Product Specification

so_ip_idte

Decision Tree Ensemble Inference Core

General Description

Machine learning is a scientific discipline that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables.

A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases.

There are many different predictive models (classifiers) in machine learning, including artificial neural networks (ANNs), decision trees (DTs) and recently introduced support vector machines (SVMs).

Recently a new way of making more accurate predictive models has emerged, ensemble learning.

Ensemble learning requires creation of a set of individually trained classifiers, typically decision trees (DTs) or neural networks, whose predictions have to be combined during the process of classification of previously unseen instances. Although simple, this idea has proved to be effective, producing systems that are more accurate then single a classifier.

All ensemble systems consist of two key components. First component is used to calculate the classifications of the current instance for every ensemble member. A second module is then needed to combine the classifications of individual classifiers that make up the ensemble into one single classification, in such a way that the correct decisions are amplified, and incorrect ones are cancelled out.

The main advantage of ensemble classifiers over single classifiers is in higher accuracy and greater robustness. The price to be paid is large amounts of memory to store the ensemble classifier and high computing power. This is because ensemble classifiers typically combine 30 or

more individual classifiers, which means that 30+ times more memory and computational power is required if we want to get the same performance in classification speed as with the single classifier.

Ensemble classifiers are typically implemented in software. But in applications that require rapid classification or ensemble creation, hardware implementation is the only solution.

So_ip_idte core can be used create an ensemble of decision trees directly in hardware. DTs that make up the ensemble can have univarite, multivariate and non-linear tests. Creating DT ensemble directly in hardware results in the significant increase of the inference speed, compared with the traditional software-based approach.

So_ip_idte core uses *bagging* algorithm to create a DT ensemble. Bagging is one of the earliest proposed ensemble-creation algorithms. It is also one of the most intuitive and simplest to implement, with surprisingly good performance. Bagging is particularly appealing when the available data is of limited size. Bagging algorithm can be easily parallelized in contrast to most other popular methods for the ensemble classifier creation. At the hart of the bagging algorithm a proprietary DT inference algorithm based on the evolutionary algorithms, developed at So-Logic, is used to infer individual DT members in parallel. This approach results in very fast DT ensemble inference times while still having acceptable resource requirements.

After the inference process is complete, complete structural information about the created DTs is transferred through the output ports. This information can be easily transferred to some of the So-Logic's DT ensemble evaluation cores enabling hardware implementation of the inferred DT ensemble. By combining these two cores a hardware-based adaptive learning ensemble systems can be easily designed.

So_ip_idte core is delivered with fully automated testbench and a compete set of tests allowing easy package validation at each stage of SoC design flow.

The so_ip_idte design is strictly synchronous with positive-edge clocking, no internal tri-states and a synchronous reset.

The so_ip_idte core can be evaluated using any evaluation platform available to the user before actual purchase. This is achieved by using a time-limited demonstration bit files for selected platform that allows the user to evaluate system performance under different usage scenarios.

Features

- Enables DT ensemble creation directly in hardware
- Speedup of inference time of over 1000x compared to the traditional software approach
- Supports classification problems that are defined by numerical attributes only
- DTs with univariate or multivariate tests are supported
- DTs with nonlinear tests are supported
- No special IP blocks are needed to implement the core, only memory, adders and multipliers
- User can specify the number format for all DT ensemble parameters in order to achieve the best performance/size ratio after implementation
- Can be easily integrated with some of the So-Logic's DT ensemble evaluation cores to create hardware-based adaptive learning ensemble systems

Applications

- Speech and handwriting recognition
- Computer vision
- Machine perception

- Pattern recognition
- Medical diagnosis
- Robot locomotion
- Adaptive systems

Deliverables

- Source code:
  - VHDL Source Code
- VHDL test bench environment
  - Tests with reference responses
- Technical documentation
  - Installation notes
  - HDL core specification
  - Datasheet
- Instantiation templates
- Example application
- Technical Support
  - IP Core implementation support
  - Variable length maintenance
    - Delivery of IP Core updates, minor and major changes
    - Delivery of documentation updates
    - Telephone & email support

Licensing

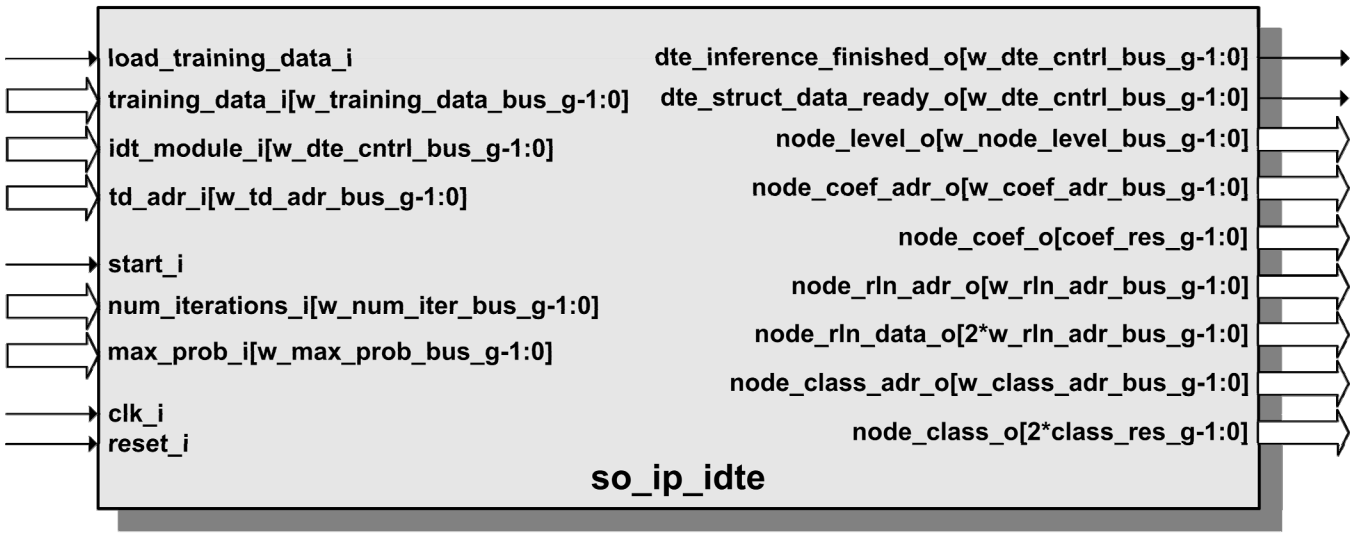
Netlist License

- Post-synthesis netlist
- Self checking testbench
- Test vectors for testing the core
- Place&Route scripts
- Constraints
- Instantiation templates
- Documentation

VHDL Source License

- VHDL RTL source code
- Complete verification plan together with testbenches needed to verify correct operation of the core
- Self checking testbench
- Vectors for testing the functionality of the core
- Simulation & synthesis scripts
- Documentation

Symbol



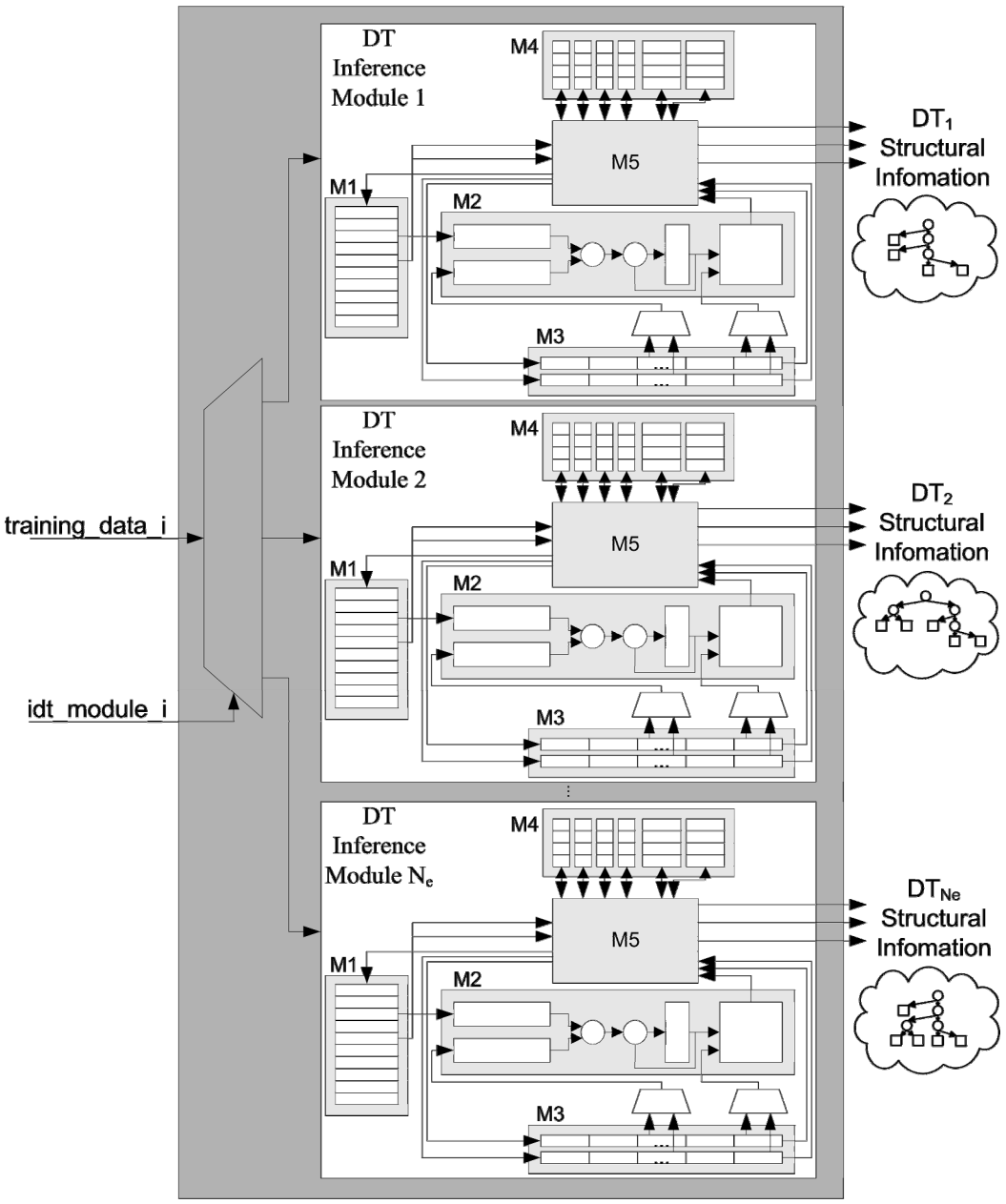
Pin Description

Name	Signal Direction	Description
Global Clocks and Reset Ports		
clk_i	Input	Main clock input
reset_i	Input	Main reset
Training Set Interface		
load_training_data_i	Input	Signal that is used to load the new training set data (attribute values and class membership information for the current instance) into the training set memory of the selected module for the DT inference
training_data_i[w_training_data_bus_g-1:0]	Input	Data bus that is used to transfer the information (attribute values or class membership value) about the instance that is being transferred to the training set memory of the selected module for the DT inference
idt_module_i[w_dte_cntrl_bus_g-1:0]		Address bus used to select which DT inference module is

		to be used in the current data transaction
td_adr_i[w_td_adr_bus_g-1:0]	Input	Address bus that is used to specify the training set memory location where the training data should be stored
Control Interface		
start_i	Input	Indication that the training set is loaded into training set memory and the process of DT ensemble inference can commence
num_iterations_i[w_num_iter_bus_g-1:0]	Input	Number of iterations of the DT optimization algorithm that should be performed during the optimization of every DT node
max_prob_i[w_max_prob_bus_g-1:0]	Input	Maximum mutation probability that should be used during the node optimization procedure
Inferred DT Structural Information Interface		
dt_inference_finished_o[w_dte_cntrl_bus_g-1:0]	Output	DT inference indication bus. Each bit that is set indicates that the inference of that particular DT member is completed
dt_struct_data_ready_o[w_dte_cntrl_bus_g-1:0]	Output	Indication bus that is used to signal that the new structural information about DT ensemble members is ready to be transmitted. Each bit is related with one ensemble member.
node_level_o[w_node_level_bus_g-1:0]	Output	Information about the DT level in which is the current node located
node_coef_adr_o[w_coef_adr_bus_g-1:0]	Output	Address of the location in the node coefficient memory where the current coefficient value should be stored
node_coef_o[coef_res_g-1:0]	Output	Coefficient value that should be stored in the node coefficient memory
node_rln_adr_o[w_rln_adr_bus_g-1:0]	Output	Address of the location in the next-node memory where the information about the current node's successors should be stored
node_rln_data_o[2*w_rln_adr_bus_g-1:0]	Output	Information about the locations of the two

		successors of the current DT node that should be stored in the next-node memory
node_class_adr_o[w_class_adr_bus_g-1:0]	Output	Address of the location in the class memory where the information about the output class membership for the current node should be stored
node_class_o[2*class_res_g-1:0]	Output	Output class membership for the current node that should be stored in the class memory

Block Diagram



Functional Description

The previous diagram shows all major modules of the so_ip_idte core that are described here in more detail.

Architecture of the so_ip_idte core consists of N_e DT inference modules that operate in parallel, N_e being the size of the DT ensemble that is being created. Each of the N_e DT inference modules is responsible for creating one DT from the ensemble. This means that all DTs from the ensemble are being created in parallel. Each DT inference module is composed from five major modules: training set instance memory (M1), instance evaluation module (M2), chromosome memory (M3), instance index and count memories (M4) and control unit (M5).

Module M1 holds the training set instances that should be used to infer this particular DT from the ensemble. Each instance is represented by the vector of *n* attribute values followed by the class membership value. Size of the memory depends on the size of the training set that is used to create a DT.

Module M2 calculates the instance position relative to the candidate hyperplane. This information is needed to calculate the value of the cost function during the hyperplane position optimization.

Module M3 is used to store both the encoding of the best so-far hyperplane and the encoding of the current candidate hyperplane (that is to store two chromosomes).

Module M4 consists of six memories, four of which are used to store the instance indexes associated with the nodes in the current and the next DT level. To store the information about the number of instances from every class that are located above and below the candidate hyperplane, two count memories are also needed. This information is needed to calculate the value of the cost function during the evolutionary optimization of the hyperplane position.

Module M5 is the control unit responsible for the correct operation of the entire system. Control unit performs the steps defined in the proprietary DT inference algorithm to build the DT, and controls the overall operation of the core.

Using the control interface user can modify the parameters of the DT ensemble inference algorithm and by doing so alter the overall DT ensemble inference process.

At regular interval, which depends on the current settings of the DT inference algorithm parameters and the training set associated to each DT that is being inferred, structural information about the next DT node that has been optimized is transferred to the user through the DT structural information interface. Each DT from the ensemble has its own structural interface port, which means that structural information about every DT from the ensemble is being transferred to the user in parallel, as it becomes available.

Verification Methods

Decision tree ensemble inference core was tested both using sophisticated verification environment and in dedicated hardware platform. Verification environment was used to extensively verify the so_ip_idte core's operation for different types and sizes of decision tree ensembles. After reaching all verification goals, IP core was next tested using dedicated hardware platform. Using this platform so_ip_idte core was implemented in FPGA and tested in real applications to estimate the performance of the core. The details about the verification

methodology that was used and performance results during hardware testing can be obtained from So-Logic upon request.

Device Utilization & Performance

In order to get a better estimation about the required resources in actual applications, following table presents the so_ip_idte core implementation results for selected datasets obtained from the UC Irvine Machine Learning Repository. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. All reported implementation results are obtained under the assumption that the ensemble is composed from 30 individual DTs.

UCI Dataset	Size of Training Dataset	Number of Problem Attributes	Number of Classes	Number of Slice LUTs	Number of BRAMs	Number of DPS48E BLocks
Australian Credit Approval	690	14	2	145740	30	270
Balance Scale	625	4	3	134670	30	270
Breast Cancer	264	9	2	131910	30	270
Breast Cancer Wisconsin	699	9	2	140850	90	270
Car Evaluation	1728	6	4	138960	30	270
Contraceptive Method Choice	1333	9	3	152880	30	270
German Credit	1000	24	2	134070	30	270
Glass Identification	214	9	6	138630	30	270
Cleveland Heart Disease	297	13	5	136680	30	270
Statlog Heart Disease	270	13	2	136440	30	270
Hepatitis	155	19	2	153630	30	270
Ionosphere	351	34	2	125880	30	270
Iris	150	4	3	133410	30	270
Liver Disorders	345	6	2	137760	30	270
Lymphography	148	18	4	149130	90	270
Page Blocks	5427	10	5	137220	30	270
Pima Indians Diabetes	768	8	2	171300	30	270
Sonar	208	60	2	126720	30	270
Thyroid Disease	215	5	3	138780	30	270
Tic-Tac-Toe Endgame	958	9	2	145350	30	270
Statlog Vehicle Silhouettes	846	18	4	139200	30	270
Vote	232	16	2	144120	30	270
Vowel Recognition	990	13	11	154770	90	270
Waveform21	5000	21	3	170880	90	270
Waveform40	5000	40	3	156660	30	270
Wisconsin Diagnostic Breast Cancer	569	30	2	133950	30	270
Wine	178	13	3	136470	30	270
Wisconsin Prognostic Breast Cancer	194	33	2	147300	30	270
Zoo	101	17	7	133050	30	270

Notes:

1. All core I/O signals are routed off chip
2. Results were obtained using Xilinx ISE 9.1.03i version of software, targeting Virtex-5 family FPGA devices
3. The synthesis results provided are for reference only. Please contact So-Logic for estimates for your particular application.

Contact Information

So-Logic
Lustkandlgasse 52/22
A-1090 Vienna
Austria/Europe
Phone: +43-1-3157777-11
Fax: +43-1-3157777-44
E-Mail: ip_idte@so-logic.net
URL: http://www.so-logic.net

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
01/10/2009	1.0	Initial release.