31.3 Brain-Inspired Computing Exploiting Carbon Nanotube FETs and Resistive RAM: Hyperdimensional Computing Case Study

Tony F. Wu[1], Haitong Li[1], Ping-Chen Huang[2], Abbas Rahimi[2], Jan M. Rabaey[2], H.-S. Philip Wong[1], Max M. Shulaker[3], Subhasish Mitra[1]

[1]Stanford University, Stanford, CA
[2]University of California, Berkeley, Berkeley, CA
[3]Massachusetts Institute of Technology, Cambridge, MA

We demonstrate an end-to-end brain-inspired hyperdimensional (HD) computing nanosystem, effective for cognitive tasks such as language recognition, using heterogeneous integration of multiple emerging nanotechnologies. It uses monolithic 3D integration of carbon nanotube field-effect transistors (CNFETs, an emerging logic technology with significant energy-delay product (EDP) benefit vs. silicon CMOS [1]) and Resistive RAM (RRAM, an emerging memory that promises dense non-volatile and analog storage [2]). Due to their low fabrication temperature (<250°C), CNFETs and RRAM naturally enable monolithic 3D integration with fine-grained and dense vertical connections (exceeding various chip stacking and packaging approaches) between computation and storage layers using back-end-of-line inter-layer vias [3]. We exploit RRAM and CNFETs to create area- and energy-efficient circuits for HD computing: approximate accumulation circuits using gradual RRAM reset operation (in addition to RRAM single-bit storage) and random projection circuits that embrace inherent variations in RRAM and CNFETs. Our results demonstrate: 1. pairwise classification of 21 European languages with measured accuracy of up to 98% on >20,000 sentences (6.4 million characters) per language pair. 2. One-shot learning (i.e., learning from few examples) using one text sample (~100,000 characters) per language. 3. Resilient operation (98% accuracy) despite 78% hardware errors (circuit outputs stuck at 0 or 1). Our HD nanosystem consists of 1,952 CNFETs integrated with 224 RRAM cells.

For language classification using HD computing, an input sentence (entered serially character by character) is first time-encoded (input characters are mapped to the delay of the rising-edge of the signal *in* with respect to the reference clock, *clk1*, Fig. 31.3.1). The time-encoded sentence is then transformed into a *query hypervector* (QV) using 4 *HD operations:* 1. *random projection* (each character in a sentence is mapped to a binary hyperdimensional vector (*HV*), Projection unit, Fig. 31.3.1); 2. *HD permutation* (1-bit rotating shift of HV, HD Permute unit, Fig. 31.3.1); 3. *HD multiplication* (bit-wise XOR of two HVs, HD Multiply unit, Fig. 31.3.1); and 4. *HD addition* (bit-wise accumulation, HD Accumulator unit, Fig. 31.3.1). During training, this QV is chosen to represent a language and stored in RRAM, either in location 0 or location 1 (RRAM-based classifier, Fig. 31.3.1). During inference, the QV is compared with all stored HVs (from training) and the language that corresponds to the least Hamming distance (current on ml0 for location 0 and ml1 for location 1) is chosen as the output language.

Figure 31.3.2 shows the overall nanosystem. Our design (exploiting CNFETs, RRAM, and monolithic 3D) provides significant EDP and area benefits vs traditional 2D silicon CMOS-based digital design (e.g., projected 35× EDP, 3× area benefits for pair-wise language classification HD when compared at 28 nm technology node, estimated using simulations after place-and-route). A key requirement for HD computing is to achieve random projections of inputs to HVs [4]. To do so, we embrace the following inherent variations in nanotechnologies using a delay cell (PMOS-only CNFET inverter with an additional RRAM in the pull-down network, Fig. 31.3.2 and Fig. 31.3.3): variations in RRAM resistance and variations in CNFET drive current resulting from variations in carbon nanotube (CNT) count (i.e., the number of CNTs in a CNFET) or threshold voltage.

To generate HVs, each possible input (26 letters of the alphabet and the space character [4]) is time-encoded and mapped to an evenly-spaced delay (maximum delay *T*, Fig. 31.3.1). To calculate each bit of the HV, random delays are added to clk1 and input (in) using delay cells. If the resulting signals are coincident (the falling edges are close enough to set the SR latch) the output is ‘1’ (Fig. 31.3.3). To initialize delay cells, the RRAM resistance is first reset to a high-resistance state (HRS) and then set to a low-resistance state (LRS). This process is performed before training. During random projection operation, the RRAM in each delay cell is static (the voltage across the RRAM (0-1V) is not enough to change its resistance).

HD computing generally requires HVs representing these 27 possible inputs to be nearly orthogonal to each other (Hamming distance close to 50% of the vector dimension). This requirement (e.g., 16 for 32 bits) is fulfilled as delay variation (σ/μ) of delay cells increases (Fig. 31.3.4, σ: standard deviation of delays, μ: mean delay). By combining the RRAM resistance variations and CNFET drive current variations, our delay cells achieve experimentally characterized delay variations with $\sigma/\mu = 1.5$, corresponding to mean Hamming distance of 16 for 32 bits (Fig. 31.3.4), sufficient for random projection in HD computing. Other approaches (e.g., operating in subthreshold voltages to exploit inherent variations [5] or pseudo-random number generators such as linear feedback shift registers and its variants) can also be used to generate HVs.

Bit-wise accumulation corresponds to counting the number of ones in each bit location of the HV. For example, bit-wise accumulation of binary vectors 0100, 0101, and 1011 produces (1,2,1,2). Thresholding is performed to transform the vector back into a binary vector [4] (e.g., (1,2,1,2) turns into 0101). The threshold value is set to half of the number of total HVs accumulated (e.g., threshold 50 for 100-character sentence) [4]. For language classification, the average input sentence contains fewer than 128 characters (requiring 7-bits of precision to accumulate 128 HVs). Here, we use an approximate accumulator with thresholding: we leverage the multiple values of RRAM resistance that can be programmed by performing a gradual reset ($V_{top} - V_{bot}$: -2.6V) (when the RRAM is in the low resistance state, multiple reset pulses (50μs pulse width, 1ms period) are applied to gradually increase the resistance to the high resistance state) to count the number of ones (Fig. 31.3.3) [2]. Since the accumulated value is thresholded to a binary value, the impact of accumulation error is somewhat mitigated (with mean cycle-to-cycle error 4%, showing consistency across time) (Fig. 31.3.4). A digital buffer is used to transform (threshold) the sum to a binary vector (when clk is 0V and $V_{top} - V_{bot}$ is +0.5V). To clear the sum, a set operation is performed on the RRAM ($V_{top} - V_{bot}$: +2.6V). Each such approximate accumulator uses 8 transistors and a single RRAM cell. In contrast, a digital 7-bit accumulator may use 240 transistors. Thus, when D (e.g. 10,000) accumulators (where D is the HV dimension) are needed, the savings can be significant.

We create multiple functional units of the HD encoder (Fig. 31.3.5). These functional units, when connected in parallel (Fig. 31.3.2), form the HD encoder.

The classifier is implemented using 2T2R (2-CNFET transistor, 2-RRAM) ternary content-addressable memory (TCAM) cells to form an associative memory [6]. During training, the matchline (ML) (i.e. ml0 or ml1) corresponding to the language (e.g., ml0 for English and ml1 for Spanish) is set to 3V, writing the QV into the RRAM cells connected to the ML (Fig. 31.3.2). During inference, the MLs (i.e. ml0 and ml1) are set to a low voltage (e.g. 0.5V), and the current on each ML is read as an output. When the QV bit is equal to the value stored in a TCAM cell (match), the current is high. Otherwise (mismatch), the current is low. An individual TCAM cell has a match / mismatch current ratio of ~20 (Fig. 31.3.6). Cell currents are summed on each ML. The line with the most current corresponds to the output class (read and compared off-chip). Figure 31.3.6 shows a distribution of ML currents that correspond to Hamming distance. Our HD computing nanosystem is resilient to errors: despite 78% of QV bits (25 out of 32) stuck at 0 or 1 (characterized by measuring functional units in Fig. 31.3.5), our overall nanosystem still achieves 98% classification accuracy.

We emulate larger HD computing systems (more bits per HV) by running our fabricated nanosystem iteratively. In each iteration, language pairs are trained (each language is trained on one text sample of 100,000 characters, which uses the same amount of time per character as an inference: one-shot learning, Fig. 31.3.6) and all inferences are performed. After each iteration, the RRAM in the delay cells are cycled (i.e. reset to HRS then set to LRS, providing new projection of HVs for each iteration). After all iterations, the ML currents of the corresponding sentences are summed and compared. For a single iteration, the accuracy is 59%. Using 256 iterations with 22% non-stuck QV bits (D=8192, with 1792 non-stuck QV bits), our HD nanosystem can categorize between 2 European languages with a mean accuracy of 98% (Fig. 31.3.6). The classification energy per iteration is measured at 540μJ (3V supply, average power 5.4mW, 1kHz clock frequency, 1μm gate length). The reported accuracy is the percentage of 84,000 sentences that were classified correctly (dataset: 420 language pairs, 200 sentences per language pair). Software HD implementations (using 8192-bit HVs with HV sparsity 0.5) on general-purpose processors achieve 99.2% accuracy. This work illustrates how properties of heterogeneous emerging nanotechnologies can be effectively exploited and combined to realize brain-inspired computing architectures that: tightly integrate computing and storage, provide energy-efficient computation, employ approximation, embrace randomness, and exhibit resilience to errors.

Acknowledgements
Work supported in part by DARPA, NSF/NRI/GRC E2CDA, STARnet SONIC, and Stanford SystemX Alliance. We thank Edith Beigne of CEA-LETI for fruitful discussions.

References

[1] L. Chang, et al., “Technology Optimization for High Energy-Efficiency Computation,” Short course, *IEDM*, 2012.
[2] H.-S. P. Wong, et al., “Metal-oxide RRAM,” *Proc. of IEEE*, vol. 100, no.6, pp 1951-1970, May 2012.
[3] M. M. Shulaker, et. al., “Three-dimensional integration of nanotechnologies for computing and data storage on a single chip,” *Nature*, no. 547, pp. 74-78, July 2017.
[4] A. Rahimi, et. al., “Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing,” *Int. Symp. on Low Power Elec. & Design*, pp. 64-69, 2016.
[5] S. Hanson, et al., "Energy Optimality and Variability in Subthreshold Design," *Int. Symp. on Low Power Elec. & Design*, pp. 363-365, 2006.
[6] L. Zheng, et. al., “RRAM-based TCAMs for pattern search,” *IEEE ISCAS*, pp. 1382-1385, 2016.

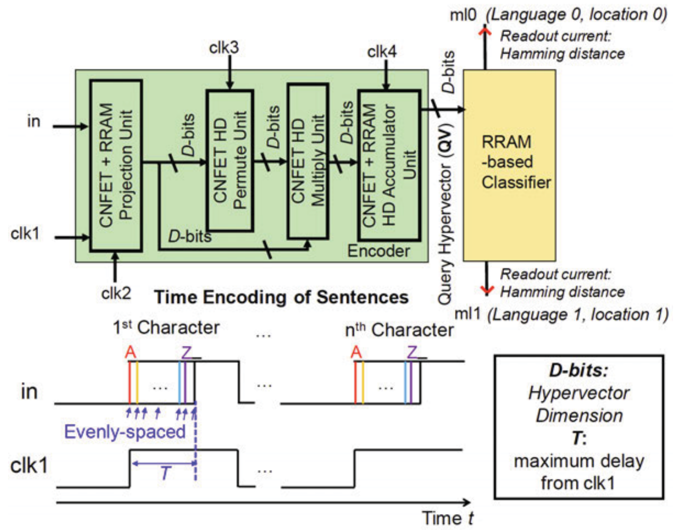

ml0 (Language 0, location 0)
Readout current: Hamming distance
clk3
clk4
in
CNFET + RRAM Projection Unit
D-bits
CNFET HD Permute Unit
D-bits
CNFET HD Multiply Unit
D-bits
CNFET + RRAM HD Accumulator Unit
D-bits
Query Hypervector (QV)
RRAM -based Classifier
clk1
D-bits
Encoder
Readout current: Hamming distance
clk2
Time Encoding of Sentences
ml1 (Language 1, location 1)
1st Character
nth Character
A
Z
in
Evenly-spaced
clk1
T
Time t
D-bits: Hypervector Dimension
T: maximum delay from clk1

Figure 31.3.1: HD computing architecture and time-encoding of input sentences.

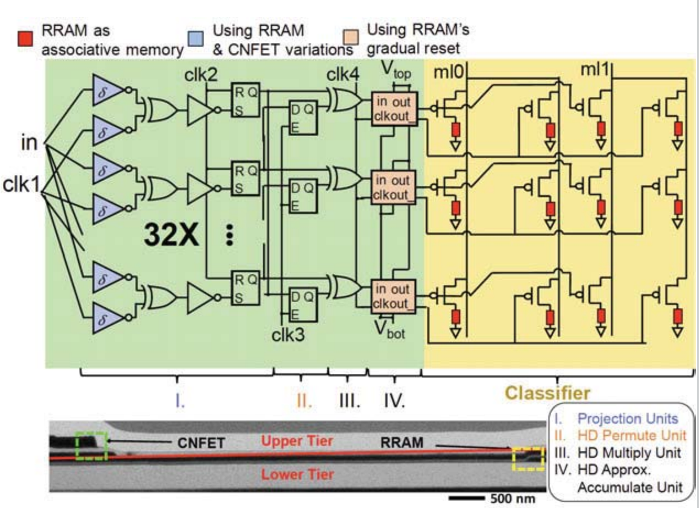
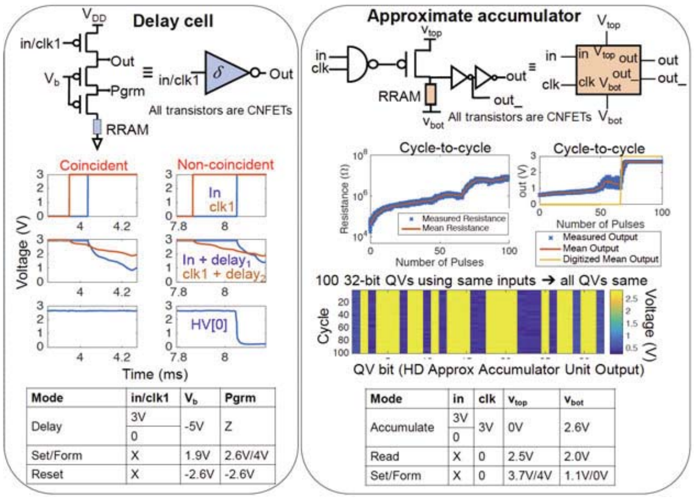


Figure 31.3.2: Schematic of HD computing nanosystem built using monolithic 3D integration of CNFETs and RRAM.



Mode	in/clk1	V_b	Pgrm
Delay	3V 0	-5V	Z
Set/Form	X	1.9V	2.6V/4V
Reset	X	-2.6V	-2.6V

Mode	in	clk	V_{top}	V_{bot}
Accumulate	3V 0	3V	0V	2.6V
Read	X	0	2.5V	2.0V
Set/Form	X	0	3.7V/4V	1.1V/0V

Figure 31.3.3: Implementation of a delay cell and approximate accumulator using CNFETs and RRAM.

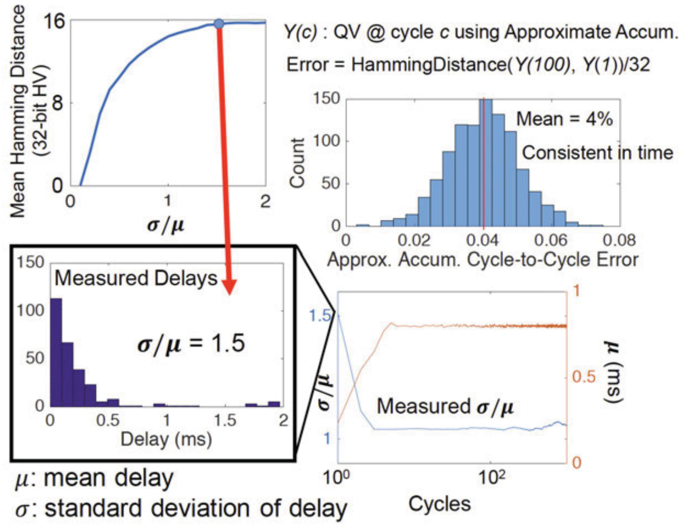


Figure 31.3.4: Characterization of the delay distribution of the delay cells.

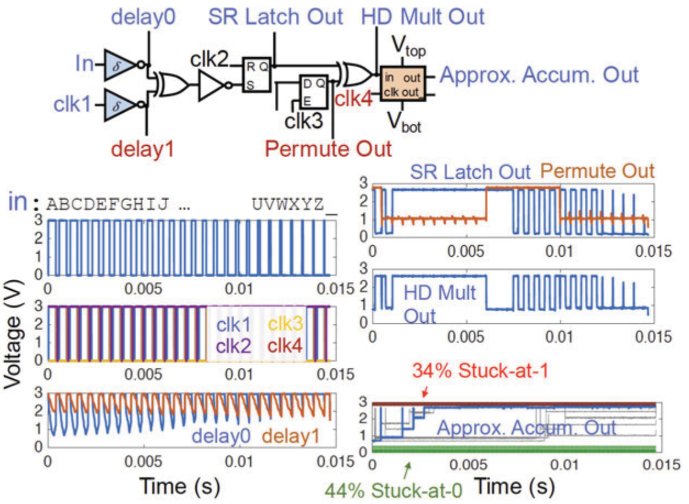


Figure 31.3.5: Input and output waveforms of a functional unit of the HD encoder with outputs of 32 approximate accumulators overlaid.

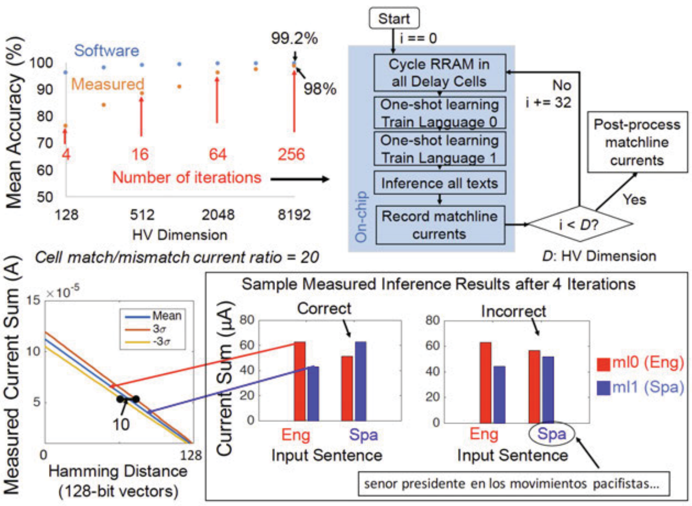
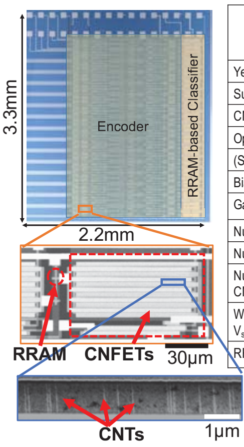


Figure 31.3.6: Pairwise classification of 21 European languages.



	This work	Science [8]	Nature [3]	Nat. Nanotech [9]	ACS Nano [7]	VLSI Tech [10]
Year	2018	2017	2017	2017	2017	2016
Supply Voltage	3V	0.4V	3V	1.9V	2V	-
CNFET Logic	PMOS	CMOS	PMOS	CMOS	CMOS	-
Operating Frequency	1 KHz	-	2 KHz	282 MHz	100 Hz	-
(S)ystem vs (T)ech Demo	S	T	S	T	S	T
Bias voltage	-5V	-	-5V	-	-	-
Gate length	1 μm	5 nm	1 μm	100nm	1 μm	-
Number of CNFETs	1952	2	>2^{21}	12	132	-
Number of RRAM	224	-	2^{20}	-	-	4
Number of Cascaded CNFET Logic Stages	16	-	>20	6	9	-
Write Condition (RRAM) V_{set}, V_{reset}	2.6V, -2.6V	-	2V, -2V	-	-	1.1, -1.5
RRAM properties utilized	I, II, III	-	I	-	-	I, III

I. Digital memory, II. Approximate accumulation, III. Stochasticity

[7] Y. Yang, et al., *ACS Nano, 11 (4),* pp 4124-4132, 2017.

[8] C. Qiu, et al., *Science*, 335, pp. 271-276, 2017.

[9] S.-J. Han, et al., *Nat. Nanotech.*, 12, pp. 861-865, Sept. 2017.

[10] H. Li, et al., *VLSI Technology*, 2016.

Figure 31.3.7: Die micrograph.

978-1-5090-4940-0/18/$31.00 ©2018 IEEE