A 19.4 nJ/decision, 364K decisions/s, In-memory Random Forest Multi-class Inference Accelerator

Mingu Kang, *Member, IEEE*, Sujan K Gonugondla, *Student Member, IEEE,*
Sungmin Lim, and Naresh R. Shanbhag, *Fellow*, *IEEE*

***Abstract*—This paper presents an IC realization of a random forest (RF) machine learning classifier in a 65 nm CMOS. Algorithm, architecture, and circuits are co-optimized to achieve aggressive energy and delay benefits by taking advantage of the inherent error resiliency derived from the ensemble nature of a RF classifier. Deterministic sub-sampling (DSS) and regularized decision trees reduce interconnect complexity, and avoid irregular memory access patterns and computations, thereby reducing the energy-delay product (EDP). The prototype IC also employs low-swing analog in-memory computations embedded in a standard 6T SRAM to enable massively parallel tree node comparisons thereby minimizing the memory fetches and reducing the EDP further. The 65nm CMOS prototype IC achieves a $3.1\times$ and $2.2\times$ improved energy efficiency and throughput leading to $6.8\times$ lower EDP compared to a conventional digital system at the same accuracies of 94% and 97.5% for two tasks: 1) 8-class traffic sign recognition, and 2) face detection, respectively.**

***Index Terms*—random forest, machine learning, accelerator, in-memory computing, analog processing.**

I. Introduction

Machine learning (ML)-based systems are transforming the way we live and interact with the world around us. In various recognition tasks, such as those in computer vision, machines have begun to exceed human performance [1]. However, the energy and delay costs of ML algorithms are very high and needs to be significantly reduced for the deployment on real-time sensor-rich platforms such as biomedical devices, wearables, autonomous vehicles, Internet-of-things (IoT), and many others. As a result, a number of integrated circuit (IC) implementations of ML kernels and algorithms have been proposed recently [2]–[8] to minimize the energy, delay, and latency of ML systems in silicon.

Many of these IC realizations have focused on efficient implementation of deep neural network (DNN) algorithms due to its state-of-the-art performance in various decision-making tasks [9]. However, the high complexity of DNNs with an irregular data flow across multiple layers limits the achievable energy efficiency and throughput making them unsuitable for severely resource-limited embedded platforms.

This work was supported by Systems On Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by SRC and DARPA.

The authors are with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Champaign, IL 61801 USA (e-mail: mkang17@illinois.edu; gonugon2@illinois.edu; sungmin3@illinois.edu; shanbhag@illinois.edu).

In contrast, the random forest (RF) algorithm [10] is an attractive alternative due to the simplicity of its computations (mainly comparisons), applicability to multi-class problems, and high-accuracy. In addition, the RF algorithm has inherent robustness to non-ideal computations due to its ensemble nature [10]. However, realizing an energy-efficient implementation of the RF algorithm is made challenging due to its high data access rate combined with its irregular data access pattern. There have been digital implementations of the RF algorithm on FPGAs, GPUs, and multi-core processors [11]. However, these fail to take advantage of the inherent tolerance of the algorithm to hardware non-idealities and the opportunities afforded by analog computations.

This paper presents an energy-efficient and high-throughput RF classifier IC in a 65 nm CMOS demonstrating a $6.8\times$ lower EDP compared to a conventional 8-b fixed point digital implementation for an 8-class traffic sign recognition and face detection tasks at the same accuracy of 94% and 97.5%, respectively. Preliminary measured results of this in-memory RF IC were reported in [12]. The energy and throughput improvements are achieved by employing: 1) deterministic sub-sampling (DSS) to reduce the interconnect complexity, 2) balanced full decision trees to regularize processing and memory access pattern, 3) deep in-memory architecture (DIMA) [13]–[17] to embed energy-efficient low-swing analog memory readout and computations in the periphery of an SRAM bitcell array (BCA), and 4) Class ADD generator (CAG) to obtain the tree-level decisions from the outputs of all the tree nodes executed in parallel.

While the previous DIMA chip [16] implemented simple ML algorithms such as the support vector machine (SVM), more than 75% of energy consumption in the RF algorithm comes from operations, which cannot be accelerated by DIMA. Therefore, this paper proposes additional techniques to complement DIMA in order to realize an RF classifier. To

Summary of notation

$\boldsymbol{P}_m = [p_{m,1}, p_{m,2}, ..., p_{m,N}]$: pseudo-random pattern
$p_{m,n}$: pixel index of n^{th} node in m^{th} tree
RSS: random sub-sampling using $\boldsymbol{P}_m$
$\tau_{m,n}$: threshold of the n^{th} node in the m^{th} tree
$x(p_{m,n})$: $p_{m,n}^{th}$ pixel of input image $\boldsymbol{X}$
$q_{m,n}$: output of the n^{th} node in the m^{th} tree
$c_{m,l}$: label of the l^{th} leaf node in the m^{th} tree
(m: 1~M, n: 1~N, l: 1~$N+1$)

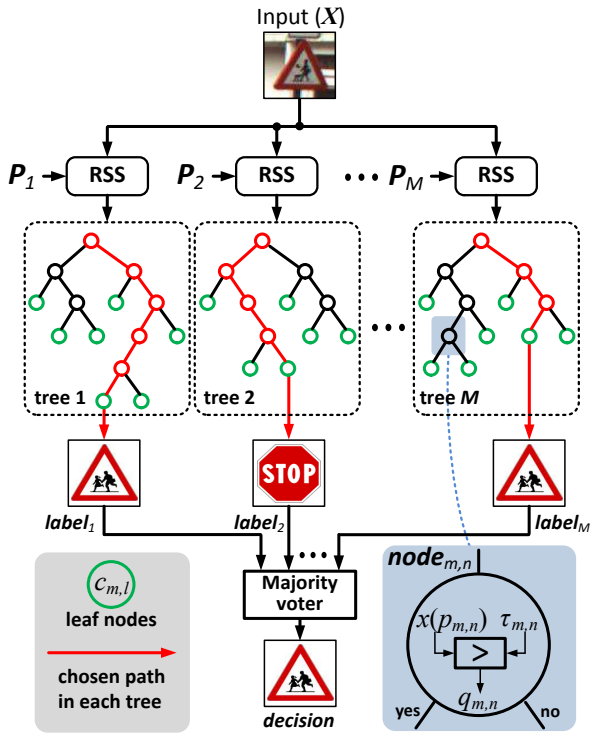


Fig. 1: The random forest (RF) algorithm.

the best of our knowledge, this is the first IC implementation of the RF algorithm.

This paper is organized as follows. Section II introduces the RF algorithm, its implementation challenges, and provides background on the deep in-memory architecture (DIMA). Section III describes the algorithm, circuit, and architecture level techniques to address the implementation challenges listed in Section II. Design details for the architecture and prototype IC are described in Section IV. Measurement results including task-level accuracy, energy and delay are presented in Section V. Section VI concludes the paper.

II. Background

This section provides background on the RF algorithm, its implementation challenges, and DIMA.

A. Random Forest (RF) Algorithm

The RF algorithm (Fig. 1) consists of M decision trees, each comprising a maximum of N nodes where $1+\log_2(N+1)$ is the tree depth including the leaf nodes. The m-th tree processes data obtained by random sub-sampling (RSS) the input image $\boldsymbol{X}$ based on a pseudo-random pattern vector $\boldsymbol{P}_m$, which is obtained during the training stage. The n-th node in the m-th tree compares the pixel (or feature) $x(p_{m,n})$ indexed by $p_{m,n}$, with a threshold $\tau_{m,n}$ to obtain a node-level binary decision $q_{m,n}$. Either the left or right branch is taken based on $q_{m,n}$. This process is repeated until a leaf node is reached. The label $c_{m,l}$ corresponding to the l-th leaf node is the tree-level decision. The final decision is obtained by majority-voting M such tree-level decisions. Although the RF algorithm is conceptually simple, it has a number of implementation challenges as described next.

B. RF Implementation Challenges

To enable energy-efficient and low-latency RF system, the following implementation challenges need to be addressed: 1) *the complex crossbar problem*, 2) *irregular trees*, 3) *parallelizing comparisons*, and 4) *LUT inefficiency*.

1) The complex crossbar problem: the RSS operation requires a complex crossbar (e.g., K:1, where $K > 256$ is the number of pixels in $\boldsymbol{X}$) to route a specific pixel of $\boldsymbol{X}$ to the corresponding tree node, i.e., to generate $x(p_{m,n})$ in Fig. 1.

2) Irregular trees: each tree can have a different shape and different number of nodes (e.g., up to $2^{(N-1)}$ possible shapes with $N \geq 31$) as shown in Fig. 1. The $q_{m,n}$s from each tree node are employed as the address to a look-up-table (LUT) to obtain $c_{m,l}$. In this case, the irregular tree shapes incur different processing delays and complex LUT logic (e.g, requires don't care logic for non-existent tree nodes).

3) Parallelizing comparisons: low-latency requires computing many tree nodes in parallel including the nodes on the non-selected paths. This requires highly parallel comparisons by fetching $2N$ bytes data and N comparisons per tree (e.g., $N \geq 31$) via the limited bit-widths of SRAM IO (B_{IO}) and the bus (B_{BUS}) (e.g., $B_{IO} = B_{BUS} = 64$-b per SRAM bank) for 64 to a few hundred trees.

4) LUT inefficiency: a brute-force implementation of the LUT to compute the labels $c_{m,l}$ from tree node decisions $q_{m,n}$ requires fetching the LUT contents of 2^N cases (e.g., $N \geq 31$) each consisting of an index and corresponding label, which needs to be processed in additional hardware.

C. Deep In-memory Architecture (DIMA)

DIMA addresses the high energy and latency costs of data movement between processor and memory by embedding analog computations deeply into the periphery of the memory core, i.e., the bitcell array (BCA) [13]–[17]. DIMA has four sequentially executed processing stages: 1) *functional read* (FR): fetches the stored data from multiple rows in the column of BCA at a time to generate linearly weighted sum of the binary data as an analog voltage of the bit-line (BL), i.e., digital-to-analog conversion (D/A), 2) *BL processing* (BLP): compute word-level arithmetic operations, e.g., addition, subtraction, or multiplication, by processing the BL voltage level generated via FR in the column pitch-matched analog processors in parallel, 3) *cross BL processing* (CBLP): aggregates multiple BLP outputs via charge-sharing to obtain a scalar output, 4) *thresholding* (TH): generates the final classifier decision. Silicon prototypes of DIMA have demonstrated significant energy and throughput benefits, e.g., up to 56× in energy-delay product (EDP) and up to

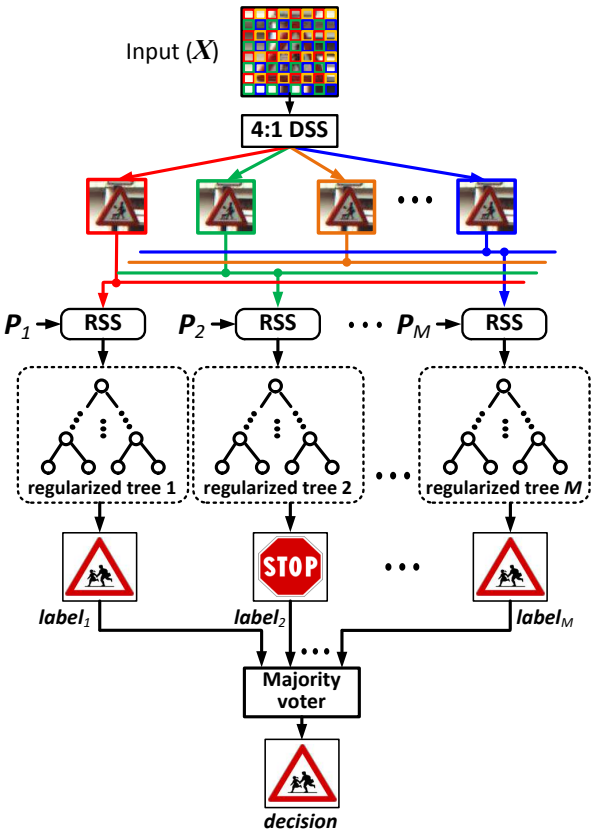


Fig. 2: The proposed RF algorithm with deterministic sub-sampling (DSS), and regularized decision trees.

9.7× energy reduction [16] over their fixed-function digital counterparts due to its low-swing analog processing and multi-row reads per BL precharge. In this paper, we employ DIMA to efficiently compare $x(p_{m,n})$ with $\tau_{m,n}$.

III. Proposed Solutions to the RF Implementation Challenges

This section proposes algorithm, architecture, and the circuit level solutions to address the following implementation challenges: 1) *irregular trees*, 2) *the complex crossbar problem*, 3) *parallelizing comparisons*, and 4) *LUT inefficiency* described in Section II-B. We propose a modified RF algorithm (in Fig. 2) to address 1) and 2) by using regularized trees and deterministic sub-sampling (DSS), respectively. We propose the use of DIMA for implementation of tree node comparisons to address 3) and a class ADD generator (CAG) to address 4). The required operations are summarized in Table I.

A. Regularized Tree

The decision trees are extended to form a balanced tree with N nodes (Fig. 2) by introducing filler nodes in order to regularize the memory access and processing patterns. Note that tree regularization does not incur any additional

TABLE I: Required operations per tree for the proposed (conventional) architecture with $N = 31$ and $K = 256$.

OPs per tree	Memory access			Compare	crossbar
Data	$p_{m,n}$	$c_{m,l}$	$\tau_{m,n}$	$x(p_{m,n}) > \tau_{m,n}$	$x(p_{m,n})$
Bit precision	6(8)	3(3)	8(8)	8(8)	Mux ratio 64 : 1 (256 : 1)
Size (Bytes)	23.3(31)	0.5(16)	0(31)	1	
# of OPs*	3(4)	0.5(2)	0(4)	31(31)	1(1)

*8 bytes per SRAM access

delay overhead or tree node hardware. This is because any RF architecture has to accommodate all possible tree shapes including the worst-case scenario of having to implement N nodes and incurring the worst-case delay of $\log_2(N+1)$ node comparisons. Furthermore, a regularized tree does not require an additional pointer address for the child node. Though one could choose to disable the filler nodes selectively to minimize computations in the regularized tree, doing so requires an additional data bit to indicate a null node and an additional memory access. Post-layout simulations show that a 1-b memory access requires 14× higher energy than a comparison operation. Thus, we allow all the nodes to be computed including filler nodes. The comparison result $q_{m,n}$ of the filler nodes do not affect the classification results as both left and right paths end up with identical labels. Therefore, for every filler node, we assign those values of $x(p_{m,n})$ and $\tau_{m,n}$ for which $|x(p_{m,n}) - \tau_{m,n}|$ is maximized. Doing so minimizes the probability of a metastability event in the analog comparator as described in Section III-C.

B. Deterministic Sub-sampling (DSS)

The modified RF algorithm (Fig. 2) employs a fixed pattern deterministic sub-sampling (DSS) step prior to RSS to solve the problem of requiring a *complex crossbar*. Fig. 3 shows that sub-sampling up to a ratio of 4:1 does not affect the misclassification rate due to the highly correlated pixel values. Therefore, a 4:1 DSS ratio is chosen to minimize the crossbar complexity without degrading P_{DET}. The optimal DSS ratio depends on the image resolution. The complexity of the subsequent RSS crossbar is reduced from 256:1 to 64:1 via DSS when the input $\boldsymbol{X}$ is a 16×16 image ($K = 256$). This results in the additional benefit of reducing the precision of $p_{m,n}$ from 8-b to 6-b.

C. DIMA-based Comparison

DIMA-based tree processing addresses the requirement of *highly parallel comparisons* (Fig. 4) and thus eliminates the need to explicitly fetch the thresholds $\tau_{m,n}$. In-memory comparison requires B-b thresholds $\tau_{m,n}$ (T in Fig. 4) and the indexed pixels $x(p_{m,n})$ (X in Fig. 4) to be stored in a *column-major* format, i.e., bits of a word are stored in a column. The comparison begins with the simultaneous

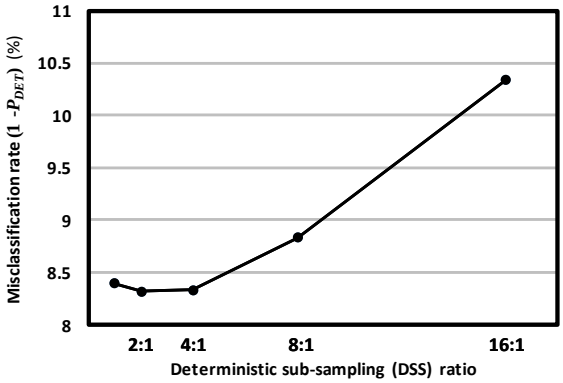


Fig. 3: DSS ratio vs. simulated misclassification rate ($1 - P_{DET}$) with test dataset including 700 images.

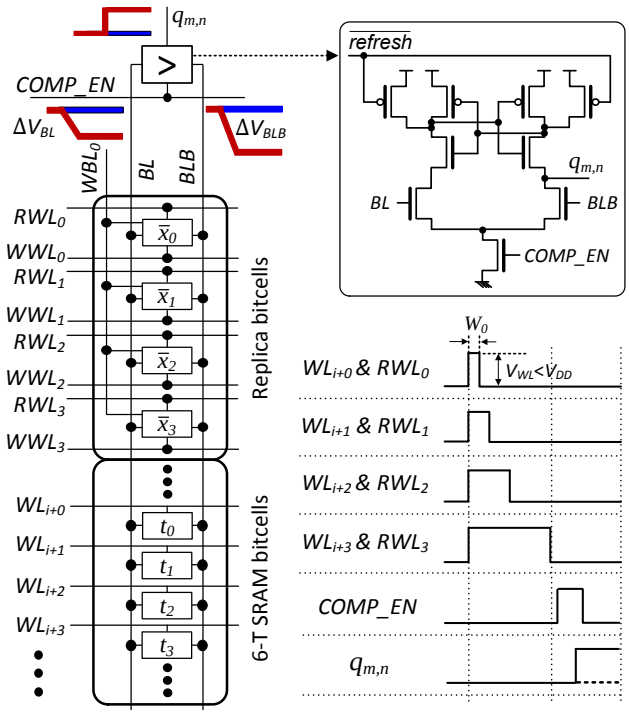


Fig. 4: In-memory comparison of $T = \sum_{i=0}^{B-1} 2^i t_i$ with $X = \sum_{i=0}^{B-1} 2^i x_i$ for bit precision $B = 4$.

application of WL access pulses with binary-weighted pulse widths to all the rows storing T and $\overline{X}$. This process is referred to as functional read (FR). Here, the pulse width is proportional to the bit position. Doing so generates a BL voltage (discharge) swing ΔV_{BL} proportional to $(X - T)$ given by:

$$\Delta V_{BL}(T, \overline{X}) = \frac{V_{PRE} W_0}{R_{BL} C_{BL}} \sum_{i=0}^{B-1} 2^i (\overline{t_i} + x_i)$$
$$= \Delta V_{lsb}(X - T - 1) \quad (1)$$

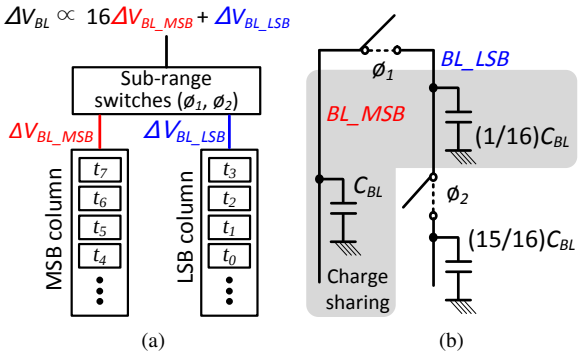


Fig. 5: Sub-ranged read [16]: (a) column pair implementation, and (b) an equivalent capacitance model.

where V_{PRE} is precharged BL voltage, W_0 is the LSB pulse width, R_{BL} is the resistance of BL discharge path comprising the access and pull-down transistors of the enabled bitcells, and $\Delta V_{lsb} = \frac{V_{PRE} W_0}{R_{BL} C_{BL}}$. The bias '-1' is generated by expressing negative numbers in 2's complement representation [13], [16]. Due to the complementary nature of SRAM bitcell, ΔV_{BLB} is also proportional to $(T - X)$. Next, BLs and BLBs are fed into analog comparators [18] to generate node-level decisions ($q_{m,n}$) for all columns in parallel.

Integral non-linearity (INL) of FR is improved up to 65% by sub-ranged read (Fig. 5), where $B/2$-b MSBs and $B/2$-b LSBs are read from adjacent columns (column pair) followed by a capacitively-weighted charge sharing step that assigns $16\times$ greater weight to the MSBs [16]. The $(1/16)C_{BL}$ in Fig. 5(b) is implemented by the parasitic capacitance of the sub-range switch and a tunable capacitor to calibrate the accuracy [16]. The WL voltage $V_{WL} < 0.65$ V to prevent destructive read and further improve linearity. The replica BCA [16] (Fig. 6) stores $\overline{X}$ via a separate write BL (WBL) and wordline (WWL) to avoid full-swing toggling of high-capacitance BL, which is required during the SRAM write operation.

In-memory comparison is a massively parallel operation as it fetches and processes $B/2$-b per column per read access bypassing the column mux whereas the conventional memory fetches only one bit per L columns per access, where L is the column muxing ratio with typical values from 4 to 16. The FR also saves precharge energy by accessing $B/2$-b per BL precharge whereas conventional read fetches single bit through the column mux per L BL precharges.

D. Class Address (ADD) generator

The *LUT inefficiency* problem is solved by employing a Class ADD generator (CAG), which converts the result of comparisons $q_{m,n}$s into a memory address for the chosen label $c_{m,l}$. This conversion is achieved by generating the memory address as $f(m) + g(q_{m,1\sim N})$, where $f(\,)$ provides an offset address and $g(\,)$ specifies the address. The functions $f(\,)$ and $g(\,)$ need to guarantee one-to-one mapping from

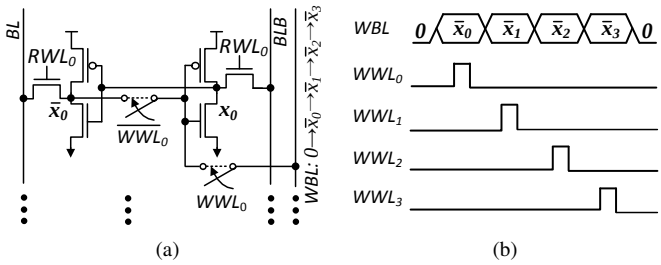


Fig. 6: Replica BCA operation [16]: (a) bitcell, and (b) write access timing diagram.

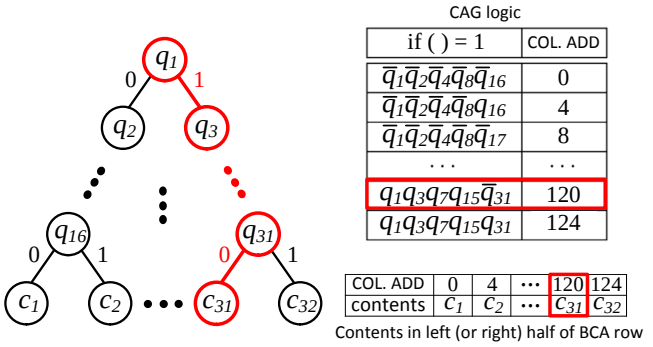


Fig. 7: CAG function $g(\ )$, where red tree nodes are an example of the selected path, and CAG logic generates the BCA column address corresponding to the chosen label, c_{31}.

$(m, q_{m,n})$ to memory address. For example, in this paper, $f(\ )$ generates the row address = $12\lfloor(m-1)/4\rfloor + 8 + \text{mod}(\lfloor(m-1)/2\rfloor, 2)$ and decides either the left or right half of BCA based on mod(m+1, 2). Function $g(\ )$ specifies the column address via simple Boolean logic with $q_{m,n}$s as shown in Fig. 7. Therefore, the only label of the chosen path (rather than the index and label of every path) is fetched.

These proposed techniques result in only 24.8 bytes/tree of data needing to be fetched as compared to 79 bytes/tree in the conventional parallel architecture as summarized in Table I.

IV. Prototype Random Forest IC

This section describes the design of the prototype RF IC architecture and its implementation in a 65 nm CMOS process.

A. Architecture and Timing

The prototype IC architecture (Fig. 8) includes a digital controller (CTRL) and a CORE consisting of a 512×256 SRAM BCA, peripherals for standard read/write operations, 64-b I/O ($B_{IO} = 64$) with a 4:1 ($L = 4$) column mux, FR WL drivers, 4:1 DSS input buffer to store the streamed 256-b $\boldsymbol{X}$ ($K = 256$), RSS crossbars, CAG, label finder, and a majority voter.

The timing diagrams in Fig. 9 shows that a group of four trees are processed in parallel requiring 171 clock cycles and

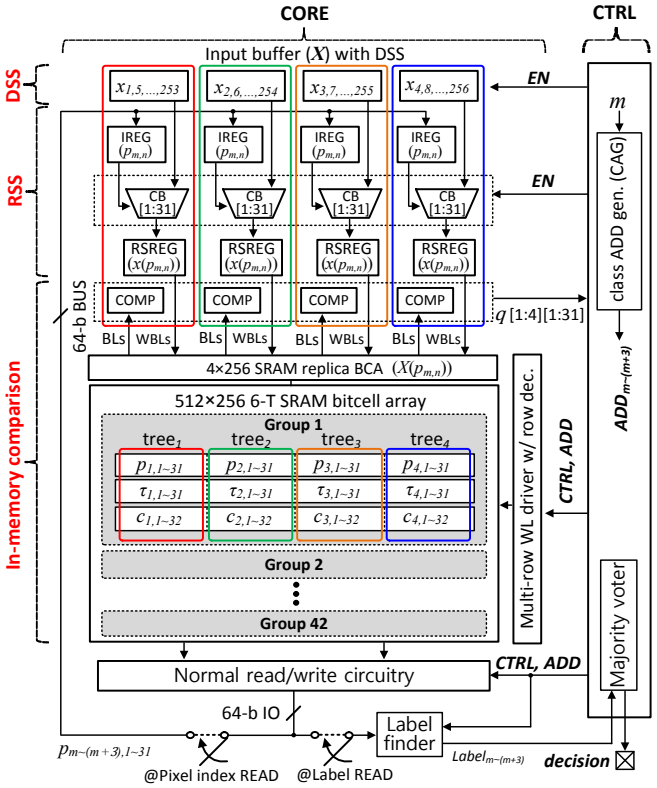


Fig. 8: Proposed RF architecture (IREG: pixel index register, RSREG: RSS register, COMP: analog comparator, and CB: crossbar).

TABLE II: Chip summary.

Technology	65 nm CMOS
Die Size	1.2 mm×1.2 mm
SRAM Capacity	16 kB (512 × 256 bitcells)
Bitcell dimension	2.11μm×0.92 μm
CTRL operating freq	1 GHz
Supply voltage (V)	CTRL: 0.75 CORE: 1
Energy per decision (pJ) (4 trees, 64 trees)	CTRL: (0.3,5.0) CORE: (0.9,14.4)
Decision throughput (decisions/s)	4 trees: 5.6 M 64 trees: 364.4 K

$M/4$ such groups are processed sequentially for a total of $M \leq 168$ trees. In the beginning, four groups of 4:1 sub-sampled 64 pixel words ($\boldsymbol{X}$) are stored in the four DSS input buffers. First, the pixel indices $p_{m,n}$s are fetched from the BCA into index registers (IREG) through 12 normal SRAM read accesses. Next, a main controller enables four RSS crossbars (CB) that places B = 8-b pixels $x(p_{m,n})$s into the RSS registers (RSREG) according to index $p_{m,n}$ stored in IREGs. Next, the $x(p_{m,n})$s are written into the replica BCA for in-memory comparison with thresholds $\tau_{m,n}$. The FR WL drivers apply binary weighted PWM pulses to WLs simultaneously and the discharged BLs and BLBs are fed to analog comparators (Fig. 4). The in-memory comparison generates 128 comparison outputs $q_{m,n}$s per precharge cycle

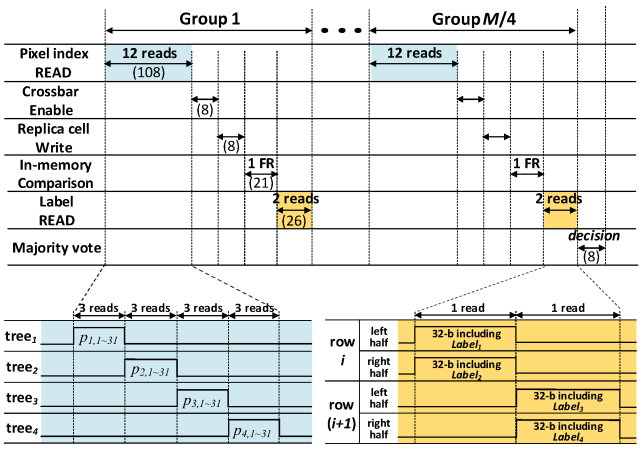


Fig. 9: Timing diagram showing the number of required clock cycles per stage.

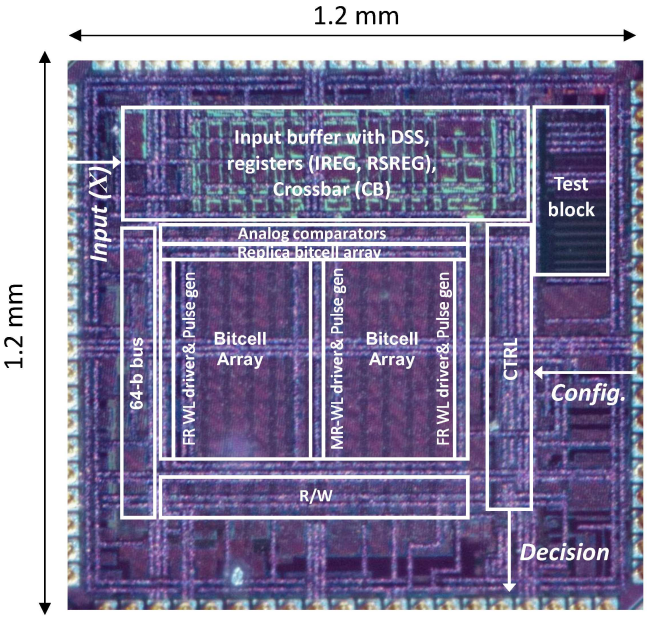
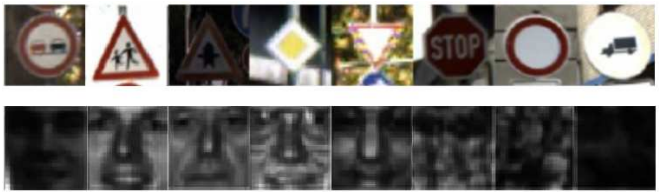


Fig. 10: Micrograph of the in-memory RF classifier chip.

at the output of the 128 pitch-matched analog comparators in parallel. The controller fetches four tree-level labels $c_{m,l}$s from the BCA via normal read operation using the address generated by CAG using the in-memory comparison outputs $q_{m,n}$. Finally, the majority voter makes a final decision based on the M tree-level labels $c_{m,l}$ after processing $M/4$ such groups.

B. Prototype IC Implementation

The prototype IC (Fig. 10) is fabricated in a 65 nm CMOS process and packaged in 88-pin QFN as summarized in Table II. The logic blocks including DSS input buffer, IREG, CB, and RSREG occupy 25% of area whereas less than 10% of area is occupied by the additional circuitry for in-memory



Task	Class	Dataset	Image Size
Traffic sign recognition [19]	8	KUL Belgium traffic sign dataset - Train: 148 images per class - Test: 200 random images	Resized 16×16 pixels (gray-scale)
Face detection [20]	2	MIT CBCL dataset - Train: 2000 images per class - Test: 200 random images	

Fig. 11: Datasets used in measurements.

comparison such as analog comparators and replica BCA. The CTRL operates with 1 GHz clock to provide fine time resolution for control signals. In this implementation, all the control signals, even for standard read/write operations, are synced with clock. On the other hand, the multi-row WL driver in Fig. 8 includes a pulse generator, whose output pulse width is set by the configuration code, which alters the number of inverter-based delay cells.

The 256-b input image $\boldsymbol{X}$ is provided serially into the input buffer and the configuration word defines the operations such as number of trees (M) to be processed. The final 3-b decision, to support a maximum of 8 classes, is fetched through the serial output port.

V. Measured Results

This section describes the measured results from the prototype IC and evaluates its energy, delay, accuracy benefits and robustness with respect to the conventional system, which employs the same architecture as in Fig. 8, but with a 256:1 RSS crossbar and no DSS, using a digital comparator, which is an 8-b subtractor with a sign detector, instead of in-memory comparison, and a digital LUT logic to store all the labels $c_{m,l}$s. The digital comparators processes eight comparisons at a time with a 64-b output (eight $\tau_{m,n}$s) from the SRAM while the next eight words are fetched. The energy and delay of the conventional architecture are estimated from: 1) measurements of the normal SRAM read access of the prototype IC, and 2) post-layout simulations of synthesized digital comparators and a 256:1 crossbar. The required operations (memory access, comparison, and enabling crossbar) of the conventional system are calculated using Table I. Layouts of the digital comparator and the 256:1 crossbar are matched to the horizontal dimension of the SRAM BCA to align well with the SRAM IOs. The area of the synthesized 256:1 RSS crossbar is found to be four times

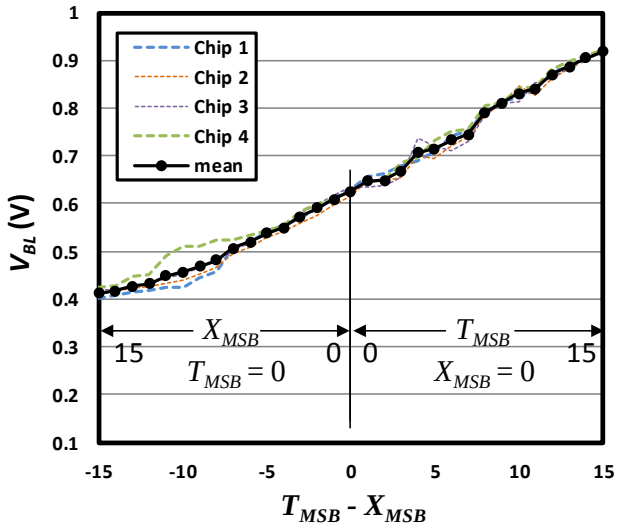


Fig. 12: Measured accuracy of in-memory subtraction from four dies where T_{MSB} and X_{MSB} are the decimal representations of the top 4 MSBs of the threshold and input pixel, respectively.

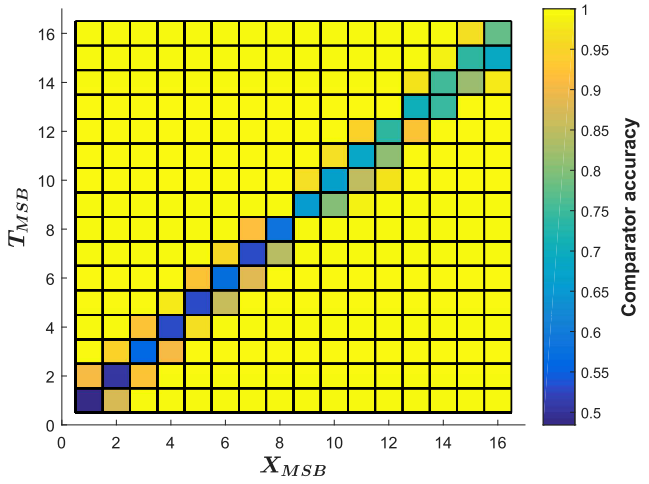


Fig. 13: Measured accuracy of DIMA comparisons with all possible combinations of (X_{MSB}, T_{MSB}) with ΔV_{lsb} = 25 mV. Each data-point is obtained by averaging 256 measurements over 256 different locations of the BCA.

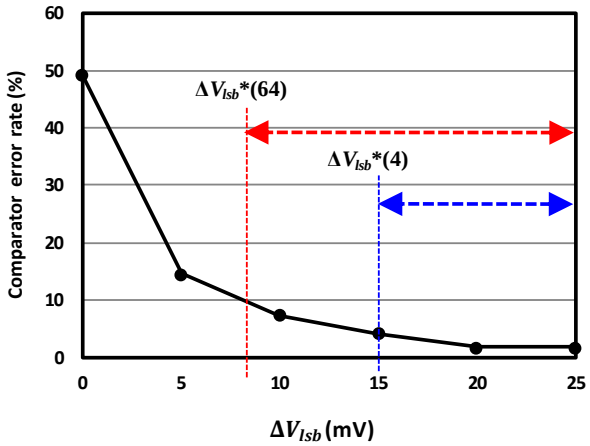


Fig. 14: Measured comparator error rate. Here $\Delta V^*_{lsb}(M)$ is the minimum ΔV_{lsb} to avoid accuracy degradation when using M trees. The ΔV_{lsb} is controlled by changing the voltage level of WL enabling signal to affect R_{BL} in (1). The ΔV_{lsb} is estimated from (1) by measuring ΔV_{BL} with $X = 15$ and $T = 0$ in the test mode.

larger than the proposed crossbar due to its higher complexity whereas the area of the digital comparators is similar to the proposed analog processors (replica BCA and analog comparators). Therefore, the conventional system occupies approximately $1.8\times$ more area than the proposed system.

A. Application Mapping and Classifier Training

The proposed architecture was tested on two datasets (Fig. 11): a) the KUL Belgium traffic sign dataset [19] for 8-class traffic sign recognition task, and b) the MIT CBCL dataset [20] for face detection task. The input pixels and the threshold values $\tau_{m,n}$ are represented in 8-b fixed-point as this precision provides approximately the same accuracy as floating point [9], [21], [22]. During off-chip training, 148 training images per class are used for traffic sign recognition whereas 2000 training images per class are used for face detection. The maximum supported tree depth is chosen to be six, which is considered optimal for the target application [11]. To evaluate the impact of the number of trees on accuracy, two different cases of $M = 4$ and $M = 64$ are tested. The classification accuracy (P_{DET}) is measured by streaming 200 randomly chosen images and counting the correct decisions.

B. Component-level Accuracy

Fig. 12 shows that the in-memory subtraction, which is realized as part of the FR process, achieves an INL < 1.85 LSB in the range of $-15 \leq T_{MSB} - X_{MSB} \leq 15$. Deviation in V_{BL} are < 25 mV over different combinations of T_{MSB} and X_{MSB}. These variations are induced by circuit non-idealities including inaccurate ratio of PWM pulse widths, asymmetry of the replica BCA, and BL voltage dependence of the discharge path resistance. Spatial transistor threshold voltage (V_t) variations caused by random dopant fluctuations lead to increased comparator errors as shown in Fig. 13, where the in-memory comparator accuracy ranges from 100% to 50% (when $T_{MSB} \simeq X_{MSB}$). The asymmetry of the replica BCA is due to the use of single-ended *WBL* as shown in Fig. 6, which results in an asymmetric discharge current between BL and BLB. This asymmetry leads to an increase in the comparator error rate for large values of X_{MSB} and T_{MSB}, e.g., $X_{MSB} > 10$ and $T_{MSB} > 10$.

TABLE III: Comparison of energy efficiency and throughput with prior art.

Prior art	Process	Algorithm	Power (mW)	Layout area (mm^2)	Decision throughput (decisions/s) ①	Decision energy (nJ/decision) ②	# of scalar node computations /decision ③*	Node energy (pJ) ④=②/③	Decision EDP (f J·s /decision) ⑤=②/①	Node EDP (10^{-17}J·s) ⑤/③
[23]	65 nm CMOS	Vocabulary tree	5.6	6.3	30	186.7K	8.4M	22	6.2G	74K
[24]	65 nm CMOS	Vocabulary forest	27.6	2.3	60	460K	320M	1.4	7.7G	2.4K
This work†	65 nm CMOS	Random forest	7.1	1.0	364.4K	19.4	1984	9.8	53.2	2.7

* (# of trees/decision)×(# of tree nodes/tree)×(dimension/tree node), e.g., 64×31×1 in this work.
† at ΔV_{lsb}= 15 mV with M = 64 trees and misclassification rate of 94%.

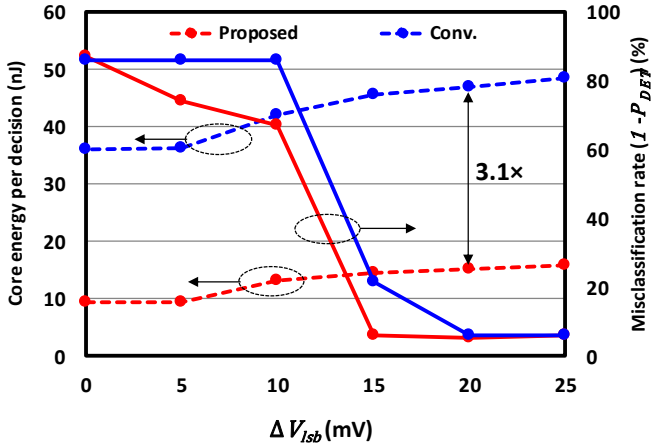


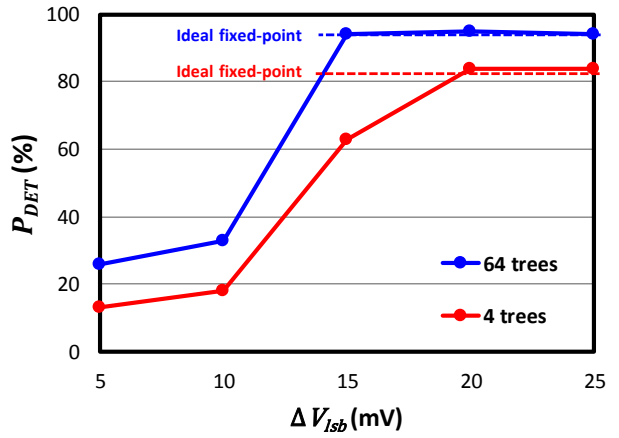
Fig. 15: Energy vs. misclassification rate with regard to ΔV_{lsb} for face detection with M = 64 trees, where $\Delta V_{BL} = 8\Delta V_{lsb}$ during normal SRAM read in order to achieve zero bit-error rate at default configuration (ΔV_{BL}= 200 mV, ΔV_{lsb}= 25 mV).

Fig. 14 shows that the measured in-memory comparison error rate increases from 1.6% to 14.5% due to the increased impact of process variations as ΔV_{lsb} (see (1)) reduces from 25 mV to 5 mV. Comparator errors were measured at each ΔV_{lsb} by counting the errors during the classification with the KUL Belgium dataset.

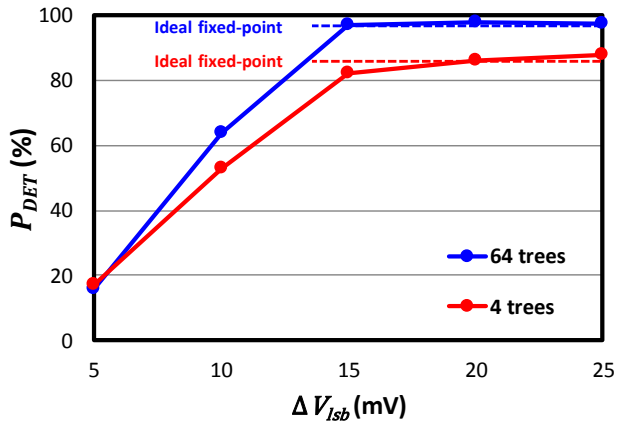
The impact of in-memory comparison errors on the misclassification rate was studied next. System simulations indicate that a comparison error rate of less than 9.5% will result in an indiscernible 8-class classification accuracy loss whereas $M = 4$ trees can tolerate a comparison error rate of only 4%. Thus, Fig. 14 shows that the minimum ΔV_{lsb} ($\Delta V^*_{lsb}(M)$) required by a M-tree RF architecture to avoid accuracy degradation is $\Delta V^*_{lsb}(4) = 15$ mV, and $\Delta V^*_{lsb}(64) = 8$ mV. This analysis indicates that ΔV_{lsb} should be assigned based on the number of trees (M).

C. Task-level Accuracy, Energy, and Throughput

Fig. 15 shows that the proposed IC achieves a maximum of 3.1× energy savings over the conventional architecture for the same misclassification rate of 6%. Fig. 16 shows that the



(a)



(b)

Fig. 16: Task-level accuracy vs. ΔV_{lsb} for different number of trees (M): a) traffic sign recognition, and b) face detection.

inherent error tolerance of the RF algorithm improves with the number of trees (M). The RF algorithm with $M = 64$ trees can operate at ΔV_{lsb} of 15 mV without any loss in accuracy as compared to an ideal fixed-point implementation (Fig. 16) while the RF with four trees observes degradation as soon as $\Delta V_{lsb} < 20$ mV for both datasets. This requirement is 5~7 mV greater than that predicted by Fig. 14 where only errors from FR and analog comparators were taken into

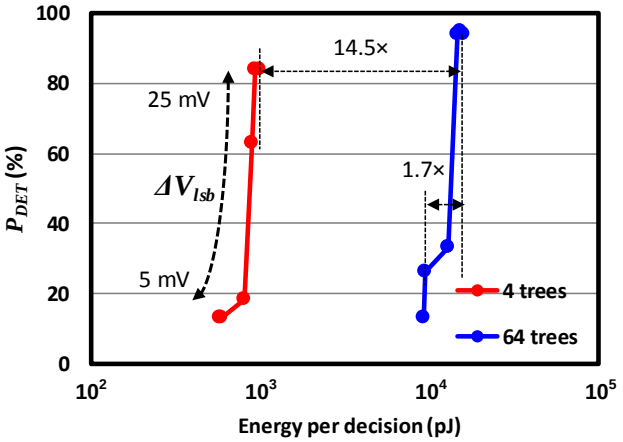


Fig. 17: Energy vs. task-level accuracy for traffic sign recognition with different number of trees (M) at $\Delta V_{lsb} = 5$ to 25 mV.

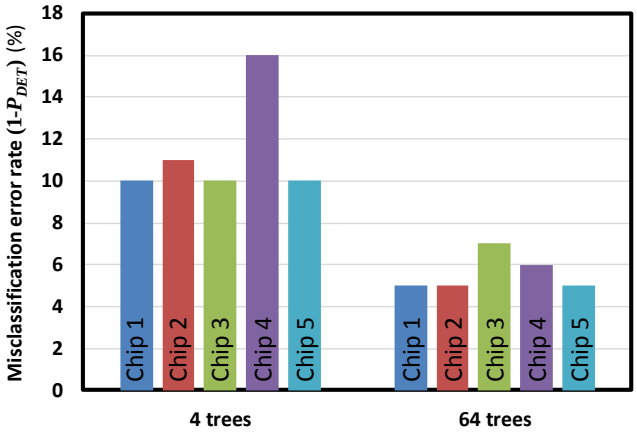


Fig. 18: Measured misclassification rates of multiple chip instances for traffic sign recognition with different number of trees (M) at $\Delta V_{lsb} = 25$ mV.

account. This discrepancy can be attributed to the presence of additional error sources in other stages such as pixel read and label read, where conventional SRAM read may also become erroneous due to ΔV_{lsb} scaling. It is expected the energy efficiency will improve in highly complex real-world tasks with a few hundreds trees as it will enable further reductions in ΔV_{lsb}. Binary face detection is less sensitive to ΔV_{lsb} reduction than 8-class traffic sign recognition as expected. This result indicates that the ΔV_{lsb} can be systematically scaled based the number of classes, the number of trees M, and the target classification accuracy.

Fig. 17 shows the energy vs. accuracy trade-off with respect to two parameters: 1) number of trees (M) and 2) BL voltage swing ΔV_{lsb}. The decision energy can be saved by reducing either M or ΔV_{lsb}. However, the P_{DET} degrades more gracefully when the energy is reduced by reducing M, e.g., P_{DET} drops by 10% achieving 14.5× energy savings with M whereas P_{DET} degrades by 68% for 1.7× energy reduction with ΔV_{lsb}. Therefore, classification with smaller M always achieves better energy efficiency for a fixed classification accuracy P_{DET}, though the maximum achievable P_{DET} also reduces.

To observe the impact of process variations, P_{DET} is measured by testing five dies. Fig. 18 shows minor differences in P_{DET}, e.g., <6% and <2% with M = 4 and 64, respectively. This result indicates that the process variations are well-compensated by the inherent error tolerance of ensemble classification in RF algorithm.

Fig. 19 shows the energy and delay breakdowns of the proposed IC as compared to the conventional architecture. The breakdowns are obtained from post-layout simulations as it is difficult to measure the energy and delay of each component from the measurement of prototype IC. The energy and delay are reduced by 3.1× and 2.2×, respectively, thereby providing an overall 6.8× lower energy-delay product (EDP). Fig. 19 indicates that DSS, in-memory comparison, CAG con-

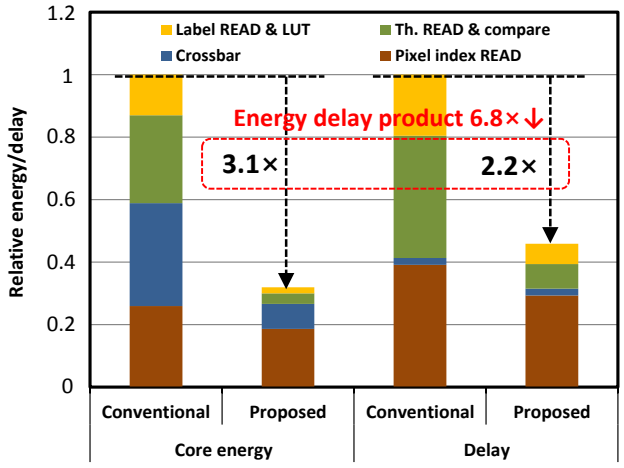


Fig. 19: Energy and delay breakdown at $\Delta V_{lsb} = 25$ mV ($\Delta V_{BL} = 200$ mV) obtained via post-layout simulations.

tribute 47%, 37%, and 16% of the total energy savings, and 18%, 58%, and 24% of the total delay reduction, respectively, thereby indicating the relative contributions of DIMA and non-DIMA techniques.

Table III shows that the prototype IC achieves a throughput of 364 K decisions/s and energy efficiency of 19.4 nJ/decision including the energy from CTRL. To the best of our knowledge, this is the first IC prototype of the RF algorithm. Therefore, we compare our work with other tree-based classifier ICs [23], [24]. Due to the vast difference in the datasets, tree complexity, and node architecture, we focus on node level metrics (last three columns in Table III). The proposed RF classifier achieves node-level EDP that is three to four orders-of-magnitude lower than [23], [24].

VI. Conclusions and Future Work

This paper has presented an IC realization of random forest (RF) algorithm to achieve energy-efficient and high

throughput by co-optimizing algorithm, architecture, and circuit design. As a result, the prototype IC achieves a $3.1\times$ energy savings and $2.2\times$ speed-up providing a $6.8\times$ lower energy-delay product (EDP) at the same accuracy of $> 93\%$ compared to conventional digital architecture.

The benefits of the proposed architecture are expected to increase with large-scale applications. This is because high resolution images have increased pixel correlation, which allows higher DSS ratio without degrading classification accuracy. The random noise components of in-memory computation also get averaged out better with higher tree numbers so that a lower voltage swing or a smaller technology node can be employed.

On the other hand, complex applications require processing a large number of tall trees. This is not an issue, as it is straightforward to parallelize the proposed RF architecture by employing multiple banks and exploiting the independence between tree executions. It is also expected that the regular RF structure provides strong scalability and reconfigurability features, e.g, stacking trees (one tree on top and $N+1$ trees at bottom) doubles the tree height and cascading the 4:1 DSS blocks generates a 16:1 DSS ratio. These features make the proposed RF classifier suitable for resource-constrained applications such as IoT devices to sustain the always-on functionality whereas DNN will be suited for higher accuracy applications.

ACKNOWLEDGMENTS

This work was supported by Systems On Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by SRC and DARPA. The authors would like to acknowledge constructive discussions with Sean. Eilert, Ken Curewitz, Naveen Verma, Boris Murmann, and Pavan Hanumolu.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017.

[3] H. Kaul, M. A. Anders, S. K. Mathew, G. Chen, S. K. Satpathy, S. K. Hsu, A. Agarwal, and R. K. Krishnamurthy, "A 21.5 M-query-vectors/s 3.37 nJ/vector reconfigurable k-nearest-neighbor accelerator with adaptive precision in 14nm tri-gate CMOS," in *IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2016, pp. 260–261.

[4] S. Park, K. Bong, D. Shin, J. Lee, S. Choi, and H.-J. Yoo, "A 1.93TOPS/W scalable deep learning/inference processor with tetra-parallel MIMD architecture for big-data applications," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2015, pp. 1–3.

[5] M. Price, J. Glass, and A. P. Chandrakasan, "A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating," in *2017 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2017, pp. 244–245.

[6] P. N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G.-Y. Wei, "A 28nm SoC with a 1.2 GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," in *2017 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2017, pp. 242–243.

[7] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "ENVISION: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm FDSOI," in *2017 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2017, pp. 246–247.

[8] K. Bong, S. Choi, C. Kim, S. Kang, Y. Kim, and H.-J. Yoo, "A 0.62 mw ultra-low-power convolutional-neural-network face-recognition processor and a CIS integrated with always-on Haar-like face detector," in *2017 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2017, pp. 248–249.

[9] D. Silver, A. Huang, and et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[10] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] B. Van Essen, C. Macaraeg, M. Gokhale, and R. Prenger, "Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA?" in *IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2012, pp. 232–239.

[12] M. Kang, S. K. Gonugondla, and N. R. Shanbhag, "A 19.4 nJ/decision 364K decisions/s In-memory Random Forest Classifier in 6T SRAM Array," in *IEEE European Solid-State Circuits Conf. (ESSCIRC)*, Sept 2017.

[13] M. Kang, M.-S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8326–8330.

[14] N. Shanbhag, M. Kang, and M.-S. Keel, *Compute Memory*. Issued July 4 2017, US Patent 9,697,877 B2.

[15] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, 2017.

[16] M. Kang, S. K. Gonugondla, A. Patil, and N. R. Shanbhag, "A multi-functional in-memory inference processor using a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 53, no. 2, pp. 642–655, 2018.

[17] S. K. Gonugondla, M. Kang, and N. R. Shanbhag, "A 42pJ/decision 3.12TOPS/W robust in-memory machine learning classifier with on-chip training," in *IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, 2018, pp. 490–492.

[18] T. Kobayashi, K. Nogami, T. Shirotori, and Y. Fujimoto, "A current-controlled latch sense amplifier and a static power-saving input buffer for low-power architecture," *IEEE J. Solid-State Circuits*, vol. 76, no. 5, pp. 863–867, 1993.

[19] V. A. Prisacariu, R. Timofte, K. Zimmermann, I. Reid, and L. Van Gool, "Integrating object detection with 3D tracking towards a better driver assistance system," in *IEEE International Conference on Pattern Recognition (ICPR)*, 2010, pp. 3344–3347.

[20] "Center for biological and computational learning (CBCL) at MIT," 2000, `http://poggio-lab.mit.edu/codedatasets`.

[21] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1737–1746.

[22] C. Sakr, Y. Kim, and N. Shanbhag, "Analytical guarantees on numerical precision of deep neural networks," in *International Conference on Machine Learning*, 2017, pp. 3007–3016.

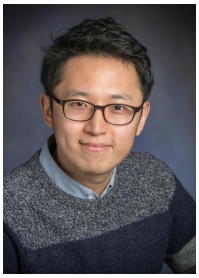
[23] T.-W. Chen, Y.-C. Su, K.-Y. Huang, Y.-M. Tsai, S.-Y. Chien, and L.-G. Chen, "Visual vocabulary processor based on binary tree architecture for real-time object recognition in full-HD resolution," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 12, pp. 2329–2332, 2012.

[24] K. J. Lee, G. Kim, J. Park, and H.-J. Yoo, "A vocabulary forest object matching processor with 2.07 M-vector/s throughput and 13.3 nJ/vector per-vector energy for full-HD 60 fps video object recognition," *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 1059–1069, 2015.

Mingu Kang (M'13) received the B.S. and M.S. degrees in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea, in 2007 and 2009, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2017.

From 2009 to 2012, He was with the Memory Division, Samsung Electronics, Hwaseong, South Korea, where he was involved in the circuit and architecture design of Phase Change Memory (PRAM). Since 2017, he has been with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, where he designs machine learning accelerator architecture. His research interests include low-power integrated circuits, architecture, and system for machine learning, signal processing, and neuromorphic computing.

Sujan K. Gonugondla (S'16) received the B.Tech. and M.Tech. degrees in Electrical Engineering from Indian Institute of Technology Madras, Chennai, India, in 2014. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, Champaign, IL, USA. His current research interest includes low-power integrated circuits specifically algorithm hardware co-design for machine learning systems on resource constrained environments. He is a recipient of the Dr. Ok Kyun Kim Fellowship 2018-19 from the ECE department at UIUC and the ADI Outstanding Student Designer Award 2018.



Sungmin Lim received the B.S. and M.S. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree in Electrical and Computer Engineering at University of Illinois at Urbana-Champaign, Champaign, IL, USA. His current research interests include design of energy-efficient architectures and implementation on integrated circuits for machine learning in resource constrained platforms.

Naresh R. Shanbhag (F'06) received the Ph.D. degree in Electrical Engineering from the University of Minnesota, Minneapolis, MN, USA, in 1993. From 1993 to 1995, he was with the AT&T Bell Laboratories, Murray Hill, NJ, USA, where he led the design of high-speed transceiver chipsets for very high-speed digital subscriber line. In 1995, he joined the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He has held visiting faculty appointments at the National Taiwan University, Taipei, Taiwan, in 2007, and at Stanford University, Stanford, CA, USA, in 2014. He is currently the Jack Kilby Professor of Electrical and Computer Engineering with the University of Illinois at Urbana-Champaign. His current research interests include the design of energy-efficient integrated circuits and systems for communications, signal processing, and machine learning. He has authored or co-authored more than 200 publications in this area and holds 13 U.S. patents.

Dr. Shanbhag was a recipient of the National Science Foundation CAREER Award in 1996, the IEEE Circuits and Systems Society Distinguished Lecturership in 1997, the 2010 Richard Newton GSRC Industrial Impact Award, 826 and multiple Best Paper Awards. In 2000, he co-founded and served as the Chief Technology Officer of Intersymbol Communications, Inc., (acquired in 2007 by Finisar Corporation) a semiconductor startup that provided DSP-enhanced mixed-signal ICs for electronic dispersion compensation of OC-192 830 optical links. From 2013 to 2017, he was the founding Director of the Systems On Nanoscale Information fabriCs Center, a 5-year multi-university center funded by DARPA and SRC under the STARnet program.