Project Summary Report

The Catalyst Foundation
50 Main Street, Suite 920, White Plains, NY 10606, USA
Fax: (914)-681-9137

Shantanu Chakrabartty
Center for Language and Speech Processing (CLSP), and
Dept. of Electrical and Computer Engineering, Johns Hopkins University
3400 N. Charles St., Baltimore, MD 21218-2686, USA
Tel: (410) 516-7701; Fax: (410) 516-5566; shantanu@jhu.edu

Gert Cauwenberghs
Center for Language and Speech Processing (CLSP), and
Dept. of Electrical and Computer Engineering, Johns Hopkins University
3400 N. Charles St., Baltimore, MD 21218-2686, USA
Tel: (410) 516-5180; Fax: (410) 516-5566; gert@jhu.edu

Hervé Bourlard
Dalle Molle Institute of Perceptual Artificial Intelligence, IDIAP
P.O. Box 592, Rue du Simplon 4, CH-1920 Martigny, Switzerland
Tel: +41-27-721.77.25; Fax: +41-27-721.77.12; bourlard@idiap.ch

Jayadeva
Dept. of Electrical Engineering
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi - 110 016 India
Tel: +91-11-6857760; Fax: +91-11-6862037; jayadeva@ee.iitd.ernet.in

Summary

Driven by the proliferation of portable devices like cellular phones, personal digital assistants (PDAs) and smart wrist watches there has been an ever increasing demand for efficient and robust user interfaces. An intelligent speech interface offers an attractive alternative to other means of communication and provides hands free communication with these portable devices. Miniature handheld and wristworn devices require extreme low power solutions to support the use of very small batteries. Micropower analog VLSI provides a viable technology to implement a speech recognition user interface efficiently enough so that it can run off a wristwatch battery. From a computational perspective, parallel analog techniques are feasible because most of the computation involved in recognition is of a probabilistic nature that does not require high precision.

In the first part of the project we designed and developed efficient speech processing and recognition algorithms for small vocabulary systems, in light of efficient implementation in analog hardware. A flexible and scalable design approach allowed to reduce the complexity of the hardware by trading implementation accuracy for reduced silicon area and power dissipation. Theoretical research in this area has resulted in forward decoding kernel machines (FDKM), a maximum-a-posteriori (MAP) based sequence decoding scheme that combines traditional hidden markov models (HMM) with support vector machines (SVMs). The SVMs process acoustic features and produce HMM transition probabilities and a HMM forward decoding block integrates these probabilities to discriminate between phonetic utterances. The performance of FDKM depends on the discriminatory ability of the SVM generating margin classifier. Further investigation in this area has led to the development of the $Gini$-support vector machine (SVM), a sparse large margin classifier that generates normalized output probability scores. Both $Gini$-SVM and FDKM have demonstrated state-of-art performance on various signal processing tasks in speech and image recognition.

In the second part of the project the $Gini$SVM and FDKM algorithms were mapped onto parallel architecture, and implemented in low-power current-mode CMOS analog VLSI. Non-volatile floating-gate MOS storage provides full analog programmability and trainability throughout all stages of the architecture. A calibration scheme, coupled with a chip-in-loop retraining procedure, cancels imprecision due to fabrication-induced mismatch in the analog circuit implementation. A $Gini$SVM/FDKM processor was prototyped and fabricated in $0.5\mu m$ CMOS technology. In experiments on a speaker verification task, the chip yielded real-time recognition accuracy at par with floating-point software, but consumed sub-microwatt power.

Table 1: *Project milestones over annual grant periods*

Grant Period	**Accomplishments**
1999-2000	- FDKM decoder design and architecture - Version I of FDKM decoder chip
2000-2001	- FDKM training algorithms - Ratio spectrum front end chip - Core SVM software package development
2001-2002	- $Gini$SVM design and development - FDKM training algorithm using Expectation-Maximization - Scalable FDKM decoder chip
2002-2003	- $Gini$SVM prototype chip - Memory efficient and sparse algorithms - Convergence proofs and theoretical foundations of FDKM algorithms
2003-2004	- Fabrication, calibration and testing of $Gini$SVM/FDKM processor chip - Compilation and interfacing with PC-in-loop training software and adaptation scripts.

1 Introduction

Embedding speech recognition in miniatured handheld and wristworn devices require extreme low power solutions to support the use of very small batteries. Design of such sub-microwatt user interfaces requires power budget optimization to be performed rigorously at several levels [1, 2, 3] as shown in decreasing level in the design hierarchy:

1. **Algorithmic and System Level**— Designing algorithms that are robust to imprecision and noise, utilization of signal statistics, floor planning, data encoding, sleep modes and reference localization.

2. **Architectural Level**— Using parallel and pipelined architectures;

3. **Circuit and Logic Level**— Logic style, current starvation, switching behavior, supply switching and sub-threshold design;

4. **Technology Level**— Supply voltage reduction and threshold voltage tuning.

It has been argued that maximum power savings can be obtained by optimizing at higher levels of the design hierarchy, which has more degrees of freedom of implementation. A successful integration of algorithm onto hardware requires joint optimization keeping in mind the sensitivity of the system to typical imperfection of the circuits. Once the specifications for the system have been derived, different circuit topologies can then be evaluated for their ability to implement the system. Depending on the available degrees of freedom, the implementation can be optimized for area, speed and power consumption for a given accuracy.

1.1 Algorithmic Optimization

Several sequence decoding implementations using Hidden Markov Models (HMMs) have been proposed and used for speech recognition [4]. However, HMMs do not map efficiently onto analog VLSI due to problems in scaling of the HMM architecture. The number of parameters required to achieve reasonable recognition performance for a task is usually large. The nemesis to analog VLSI implementation [5] is the complexity in decoding using Viterbi-like algorithms [6]. As the decoding depth increases, the dynamic range of path scores decreases and may exceed the noise margin of the system, making it harder to discriminate between classes. A connectionist approach to *Maximum a Posteriori* (MAP) speech recognition [7] offers an attractive alternative to HMM maximum likelihood decoding that relaxes the need for a backward pass over the data and allows one to perform decoding in forward time. Our work in this direction has led to the Forward Decoding Kernel Machine (FDKM), a sequence decoding architecture based on statistical learning theory and Bayesian belief propagation. FDKMs are attractive for sequence classification because they provide a framework to elegantly trade between complexity and power-consumption of the system with limited dynamic range and system accuracy. Unlike traditional Viterbi decoding of HMMs, FDKMs use MAP (maximum a posteriori) decoding where the propagation probabilities are normalized at each stage, thus maintaining dynamic range [7]. The FDKM architecture directly leads to analog VLSI implementation offering real-time forward decoding.

At the core of FDKM, we developed the $Gini$-support vector machine ($Gini$SVM), a kernel-based probability regressor which maximizes noise margin and provides sparse output encoding based on a rate-distortion criterion. $Gini$SVM extends the functionality of support vector machines (SVMs) [8] for classification by providing class probabilities derived from maximizing a regularized form of quadratic-entropy. Training a $Gini$SVM entails solving a quadratic programming problem, where a unique solution is warranted, unlike conventional neural-network implementations. The power-consumption of the system is determined by the number of template vectors or support vectors stored which in turn is determined by the complexity of the discrimination boundary and signal-to-noise ratio of the sensor interface.

Our $Gini$SVM formulation led to an alternative novel sequence decoding algorithm called margin propagation. Instead of using conditional probabilities, margin propagation uses normalized margin scores generated by $Gini$SVM and can also be embedded into the FDKM architecture. Margin-based decoding requires only summation and subtraction as opposed to conventional probability decoding which requires multiplication, division and exponentiation, and therefore can be mapped onto architecture independent of its device characteristics. We also showed margin-based decoding to be more robust to effects of noise and mismatch prevalent in hardware implementation.

1.2 Architecture and Hardware Optimization

Evaluating an SVM decision function is computationally intensive, due to the fact that the input features have to be matched with several stored templates. Dedicated hardware would optimize the decision function by mapping it onto a computationally efficient architecture thus enhancing speed and performance. Previous SVM architecture [9] exploited the inherent parallelism in SVMs to implement hardware accelerators, for use in real-time video classification.

We formulated an array-based single instruction multiple data (SIMD) architecture implementing FDKM, which exploits a high degree of regularity in SVM and forward decoding computation. The parallel architecture conforms to a two-dimensional grid of computing elements interconnected so that shared inputs are along one dimension and shared outputs are along another dimension [10]. We achieved significant gains in computational efficiency by using primimitves of analog computation inherent in physical properties of devices and structural properties of circuits, such as the translinear principle and charge or current conservation [11, 12].

The FDKM architecture comprises matrix-vector multipliers (MVM), where each element in the MVM array computes the unsigned product of two variables: a varying input, usually provided along an input column, and a weight, stored in the computing element. The weights can be dynamic or static. Dynamic weight storage schemes use DRAM (dynamic random access memory) based principles [13], and allow the weights to be changed continuously without any impact on the device's computational throughput. The advantages of dynamic weight storage is its ability to incorporate active learning onto the system architecture which is an important consideration when systems are required to adapt to different environments. The disadvantage however is the use of refresh techniques to continuously update all the weights, increasing the standby power consumption of the system. For static weight storage, the weights are stored either as static digital memory or non-volatile analog memory. Static digital memory provide the flexibility of ease of adaptation of the weight parameters, but has a low storage density ($bits/\mu m^2$) as compared to non-volatile analog memory. Also the weights have to be re-initialized in the event of a system power failure, a frequent scenario for autonomous sensors powered by ambient energy sources.

Non-volatile analog memory [14, 15, 16] provides a compact storage cell, where the weights are stored as charge on a floating polysilicon gate of an MOS transistor. Because the gate is surrounded by high quality oxide, the charge on the gate is retained for long intervals of time, even after the system has been powered off. By biasing the floating gate MOS transistor in weak-inversion, the exponential relationship between the gate to source voltage and the drain current can be exploited to implement a single quadrant multiply accumulate (MAC) cell. Operating the cells in weak-inversion, maximum power-delay product is achieved at the expense of higher threshold mismatch and noise [17]. The mismatch between different cells is calibrated by using floating gate cells as adaptive elements, compensating for the imperfections inherent in fabrication. The effect of non-linearities in system implementation is then alleviated through a PC-in-the-loop re-training procedure, where the non-linearities are inherently parameterized as training variables.

2 Summary of Contributions and Accomplishments

The main contributions of this project consisted of developing the theory and design of robust pattern recognition and sequence decoding algorithms and architectures, and efficiently implementing them in analog VLSI achieving sub-microwatt operation. A summary of contributions with reference to resulting publications are described below.

2.1 Forward Decoding Kernel Machines

Forward-Decoding Kernel Machines (FDKM) provide an adaptive framework for general *maximum a posteriori* (MAP) sequence decoding, that avoid the need for backward recursion over the data in Viterbi and HMM-based sequence decoding. At the core of FDKM is a support vector machine (SVM) for large-margin trainable pattern classification, performing noise-robust regression of transition probabilities in forward sequence estimation. The achievable limits of FDKM power-consumption are determined by the number of support vectors (*i.e.*, regression templates), which in turn are determined by the complexity of the discrimination task and the signal-to-noise ratio of the sensor interface [18]. FDKMs have shown superior performance for wide range of applications ranging from channel equalization in digital communication [21] to phone sequence identification [19, 20].

2.2 Generalized Dual Framework and $Gini$ Support Vector Machine

Support Vector Machines are known to generate biased estimates of conditional probabilities which limits their use to developing hybrid classification systems in conjunction with other inference models. A generalized dual framework has been formulated for support vector machine like classifiers based of different dual potential functions. Within this framework several classifiers can be designed that produce normalized outputs and therefore be readily embedded in other probabilistic models like hidden Markov models [22]. One classifier of interest within this general class is the $Gini$ support vector machine ($Gini$SVM) which is based on quadratic entropy duals. Not only does $Gini$SVMs generate normalized output, further its solution is sparse compared to kernel logistic regression and is more suitable for large scale problems. $Gini$SVM showed similar or superior classification performance over other SVM approaches for tasks ranging from face detection to phone recognition [19]. The robustness of $Gini$SVMs particularly stands out, when classifiers are subjected to precision and mismatch errors. $Gini$SVM have found use in designing cephalometric landmark identification systems [23], where its performance is comparable or even superior than expert dentists. Other hybrid systems have used $Gini$SVM output scores in voting based architecture to achieve state-of-art performance in speech recognition [24]. Dissemination of our work on developing the $Gini$SVM classifier is supported by a publicly available software depository *http://bach.ece.jhu.edu/ ginisvm*. The software posted for public-domain downloading has been configured to perform $Gini$SVM based training and decoding for large scale real life classification problems.

2.3 Margin Propagation Networks

When embedded into graphical models, $Gini$SVM naturally lends to a novel decoding framework which we termed “margin propagation”. Instead of propagating probabilities, which form the backbone of all state-of-art analog decoders, margin decoders propagate classification margins. As a result the basic operations used for decoding are additions, subtractions and thresholding, operations which are independent of device characteristics, as opposed to probability propagation using translinear based principles. Margin based normalization and decoding has been shown to be more robust to additive and scaling noise than conventional probability propagation methods [25], making it attractive for analog VLSI implementation, and also for digital implementation in VLSI and in software on general-purpose computing platforms. A new learning algorithm using margin based decoding has shown to extract spatio-temporal information from training data.

2.4 Partition Label Machines

The efficiency of real-time classification for SVMs depends on the degree of sparsity as measured by the number of support vectors required for a given task, which for most practical scenarios is large. This is compounded by the fact that the complexity of SVM training scales with the square of the number of support

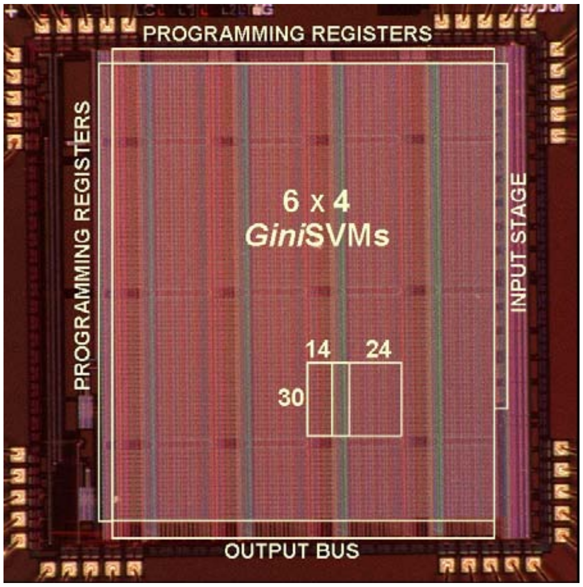


Figure 1: *Micrograph of the FDKM chip.*

vectors, and becomes prohibitively slow for huge datasets with significant class overlaps. Partition label machines [26] tackle the problem of sparsity by mapping the labels, rather than the data vectors, into a higher dimensional space. Through partitioning of the labels, the resulting partitioned classes can be linearly classified. The linear form gives rise to a sparse representation in the primal formulation, with an expansion that scales not with the number of training samples, but with the number of label partitions. Experiments show that label partitioning is effective in modeling non-linear decision boundaries with same, and in some cases superior, generalization performance to Support Vector Machines with significantly reduced memory and run-time requirements [26].

2.5 Sub-Microwatt FDKM in Analog VLSI

Optimization at combined algorithmic, system, circuit, logic and device levels resulted into an analog system-on-chip for kernel-based pattern classification and sequence estimation implementing the FDKM architecture. A prototype FDKM chip has been fabricated for use in adaptive sequence detection and pattern recognition [27]. State transition probabilities conditioned on input data are generated by an integrated support vector machine. Dot product based kernels and support vector coefficients are implemented in analog programmable floating gate translinear circuits, and probabilities are propagated and normalized using sub-threshold current-mode circuits. A 14-input, 24-state, and 720-support vector forward decoding kernel machine was integrated on a 3mm×3mm chip in 0.5μm CMOS technology whose micrograph is shown in Figure 1 and specifications summarized in table 2. The chip is fully configurable with parameters directly downloadable onto an array of floating-gate CMOS computational memory cells. By means of calibration and chip-in-loop training, the effect of mismatch and non-linearity in analog implementation is significantly reduced. Experiments with the processor trained for speaker verification and phoneme sequence estimation demonstrated real-time

Table 2: FDKM Chip Summary

Technology	**Value**
Area	3mm×3mm
Technology	0.5μ CMOS
Supply Voltage	4 V
System Parameters	
Floating Cell Count	28814
Number of Support Vectors	720
Input Dimension	14
Number of States	24
Power Consumption	80nW - 840nW
Energy Efficiency	1.6pJ/MAC

recognition accuracy at par with floating-point software, at sub-microwatt power. A scope plot demonstrating speaker verification being performed by the FDKM chip is shown in Figure 2(a) and (b) where a correct speaker is accepted and an imposter is rejected.

3 Conclusion

The five year term of this project has led to development of several key concepts bridging the field of pattern recognition and sequence estimation with analog VLSI. Robust pattern recognition and sequence decoding architectures have been developed and have been prototyped onto a ultra-low power analog VLSI system-on-chip. The research has laid the foundation of core FDKM based technology and its analog VLSI implementation which can be extended to design smart user interfaces to be embedded into portable devices.

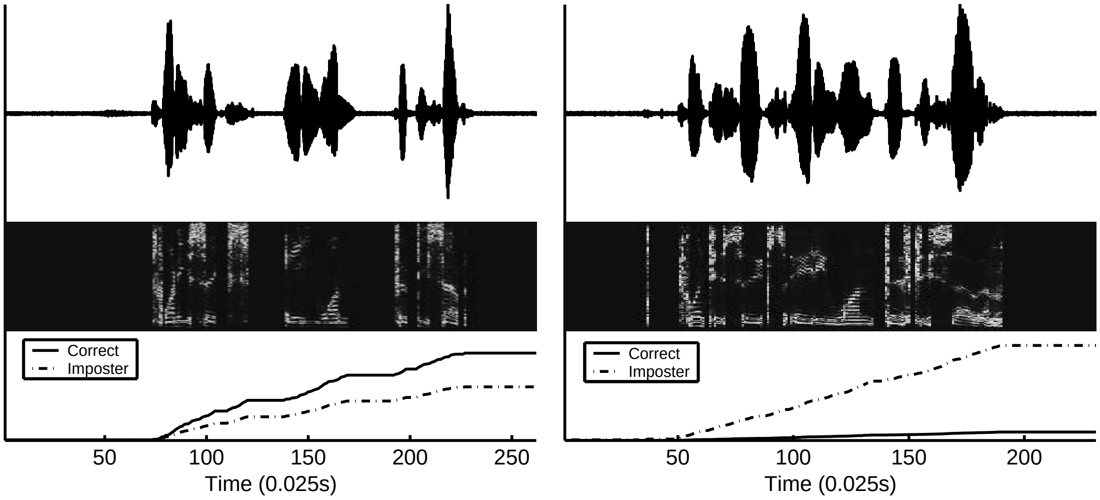
While low power dissipation is a virtue in many applications, increased power can be traded for increased bandwidth. For instance, the presented circuits could be adapted using heterojunction bipolar junction transistors in a SiGe process for ultra-high speed MAP decoding applications in digital communication, using essentially the same FDKM architecture as presented here. Similarly margin propagation network are based on principles that do not rely on device characteristics, making it more applicable to emerging newer generations of silicon and other devices.

References

[1] Chandrakasan, A.P, and Brodersen, R.W, "Minimizing power consumption in digital CMOS circuits," *Proceedings of the IEEE* , vol. 83, No.4 April 1995.

[2] Meindl, J, D, "Low Power Microelectronics: Retrospect and Prospect," *Proceedings of the IEEE,* vol. **83** (4), April 1995.

[3] Shanbhag, N.R, "A Mathematical Basis for Power-Reduction in Digital VLSI Systems," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing,* vol. **44**, pp. 935-951, 1997.



(a) *(b)*

Figure 2: *Experimental speaker verification with FDKM chip. Spectral features are first extracted from speech signal and then presented to the FDKM chip which computes confidence of correct and imposter states and integrates their values over time. (a) Verification of a correct speaker, whose confidence values is integrated over time. (b) Rejection of an imposter speaker.*

[4] Bahl, L.R., Cocke J., Jelinek F. and Raviv J. “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Inform. Theory,* vol. **IT-20**, pp. 284-287, 1974.

[5] K. He and G. Cauwenberghs “An Area-Efficient Analog VLSI Architecture for State-Parallel Viterbi Decoder,” *Proc. IEEE Int. Symp. Circuits and Systems,* Orlando FL,vol 2,pp.432-435,1999.

[6] G. David Forney. Jr, ”The Viterbi Algorithm,” *Proceedings of the IEEE,* vol.**61**,No.3,pp 268-278, March 1973.

[7] Bourlard, H. and Morgan, N., *Connectionist Speech Recognition: A Hybrid Approach,* Kluwer Academic, 1994.

[8] Vapnik, V. *The Nature of Statistical Learning Theory,* New York: Springer-Verlag, 1995.

[9] Genov, R. “Massively Parallel Mixed-Signal VLSI Kernel Machines,” *Ph.D Thesis, The Johns Hopkins University* , May 2003.

[10] A. Kramer, “Array-based analog computation,” *IEEE Micro,* vol. **16** (5), pp. 40-49, 1996.

[11] Andreou, A.G “On physical models of neural computation and their analog VLSI implementation,” *Proceedings., Workshop on Physics and Computation* , pp. 255-264,Nov. 1994.

[12] C. Mead, *Analog VLSI and Neural Systems,* Addison-Wesley, 1989.

[13] Cauwenberghs, G., and Yariv, A., “Fault-tolerant dynamic multilevel storage in analog VLSI,” *IEEE transactions on Circuits and Systems II* , vol. **41** (12), pp. 827-829, Dec. 1994.

[14] Shibata, T., and Ohmi, T., “A functional MOS transistor featuring gate-level weighted sum and threshold operations,” *IEEE Transactions on Electron Devices,* vol. **39** (6), pp. 1444-1455, Jun. 1992.

[15] P. Hasler, B. Minch, J. Dugger and C. Diorio, *Adaptive Circuits and synapses using PFET Floating Gate Devices,* Learning on Silicon: Kluwer Academic Publishers, Boston 1999.

[16] C. Dorio, P. Hasler, B. Minch and C.A. Mead, “A Single-Transistor Silicon Synapse,” *IEEE Trans. Electron Devices,* vol. 43 (11), Nov. 1996.

[17] Vittoz, E.A., “Low-Power Design: Ways to Approach the Limits,” *Dig. 41st IEEE Int. Solid-State Circuits Conf. (ISSCC)*, San Francisco CA, 1994.

[18] Chakrabartty, S., and Cauwenberghs, G. “Power Dissipation Limits and Large Margin in Wireless Sensors,” *Proc. IEEE Int. Symp. Circuits and Systems(ISCAS2003)*, vol. 4, 25-28, May 2003.

[19] Chakrabartty, S. and Cauwenberghs, G. “Forward Decoding Kernel Machines: A hybrid HMM/SVM Approach to Sequence Recognition,” *IEEE Int. Conf. of Pattern Recognition: SVM workshop. (ICPR’2002)*, Niagara Falls, 2002.

[20] Chakrabartty, S. and Cauwenberghs, G. “Forward-Decoding Kernel-Based Phone Sequence Recognition,” *Adv in Neural Information Processing Systems* Cambrige: MIT Press, vol.15, 2003.

[21] S. Chakrabartty and G. Cauwenberghs. “Sequence Estimation and Channel Equalization using Forward Decoding Kernel Machines,” *Proc. IEEE Int. Conf. of Acoustics, Speech and Signal Processing(ICASSP2002)* Orlando FL, 2002.

[22] S. Chakrabartty and G. Cauwenberghs. “Expection-Maximization of Forward Decoding Kernel Machines,” *Proc. 9th Int. Workshop Artificial Intelligence and Statistics* Key West,Florida, Jan 3-6 2003.

[23] S. Chakrabartty, M. Yagi, T. Shibata and G. Cauwenberghs. “Robust Cephalometric Landmark Idenfication Using Support Vector Machine,” *Proc. IEEE Int. Conf. of Acoustics, Speech and Signal Processing(ICASSP2003)* Hong kong, 2003.

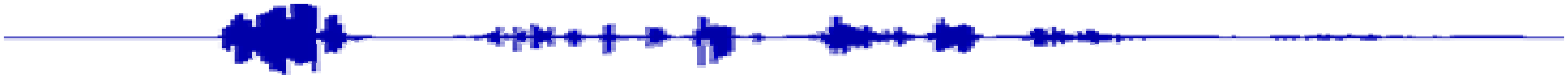
[24] V. Venkataramani, S. Chakrabartty and W. Byrne., “Support Vector Machines for Segmental Minimum Bayes Risk Decoding of Continuous Speech,” IEEE Automatic Speech Recognition and Understanding Workshop, 2003. To Appear.

[25] Chakrabartty, S., and Cauwenberghs, G. “Margin Propagation and Forward Decoding in Analog VLSI,” *Proc. IEEE Int. Symp. Circuits and Systems(ISCAS2004)*, Vancouver Canada, May 23-26, 2004.

[26] S. Chakrabartty, G. Cauwenberghs and Jayadeva. “Sparse Probability Regression by Label Partitioning,” *Proc. 16th Conf. Computational Learning Theory (COLT’03)* Washington DC, 2003.

[27] Chakrabartty, S. and Cauwenberghs, G. “Sub-Microwatt Analog VLSI Support Vector Machine for Pattern Classification and Sequence Estimation,” *Adv in Neural Information Processing Systems* Cambrige: MIT Press, vol.17, 2005.

Sub-Microwatt Analog VLSI Support Vector Machine for Pattern Classification and Sequence Estimation



Shantanu Chakrabartty

http://egr.msu.edu/~shantanu

Michigan State University

Gert Cauwenberghs

http://bach.ece.jhu.edu

The Johns Hopkins University

NIPS*2004

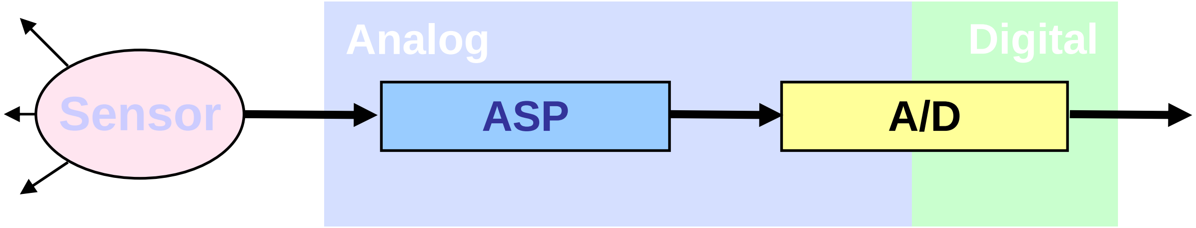
Vancouver, Canada, 12/15/2004

Sub-Microwatt Analog VLSI Support Vector Machine for Pattern Classification and Sequence Estimation

Outline

- **Motivation**
- **Kernel Classifiers and Sequence Decoders**
- **Architecture**
- **Building Blocks**
- **Calibration and Adaptation**
- **Conclusions**

“Smart” Sensors

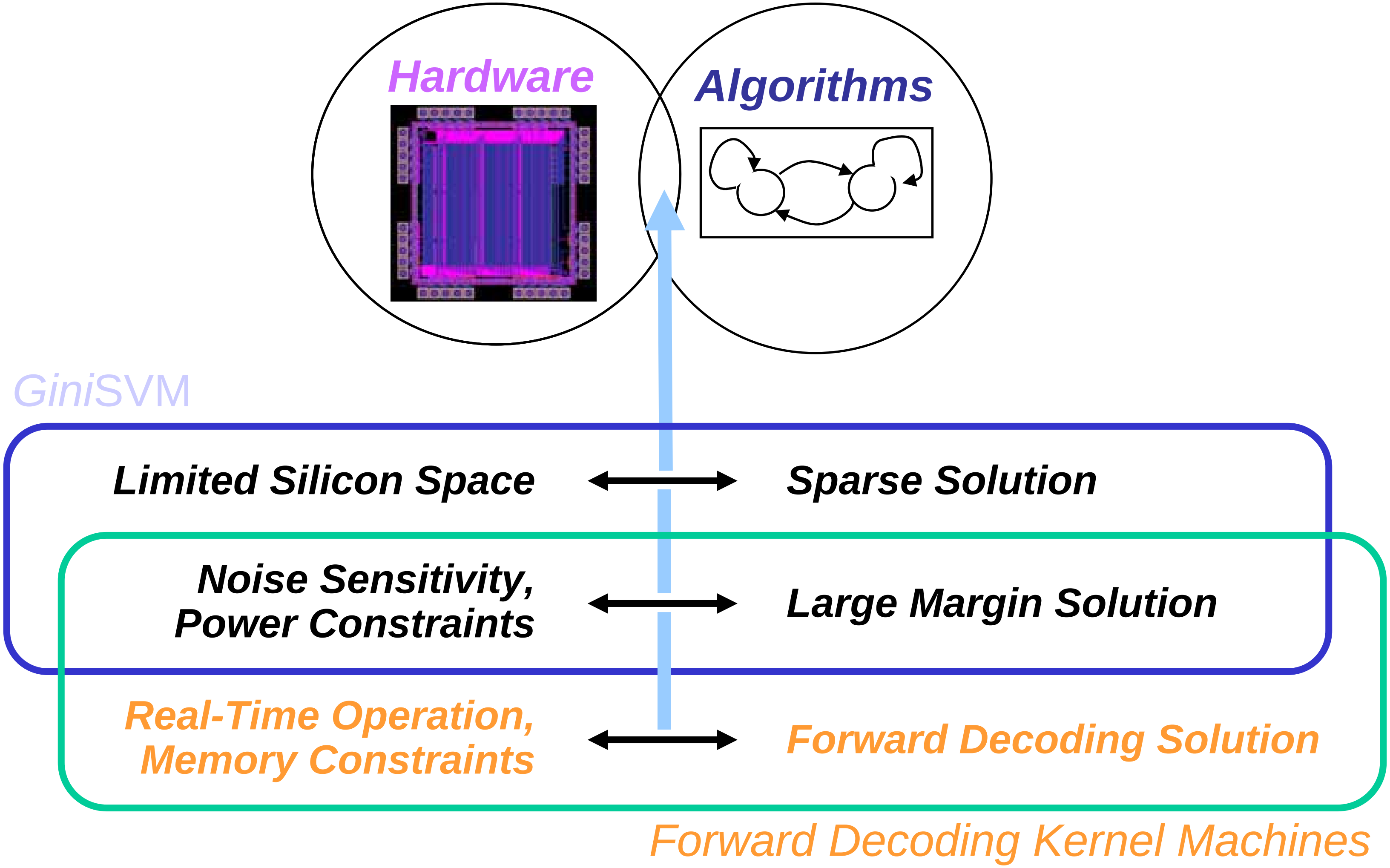


Classifier / MAP Decoder

e.g. Smart RFID tags; implantable biosensors.

- Reduced power dissipation is critical for autonomous sensors.
  - Solution: make the sensors “smart” – transmit only relevant information
  - Requires very low power classifiers and decoders
- Power budget optimization has to be performed at several levels:
  - Algorithms and architecture
  - Circuits and devices

*Gini*SVM and FDKM



*Gini*SVM

Limited Silicon Space ↔ ***Sparse Solution***

Noise Sensitivity, Power Constraints ↔ ***Large Margin Solution***

Real-Time Operation, Memory Constraints ↔ ***Forward Decoding Solution***

Forward Decoding Kernel Machines

Forward Decoding Kernel Machines (FDKM)

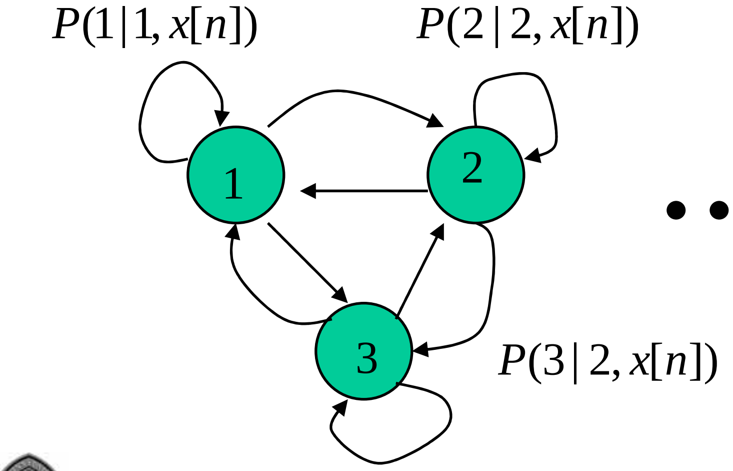
*Chakrabartty and Cauwenberghs, NIPS*2002*

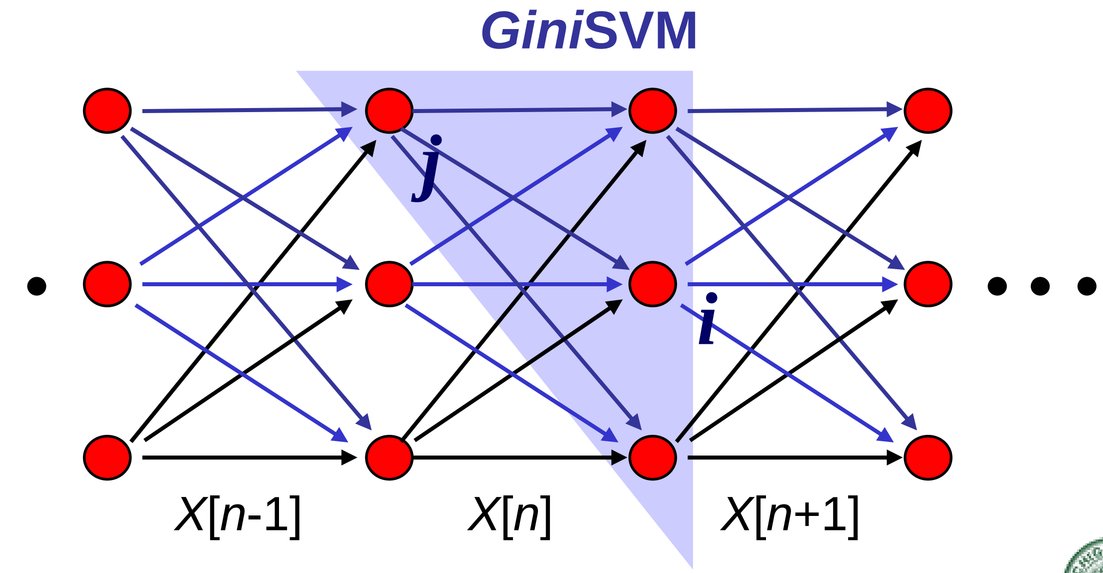
- **Forward decoding of posterior probabilities α_i**

$$\alpha_i[n] = \sum_j \alpha_j[n-1] P_{ij}[n]$$

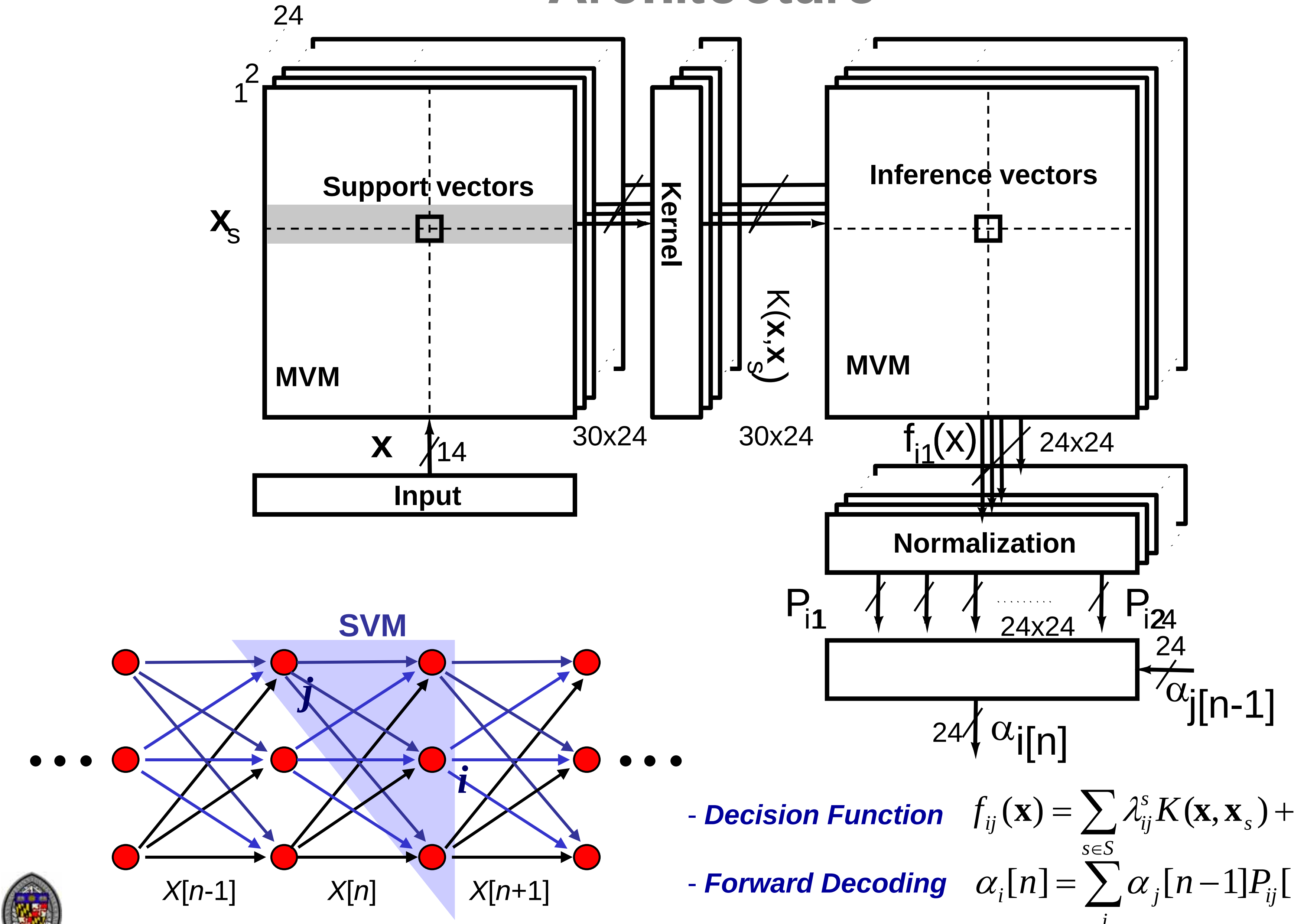
- **Transition probabilities P_{ij} generated by SVM conditioned on input data X**

$$P_{ij}[n] = P(i \mid j, X[n]) \propto f_{ij}(X[n])$$





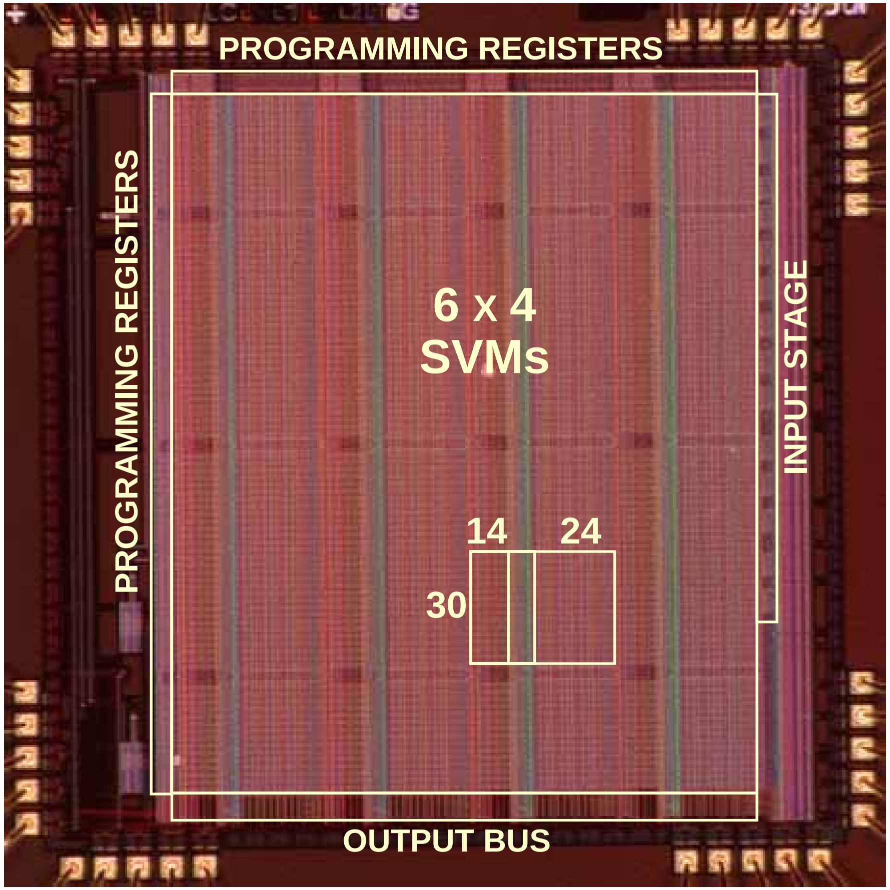
Architecture



- ***Decision Function*** $$f_{ij}(\mathbf{x}) = \sum_{s\in S} \lambda_{ij}^{s} K(\mathbf{x},\mathbf{x}_s) + b$$

- ***Forward Decoding*** $$\alpha_i[n] = \sum_{j} \alpha_j[n-1] P_{ij}[n]$$

*Gini*SVM/FDKM Processor



- 24 classes, 14 dimensions
  - current-mode analog input features and output probabilities
  - digital class outputs
- 720 support vectors
  - FDKM mode: 30/class
- 28,814 parameters stored on analog EEPROM cells
  - hot-electron injection and tunneling
- 3mm X 3mm in 0.5um CMOS
- Energy efficiency 1.6pJ/MAC
  - sub-microwatt power for speaker verification

Building Blocks

- **Inner-product computation**

  (multiply-accumulate operation):

$$\mathbf{x} \cdot \mathbf{x}_s$$

- **Kernel computation:**

$$K(\mathbf{x}, \mathbf{x}_s) = (\mathbf{x} \cdot \mathbf{x}_s)^2$$

$$f_{ij}(\mathbf{x}) = \sum_{s \in S} \lambda_{ij}^s K(\mathbf{x}, \mathbf{x}_s) + b$$

- **Normalization:**

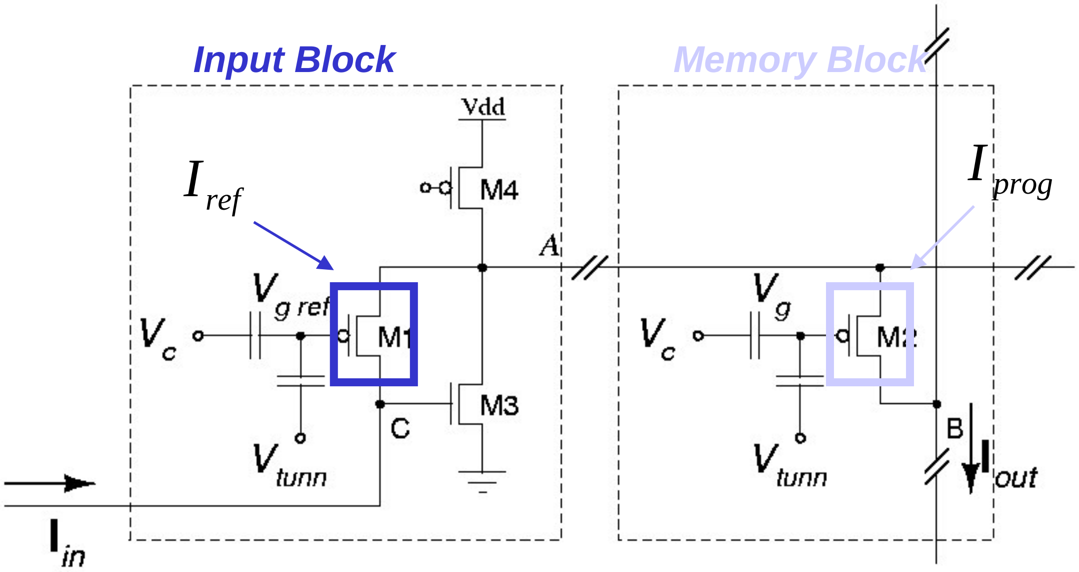
$$\sum_i [f_{ij} - z]_+ = \gamma$$

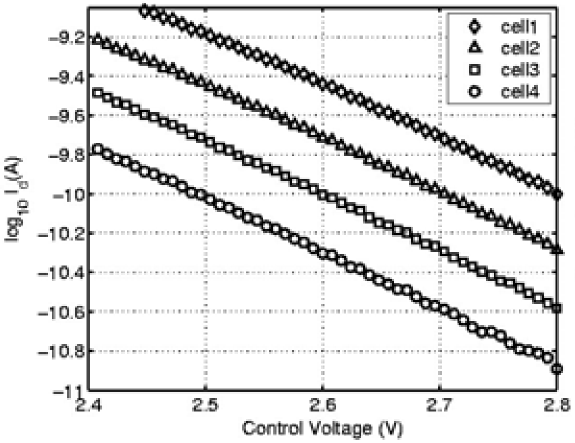
$$P_{ij} = \tfrac{1}{\gamma}[f_{ij} - z]_+$$

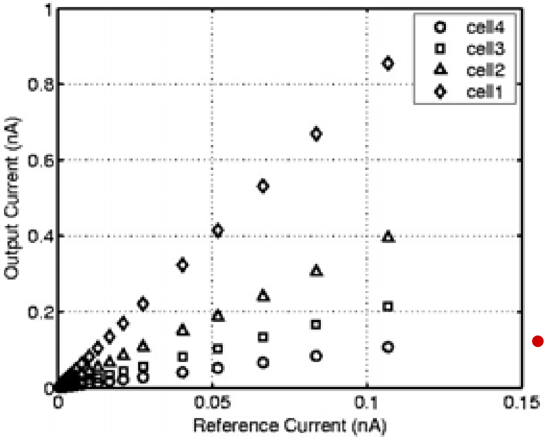
- **Forward decoding:**

$$\alpha_i[n] = \sum_j \alpha_j[n-1] P_{ij}[n]$$

Multiply-Accumulate Cell





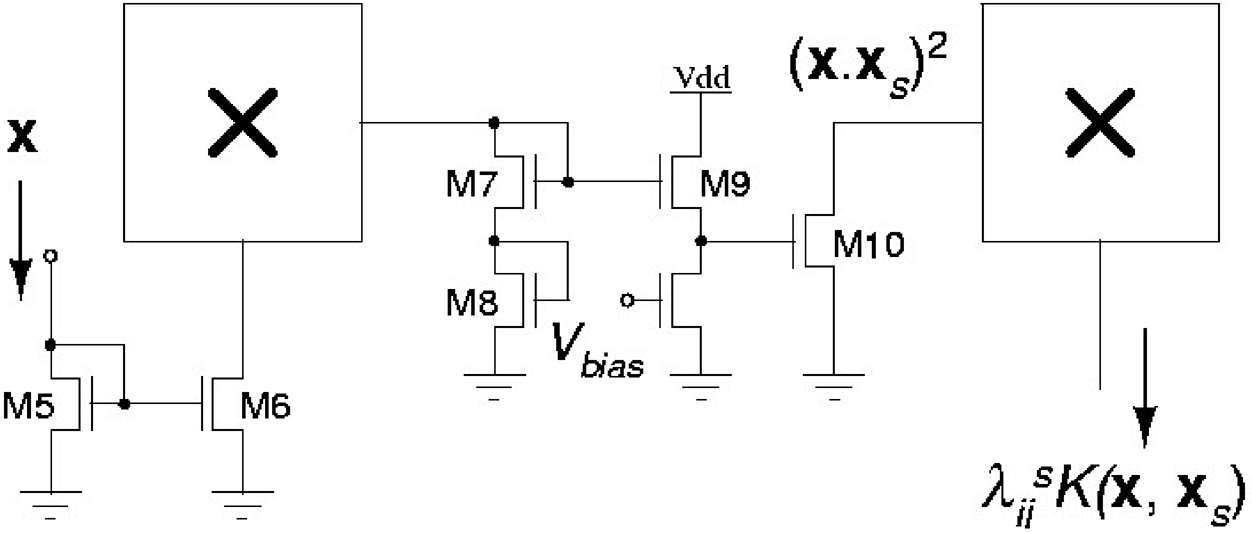


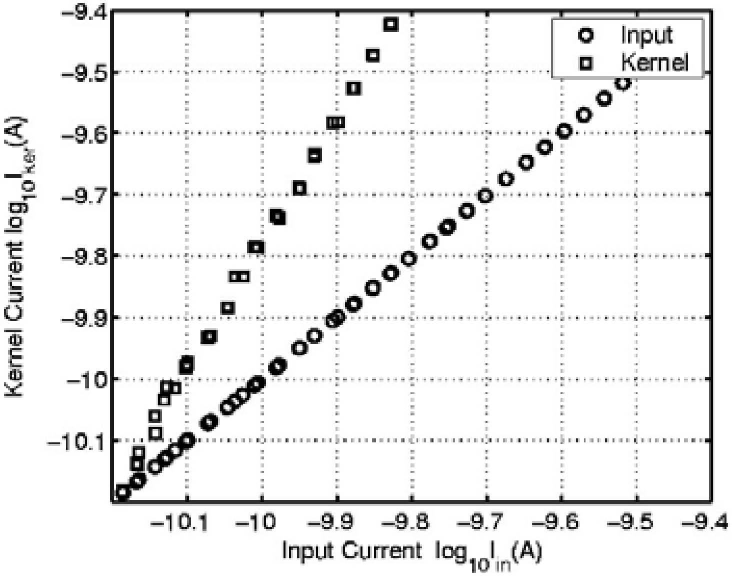
- **Sub-threshold design:**

$$I_{out} = I_{in} e^{-\kappa(V_g - V_{gref})/U_T} = I_{in} \frac{I_{prog}}{I_{ref}}$$

- **Addition operation is performed by current summation.**
- **Continuous-time current-mode architecture**
  - **Eliminates effects of substrate bounce on small currents.**
  - **Cancels out common mode effects (30dB Rejection).**
  - **Low impedance at node *A* using feedback transistors M3 enables large fan-out.**
  - **Low impedance nodes *C* and *B* reduces drain coupling onto floating gate.**
- **Calibration and retraining compensate for offset and gain mismatch in the analog implementation.**

Kernel Computation


x
M5
M6
M7
M8
V_{bias}
Vdd
M9
M10
$(\mathbf{x}.\mathbf{x}_s)^2$
$\lambda_{ii}^s K(\mathbf{x}, \mathbf{x}_s)$



- **Quantization at 6-7 bits of precision does not impact classification performance.**
- **Non-linearity in analog computation is accounted for by the kernel.**

$$K(\mathbf{x}, \mathbf{x}_s) = (\mathbf{x} \cdot \mathbf{x}_s)^{\frac{\kappa}{\kappa+1}}$$

SVM Output Normalization

Logistic Normalization:

$$P_{ij} = \frac{\exp(f_{ij})}{\sum_p \exp(f_{pj})}$$

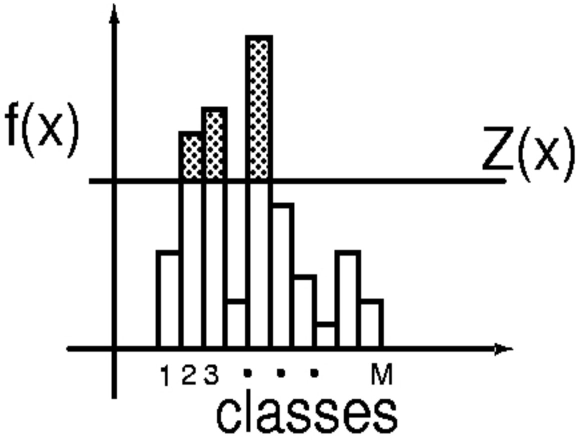
- **Exponentiation of currents difficult and sensitive to process parameters.**
- **Non-sparse**

Margin Normalization:

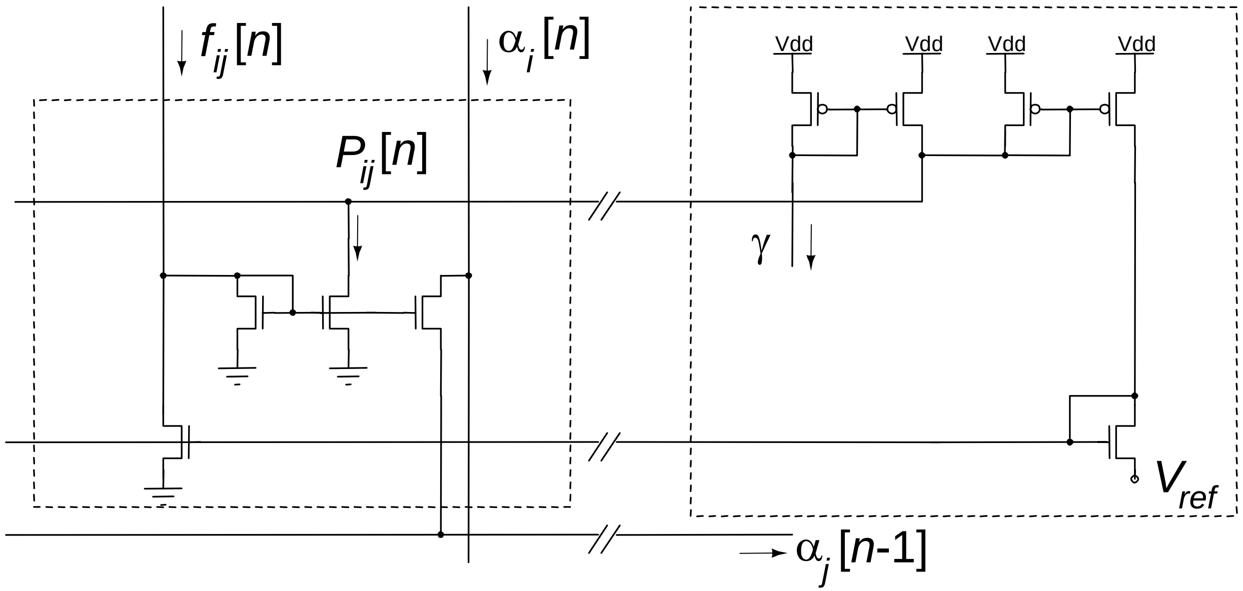
$$\sum_i [f_{ij} - z]_+ = \gamma$$

$$P_{ij} = \tfrac{1}{\gamma}[f_{ij} - z]_+$$

- **Requires only addition, subtraction and thresholding.**
- **Sparsely trained with *Gini*SVM**
  - **Reverse water filling:**



Normalization and Forward Decoding

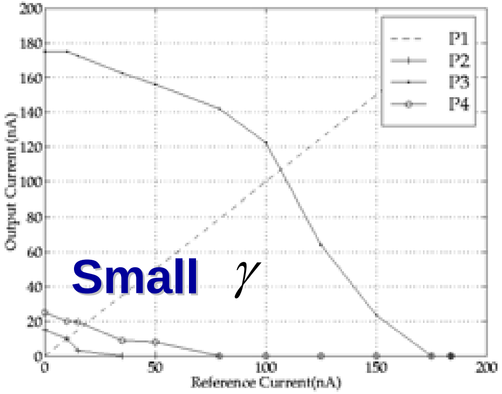


Forward Decoding

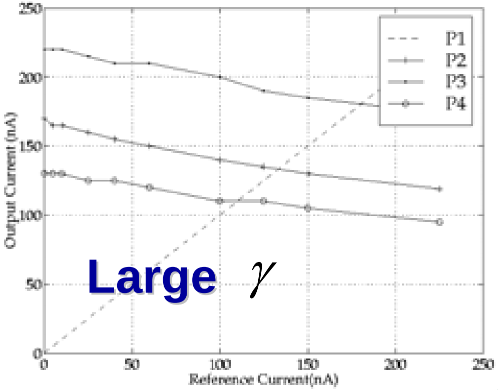
$$\alpha_i[n] = \sum_j \alpha_j[n-1] P_{ij}[n]$$

Reverse Water-Filling Normalization

$$P_{ij} = \tfrac{1}{\gamma}[f_{ij} - z]_+ \qquad \sum_i [f_{ij} - z]_+ = \gamma$$

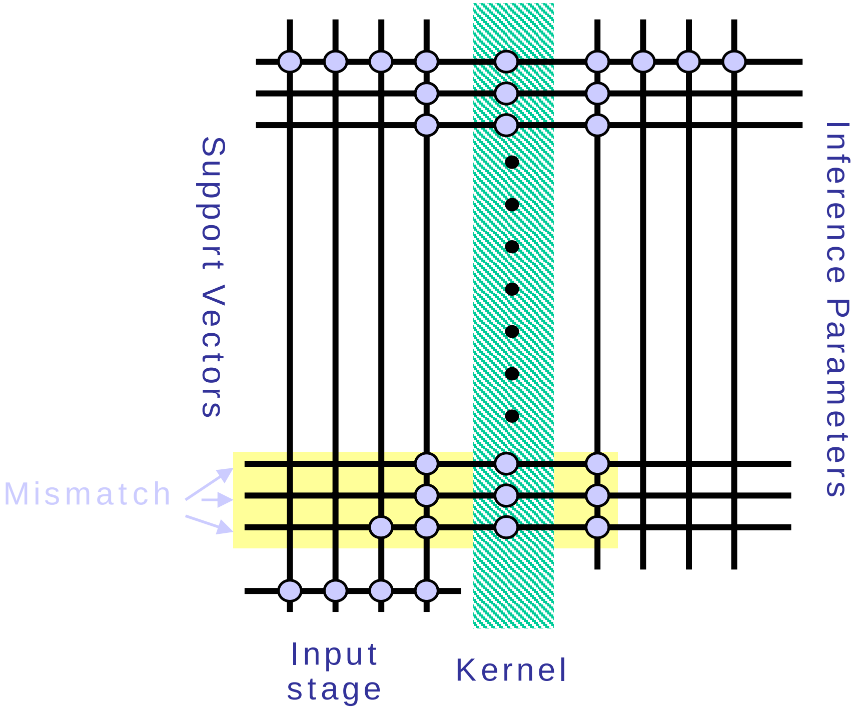


Small γ



Large γ

Calibration

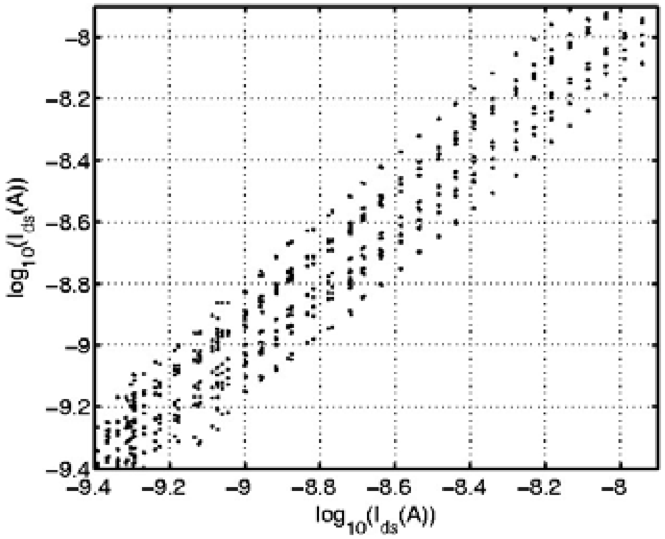

Support Vectors

Inference Parameters

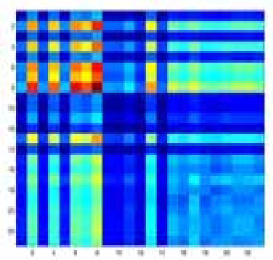
Mismatch

Input stage

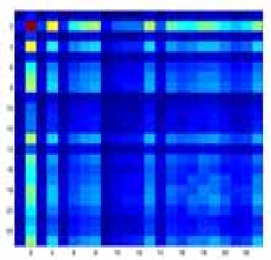
Kernel



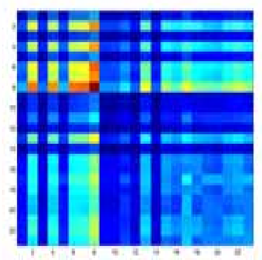
Before calibration



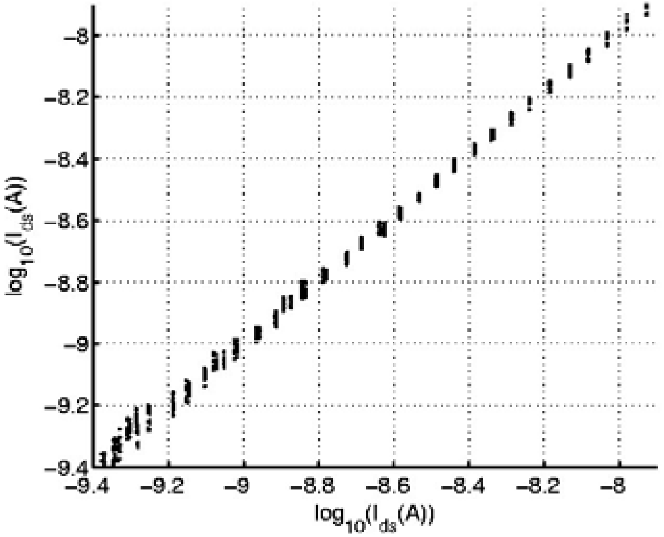
True kernel



Before Calibration



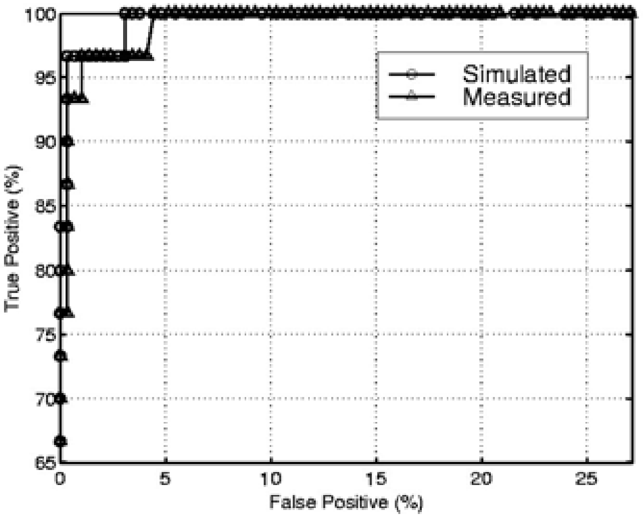
After Calibration



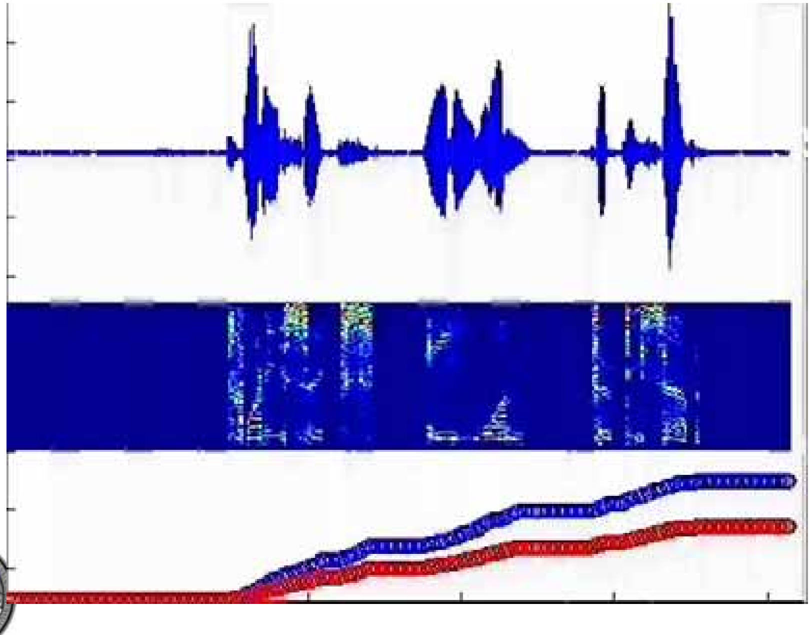
After calibration

Speaker Verification (840nW)

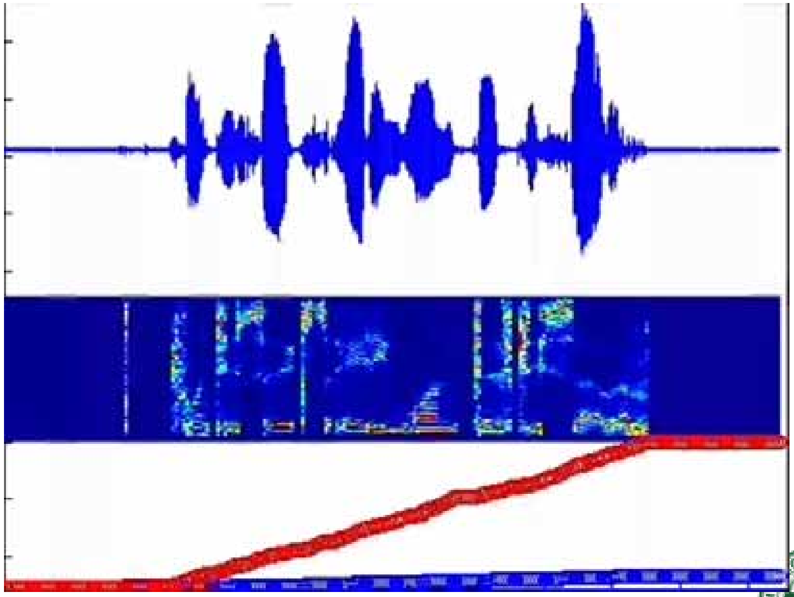
- 1 speaker and 10 imposters from YOHO dataset
- 92% recognition accuracy on 48 true and 432 imposter out-of-sample utterances
- 352 support vectors (47% FDKM chip capacity)
- 840 nW power at 25msec frame rate



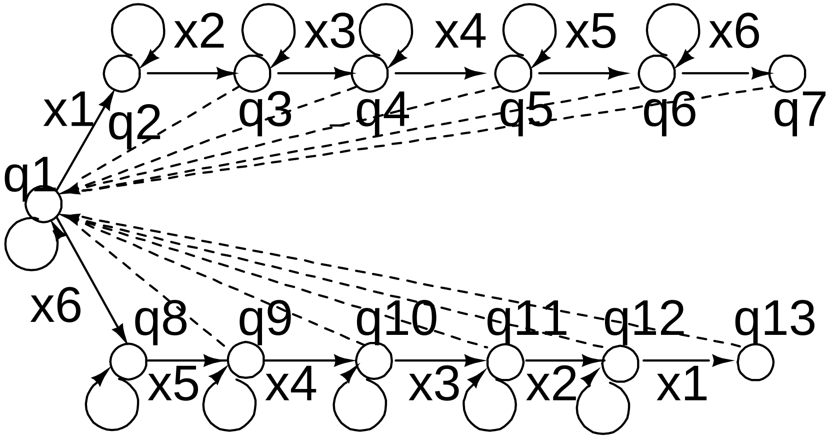
Correct Speaker

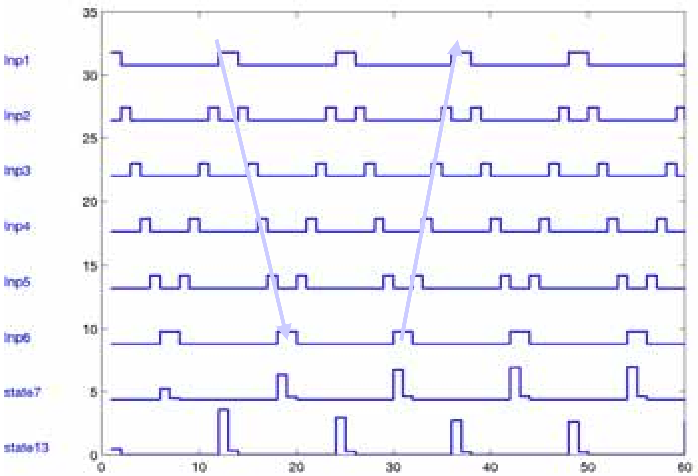


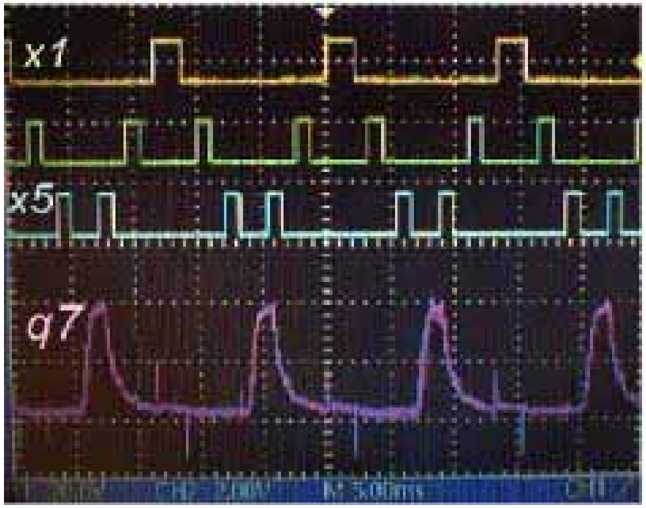
Imposter



FDKM Dynamic Sequence Detection

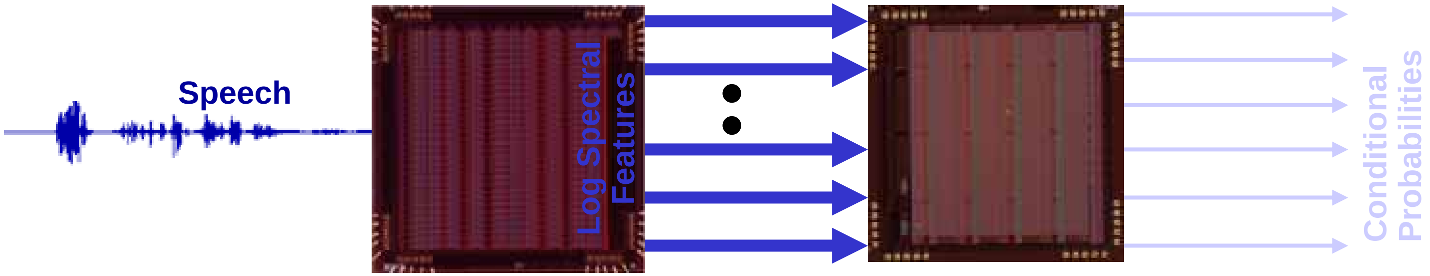






80 nW power

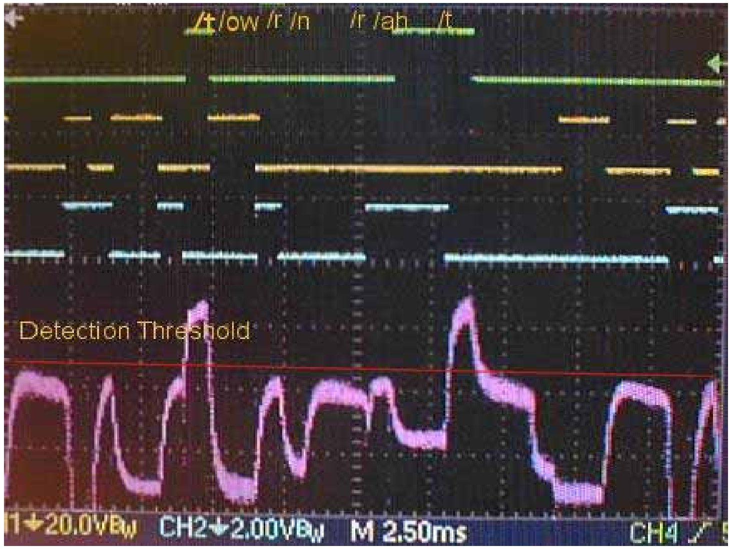
TIMIT Phone Recognition



Programmable Analog Filter bank (Deng et al, 2004)

FDKM

- 6 phones /t/n/r/ow/ah/eh/ from TIMIT corpus
- Thresholded Mel-cepstral features from log-compressed analog filterbank



Summary

- **Integrating adaptive recognition and intelligence at the sensor interface leads to significant savings in power and bandwidth, leading to greater autonomy in human machine interfaces.**
- **FDKM architecture for pattern classification and sequence decoding naturally lends to analog VLSI implementation using floating-gate translinear CMOS current-mode circuits.**
- **Calibration and retraining significantly reduce effects of fabrication-induced analog imprecision in the implementation.**
- **Classification and sequence decoding at sub-microwatt power levels have been experimentally demonstrated on tasks of speaker verification and phone classification.**

This work was supported by a grant from the Catalyst Foundation, New York (http://www.catalyst-foundation.org).