ARC '16

مؤتمر مؤسسة قطر
السنوي للبحوث
QATAR FOUNDATION
ANNUAL RESEARCH
CONFERENCE

Towards World-class
Research and Innovation

Information Communications Technology Pillar

Fast Prototyping of KNN Based Gas Discrimination System on the Zynq SoC

Amine Ait Si Ali[1], Muhammad Ali Akbar[1], Abbes Amira[1], Faycal Bensaali[1], Mohieddine Benammar[1],

Muhammad Hassan[2], Amine Bermak[2]

[1]Qatar University, QA
[2]Hong Kong University of Science and Technology, HK

Email: amine.aitsiali@qu.edu.qa

Electronic noses (EN) or machine olfaction are systems used for the detection and identification of odorous compounds and gas mixtures. The accuracy of such systems is as important as the processing time. Therefore, the choice of the algorithm and the implementation platform are both crucial. In this abstract, a design and implementation of a gas identification system on the Zynq platform which shows promising results is presented. The Zynq-7000 based platforms are increasingly being used in different applications including image and signal processing. The Zynq system on chip (SoC) architecture combines a processing system based on a dual core ARM Cortex processor with a programmable logic (PL) based on a Xilinx 7 series field programmable gate arrays (FPGAs). Using the Zynq platform, real-time hardware acceleration of classification algorithms can be performed on the PL and controlled by a software running on the ARM-based processing system (PS). The gas identification system is based on a 16-Array SnO2 in-house fabricated gas sensor and k-Nearest Neighbors (KNN) for classification. The KNN algorithm is executed on the PL for hardware acceleration. The implementation takes the form of an IP developed in C and synthesized using Vivado High Level Synthesis (HLS), the synthesis includes the conversion from C to register transfer level (RTL). The implementation requires the creation of a hardware design for the entire system that allows the execution of the IP on the PL and the remaining parts of the identification system on the PS. The hardware design is developed in Vivado using IP Integrator. The communication between the PS and PL is performed using advanced extensible interface protocol (AXI). A software application is written and executed on the ARM processor to control the hardware acceleration on the PL of the previously designed IP core and the board is programmed using Software Development Kit (SDK). An overview of the system architecture can be seen in Figure 1. The system is designed to discriminate five types of gases including C_6H_6, CH_2O, CO, NO_2 and SO_2 at various concentrations, from 0.25 to 5 parts per million (ppm) for C_6H_6 and CH_2O, from 5 to 200 ppm for CO, from 1 to 10 ppm for NO_2 and finally from 1 to 25 ppm for SO_2. The experimental setup used in the laboratory to collect the data is shown in Figure 2. It consists of a

gas chamber where the sensor array is placed. The gas chamber has two orifices, one to serve as an input for the in-flow of gases and the other one as an exhaust to evacuate the gases. Multiple gases are stored in various cylinders and connected to the gas chamber individually through several Mass Flow Controllers (MFCs). A control unit is connected to the MFCs to control the in-flow of gases and to the sensor array via a Data Acquisition (DAQ) system to collect and sample the response of the sensor array. In total, 192 samples are collected, 50% is used for training and the other 50% is used for testing. Simulations were performed in MATLAB environment prior to the implementation on the hardware where different k values have been used. The Euclidean distance has been used as a metric for the computation of distances between various points. The best results were obtained for k=1 and k=2 with a classification accuracy of 97.91% and 98.95% respectively. The system implemented on hardware is based on k=1 since the accuracies are almost similar while the hardware resources required for k=2 are much higher than for k=1. This can be explained by the fact that in the case of k=2 we need to sort the vector of distances to be able to find the nearest two neighbours while in k=1 we only need to find the smallest distance. The target hardware implementation platform of the proposed KNN is the heterogeneous Zynq SoC. The implementation is based on the use of Vivado HLS. A summary of the design flow is presented in Figure 3. The starting point is Vivado HLS where the KNN block is converted from C/C++ implementation to a RTL based IP core. This allows a considerable gain in development time without scarifying on high parallelism characteristics because Vivado HLS provides a large number of powerful optimization directives. The generated IP-core is then exported and stored in the Xilinx IP Catalog before being used in Vivado IP Integrator to create the hardware block design with all needed components and interconnections. The next step is to export the generated hardware along with IP drivers to the SDK tool. The SDK tool is used to program the Xilinx ZC702 prototyping board via joint test action group (JTAG) interface and the terminal in SDK is used to communicate with the board via universal asynchronous receiver/transmitter (UART) interface. The KNN IP is implemented on the PL of the Zynq SoC and communicates with the PS part via the Xilinx AXI-Interconnect IP. A software is written in C/C++ and executed on the PS to manage the IP present in the PL in terms of sending the input data, waiting for the interrupt and then reading the output data. The block design and the resulting chip layout are shown in Figure 4. It is worth mentioning that the running frequency for the ARM processor is set to the maximum 667MHz while the PL frequency is set to 100 MHz which is the maximum for the KNN IP generated in HLS. The real execution of KNN on the PL side of the ZC702 board shows that one sample can be processed for gas identification in 0.0078 ms while the same sample requires 0.9228 ms if executes on the PS side in the ARM processor in a pure software manner. This means that a speed up of 118 times has been achieved. The main directive in Vivado HLS that helped to reach these performances is the "Loop pipelining" which allows the operations in a loop to be implemented in a concurrent manner. The hardware resources usage can be seen in Figure 5, it shows that 24% of lookup tables (LUT), 12% of flip-flops (FF), 6% of BRAM and 58% DSP have been used. As shown in Figure 6, the total power consumption is 1.895 W, 1.565 W is consumed by the PS and the remaining 0.33W is consumed by the PL.