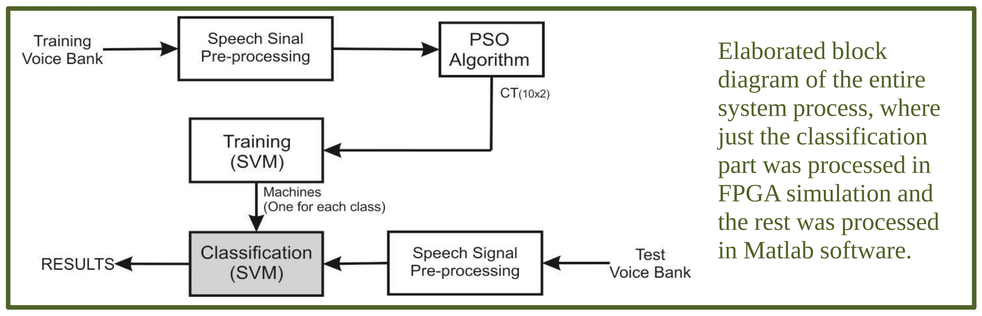
New Approach of Pipelined Architecture of SVM Classifier for ASR System in FPGA Implementation

Gracieth C. Batista, Duarte L. Oliveira, Osamu Saotome, Leonardo Romano

Technological Institute of Aeronautics – IEEA – ITA (SP – Brazil) / **E-mails:** gracieth@ita.br, duarte@ita.br, osaotome@ita.br, leoroma@uol.com.br

ABSTRACT: In this paper, it is proposed an pipelined architecture for classification of SVM (Support Vector Machine) algorithm for an Automatic Speech Recognition (ASR) application that it has the following features: multi-class (30 classes), application of RBF kernel function, 3 stages in pipeline, application of two different gated-clock control circuits to synchronize the pipelining. About these control circuits, one is based on an Extended Burst Mode (XBM) specification and the other one is based on transparent latches, i. e., the first one is an asynchronous circuit and the second one is synchronous. In addition, the training data is optimized by Particle Swarm Optimization (PSO) algorithm that it is applied before SVM training, i.e., the training data is obtained from a hybrid training (PSO-SVM training) in entrance. Then, the tests are from 60 speeches and 20 speakers, so it is a diversified dataset of test. The training/learning part was implemented in software (Matlab) and the testing part (classification) was implemented in VHDL (VHSIC Hardware Description Language) applied in FPGA (Field Programmable Gate Array). The main goal here was to obtain the fastest response time which was about 50 – 500 GOPS of throughput; the accuracy in recognition success rate was other preoccupation and it was very successful, 99% of success; the operating frequency was 100 – 500MHz.

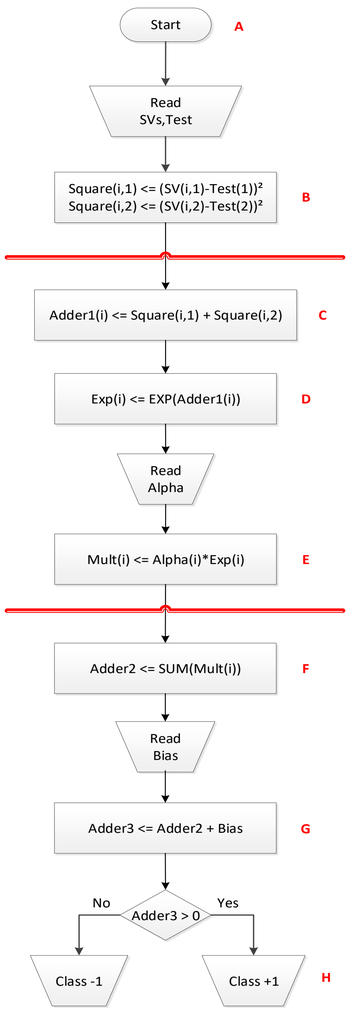
INTRODUCTION



Elaborated block diagram of the entire system process, where just the classification part was processed in FPGA simulation and the rest was processed in Matlab software.

PSO-SVM HYBRID TRAINING IN SOFTWARE

The hybrid training (PSO-SVM algorithm) is initialized from extracted parameters from the pre-processed speech signal. This pre-processing was made from data acquisition of the distinct voice banks, as it follows of spectral analysis of speech signals, then a Hamming windowing application and Mel-Frequency Cepstral Coefficients (MFCCs) are obtained from a series of calculations; finally, in order to reduce those coefficients, the Discrete Cosine Transform (DCT) is used. Parameters from this pre-processing are re-allocated in matrices of 2x2. Those 2x2 matrices are entered in PSO algorithm at the same time inside of the CTNx2 matrix as it follows in $g(\alpha,\beta) = (\alpha - a)^2 + (\beta - b)^2$, where α and β are pairs from the CT matrix that it is transformed in a only one pair at the end of PSO process, *a* and *b* are points that are localized in the center of each class, *a* related to the *x* axis and *b* related to *y* axis [6]. Once the optimized CT matrix is obtained, then the SVM multi-class training (one vs. all) [1] is realized from Radial Basis Function (RBF) kernel [1] with 0.09 sigma constructing an optimum hyperplane that obeys to $w^T x + b = 0$, where *w* is a matrix of adjustable weight values, *b* is a matrix of bias values and *x* is a matrix of entrance data values of SVM algorithm.



Flowchart of the SVM classifier divided in 3 parts (A and B / C, D and E / F, G and H) according to its pipelined architecture of 3 stages.

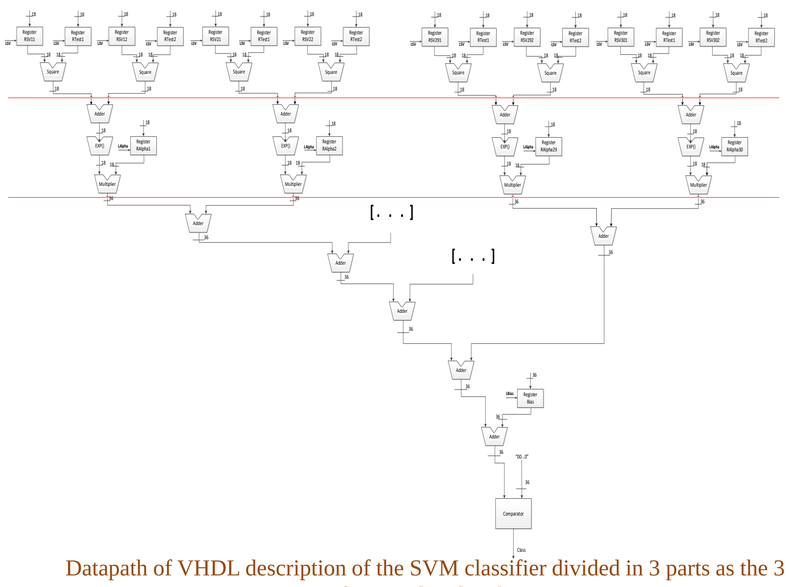
SVM CLASSIFIER IN HARDWARE

The SVs and Test matrices are the data entrances showed on the top and the Alpha matrix entrance (Lagrange Multipliers coefficients) is showed on the bottom. All these matrices came from the training processed in Matlab software.

The *ieee_proposed.fixed_pkg.all* package was used for converting the floating point to 18 bit fixed point format [3]. The *EXP()* component is a look up table described by Equation 3.

$$\text{EXP}\left(^{-(SV-Test)^2}/_{2*\sigma^2}\right) \qquad (3)$$

where $\sigma = 0.9$, *SV* are the Support Vectors values and *Test* are values from the speech signals parameters of voice bank test.



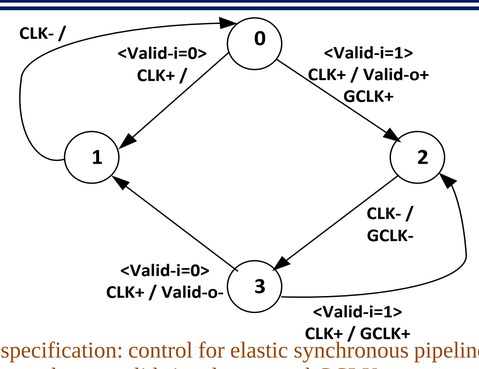
Datapath of VHDL description of the SVM classifier divided in 3 parts as the 3 stages of its pipelined architecture.

In this paper, it is proposed a new basic gated-clock architecture to implement synchronous pipeline systems that is insensitive to global clock. One of its proposed architecture has as main characteristic when compared with other pipeline architectures, the use of a simple asynchronous control to perform the operations (XBM control circuit) different from basic pipelined architectures. Basic pipelined architectures only activate the registers when there is valid data to store in the different stages, it indicates when there is valid data in the output and the input accepts data that arrives at a frequency unrelated to the frequency of operation of the pipeline and it may require several clock cycles or even a fraction of clock cycles.

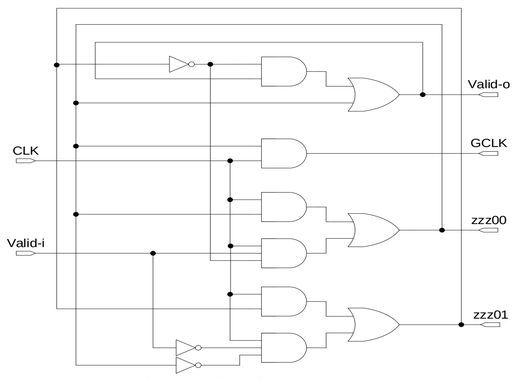
RESULTS AND ANALYSIS

The tests were made from a dataset size of 60 speeches from 20 speakers (male and female). The proposed architecture was targeted in Altera's Cyclone II EP2C70F896C6 with operating frequency of 100 –500MHz, but it can be changed to any other targeted device. Then, it is made comparisons of four versions of the SVM classifier, as it follows:

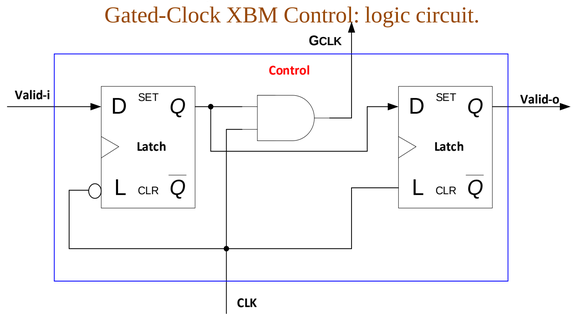
1. Batista and Silva's implementation processed in Matlab software [1];
2. Song et al.'s implementation in Combinational Circuit (CC) processed in FPGA simulation [2];
3. Proposed pipelined architecture with control based on XBM specification processed in FPGA simulation;
4. Proposed pipelined architecture with control based on transparent latches processed in FPGA simulation.



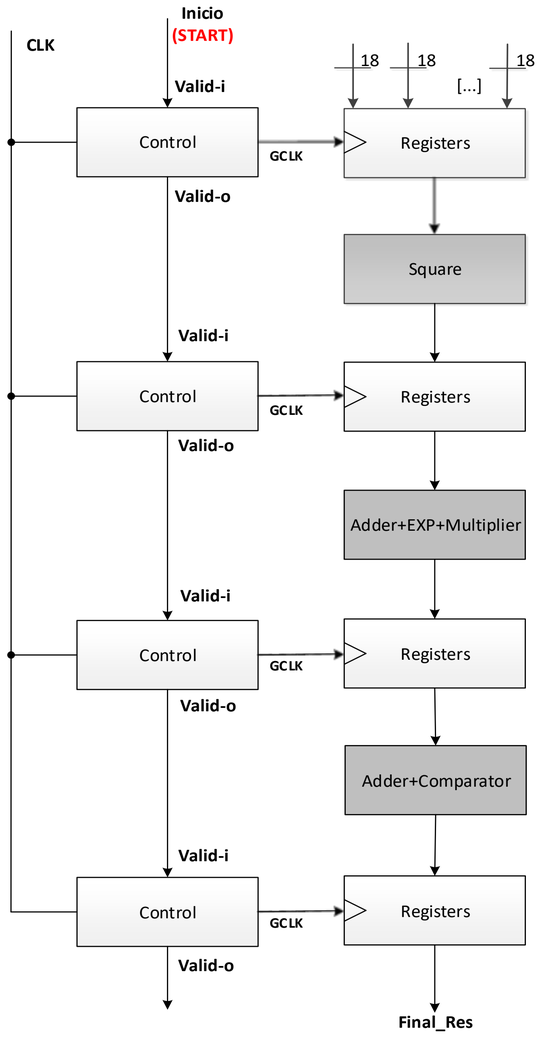
XBM specification: control for elastic synchronous pipeline with only one valid signal to control GCLK output.



Gated-Clock XBM Control: logic circuit.



Gated-Clock Control: logic circuit.



Proposed basic synchronous linear pipeline: it only storages data when there is a valid signal in the first control circuit independently from clock cycles.

TABLE I. COMPARISON BETWEEN FPGA AND PC

FEATURES	FPGA (HARDWARE)	PC (SOFTWARE)
FREQUENCY (HZ)	150,000,000 – 250,000,000	15 – 170
RECOGNITION SUCCESS RATE (%)	98,5 – 100	89 – 92

TABLE III. RESULTS OF THE PROPOSED PIPELINED ARCHITECTURES

FEATURES	XBM - CONTROL CIRCUIT	LATCH - CONTROL CIRCUIT
THROUGHPUT (GOPS)	50	500
LUTS	6,198	6,198
RECOGNITION SUCCESS RATE (%)	98.5 – 100	98.5 – 100
DYNAMIC POWER DISSIPATION (MILIWATTS)	174.15	224.92
MAXIMUM OPERATION FREQUENCY (MHZ)	500	250

TABLE II. DIFFERENCE BETWEEN THE COMPARED ARCHITECTURES

FEATURES	SONG'S	PROPOSAL IN CC CONFIGURATION
SIGMA RBF KERNEL FUNCTION	0.6	0.9
CLASSES (PATTERNS)	10	30
MODELS (SPEAKERS AMOUNT)	4	20
TESTING SIZE	40	60
REGISTERS	0	1
CLOCK	THERE ISN'T IT	YES
MAXIMUM OPERATION FREQUENCY (MHZ)	250	240
RECOGNITION SUCCESS RATE (%)	97 – 100	98.5 – 100

TABLE IV. RESULTS OF THROUGHPUT BETWEEN IMPLEMENTATIONS OF SVM CLASSIFERS

	THROUGHPUT
1) MATLAB SOFTWARE	170 OPS
2) SONG ET AL' IMPLEMENTATION	150 MOPPS
3) PIPELINED ARCHITECTURE IMPLEMENTATION (ASSYNCHRONOUS CONTROL CIRCUIT - XBM)	50 GOPS
4) PIPELINED ARCHITECTURE IMPLEMENTATION (SYNCHRONOUS CONTROL CIRCUIT - LATCHES)	500 GOPS

CONCLUSION

It is possible to observe that the proposed architecture is better, because of the following items:

- Sigma RBF kernel function is higher which provides even better classification results, in reason of the exponential calculation which has the same 14 bits of precision in its decimal part;
- It gives a system recognition more applicable because it has more patterns to be recognized, i.e., more understandable voice commands;
- It is an automatic system because training dataset is applied in the classification/test architecture (in FPGA) and not manually as it is in Song's architecture through a testbench VHDL file, for example.

Besides, it is observed in pipelined architecture from comparison between both control circuits that asynchronous circuit (XBM specification) presented worse throughput however it showed better performance in power consumption, latency and operating frequency rate.

[1] G. C. Batista and W. L. S. Silva, "Using support vector machines and two dimensional discrete cosine transform in speech automatic recognition", *Proceedings of the IEEE IJCNN*, July 2015.

[2] X. Song, H. Wang and L. Wang, "FPGA Implementation of a Support Vector Machine based Classification System and its Potential Application in Smart Grid", *Proceedings of the IEEE 11th International Conference on Information Technology: New Generations*, pp.397-402, June 2014.

[3] D. Bishop, "Fixed and floating point packages for VHDL 2005", Eastman Kodak Company, Rochester, NY, 2005.