Fig. 6 show the power dissipation of the ABFT logic augmented systolic array when the operating voltage is gradually reduced by 0.01V steps, and the frequency is adjusted after error detection. On an average the highest power consumption per all types of PEs is around $74\mu W$ at $V_{dd} = 0.9v$ and $f_{clk} = 400MHz$. The recalculations can be observed as momentary power dissipation peaks taking place after frequency drop. In a realistic scenario the frequency and the voltage are adjusted in tandem to meet the desired performance requirement at operating temperature. The peaks in Fig. 6, represents cases where an error is detected and corrected by re-computing the matrix multiplication and the associated energy dissipation is topped up. The lower part of Fig. 6 shows the correct throughput rates (“goodput”) for the 32-by-32 matrix multiplication systolic array with ABFT logic. These results are for long runs at each voltage after frequency adjustment. The impacts of process, voltage and temperature variations were simulated by adding timing variations, following the approach presented in [22] with a fixed relative variance of clock period ($\sigma_T/\mu_T = 0.5\%$).

Voltage reduction from 1V to 0.7V saves half of the energy for a matrix multiplication, emulating elimination of vendor inserted “guardband” voltage. When the near threshold region is approached at 0.5V, the energy use is reduced by further 70%, while the goodput drops to half. Aiming at higher energy

Fig. 7. Top: Error rates of PE output bits. Bottom: Total error and silent error rates for 32 point row-column and 32-by-32 matrix multiplication. The horizontal axis represents clock period variations (σ_T , nanoseconds).

efficiency sacrifices the goodput: at the near-threshold region in vicinity of 0.4V the goodput is only 9% of the one at 0.7V, and the energy per 32×32 matrix multiplication is only 8%.

The silent error rate or the share of errors that can find their way into the output without being detected is a relevant parameter for applications. It can be impacted by the design of error detection logic that in our case is fairly sensitive.

ABFT error checking is carried out row-by-row. Assuming error probability of P_i for i th row, the silent error probability is $S_{matrix} = \prod_{i=1}^N P_i$. If errors strike B_i bits of elements on i th row, the silent error probability of the row is $P_i = 2^{-B_i}$. Assuming n rows are being affected by errors, the probability of undetected error in the matrix is exponentially decaying $S_{matrix} = 2^{(-\sum_i B_i)}$. Assuming silent error rate s and a task that requires q matrix multiplications, the probability of silent errors is $S_{task} = s^{-q}$. For example, AlexNet inference requires around 22 000 32×32 matrix multiplications [15], so the silent error probability is very low.

Fig. 7 shows the error rate for different variations and the error rate in the final output of a row of PEs. To estimate the silent error rate we first modeled the bit error rate of a PE using a fixed voltage ($V_{dd} = 0.7V$) and clock period ($f_{clk} = 280MHz$) under different random variations in HSPICE. The variations were introduced as clock jitter and were considered to represent sources for timing-errors collectively. The top plot in Fig. 7 depicts bit error rates for outputs of PEs. The resulting error probability density was then used within PEs of the systolic array to flip their output bits. In the bottom plot, the “Total error rate in matrix”, are errors detected by comparing every element of the resultant matrix with pre-computed result. Thus, it includes even the silent ones not detected by ABFT. “Silent error rate in a row” shows the failures of ABFT for an individual row, while “Silent error rate in matrix” illustrates that those detection failures in case of a whole matrix are negligible. In our co-simulations with millions of trials not a single silent error was observed.

In the worst case for ABFT error detection, only a single row is struck by errors, whereas in a more realistic case

the PEs in all rows undergo similar variations. This means lowering chances of a silent error passing through to the output.

Based on the gate count and power estimates obtained from simulations using the HSPICE model for 32×32 matrix the ABFT scheme resulted in 8.5% logic overheads and 9% power overheads at fixed voltage and frequency. As the ABFT overheads are $O(1/N)$ our solution scales sub-linearly with respect to the size of the systolic array. With 128×128 matrix the overheads are 2.2% and 2.5% in gate count and power dissipation, respectively. Around 20% of the overheads are due to input checksum augmentation and 30% due to output checksum inspection circuitry, while 50% comes from the needed extra column of PEs.

FPGA implementation on a Xilinx Zynq-7000 FPGA verified the co-simulation results concerning the silent errors. The operating voltage could be reduced from the default 1.0V to 0.78V cutting energy dissipation to around a half.

VI. DISCUSSION

Implementations of Razor [23][6] have exhibited 16% to 64% logic overheads with systolic array depending on whether Razor Flip Flops are only in the most critical timing paths or all. In Zhang's [21] approximate computing experiment the power overheads were 10.3% in the MAC units while 1% drop in accuracy with neural models was observed. The co-simulations of Gundi et al. [24] with Razor augmented MAC units in a systolic array for approximate computing showed 2.5x improvement in energy efficiency at the cost of 20.8% logic overheads.

The proposed ABFT approach enables removing the extra voltage guard bands determined by the circuit manufacturers, without performance compromises. Furthermore, efficient near-threshold operation points can be reached when the clock rate is adjusted as well. As a future work, utilization of the proposed solution for approximate computing applications can be explored.

VII. SUMMARY

The energy savings potential indicated by transistor level HSPICE simulations was experimentally confirmed with an FPGA. Integration of the proposed ABFT scheme into a systolic array allows on-the-fly error detection, and can be employed to find low-power operation points. With its low silent error rate, the scheme fits a wide field of applications from wireless communications to artificial intelligence[†].

REFERENCES

- [1] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 7112–7139, 2014.
- [2] G. Papadimitriou, A. Chatzidimitriou, D. Gizopoulos, V. J. Reddi, J. Leng, B. Salami, O. S. Unsal, and A. C. Kestelman, "Exceeding conservative limits: A consolidated analysis on modern hardware margins," *IEEE Transactions on Device and Materials Reliability*, 2020.
- [3] R. M. Swanson and J. D. Meindl, "Ion-Implanted Complementary MOS Transistors in Low-voltage Circuits," *IEEE Journal of Solid-State Circuits*, vol. 7, no. 2, pp. 146–153, 1972.
- [4] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, 2005.
- [5] M. Hienkari, N. Gupta, J. Teittinen, J. Simonsson, M. Turnquist, J. Eriksson, R. Anttila, O. Myllynen, H. Rämäkkö et al., "A 0.4–0.9 V, 2.87 pJ/cycle near-threshold ARM Cortex-M3 CPU with in-situ monitoring and adaptive-logic scan," in *2020 IEEE Symposium on Low-Power and High-Speed Chips (COOL CHIPS)*. IEEE, 2020, pp. 1–3.
- [6] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, "RAZOR: Circuit-level Correction of Timing Errors for Low-power Operation," *IEEE Micro*, vol. 24, no. 6, pp. 10–20, 2004.
- [7] B. Salami, O. S. Unsal, and A. C. Kestelman, "Comprehensive evaluation of supply voltage undervoltage in FPGA on-chip memories," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2018, pp. 724–736.
- [8] M. Sarmiento, K.-D. Nguyen, C. Duran, T.-T. Hoang, R. Serrano, V.-P. Hoang, X.-T. Tran, K. Ishibashi, and C.-K. Pham, "A sub- μ W reversed-body-bias 8-bit processor on 65-nm silicon-on-thin-box (SOBT) for IoT applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2021.
- [9] "Dynamic Margining: The Minima Approach to Near-threshold Design," Minima Processor Oy, Tech. Rep., November 2017. [Online]. Available: <https://minimaprocessor.com/wp-content/uploads/2017/11/minima-margining-white-paper.pdf>
- [10] K. Singh and J. P. de Gyvez, "Twenty years of near/sub-threshold design trends and enablement," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 5–11, 2020.
- [11] H. David, C. Fallin, E. Gorbato, U. R. Hanebutte, and O. Mutlu, "Memory power management via dynamic voltage/frequency scaling," in *Proceedings of the 8th ACM international conference on Autonomic computing*, 2011, pp. 31–40.
- [12] G. Bosilca, R. Delmas, J. Dongarra, and J. Langou, "Algorithm-based fault tolerance applied to high performance computing," *Journal of Parallel and Distributed Computing*, vol. 69, no. 4, pp. 410–416, 2009.
- [13] X. Zhang, F. Boussaid, and A. Bermak, "32 bit $\times$ 32 bit multiprecision razor-based dynamic voltage scaling multiplier with operands scheduler," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 4, pp. 759–770, 2014.
- [14] I. Ahmed, S. Zhao, O. Trescases, and V. Betz, "Measure twice and cut once: Robust dynamic voltage scaling for FPGAs," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1–11.
- [15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [16] J. Xu, "Systolic array for universal matrix arithmetic," Ph.D. dissertation, Northeastern University Boston, 2020.
- [17] H.-T. Kung, "Why systolic architectures?" *Computer*, vol. 15, no. 1, pp. 37–46, 1982.
- [18] Y. Parmar and K. Sridharan, "A resource-efficient multiplierless systolic array architecture for convolutions in deep networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 2, pp. 370–374, 2020.
- [19] K.-H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE transactions on computers*, vol. 100, no. 6, pp. 518–528, 1984.
- [20] M. Vijay and R. Mittal, "Algorithm-based fault tolerance: a review," *Microprocessors and Microsystems*, vol. 21, no. 3, pp. 151–161, 1997.
- [21] J. J. Zhang, K. Basu, and S. Garg, "Fault-tolerant systolic array based accelerators for deep neural network execution," *IEEE Design & Test*, vol. 36, no. 5, pp. 44–53, 2019.
- [22] M. Rathore, P. Milder, and E. Salman, "Error probability models for voltage-scaled multiply-accumulate units," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.
- [23] K. Flautner, T. M. Austin, D. T. Blaauw, T. N. Mudge, and D. Bull, "Error recovery within integrated circuit," Mar. 3 2020, U.S. Patent 10579463.
- [24] N. D. Gundi, T. Shabanian, P. Basu, P. Pandey, S. Roy, K. Chakraborty, and Z. Zhang, "Effort: Enhancing energy efficiency and error resilience of a near-threshold tensor processing unit," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020, pp. 241–246.

[†]A patent application based on this research has been filed by the University of Oulu early 2021