A Committee Machine Gas Identification System Based on Dynamically Reconfigurable FPGA

Minghua Shi, Amine Bermak, *Senior Member, IEEE*, Shrutisagar Chandrasekaran, *Student Member, IEEE*, Abbes Amira, *Senior Member, IEEE*, and Sofiane Brahim-Belhouari

***Abstract*—This paper proposes a gas identification system based on the committee machine (CM) classifier, which combines various gas identification algorithms, to obtain a unified decision with improved accuracy. The CM combines five different classifiers: K nearest neighbors (KNNs), multilayer perceptron (MLP), radial basis function (RBF), Gaussian mixture model (GMM), and probabilistic principal component analysis (PPCA). Experiments on real sensors' data proved the effectiveness of our system with an improved accuracy over individual classifiers. Due to the computationally intensive nature of CM, its implementation requires significant hardware resources. In order to overcome this problem, we propose a novel time multiplexing hardware implementation using a dynamically reconfigurable field programmable gate array (FPGA) platform. The processing is divided into three stages: sampling and preprocessing, pattern recognition, and decision stage. Dynamically reconfigurable FPGA technique is used to implement the system in a sequential manner, thus using limited hardware resources of the FPGA chip. The system is successfully tested for combustible gas identification application using our in-house tin-oxide gas sensors.**

***Index Terms*—Committee machine (CM), dynamically reconfigurable field programmable gate array (FPGA), gas identification, pattern recognition.**

I. Introduction

THE PAST DECADE has seen a significant research activity in the development of electronic nose (EN) for a wide range of applications in civil and military environments. Most of this work has been focused on systems using microelectronic gas sensors featuring small size and low-cost fabrication, making them attractive for consumer electronic applications. Unfortunately, gas sensors present a lack of selectivity and, therefore, respond similarly to a wide variety of gases. Thus, the general structure of an EN, which combines an array of sensors with signal preprocessing and pattern recognition algorithms, has been widely accepted and being used by researchers in this field [1]. Combining multiple classifiers to build an ensemble classifier is an advanced pattern recognition technique which has gained increasing attention within the machine learning community [2]. Ensemble-based classifier is generally more robust and accurate than a single classifier trained on the original dataset. However, ensembles suffer from some shortcomings as reported in [3]: "*While ensembles provide very accurate classifiers, there are problems that may limit their practical applications. One problem is that ensembles can require large amounts of memory to store and large amounts of computation to apply*." Thus, this scheme can be put to efficient practical use only if efficient hardware implementation strategies are developed. The appearance of fast reconfigurable field programmable gate arrays (FPGAs) brings about a new path for the design of such systems. FPGA-based implementation has the advantage of short design period and low cost of fabrication, which are suitable for the implementation and prototyping of new algorithms. However, the FPGAs present limited resources to implement complex hardware such as the EN system. Dynamically reconfigurable FPGA techniques [4]–[6] provide the possibility of implementing such complex system by using time multiplexing strategies.

In this paper, an EN system using a committee machine (CM) classifier is implemented using a dynamically reconfigurable FPGA-based system. The proposed CM combines five different pattern recognition algorithms: K nearest neighbors (KNNs), multilayer perceptron (MLP), radial basis function (RBF), Gaussian mixture model (GMM), and probabilistic principal component analysis (PPCA) [1], [7]–[9]. The CM combines the results of all the individual classifiers by applying a novel combination rule and achieves improved classification performance [10], [11]. This paper is organized as follows. Section II presents the general architecture of the EN system. Section III describes the proposed CM and evaluates its classification performance. Section IV describes the implementation of dynamically reconfigurable FPGA-based gas identification system and presents the detailed architecture of its building blocks. Section V presents the implementation results and discussion. Section VI concludes this paper.

II. Architecture of EN System

An EN is an instrument which comprises an array of electronic chemical sensors with partial specificity and an appropriate pattern recognition system, capable of recognizing simple or complex odors. As illustrated in Fig. 1, this process can be split into five stages: sensing, signal preprocessing, dimensionality reduction, prediction, and validation [1]. Sensors that can transform different gases into measurable electrical signals are fundamentally required in the application-specific sensing system. After the sensor signals have been acquired, the

Manuscript received July 5, 2007; revised November 24, 2007; accepted December 4, 2007. This work was supported by a grant from the Research Grant Council of Hong Kong SAR, China, under Project HKUST6162/04E. The associate editor coordinating the review of this paper and approving it for publication was Prof. Ralph Etienne-Cummings.

M. Shi, A. Bermak, and S. Brahim-Belhouari are with the Smart Sensory Integrated Systems (S2IS) Laboratory, Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: eebermak@ust.hk).

S. Chandrasekaran and A. Amira are with Brunel University, Uxbridge, Middlesex UB8 3PH, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2008.917124

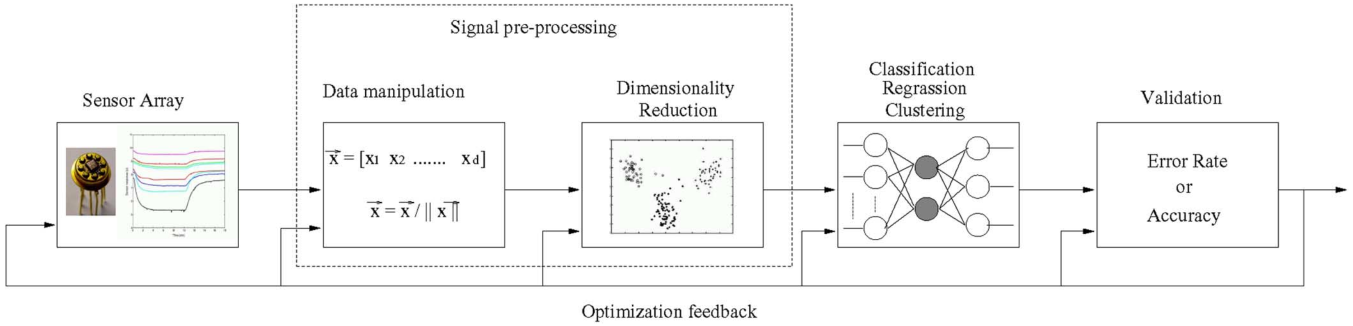


Fig. 1. Building block of the pattern analysis system for an EN.

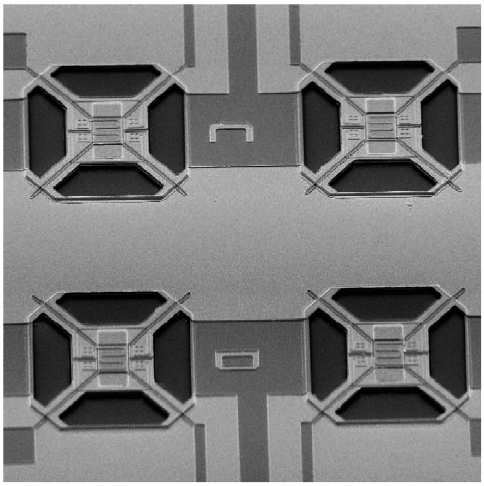


Fig. 2. Microphotograph of the integrated gas sensor array.

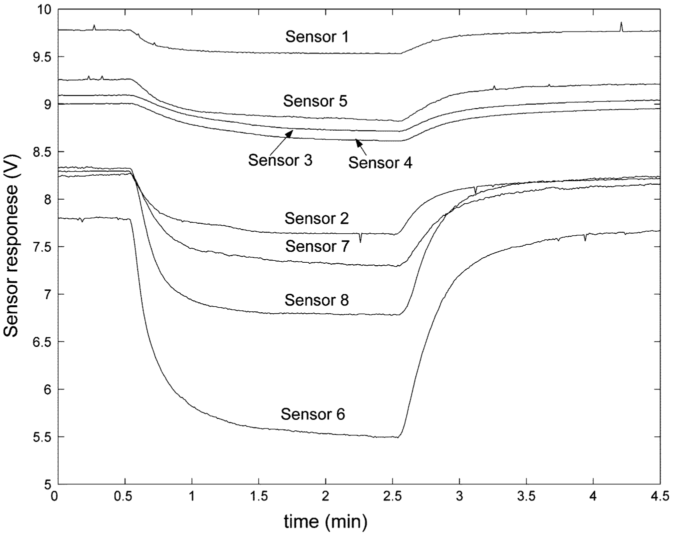


Fig. 3. Typical response of the sensor array. The voltage measurement of the sensor decrease when exposed to the analyte gases.

preprocessing stage is used to extract descriptive information from the sensor array response and generate the feature vector for further processing. Dimensionality reduction is an optional stage used to project the initial feature vector to a lower dimensional space. The prediction stage is used to solve the problem of classification, regression, or clustering depending on the application requirement. Here, we focus on the application of combustible gas recognition, this stage can be called pattern recognition stage. The final stage is used to estimate the true error rates or select the parameter settings for a trained model by means of validation techniques but it is not necessary for the practical EN system.

In our application, microhotplate tin oxide gas sensors are used for combustible gas identification application. Tin-oxide gas sesnors feature the advantage of good sensitivity to combustible gases, low-power consumption, low fabrication cost, and compatibility with semiconductor technology [12]. Fig. 2 shows a microphotograph of the manufactured chip including four sensors on a single chip. Each sensor has its own heater and temperature sensor. Three different sensing films are used to implement the sensor array. One sensor is based on Au/SnO (sensor 1), another sensor is based on Pt/Cu(0.16 wt%)-SnO (sensor 2), and the remaining two sensors are based on Pt/SnO (sensors 3 and 4) [13]. Totally, two chips were used and calibrated by tuning their selectivity to a given set of gases using the temperature parameter. A good sensitivity to H was obtained at about 260 °C, while a good sensitivity to CH was obtained at about 300 °C. It should be noticed that the two chips are identical; however, the operating temperature is different, allowing us to tune the selectivity of the two chips to different gases. The two chips provide eight responses, which could be seen as a fingerprint or a signature corresponding to a given gas mixture, which can then be exploited by a pattern recognition system in order to build a selective detection system, as will be described in the following sections.

When the sensors are exposed to the analyte gases, the resistance of the sensor will change. Thus, the voltage response of the sensor is measured using a simple voltage divider. Fig. 3 shows the row response of the two sensor chips (operated at different temperature) to the an input gas. The steady-state of the sensors are used as features. Thus, a gas pattern is a vector with dimension of eight. After the gas pattern is obtained from the sensor array, Euclidean normalization is performed to reduce the pattern dispersion induced by concentration changes. Normalization has been previously employed in gas discrimination applications where the identification must be based on signature pattern, and not on the concentration dependent amplitudes [13]. Normalization is also useful to set the range of values for sensors output. Principal Component Analysis (PCA)

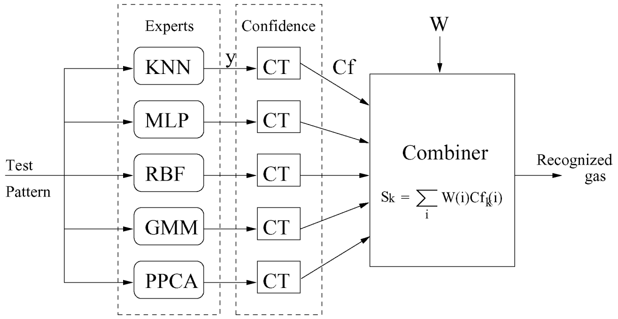


Fig. 4. Architecture of the CM (CT stands for CT).

is then used to remove the redundant information of the gas pattern and reduce the complexity of the pattern recognition stage by decreasing the dimension of the gas pattern [14], [15]. Finally, the pattern recognition stage will process the gas pattern to perform the classification. CM is used as a pattern recognition stage, which will be described in the next section.

III. Committee Machine (CM) and Weighted Combination

A. Committee Machine (CM) Architecture

A CM is a classical ensemble method which combines a mixture of experts and effectively make use of the results produced by individual classifiers to improve the classification performance. Fig. 4 shows the system overview of our CM combining five classifiers (KNN, MLP, RBF, GMM, and PPCA). In order to combine the results from each individual classifier, the outputs of the classifiers are first transformed to confidences using confidence transform functions (CT as shown in Fig. 4). The confidences are then combined using weighted combination rule to generate the score for each gas. The class with the highest score would be selected as the recognized gas.

B. Confidence Transform (CT) Functions

The CT method is the combination of a scaling function and an activation function [16]. The scaling and activation functions are used to normalize the outputs from each classifier in order to make the outputs comparable and meaningful by transforming the classifiers output to a moderate range. In order to find the confidences of various classifiers, we used different functions depending on the classifier [17], as follows.

- K nearest neighbor: The classifier calculates the distance between the unknown pattern $\boldsymbol{x}$ and each data pattern. The closest K examples will be found in the data set and the predominant class $C_{\hat{k}}$ is selected among those K neighbors. Thus, the confidence of the result is defined as the number of neighbors belonging to each class divided by K, i.e.,

  $$Cf_k = \frac{K_k}{K}. \tag{1}$$

  In our experiment, we set $K = 3$. This choice of $K = 3$ in implementing the KNN was purely experimental and based on the classification performance. KNN was tested for different values of K and $k = 3$ enables the best performance for our specific application.
- Density models (GMM and PPCA): The classifier will model the class conditional density $\wp(\boldsymbol{x}|C_k)$ (i.e., the model is trained for each class) and then by applying Bayes' rule to compute the posterior distribution, we obtain

  $$\wp(C_k|\boldsymbol{x}) = \frac{\wp(\boldsymbol{x}\,|\,C_k)\wp(C_k)}{\sum_{l=1}^{c}\wp(\boldsymbol{x}\,|\,C_l)\wp(C_l)}. \tag{2}$$

  The posterior probability $\wp(C_k|\boldsymbol{x})$ is considered as the confidence for our density models.
- Neural networks: The output of MLP approximates the posterior probabilities. Although, in practice the number of training data patterns is limited, the outputs will not always represent probabilities. In order to interpret MLP outputs as probabilities (confidence), the following function is used:

  $$Cf_k = \frac{\exp(y_k)}{\sum_{i=1}^{c}\exp(y_i)} \tag{3}$$

  where y_k is the output related to k_{th} class. For RBF, the outputs can also be interpreted by the posterior probabilities of class membership.

C. Combination Rules

In order to reduce the risk of any algorithm that performs poorly on average from affecting the ensemble decision, we propose to use the weighted combination rule to combine the confidences computed from the outputs of each individual classifier. The weighting and combination block shown in Fig. 4 assembles the results by calculating the score S of each class as follows:

$$S_k = \sum_i W(i)Cf_k(i). \tag{4}$$

The class with the highest score would be selected as the recognized class of the CM. We propose a weighting function to calculate weights for individual classifier [17]

$$NW_i = \frac{P_i - P_w}{P_b - P_w} \tag{5}$$

where NW_i is the weight of a given classifier i with a given performance P_i. P_b and P_w are the best and the worst classification performance within the CM, respectively.

To evaluate the performance of the proposed CM and individual classifiers, we used a gas data set, which is collected from our tin-oxide gas sensor array. Table I summarizes the analyte gases and their concentration ranges used in the experiment. The steady states of the sensors are used as features. The original gas data set with dimension of eight is projected to a lower dimension space by using PCA. Tenfold cross validation method is used to obtain the average classification performance of all the classifiers. Table II reports the average accuracy of the trained classifiers based on gas data set with different principal components (PCs).

Using the normalized weights, the impact of each classifier is normalized with respect to its performance with the CM. At the same time, the worst classifier is removed from the ensemble

TABLE I
GASES AND THEIR CONCENTRATION RANGES

Gas	concentration range (ppm)
CO	25-200
H_2	500-2000
CH_4	500-4000
$CO-H_2$	25-200&500-2000
$CO-CH_4$	25-200&500-4000

TABLE II
GAS IDENTIFICATION RESULTS BASED ON GAS DATA SET WITH DIFFERENT PRINCIPAL COMPONENTS (%)

No. of PCs	2	3	4	5	6	7	8
KNN	79.1	86.4	88.6	87.7	88.2	89.1	89.1
MLP	81.4	88.2	91.8	90.5	93.2	93.6	92.3
RBF	70.9	65.5	86.4	86.8	81.8	83.2	82.3
GMM	75	86.4	90.9	94.5	90	91.8	90.9
PPCA	70.5	81.8	84.5	84.1	79.1	79.1	79.1
CM	82.7	**90.9**	**93.2**	**95.5**	**94.5**	**95.5**	**95.9**

because its corresponding normalized weight will be zero. The results demonstrate that with the use of confidence and normalized weighting function, poor result from individual classifiers would not affect the ensemble result.

IV. IMPLEMENTATION OF EN SYSTEM BASED ON DYNAMICALLY RECONFIGURABLE FPGA

A. System Overview

We have seen in the previous section that the EN system consisted of data acquisition, signal preprocessing, and pattern recognition stages. Our proposed CM can achieve over 95% classification performance. However, the implementation of the whole system will occupy very large hardware resources. Thus, using full-custom IC design will lead to high fabrication cost. In addition, the response of the sensor is very slow (the time to achieve steady-state is around a few hundred seconds to a few minutes) [18], [19], thus, full-custom IC design, which is generally used to achieve high-speed operation, is not necessary for our application. Semi-custom design such as DSP, microcontroller, or FPGA is another viable option but only if sufficient hardware resources are available.

Due to the slow response of the sensor array, the time constraint is relaxed allowing to operate each stage of the system sequentially. This relaxed requirement will permit to use all the hardware resources for different stages, while the gas identification system is implemented, as shown in Fig. 5.

This system can be regarded as a dynamically reconfigurable system which can implement different stages at different times. Reconfigurable FPGA presents a good solution to implement such dynamic reconfigurable system. For the whole system, we can first implement data acquisition and signal preprocessing stage on the FPGA to generate a valid pattern to be processed at a later stage by the pattern recognition system. After signal preprocessing is performed, the gas pattern will be stored in the SRAM and the FPGA will be automatically reconfigured to implement the CM. The CM will first read out the gas pattern stored previously in the SRAM and perform the classification. All the results will be stored again in the SRAM before the FPGA is reconfigured to implement the final decision stage. Finally, the decision stage will output the results from the CM and provide the classification results.

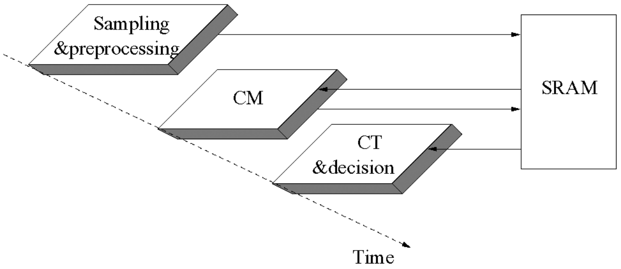


Fig. 5. Gas identification system based on dynamically reconfigurable design.

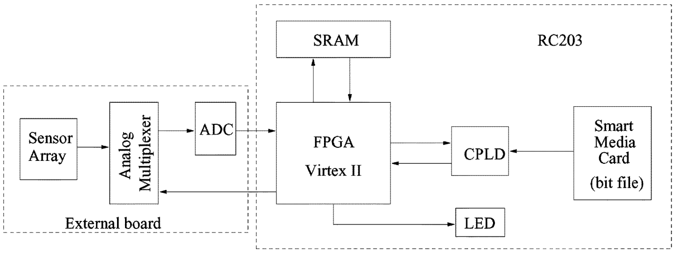


Fig. 6. The gas identification system platform.

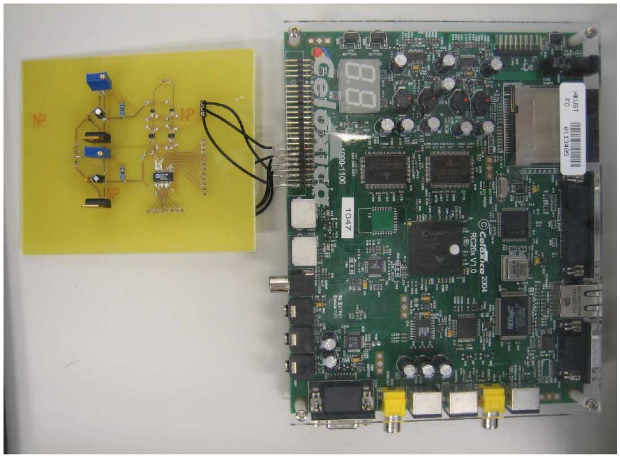


Fig. 7. Photograph of the gas identification system board.

The system platform is shown in Fig. 6. The platform consists of two boards: the sensor interface board and the FPGA board. A gas sensor array chip, analog multiplexer, and ADCs are located on the sensor interface board, which is used to sample the responses of the sensor array and convert them into digital data. The gas identification system was developed based on the RC203 FPGA platform provided by *Celoxica* [20]. The RC203 is suitable for evaluation and development of high-performance FPGA and soft-core microprocessor-based applications. The FPGA on RC203 belongs to *Xilinx virtex II* family, which is shown in Fig. 7.

There are two banks of SRAM providing a total of 4 MBytes and a *SmartMedia* socket used to configure the FPGA. All the bit files for the FPGA reconfiguration are stored in the *SmartMedia* card. The FPGA is reconfigured through CPLD by downloading bit files from the *SmartMedia* card. The SRAM is used

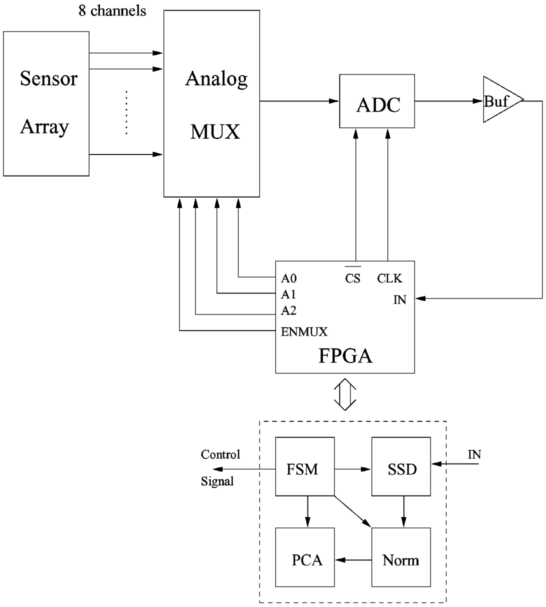


Fig. 8. Sensor interface with the data acquisition and signal preprocessing circuits implemented on FPGA.

to store intermediate results and the LED is used to display the final classification result. The following section will introduce the FPGA implementation of all building blocks including data acquisition, signal preprocessing, pattern recognition, and the final decision stage.

B. Architecture of the CM's Building Blocks

1) Data Acquisition and Signal Preprocessing Stage: Data acquisition and signal preprocessing stages are used to extract the steady-state of the sensors and prepare the gas patterns for pattern recognition stage. Fig. 8 shows the architecture of the data acquisition and signal preprocessing block.

An analog multiplexer with eight input channels, a 12-bit serial ADC and a buffer are used to build an interface between the sensor array chip and the FPGA. As illustrated in Fig. 8, the FPGA is configured to implement the circuits including a steady-state detection (SSD) circuit , normalization circuit, PCA circuit, and the control circuit implemented as a finite-state machine (FSM). The FSM is used to generate control signals for the analog multiplexer, clock signal for the ADC. In addition, the FSM is responsible for controlling all the other three blocks mentioned above.

Fig. 9 shows the timing diagram of the sampling circuit for the sensor array using an analog multiplexer and a 12-bit serial ADC. A 20 MHz clock signal generated by the FPGA chip is used to control the conversion and readout processes of the 12-bit serial ADC. The process of sampling the response of one gas sensor takes 20 clock cycles (Fig. 9). The whole process can be described as follows: i) at the first clock cycle, the *ENMUX* signal is set high and a 3-bit address is generated to determine which gas sensor will be sampled; ii) at the second clock cycle, the ADC is enabled by setting the *CS* signal to low and the *CLK* is used to generate *SCLK* which controls the readout process; iii) during the first three cycles of *SCLK*, 3 leading zero bits are generated, 12 data bits will be generated serially after 3 zero bits; and iv) at the last clock cycle of *SCLK*, the output of the ADC is set to tri-state and a new sampling process will start after three clock cycles. Data can be periodically sampled every 1 s and the sensors in the array are scanned at a rate of 1 MHz. The circuit shown in Fig. 10 is used to detect the steady-state of the sensors. The detection process is explained in Fig. 11. When the response from ith sensor is sampled, a shift register (*SR*) is used to receive the digital word from the serial ADC, then a subtractor will calculate the difference between the sampled data and the previously sampled one from the same sensor stored in *RDi*.

A comparator will compare the difference with a certain threshold and the output of the comparator will control a switch to decide whether the sampled data will be stored in *RDi* or *RSi* registers which are used to store the dynamic response or the steady-state value, respectively. If the steady-state of the ith sensor has been detected, the FSM will receive a low signal from the comparator and will generate a high signal to disable the switch, which means that the steady-state is reached for the ith sensor. Finally, the steady-state values of the sensor array will be ready in *RS1-RS8* for normalization. The normalization circuit consists of a group of adders and a divider to realize the city block normalization expressed by

$$x_{\text{in}} = \frac{x_i}{\sum_{i=1}^{8} x_i}. \tag{6}$$

The architecture of the normalization circuit is shown in Fig. 12. The sum of all the components of the pattern vectors is first calculated by a group of adders. Then, a divider is applied to normalize each component one by one and the results are stored in *RN1-8* registers.

The PCA circuit is used to realize the linear transform of the gas pattern to a lower dimensional space. The transform is actually a vector-matrix multiplication, which can be expressed as

$$\boldsymbol{z} = \boldsymbol{x}\boldsymbol{T} \tag{7}$$

where $\boldsymbol{z}$, $\boldsymbol{x}$, and $\boldsymbol{T}$ are the transformed pattern, the original pattern, and the transform matrix (a constant matrix), respectively. The implementation of the PCA circuit is based on Distributed Arithmetic (DA) [21]. ROM-based DA uses a ROM table to store the precomputed data, which makes it regular and efficiently implemented into an FPGA hardware [22]. The basic operations required are storage of precomputed coefficients in a ROM, addition, subtraction, and shift operations of the input data sequence [21]. All of these functions can be efficiently mapped to FPGA structures. Consider an inner product of two vectors A and B of length K

$$Y = \sum_{k=1}^{K} A_k B_k \tag{8}$$

where B_k is in $2's$ complement binary number scaled such that $|B_k| < 1$, with wordlength N and is defined as B_k: b_{k0}, $b_{k1}, \ldots, b_{k(N-1)}$

$$B_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \tag{9}$$

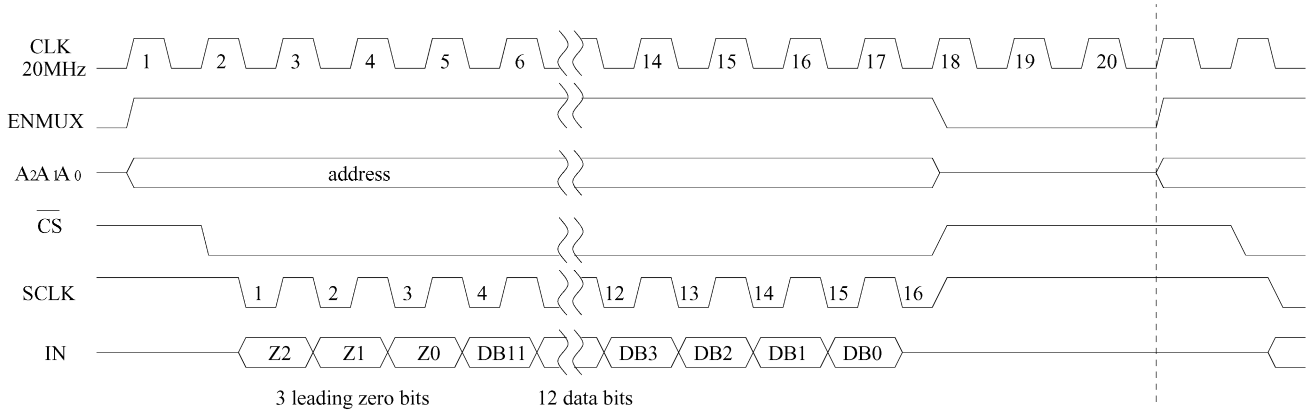


Fig. 9. Timing diagram of the sampling circuit for the sensor array using an analog multiplexer and a 12-bit serial ADC.

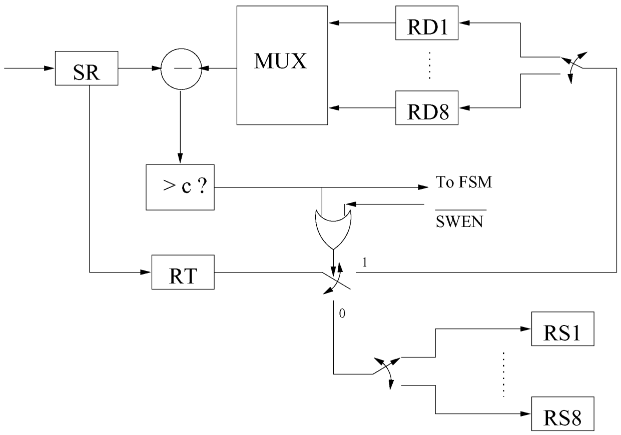


Fig. 10. The SSD circuit used to extract the steady-state of the gas sensor array.

where b_{k0} is the sign bit. Substituting (8) in (9), we obtain

$$Y = \sum_{k=1}^{K} A_k \left[-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right] \tag{10}$$

$$= \sum_{k=1}^{K} A_k \sum_{n=1}^{N-1} b_{kn} 2^{-n} + \sum_{k=1}^{K} A_k(-b_{k0}) \tag{11}$$

$$= \sum_{n=1}^{N-1} \left[\sum_{k=1}^{K} A_k b_{kn} \right] 2^{-n} + \sum_{k=1}^{K} A_k(-b_{k0}). \tag{12}$$

The term $\sum_{k=1}^{K} A_k b_{kn}$ and $\sum_{k=1}^{K} A_k(-b_{k0})$ have only 2^k possible values each, and can thus be stored in a ROM of size 2^{k+1}. However, if an adder/subtractor block is used, instead of just an adder, the second term in (12) can be eliminated, and the size of the ROM is reduced to 2^k. Hence, vector dot product can be efficiently implemented in DA form using ROMs and adders. Fig. 13 shows a simple structure that can be used to calculate (12) with $K = 3$. B_k is delivered in a one-bit-at-a-time fashion with LSBs first to access the ROM. The sign bits are the last bits to be received. *S* is the sign control signal (when the sign bits are received, *S* will be set high). This sign control signal is used to configure the add/subtract unit and toggle the switch shown in Fig. 13. During each clock cycle, components of B_k will shift their LSBs out to access the ROM and the corresponding value stored in the ROM will add the previous partial results (which have been right shifted one bit). After the sign bits access the ROM, the result is output. The content of the ROM is shown on the right side of Fig. 13. The size of the ROM is dependant on the vector's length K.

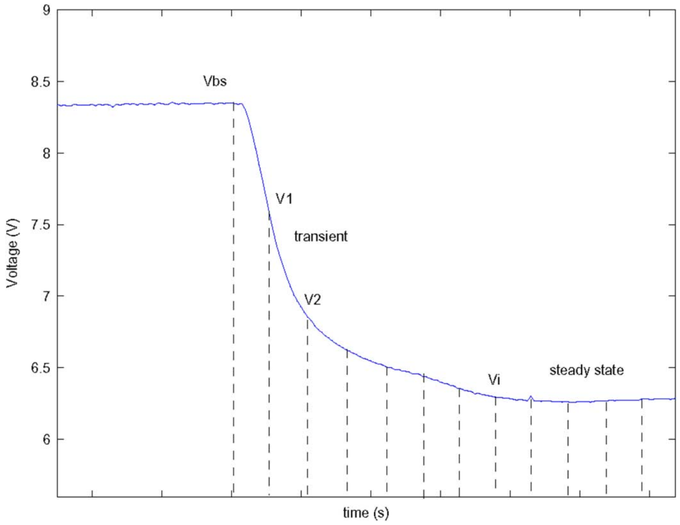


Fig. 11. Sampling process controlled by the FPGA. The difference between two sampled values is compared with a threshold until a steady-state is detected.

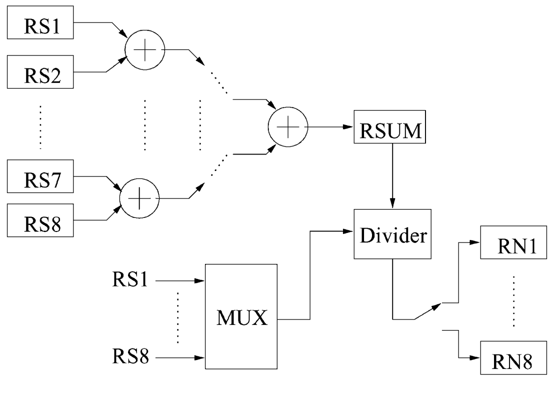
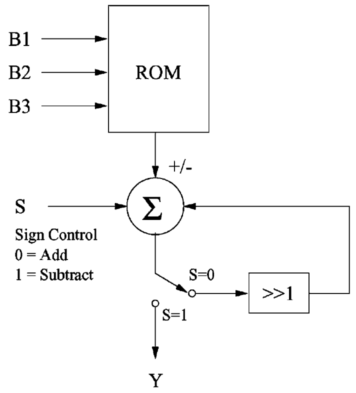


Fig. 12. City block normalization circuit.

In our case, the original gas pattern $\boldsymbol{x}$ has a dimension of eight $(K = 8)$ and $\boldsymbol{z}$ has a dimension of five. Thus, five ROMs with size of $2^8 \times 11$ bits ($\boldsymbol{x}$ is 8 bits) are required for DA-based



Input b_{1n}	b_{2n}	b_{3n}	ROM Content
0	0	0	0
0	0	1	A3
0	1	0	A2
0	1	1	A2+A3
1	0	0	A1
1	0	1	A1+A3
1	1	0	A1+A2
1	1	1	A1+A2+A3

Fig. 13. Simple structure of DA unit.

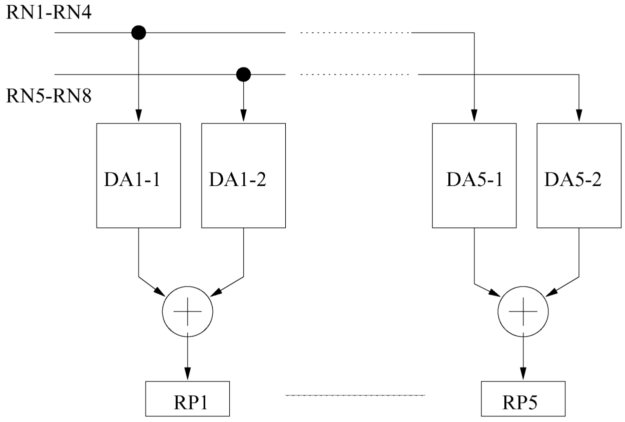


Fig. 14. PCA circuit implemented based on DA algorithm.

implementation. In order to reduce the size of the ROM, we divided $\boldsymbol{x}$ and $\boldsymbol{T}$ into two parts: $\boldsymbol{x}^1$ with the first four components and $\boldsymbol{x}^2$ with the last four components, $\boldsymbol{T}^1$ with the first four rows and $\boldsymbol{T}^2$ with the last four rows. Now, (6) and (7) can be rewritten as

$$z = \boldsymbol{x}^1\boldsymbol{T}^1 + \boldsymbol{x}^2\boldsymbol{T}^2. \tag{13}$$

Thus, ten ROMs with the size of $2^4 \times 10$ bits are required for DA-based implementation. The architecture of PCA circuit is shown in Fig. 14. The calculation of each principal component requires two DA units and one adder. The architecture of the DA unit is shown in Fig. 13. The 4-bit address used to access the ROM is composed of the LSBs of four components of $\boldsymbol{x}^1$ or $\boldsymbol{x}^2$. The DA unit requires eight clock cycles to finalize the calculation. $\boldsymbol{z}$ will be stored in *RP1–RP5*.

All the circuits mentioned above are implemented in the same FPGA chip but SSD, normalization circuit and PCA circuit are enabled sequentially due to the slow response of the sensor.

C. CM Stage

The CM consists of the five individual classifiers, which we introduced in Section III. All of the five individual classifiers are designed and implemented on the same FPGA chip. However, each individual classifier can be considered as an independent system and operates in parallel on the FPGA chip. The implementation of each individual classifier are introduced next.

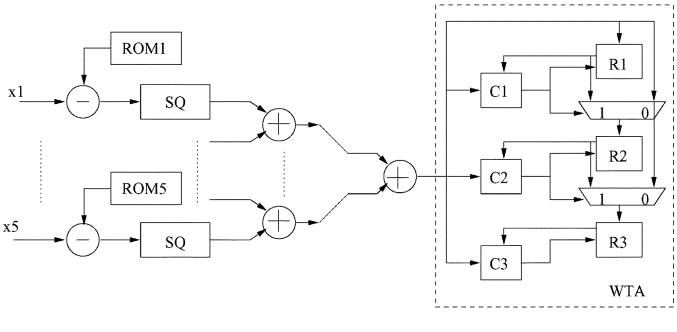


Fig. 15. Architecture of KNN classifier.

1) KNN: KNN classifier calculates the distance between an unknown pattern and all the data patterns and find the K nearest neighbors. Thus, the operation of KNN classifier can be divided into two parts: i) calculating the distance between unknown pattern and each data pattern in the training data-set and ii) finding the K nearest neighbors for the input pattern.

The architecture of KNN classifier is shown in Fig. 15. All the components of the unknown pattern $\boldsymbol{x}$ are fed into the system in parallel. The ith elements of the data pattern are stored in ROMi. There are five subtractors, five SQuare units (SQ), and four adders used to calculate the squared distance between $\boldsymbol{x}$ and one data pattern. A winner-takes-all (WTA) circuit consists of three comparators (*C1-C3*) and three registers (*R1-R3*) which is used to find three nearest neighbors ($K = 3$ in our application). The KNN classifier operates using pipelining strategy and hence the new squared distance can be calculated every clock cycle. The WTA circuit compares the distances one by one and update the three nearest neighbors in *R1–R3* every clock cycle until the last distance is processed. The operation of the circuit can be described as follows: when a new distance is calculated, *C1-C3* will compare this distance with the previous three smallest distances stored in *R1–R3* (the first smallest distance is stored in *R1* and there are 5 bits used to store the label of the corresponding data pattern corresponding to the class label). The output of *C1–C3* will dictate whether the information of *R1–R3* will be updated or not.

The new distance and the label of the corresponding class will be stored in *R1* and the information in *R1* and *R2* are shifted to *R2* and *R3*, respectively. If the new distance is just smaller than the one stored in *R2* and *R3*, the new distance will be stored in *R2* and the information in *R2* is shifted to *R3*. The comparison is repeated until all the distances are compared. Finally, the labels of the three nearest neighbors stored in *R1-R3* will be stored in the SRAM.

2) MLP: In our application, the MLP network consists of two layers. The computations involved in the MLP classifier can be described as two vector-matrix multiplications and the calculation of the activation function described by

$$y_k(\boldsymbol{x}) = \sum_{j=1}^{M} w_{kj}\Phi_j(\boldsymbol{x}) + w_{k0}. \tag{14}$$

The basis functions can be given by $\tan h$ activation functions

$$\Phi_j(\boldsymbol{x}) = \tan h\left(\theta_j^T\boldsymbol{x} + \theta_{j0}\right). \tag{15}$$

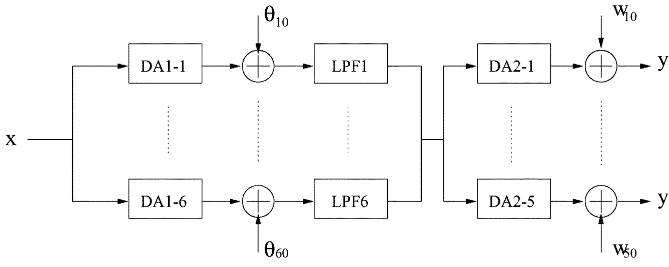


Fig. 16. Architecture of MLP classifier.

The vector-matrix multiplication is implemented using ROM-based DA. The calculation of the activation function are implemented based on linear piecewise function (LPF) approximation which is proven to be hardware friendly [23]–[25]. The architecture of MLP classifier is shown in Fig. 16. The number of hidden nodes of the MLP in our application is six. Thus, six DA units DA1-1 to DA1-6 and 6 LPF units are used to realize (14). The outputs of the 6 LPF units will access the other five DA units DA2-1 to DA2-5 to obtain $\boldsymbol{y}$ in (15). Finally, $\boldsymbol{y}$ will be stored in the SRAM for final decision.

3) RBF: The RBF classifier is similar to MLP and the main computation is expressed by

$$\Phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{\sigma_j^2}\right) \tag{16}$$

$$\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{\Phi}(\boldsymbol{x}) \tag{17}$$

where $\boldsymbol{\Phi}(\boldsymbol{x})$ are M fixed basis functions $\Phi_j(\boldsymbol{x})$ and $\boldsymbol{W}$ is a $c \times M$ matrix of the adjustable network weights [8]. The number of hidden nodes of RBF $(n = 13)$ is much higher than that of MLP in our application. Thus, calculating the activation function of each hidden node in parallel would result in excessive hardware resources. DA-based RBF implementation was not selected because it requires all the vector components to access the ROM in parallel, which is prohibitively expensive. The RBF circuit architecture is shown in Fig. 17. The components of $\boldsymbol{x}$ are fed into the system serially and a subtractor, a square unit, and an accumulator are used to calculate the distance between $\boldsymbol{x}$ and the centers. In (16), σ_j^2 is approximated by using power-of-two coefficients in order to use shift operation instead of a division. The dimension of $\boldsymbol{x}$ is 5, $\Phi_j(\boldsymbol{x})$ is generated by the LPF unit every five clock cycles. During these five clock cycles, $\Phi_j(\boldsymbol{x})$ will multiply w_{1j} to w_{5j} one by one to obtain the partial results of y_1 to y_5. Finally, five accumulators are used to obtain the final $\boldsymbol{y}$ result.

4) GMM and PPCA: In a Gaussian mixture model, a classifier can be constructed by evaluating the posterior probability of an unknown input pattern $\boldsymbol{x}$ belonging to a given class C_k expressed as $\wp(C_k|\boldsymbol{x})$. Based on Bayes' theorem, $\wp(C_k|\boldsymbol{x})$ can be written as

$$\wp(C_k|x) = \frac{\wp(C_k)\wp(\boldsymbol{x}|C_k)}{\wp(\boldsymbol{x})} \tag{18}$$

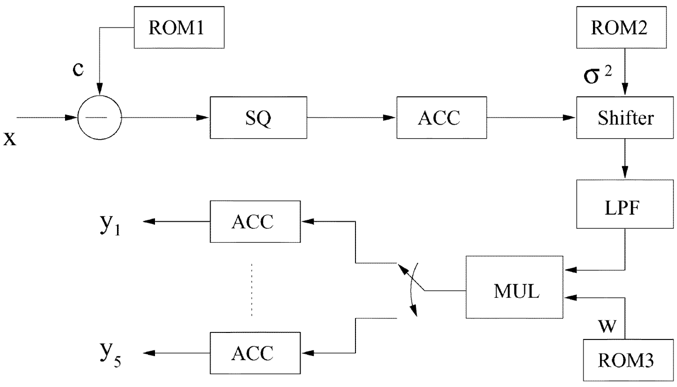


Fig. 17. Architecture of RBF classifier.

where $\wp(C_k)$ and $\wp(\boldsymbol{x})$ are the frequency of a given training sample in the data set and the unconditional density, respectively. In GMM case, the class conditional densities $\wp(\boldsymbol{x}|C_k)$ can be expressed as a linear combination of basis functions $\wp(\boldsymbol{x}|j)$. A model with M components is described as a mixture distribution [9]

$$\wp(\boldsymbol{x}|C_k) = \sum_{j=1}^{M} \wp(j)\wp(\boldsymbol{x}|j) \tag{19}$$

where $\wp(j)$ are the mixing coefficients of the component density functions $\wp(\boldsymbol{x}|j)$. Each mixture component is defined by a Gaussian parametric distribution in d dimensional space

$$\wp(\boldsymbol{x}\,|\,j) = \frac{\exp\left\{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_j\right)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right\}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}}. \tag{20}$$

Since the unconditional density $\wp(\boldsymbol{x})$ is independent of the class, it may be omitted from the Byes' formula as the classification process consists of comparing the posterior probabilities. $\wp(\boldsymbol{x}|C_k)\wp(C_k)$ is, therefore, used to find the decision boundaries in the evaluation period, which can be expressed as

$$\wp(\boldsymbol{x}|C_k)\wp(C_k) = \wp(C_k)\sum_{j=1}^{M} \wp(j)\wp(\boldsymbol{x}|j). \tag{21}$$

When implementing GMM classifier, a new set of parameters (constant K_j and a triangular matrix $\boldsymbol{G}_j$) are defined and used instead of $\wp(C_k)$, $\wp(j)$, $|\boldsymbol{\Sigma}_j|^{1/2}$, and $\boldsymbol{\Sigma}_j^{-1}$($\boldsymbol{\Sigma}_j^{-1}$ is a full matrix) in order to reduce the memory size. The new coefficients K_j and $\boldsymbol{G}_j$ are given by

$$K_j = \frac{\wp(C_k)\wp(j)}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} \tag{22}$$

$$\boldsymbol{G}_j^\top \boldsymbol{G}_j = \frac{1}{2}\boldsymbol{\Sigma}_j^{-1}. \tag{23}$$

$\boldsymbol{G}_j$ is a triangular matrix introduced in order to reduce the complexity as compared with dealing with a full matrix when it comes to (20) calculation [25]. If we assume that

$$z_j = [(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{G}_j][(\boldsymbol{x}-\boldsymbol{\mu}_j)^\top \boldsymbol{G}_j]^\top \tag{24}$$

(21) can be rewritten as

$$\wp(\boldsymbol{x}|C_k)\wp(C_k) = \sum_{j=1}^{M} K_j \exp\{-z_j\}. \tag{25}$$

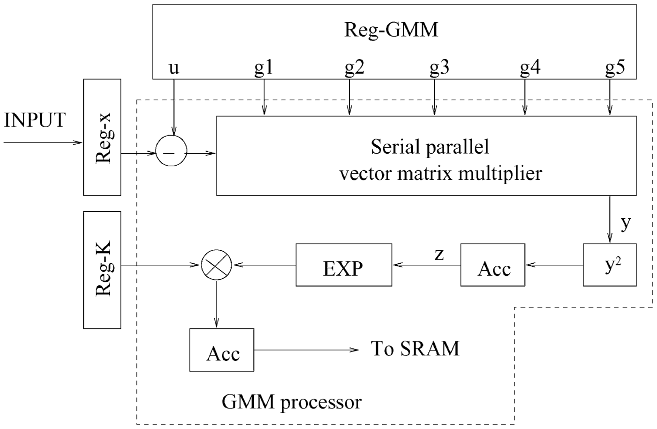


Fig. 18. Functional blocks of the GMM classifier system [25].

Fig. 18 shows the functional block diagram of the overall GMM classifier system. The architecture includes two main registers Reg-X and Reg-GMM used to store the input pattern $\boldsymbol{x}$ and the GMM parameters ($\boldsymbol{\mu}$, $\boldsymbol{G}$, K), respectively.

The GMM processor includes a serial parallel vector matrix multiplier, a square and multiplier units, two accumulators, and a LPF unit, which is used to approximate the exponential function. Initially, the $\boldsymbol{G}_j$ Matrix is multiplied by the $\boldsymbol{s}_j = \boldsymbol{x} - \boldsymbol{\mu}_j$ vector. The resulting vector $\boldsymbol{y}_j$ is then feed to a square unit followed by an accumulator, which performs the summation of all the squared components of the vector resulting in the value z_j, as described in (24). The result is fed to the LPF unit and is multiplied by a constant K_j. The multiplication result represents a single parameter $K_j \exp\{-z_j\}$, which when accumulated M times will lead to the value of $\wp(\boldsymbol{x}|C_k)\wp(C_k)$, as described by (25). An accumulator is, therefore, required at the output of the exponential block which is iterated M times. The values $\wp(\boldsymbol{x}|C_k)\wp(C_k)$ are then stored in the SRAM.

PPCA classifier is similar to GMM in the evaluation period. Thus, the implementation of PPCA is the same with GMM except for the parameters stored in the ROM which are obviously different.

D. Confidence Transform (CT) and Decision Stage

We have seen that the CM will first process the outputs of each individual classifier in order to evaluate the confidences. Next, the weighted confidence are calculated to obtain the scores for each class. The class with the highest score will be assigned to the unknown pattern. The architecture of the CT and the decision unit is shown in Fig. 19(A). Five CT units are used to process the outputs of the individual classifier and to evaluate confidence values. Five multipliers and adders are used to obtain the final score. A digital winner-takes-all (WTA) circuit is used to make the final decision.

We can note from Fig. 19 that the CT unit for different classifiers is different. For KNN, the classifier provides 5 bits labels of 3 nearest neighbors. The 5 bits label will control the multiplexer to decide whether the confidence of each class will be increased or not [Fig. 19(B)]. Since K is equal to 3 for our application, the confidence will be increased by 1/3 for 1 nearest neighbor. If three nearest neighbors belong to the same class, the confidence will be 1. For the other four classifiers, the output $\boldsymbol{y}$ will be normalized into the range of [1 0] to approximate the post probability of each class. Thus, adders and a divider are used to calculate the corresponding confidence. Finally, a digital WTA circuit [25] is used to make the final decision based on a weighted majority vote function.

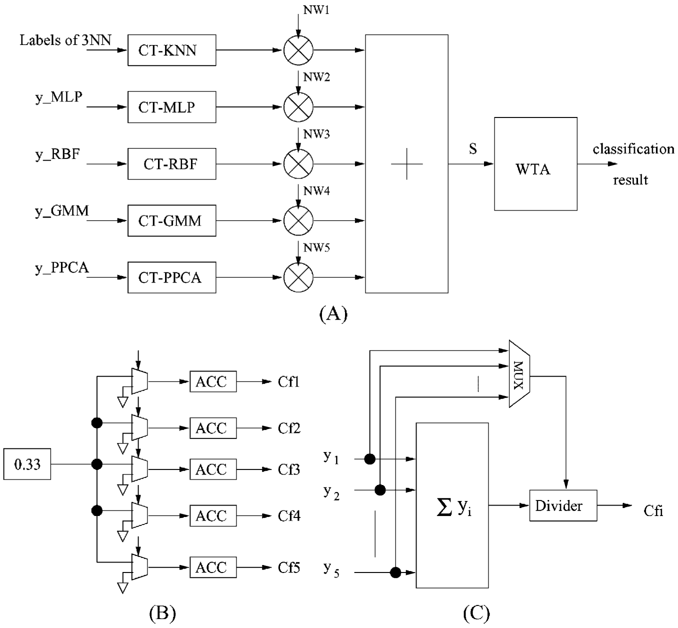


Fig. 19. (a) Architecture of CT and decision unit. (b) CT unit for KNN. (c) CT unit for MLP, RBF, GMM, and PPCA.

TABLE III
IMPLEMENTATION RESULTS ON XILINX VIRTEX II FPGA CHIP

	Data acquisition & signal preprocessing	Committee machine	CT & decision
No. of slices (% of resources)	2723 (19%)	12146 (84%)	3075 (21%)
No. of 4-in LUT (% of resources)	3812 (13%)	20115 (70%)	4236 (15%)
Frequency (MHz)	50		
Reconfiguration time (ms)	26.26		

V. IMPLEMENTATION RESULTS

A. System Implementation Results

Hybrid design entry based on a top level Handel-C wrapper and optimized cores from the Xilinx Coregen library for standard arithmetic operations such as addition/subtraction, multiplication, division and comparison has been adopted. The EDIF netlists are generated using the the *Celoxica DK 3 Design Suite*. ISE 7.1 [26] has been used to synthesize the design and generate the configuration bitstreams.

The gas identification system is divided into three parts, as illustrated in Fig. 5. The three main building blocks of the CM are implemented into *Xilinx virtex II* [26] FPGA chip separately and the corresponding bit files are stored in the *SmartMedia* card. The implementation results are summarized in Table III. A clock frequency of 50 MHz is applied to the system, which is the maximum frequency available on RC203 board. It can be

TABLE IV
OCCUPIED RESOURCES ON XILINX VIRTEX II, SPARTAN 3, AND VIRTEX FPGA CHIP. FOR VIRTEX, STAGE 2.A CONSISTS OF KNN, GMM, AND RBF CLASSIFIER AND STAGE 2.B CONSISTS OF MLP AND PPCA CLASSIFIER

	Stage 1	Stage 2		Stage 3
Virtex II	19%	84%		21%
Spartan 3	13%	59%		15%
	Stage 1	Stage 2.a	Stage 2.b	Stage 3
Virtex	39%	92%	83%	31%

noted from Table III that the dynamic reconfiguration concept permits us to implement computationally intensive processing, which otherwise would not be possible on a single FPGA. The overall system requires 24% extra resources, which is possible to accommodate using dynamic reconfiguration, as illustrated in Table III.

The FPGA chip is automatically reconfigured by downloading the bit file in the *SmartMedia* card through CPLD. The reconfiguration process can be described in the following basic operation. i) Once the FPGA completes the current task, it will communicate the address where the bit file is located in the *SmartMedia* card to the CPLD. ii) The CPLD sets up the FPGA for the next reconfiguration and checks if the *SmartMedia* card is inserted properly. iii) The bit file is copied from the *SmartMedia* card through the CPLD. The reconfiguration time is proportional to the size of the bit file, which depends on the type of FPGA chip. The size of the *Xilinx virtex II* bit file is 1.25 MByte. The reconfiguration time was measured for our application to be around 26 ms. It should be noted that the system is easily portable to other FPGA platforms and chips. For the sake of comparison, we implemented our system using different FPGA chips and the required resources are reported in Table IV. A less powerful processor will require partitioning the processing to a large number of stages. For example, in the case of *Virtex*, the CM stage cannot be implemented on the FPGA chip in a single iteration. The CM stage is, therefore, further divided into two stages: stage (a) which consists of KNN, GMM, and RBF and stage (b) which consist of MLP and PPCA (refer to Table IV). It should be noted, however, that a more powerful FPGA such as *Spartan 3* can handle all required processing stages in one iteration. It is, however, important to note that the limiting factor in the design is the speed of the sensor. Hence, time multiplexing by reconfiguration does not decrease the overall performance. Conversely, apart from a simpler design, newer and larger FPGAs, which are significantly more expensive, do not offer any other performance advantages that can be effectively leveraged, in our application. In addition, there is a significant surge of algorithmic solutions using boosting strategies, whereas a cascade or a parallel configuration of "basic classifiers" are used in order to improve the overall detection performance [27], [28]. Dynamically reconfigurable hardware can be very effectively used for such applications as the number of basic classifiers can be prohibitively large and may not fit on a single chip FPGA.

B. Gas Recognition Results

An automated experimental setup was built in order to perform gas sensing characterization. The setup can be used to measure gas-sensing characteristics in well-defined temperature cycles and gas concentration levels. The system includes a gas chamber, the gas delivery system, as well as the data acquisition system. The sensor chip is placed inside the chamber with feed-through wires used for resistance measurement and temperature control. The data from the sensor are sampled and processed using the FPGA platform. The gas concentrations in the sensor chamber are adjusted by selecting the correct flow rate for different gases. The temperature of the sensors is constantly monitored by periodically reading data from the integrated temperature sensor. The microhotplates of each chip are heated to a particular temperature by flowing the precalibrated current through the heating element. After the training procedure, the parameters for each classifier are quantized and loaded into the FPGA board. The test performance was experimentally measured using the FPGA platform and the performance are compared with the performance obtained by simulation. 94% accuracy was obtained for the CM using 5 PCs. This represents a 2% drop in performance as compared with the simulation results reported in Table II. This performance degradation is mainly due to the quantization error and linear piecewise approximation used to implement some of the complex nonlinear functions needed for some classifiers such as MLP, RBF, and GMM. It should be noted, however, that the performance of individual classifiers generally suffer more than a 4% drop in performance suggesting that CMs are robust against quantization error.

VI. CONCLUSION

In this paper, the implementation of gas identification system based on dynamically reconfigurable FPGA was introduced. Due to slow response of the gas sensor, the time constraint is relaxed allowing to partition the computation requirement of our system into different stages and FPGA implementation is carried out sequentially. The proposed gas identification system consists of three stages: data acquisition and signal preprocessing, CM, CT, and decision. Data acquisition and the signal preprocessing unit includes a SSD circuit, normalization circuit, and PCA circuit to extract the steady-state of the sensors and generate the normalized gas pattern with reduced dimensionality. For the implementation of the CM which consists of five classifiers (KNN, MLP, RBF, GMM, and PPCA), each individual classifier is implemented as an independent system but all classifiers operate in parallel within the FPGA. Novel ROM-based DA approach was applied to implement the vector-matrix multiplication required in most preprocessing and pattern recognition algorithms. The results from the CM are further processed in order to evaluate the confidence using 5 CT units operating in parallel. Finally, a digital WTA circuit [25] is used to make the final decision based on a weighted majority vote function. The implementation of different stages of the gas identification system shared the same FPGA chip in a dynamic way. The system can operate at a frequency of 50 MHz. The reconfiguration time of the FPGA chip is around 26 ms, which can be neglected compared with the speed of the gas sensors (in the range of a few seconds). It was shown that the dynamic reconfiguration concept enables the implementation of computationally intensive processing, which would not be possible on some FPGA platforms. Dynamically reconfigurable FPGA offers a very viable implementation mean which can be used for implementing new generation of boosting and cascading-based pattern recognition algorithms. Indeed, these algorithms feature a prohibitively large number of basic classifiers which may not fit on a single FPGA chip.

REFERENCES

[1] R. Gutierrez-Osuna, "Pattern analysis for machine olfaction: A review," *IEEE Sensors J.*, vol. 2, no. 3, pp. 189–202, 2002.

[2] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, 2006.

[3] T. G. Dietterich, "Machine learning research: Four current directions," *AI Magazine*, vol. 18, no. 4, pp. 97–136, 1997.

[4] F. M. Ali and A. S. Das, "Hardware-software co-synthesis of hard real-time systems with reconfigurable FPGAs," *Comput. Elect. Eng.*, vol. 30, no. 7, pp. 471–489, 2004.

[5] R. Tessier, S. Swaminathan, R. Ramaswamy, D. Goeckel, and W. Burleson, "A reconfigurable, power-efficient adaptive Viterbi decoder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 484–488, 2005.

[6] D. Kim, "An implementation of fuzzy logic controller on the reconfigurable FPGA system," *IEEE Trans. Ind. Electron.*, vol. 47, no. 3, pp. 703–715, 2000.

[7] M. Pardo and G. Sberveglieri, "Learning from data: A tutorial with emphasis on modern pattern recognition methods," *IEEE Sensors J.*, vol. 2, no. 3, pp. 189–202, 2002.

[8] C. M. Bishop*, Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon Press, 1995.

[9] D. M. Titterington, A. F. M. Smith, and U. E. Makov*, Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.

[10] T. K. Ho, J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, 1994.

[11] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 3, pp. 226–239, 1998.

[12] P. C. H. Chan, G. Yan, L. Sheng, R. K. Sharma, Z. Tang, J. K. O. Sin, I. Hsing, and Y. Wang, "An integrated gas sensor technology using surface micro-machining," *Sens. Actuators B*, vol. 82, no. 3, pp. 277–283, 2002.

[13] S. Brahim-Belhouari, A. Bermak, M. Shi, and P. C. H. Chan, "Fast and robust gas identification system using an integrated gas sensor technology and Gaussian mixture models," *IEEE Sensors J.*, vol. 5, no. 6, pp. 1433–1444, Dec. 2005.

[14] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc, B 61*, pp. 611–622, 1999.

[15] M. Shi, B. Guo, and A. Bermak, "Redundancy analysis for tin oxide gas sensor array," in *Proc. 3rd IEEE Int. Workshop on Electronic Design, Test, Appl.*, Kuala Lumpur, Malaysia, Jan. 2006, pp. 17–19.

[16] C. L. Liu, "Classifier combination based on confidence transformation," *Pattern Recogn.*, vol. 38, pp. 11–28, 2005.

[17] S. Minghua, A. Bermak, S. B-Belhouari, and P. Chan, "Gas identification based on a committee machine for microelectronics gas sensors," *IEEE Trans. Instrum. Measurements*, vol. 55, no. 5, pp. 1786–1793, Oct. 2006.

[18] P. Althainz, A. Dahlke, M. Frietsch-Klarhof, J. Goschnick, and H. J. Ache, "Reception tuning of gas-sensor microsystems by selective coatings," *Sens. Actuators B*, pp. 366–369, 1995.

[19] G. G. Mandayo, E. Castano, and F. J. Gracia, "Carbon monoxide detector fabricated on the basis of a tin oxide novel doping method," *IEEE Sensors J.*, vol. 4, pp. 32–328, 2002.

[20] [Online]. Available: http://www.celoxica.com

[21] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Mag.*, pp. 4–19, 1989.

[22] S. Chandrasekaran and A. Amira, "An area efficient low power inner product computation for discrete orthogonal transforms," in *Proc. IEEE Int. Conf. Image Process., (ICIP 2005)*, Genoa, Italy, Sept. 2005, vol. 3, pp. 1024–1027.

[23] A. Bermak and A. Bouzerdoum, "VLSI implementation of a neural network classifier based on the saturating linear activation function," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP'02)*, 2002, vol. 2, pp. 981–985.

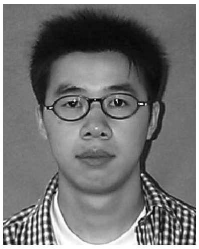
[24] Z. Ruan, J. Han, and Y. Han, "Bp neural network implementation on real-time reconfigurable FPGA system for a soft-sensing process," in *Proc. Int. Conf. Neural Netw. Brain*, 2005, vol. 2, pp. 959–963.

[25] M. Shi and A. Bermak, "An efficient digital VLSI implementation of gaussian mixture models-based classifier," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 9, Sep. 2006.

[26] [Online]. Available: http://xilinx.com/

[27] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[28] A. Bermak and D. Martinez, "A compact 3-D VLSI classifier using bagging threshold network ensembles," *IEEE Trans. Neural Networks*, vol. 14, no. 5, pp. 1097–1109, Sep. 2003.



Minghua Shi received the B.S. degree in electronic engineering from the East China University of Science and Technology, Shanghai, China, in 2002 and the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, in 2006.

He is currently a Senior Engineer in FPGA hardware multimedia technology at the Hong Kong Applied Science and Technology Research Institute Company Limited. His research interests are related to hardware implementation of pattern recognition algorithms, as well as multimedia and consumer electronic applications.

Amine Bermak (M'99–SM'04) received the M.Eng. and Ph.D. degrees in electronic engineering from Paul Sabatier University, Toulouse, France, in 1994 and 1998, respectively.

While working towards his Ph.D., he was part of the Microsystems and Microstructures Research Group at the French National Research Center LAAS-CNRS, where he developed a 3-D VLSI chip for artificial neural network classification and detection applications. He then joined the Advanced Computer Architecture Research Group, York University, York, U.K., where he was working as a Postdoc on VLSI implementation of CMM neural network for vision applications in a project funded by British Aerospace. In 1998, he joined Edith Cowan University, Perth, Australia, first as a Research Fellow working on smart vision sensors, then as a Lecturer and a Senior Lecturer in the School of Engineering and Mathematics. He is currently an Associate Professor with the Electronic and Computer Engineering Department, Hong Kong University of Science and Technology (HKUST), where he is also serving as the Associate Director of Computer Engineering Program. His research interests are related to VLSI circuits and systems for signal, image processing, sensors, and microsystems applications. He has published extensively on the above topics in various journals, book chapters, and refereed international conferences.

Dr. Bermak has received many distinguished awards including the 2004 IEEE Chester Sall Award, the HKUST Bechtel Foundation Engineering Teaching Excellence Award in 2004, and the Best Paper Award at the 2005 International Workshop on System-On-Chip for Real-Time Applications. He is a member of technical program committees of a number of international conferences including the IEEE Custom Integrated Circuit Conference CICC2006, CICC2007, the IEEE Consumer Electronics Conference CEC2007, and Design Automation and Test in Europe DATE2007. He is the General Co-Chair of the 2008 IEEE International Symposium on Electronic Design Test and Applications. He is also on the Editorial Board of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS and the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS. He is a member of the IEEE CAS Committee on Sensory Systems.

Shrutisagar Chandrasekaran (S'04) received the B.S. degree in electronics and communications (with Distinction) from Madurai Kamaraj University, India, in 2004, and the Ph.D. degree in the parallel and reconfigurable computing from the Computer Vision Research Group, Division of Electronics and Computer Engineering, School of Engineering and Design, Brunel University, West London, U.K. Previously, he was working towards the Ph.D. at the School of Computer Science, Institute of Electronics, Communications and Information Technologies (ECIT), Queen's University, Belfast (QUB), Ireland.

He is currently working in the financial sector in the U.K. as a Quantitative Analyst. His research interests include custom computing using FPGAs, hardware–software codesign, power and energy aware design techniques, modeling of power and performance for FPGA-based designs.

Dr. Chandrasekaran is a recipient of the IEEE R10 RAB Larry K. Wilson Award.

Abbes Amira (M'99–SM'06) received the Ph.D. degree in computer science from Queen's University, Belfast (QUB), Ireland, in 2001.

He is a Senior Lecturer at Brunel University, West London, U.K., within the Division of Electronic and Computer Engineering, School of Engineering and Design. Before he joined Brunel University in May 2006, he has held a Lectureship in Computer Science at Queen's University (QUB) since November 2001. He has been awarded a number of grants from government and industry, has published over 100 publications, and supervised five Ph.D. students during his career to date. He has been invited to give talks at universities in the U.K., Europe, the U.S., and North Africa, at international conferences, workshops, and exhibitions. He has held positions as chair, and program committee member for a number of well-known conferences. His research interests include reconfigurable computing, image and vision systems, system on chip, custom computing using FPGAs, medical image analysis, multiresolution analysis, biometrics technologies, and information retrieval.

Dr. Amira is a member of ACM, IET, and a Fellow of the Higher Education Academy.

Sofiane Brahim-Belhouari received the Engineer Diploma in electrical engineering from the Polytechnic Institute of Algiers, Algiers, Algeria, in 1993 and the Ph.D. degree in automatic control and signal processing from the University of Paris XI, France, in 2000.

Since 2000, he held Postdoctoral positions at Ecole Polytechnique Federale de Lausanne (EPFL) and Hong Kong University of Science and Technology (HKUST). His main research interests are in data analysis, statistical signal processing, and pattern recognition.