# Functional Decomposition Machine

# PLAN OF EVOLVABLE AND LEARNING HARDWARE LECTURES

- **Our hardware** : the **DEC-PERLE-1** board.
  - Programming/designing environment for DEC-PERLE/XILINX.
  - Two different concepts of designing Learning Hardware using the DEC-PERLE-1 board.
- Compare **logic** versus ANN and GA approaches to learning.
- Introduce the concept of **Learning Hardware**
- Methods of **knowledge representation** in the **Universal Logic Machine (ULM):**
  - **variants of Cube Calculus**.
- A general-purpose computer with instructions specialized to operate on logic data: **Cube Calculus Machine**.
  - Variants of cube calculus - arithmetics for combinatorial problems
  - Our approach to Cube Calculus Machine

We are here

- A processor for only one application: **Curtis Decomposition Machine.**

# DECOMPOSITION IN HARDWARE

- Function Decomposition is at least an NP-hard problem.
- Most its stages are NP-hard problems.
- One approach to find solutions to NP-hard problem is not to attempt at the exact solution, but be satisfied with one which is near exact but obtainable in a reasonable time.
- This type of algorithm is based on heuristics, or rules which can be applied which are likely to improve the solution.
- Such algorithms, when implemented in hardware, can bring orders of magnitude speed-up
- We have chosen algorithms that are simple, easy, fast and can be relatively easy implemented in hardware.
- We showed that decomposition of fuzzy functions and relations can be reduced to decomposition of multi-valued functions and relations.

# LEARNING HARDWARE METHODOLOGY

# LEARNING BY FUNCTIONAL DECOMPOSITION MACHINE

- Design philosophy of the FPGA implementation of a point algorithm.
- Phases of the algorithm are executed sequentially, they are then loaded from the host memory, while the intermediate data are stored in DEC-PERLE-1 memories between stages.
- We will show also how generic combinatorial problems are used in logic learning algorithms.
- The ideas of graph coloring will be used for decomposing functions, and thus in Machine Learning

The decision table represents a data set, with labeled instances, each relating a set of attribute values to a class (the **output concept**).

Decomposition of the table is to decompose the initial table into a hierarchy of decision tables, each of them no more decomposable.

Thus, each of these new tables, as well as the entire network are less complex and easier to interpret than the original table.

Some regularities not seen in the original table can be found,

and the intermediate functions correspond to some features (concepts) of the data set.

Ashenhurst/Curtis Decomposition has been adopted to multiple-valued logic (ISMVL'97).

It applies iteratively the **single decomposition step**, whose goal is to decompose a function $y = F(X)$ into $y = \mathbf{G(A,H(B))}$, where $\mathbf{X}$ is a set of input attributes $x_1, x_2, \ldots, x_n$, and $y$ is the class.

**F,G, and H** are functions represented as decision tables, i.e. possibly incomplete sets of attribute-value vectors with assigned classes.

**A and B** are subsets of input attributes, called free and bound set, respectively, such that $A \cup B = X$.

Functions G and H are developed in the decomposition process and not predefined in any way

New concept $c_1 = H(B)$ has been found.

The goal is to find the decomposition of the smallest complexity (DFC - Abu-Mostafa).

# Example of Decomposition

Three possible non-trivial partitions of attributes that yield three different decompositions

$y = G_1(x_1, H_1(x_2, x_3)),$ $\qquad y = G_2(x_2, H_2(x_1, x_3)),$

$y = G_3(x_3, H_3(x_1, x_2)).$

The comparison shows that:

1. decision tables in the decomposition $y = G_1(x_1, H_1(x_2, x_3))$ are smaller than those for $y = G_2(x_2, H_2(x_1, x_3))$,
2. the new concept $c_i = H_1(x_2, x_3)$ uses only three values, whereas that for $H_2(x_1, x_3)$ uses four,
3. we found it hard to interpret decision tables $G_2$ and $H_2$, whereas by inspecting $H_1$ and $G_1$ it can be easy to see that $c_1 = MIN(x_2, x_3)$ and $y = MAX(x_1, c_1)$.
4. This can be even more evident with the assignment of values 0,1, and 2 of a multi-valued variable $X_i$: ${X_i}^0 = lo$, ${X_i}^1 = me$, ${X_i}^2 = hi$.

# An example decision table $y = F(x_1, x_2, x_3)$

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| lo | lo | lo | lo |
| lo | lo | hi | lo |
| lo | me | lo | lo |
| lo | me | hi | me |
| lo | hi | lo | lo |
| lo | hi | hi | hi |
| lo | hi | hi | hi |
| me | lo | lo | me |
| me | lo | hi | me |
| me | me | lo | me |
| me | me | hi | me |
| me | hi | lo | me |
| me | hi | hi | hi |
| hi | lo | lo | hi |
| hi | lo | hi | hi |
| hi | me | lo | hi |
| hi | me | hi | hi |
| hi | hi | lo | hi |
| hi | hi | hi | hi |

# Two one-step decompositions

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| lo | lo | lo | lo |
| lo | lo | hi | lo |
| lo | me | lo | lo |
| lo | me | hi | me |
| lo | hi | lo | lo |
| lo | hi | hi | hi |
| me | lo | lo | me |
| me | lo | hi | me |
| me | me | lo | me |
| me | me | hi | me |
| me | hi | lo | me |
| me | hi | hi | hi |
| hi | lo | lo | hi |
| hi | lo | hi | hi |
| hi | me | lo | hi |
| hi | me | hi | hi |
| hi | hi | lo | hi |
| hi | hi | hi | hi |

(a)

| $x_1$ \ $x_2$, $x_3$ | lo, lo | lo, hi | me, lo | me, hi | hi, lo | hi, hi |
|---|---|---|---|---|---|---|
| lo | lo | lo | lo | me | lo | hi |
| me | me | me | me | me | me | hi |
| hi | hi | hi | hi | hi | hi | hi |
| c | 1 | 1 | 1 | 2 | 1 | 3 |

(b)

| $x_2$ \ $x_1$, $x_3$ | lo, lo | lo, hi | me, lo | me, hi | hi, lo | hi, hi |
|---|---|---|---|---|---|---|
| lo | lo | lo | me | me | hi | hi |
| me | lo | me | me | me | hi | hi |
| hi | lo | hi | me | hi | hi | hi |
| c | 1 | 2 | 3 | 4 | 5 | 5 |

(c)

| $x_3$ \ $x_1$, $x_2$ | lo, lo | lo, me | lo, hi | me, lo | me, me | me, hi | hi, lo | hi, me | hi, hi |
|---|---|---|---|---|---|---|---|---|---|
| lo | lo | lo | lo | me | me | me | hi | hi | hi |
| lo | lo | me | hi | me | me | hi | hi | hi | hi |
| c | 1 | 2 | 3 | 4 | 4 | 5 | 6 | 6 | 6 |

(d)

| $x_1$ | $c_1$ | $y$ |
|---|---|---|
| lo | 1 | lo |
| lo | 2 | me |
| lo | 3 | hi |
| me | 1 | me |
| me | 2 | me |
| me | 3 | hi |
| hi | 1 | hi |
| hi | 2 | hi |
| hi | 3 | hi |

| $x_2$ | $x_3$ | $c_1$ |
|---|---|---|
| lo | lo | 1 |
| lo | hi | 1 |
| me | lo | 1 |
| me | hi | 2 |
| hi | lo | 1 |
| hi | hi | 3 |

(e)

| $x_1$ | $x_3$ | $c_2$ |
|---|---|---|
| lo | lo | 1 |
| lo | hi | 2 |
| me | lo | 3 |
| me | hi | 4 |
| hi | lo | 5 |
| hi | hi | 5 |

| $x_2$ | $c_2$ | $y$ |
|---|---|---|
| lo | 1 | lo |
| lo | 2 | lo |
| lo | 3 | me |
| lo | 4 | me |
| lo | 5 | hi |
| me | 1 | lo |
| me | 2 | me |
| me | 3 | me |
| me | 4 | me |
| me | 5 | hi |
| hi | 1 | lo |
| hi | 2 | hi |
| hi | 3 | me |
| hi | 4 | hi |
| hi | 5 | hi |

(f)

# DECOMPOSITION (cont)

- The following problems must be solved by an efficient decomposition algorithm:

1. how to select sets A and B?

2. how to evaluate the quality of decompositions?

All known methods require nearly exhaustive searches that involve huge repetitions of basic operations.

•The assignment of values of c is trivial in case of a completely specified function, which is, when decision table instances completely cover the attribute space.

Otherwise, when the function is incompletely specified, the relation of compatibility of columns is no longer transitive, and the **graph coloring approach** is used.

Column functions are calculated by a cofactor operation on the original function f .

The **cofactor** $f_{PROD}$ of function f with respect to the literals from PROD is this function with all literals from PROD substituted to maximum constant value (constant value 1 in case of binary logic).

All functions are represented by arrays of cubes.

# BASIC OPERATIONS: COMPLETE TAUTOLOGY OF COFACTORS

For a completely specified binary function, two columns n_1 and n_2 are compatible if the Boolean functions corresponding to them are a Boolean Tautology:

**n_1 compatible n_2 iff f_{Prod_1} = f_{Prod_2}**

which is equivalent to:

n_1 compatible n_2 iff (ON(n_1) # ON(n_2) = emptyset)
and (ON(n_2) # ON(n_1) = emptyset

where # denotes the sharp (difference) operation on arrays of cubes, and ON is the set of true cubes in SOP form.

# BASIC OPERATIONS: INCOMPLETE TAUTOLOGY OF COFACTORS

For an incompletely specified binary function, two nodes of the graph for coloring are incompatible if the corresponding columns are not compatible (cannot be merged into one column):

n_1 incompatible n_2 iff ( ON(n_1) /\ OFF(n_2) /= emptyset or ( ON(n_2) /\ OFF(n_1) /= emptyset

# DECOMPOSITION IN HARDWARE

- Only two basic operations, **cofactor** and **sharp** are used for complete functions.
- Only cofactor and **intersection** are used for incomplete functions.
- In both cases, these operations are repeated many times on cubes from the cube arrays
- Basic (mv) logic operators used for checking compatibility of columns of multiple-valued functions while creating the graph for coloring.

# DECOMPOSITION IN HARDWARE

- After creation, the graph is colored in such a way that every two nodes linked by an edge obtain different colors, and the minimum number of colors is used.
- Graph coloring can be reduced to sequences of basic logic operators.
- Concluding, in addition to cofactoring, the partial combinatorial problems that are solved by our hardware decomposition processor DP are:
  - **set covering,**
  - **graph coloring,**
  - **maximum clique.**
- They are all NP-hard, and they all have many other applications in ML.

# OTHER VIRTUAL PROCESSORS

- **Rough Set Machine (RSM).**
  - A SIMD processor that realizes the basic operations of Rough Sets theory of Zdzislaw Pawlak.
- **Satisfiability Machine (SM).**
  - A systolic processor to solve satisfiability and related problems that occur in many combinatorial optimization problems.

# CONCLUSIONS

- Principles of Learning Hardware as a competing approach to Evolvable Hardware, and also as its generalization.
- Data Mining machines.
- Universal Logic Machine with several virtual processors.
- DEC-PERLE-1 is a good medium to prototype such machines, its XC3090A chip is now obsolete.
- This can be much improved by using XC4085XL FPGA and redesigning the board.
- Massively parallel architectures such as CBM based on new Xilinx series 6000 chips will allow even higher speedups.

# LITERATURE


- Y. Abu-Mostafa (ed.), `` Complexity in Information Theory,'' Springer Verlag, New York, 1988, p. 184.
- R.L. Ashenhurst, ``The Decomposition of Switching Functions'', Proc. Int. Symp. of Th. of Switching,} 1957.
- A. Buller, ``Artificial Brain. Phantasies no more,''Proszynski i Ska, Warsaw, 1998, (in Polish).
- P. Burkey, M. Perkowski, and A. Wielgus, ``Ashenhurst/Curtis Decomposition of Fuzzy Functions and Relations,'' submitted to Multiple-Valued Logic. An International Journal, Gordon and Breach Science Publishers, 1999.
- R.E. Bryant, ``Graph-based algorithms for boolean function manipulation'', IEEE Transactions on Computers, C-35, No. 8, pp. 667-691, 1986.
- E.F. Codd, ``A Relational Model of Data for Large Shared Data Banks,'' Comm. ACM, 13, pp. 377-387.
- H.A. Curtis, ``A New Approach to the Design of Switching Circuits,'' Princeton, N.J., Van Nostrand, 1962.
- D.L. Dietmeyer, ``Logic Design of Digital Systems,'' Allyn and Bacon, Boston, MA, 1971.
- K. Dill, and M. Perkowski, ``Minimization of Generalized Reed-Muller Forms with Genetic Operators,'' Proc. Genetic Programming '97 Conf., July 1997, Stanford Univ., CA.
- K. Dill, J. Herzog, and M. Perkowski, ``Genetic Programming and its Application to the Synthesis of Digital Logic,'' Proc. PACRIM '97, Canada, August 20-22, 1997.
- K. Dill, and M. Perkowski, ``Evolutionary Minimization of Generalized Reed-Muller Forms,'' Proc. ICCIMA'98 Conference}, pp. 727-733, February 1998, Australia, published by World Scientific.

- B. Falkowski, I. Schaefer, M. Perkowski, ``Effective Computer Methods for the Calculation of Rademacher-Walsh Spectrum for Completely and Incompletely Specified Boolean Functions,'' IEEE Trans. on Computer-Aided Design , pp. 1207 - 1226, October 1992.
- C. Files, M. Perkowski, ``An Error Reducing Approach to Machine Learning Using Multi-Valued Functional Decomposition,'' Proc. ISMVL'98, pp. 167 - 172, May 1998.
- C. Files, M. Perkowski, ``Multi-Valued Functional Decomposition as a Machine Learning Method,'' {\em Proc. ISMVL'98, pp. 173 - 178, May 1998.
- J.M. Francioni, and A. Kandel, ``Decomposable Fuzzy-valued Switching Functions,'' Fuzzy Sets and Systems, Vol. 9, No. 1, pp. 41-68, 1983.
- H. DeGaris, ``Evolvable Hardware: Genetic Programming of a Darwin Machine,'' In ``Artificial Nets and Genetic Algorithms,'' R.F. Albrecht, C.R. Reeves and N.C. Steele (eds), Springer Verlag, pp. 441-449, 1993.
- H. DeGaris, ``Evolvable Hardware: Principles and Practice,'' CACM Journal, August 1997.

- $http://www.hip.atr.co.jp/\sim degaris
- S. Grygiel, and M. Perkowski, ``New Compact Representation of Multiple-Valued Functions, Relations, and Non-deterministic State Machines,'' Proc. ICCD'98, October 1998.
- T. Higuchi, M. Iwata, and W. Liu (eds), ``Evolvable Systems: From Biology to Hardware,'' Lecture Notes in Computer Science,} No. 1259, Proc. First Intern. Conf. ICES'96, Tsukuba, Japan, October 1996, Springer Verlag, 1997.
- L. Jozwiak, M.A. Perkowski, D. Foote, ``Massively Parallel Structures of Specialized Reconfigurable Cellular Processors for Fast Symbolic Computations,'' Proc. MPCS'98 - The Third International Conference on Massively Parallel Computing Systems, Colorado Springs, Colorado - USA, April 6-9, 1998.
- T. Luba, J. Rybnik, ``Algorithmic Approach to Discernibility Function with Respect to Attributes and Object Reduction,'' Int. Workshop on Rough Sets,} Poznan 1992.

- T. Luba, ``Decomposition of multiple-valued functions'', {\em Proc. 25th ISMVL, 1995, pp. 256-261.
- R. Malvi, M. Perkowski, and L. Jozwiak, ``Exact Graph Coloring for Functional Decomposition: Do we Need it?,'' pp. 1-10, Proceedings of 3rd International Workshop on Boolean Problems, Freiberg University of Mining and Technology, Institute of Computer Science, September 17-18, 1998.
- C. Mead, ``Analog VLSI And Neural Systems,'' *Addison Wesley Pub.,* April 1989.
- R.S. Michalski and J.B. Larson, ``Inductive inference of vl decision rules,'' in *Workshop in Pattern-Directed Inference Systems,* Hawaii, May 1977.
- R.S. Michalski, I. Bratko, and M. Kubat, ``Machine Learning and Data Mining: Methods and Applications,'' *Wiley and Sons,* 1998.
- D. Michie, ``Machine Learning in the next five years,'' *Proc. EWSL'88, 3rd European Working Session on Learning,* Glasgow, *Pitman*, London, 1988.

- L. Nguyen, M. Perkowski, N. Goldstein, ``PALMINI - Fast Boolean Minimizer for Personal Computers,''*Proc. of the IEEE/ACM 24th Design Automation Conference,* pp. 615 - 621, Miami, Florida, June 28 - July 1, 1987.
- Z. Pawlak, ``Rough Sets. Theoretical Aspects of Reasoning about Data,'' Kluwer Academic Publishers, 1991.
- M. Perkowski, ``Systolic Architecture for the Logic Design Machine,'' Proc. of the IEEE and ACM International Conference on Computer Aided Design - ICCAD'85, pp. 133 - 135, Santa Clara, 19 - 21 November 1985
- M. Perkowski, S. Wang, W.K. Spiller, A. Legate, E. Pierzchala, ``Ovulo-Computer: Application of Image Processing and Recognition to Mucus Ferning Patterns,'' Proc. of the Third IEEE Symposium on Computer-Based Medical Systems, pp. 52 - 59, Chapel Hill, North Carolina, June 3-6, 1990.

- M.A. Perkowski, ``A Universal Logic Machine,'' invited address, Proc. of the 22nd IEEE International Symposium on Multiple Valued Logic, ISMVL'92, pp. 262 - 271, Sendai, Japan, May 27-29, 1992.
- M. A. Perkowski, M. Chrzanowska-Jeske, ``Multiple-Valued-Input TANT Networks,'' Proc. ISMVL'94, pp. 334-341, Boston, MA, May 25-27, 1994.
- M. A. Perkowski, T. Ross, D. Gadd, J.A. Goldman, and N. Song, ``Application of ESOP Minimization in Machine Learning and Knowledge Discovery,'' Proc. of the Second Workshop on Applications of Reed-Muller Expansion in Circuit Design, Chiba City, Japan, 27-29 August 1995, pp. 102-109.
- M. Perkowski, M. Marek-Sadowska, L. Jozwiak, T. Luba, S. Grygiel, M. Nowicka, R. Malvi, Z. Wang, and J. S. Zhang, ``Decomposition of Multiple-Valued Relations,'' Proc. ISMVL'97, Halifax, Nova Scotia, Canada, May 1997, pp. 13 - 18.

- M. Perkowski, P. Lech, Y. Khateeb, R. Yazdi, and K. Regupathy, ``Software-Hardware Codesign Approach to Generalized Zakrevskij Staircase Method for Exact Solutions of Arbitrary Canonical and Non-Canonical Expressions in Galois Logic,'' Booklet of 6th Intern. Workshop on Post-Binary ULSI Systems, Nova Scotia, Canada, May 27, 1997, pp. 41 - 44.
- M. A. Perkowski, L. Jozwiak, and D. Foote, "Architecture of a Programmable FPGA Coprocessor for Constructive Induction Approach to Machine Learning and other Discrete Optimization Problems", in Reiner W. Hartenstein and Victor K. Prasanna (ed) ``Reconfigurable Architectures. High Performance by Configware,'' IT Press Verlag, Bruchsal, Germany, 1997, pp. 33 - 40.
- M. Perkowski, L. Jozwiak, and S. Mohamed, ``New Approach to Learning Noisy Boolean Functions,'' Proc. ICCIMA'98 Conference, February 1998, Australia, published by World Scientific, pp. 693 - 706. Australia, published by World Scientific.

- M. Perkowski, ``Do It Yourself Reconfigurable Supercomputer that Learns,'' book preprint , Portland, Oregon, 1999.
- PSU POLO Directory with DM/ML Benchmarks, software and papers: http://www.ee.pdx.edu/polo/
- E. Pierzchala and M. Perkowski, ``A High-Frequency Field-Programmable Analog Array (FPAA), Part 1: Design, Part 2: Applications,'' Field-Programmable Analog Arrays, (E. Pierzchala, ed.), Kluwer Academic Publishers, 1998.
- L.O. Chua and T. Roska, ``The CNN paradigm,'' IEEE Trans. on Circuits and Systems-I, Vol. 40, No. 3, pp. 148-156, March 1993.
- T. D. Ross, M.J. Noviskey, T.N. Taylor, and D.A. Gadd, ``Pattern Theory: An Engineering Paradigm for Algorithm Design,'' Final Technical Report WL-TR-91-1060 Wright Laboratories, USAF, WL/AART/WPAFB, OH 45433-6543, August 1991.

- P. Sapiecha, M. A. Perkowski, and T. Luba, ``Decomposition of Information Systems Based on Graph Coloring Heuristics,'' Symposium on Modelling, Analysis and Simulation, CESA'96 IMACS Multiconference, Lille, France, July 9-12, 1996.
- T. Sasao (editor), ``Representation of Boolean Functions,'' Kluwer Academic Publishers, 1996
- K.B. Stanton, P.R. Sherman, M.L. Rohwedder, Ch.P. Fleskes, D. Gray, D.T. Minh, C. Espinosa, D. Mayi, M. Ishaque, M.A. Perkowski, ``PSUBOT - A Voice-Controlled Wheelchair for the Handicapped,'' Proc. of the 33rd Midwest Symp. on Circuits and Systems, pp. 669 - 672, Alberta, Canada, August 1990.
- Y.H. Su and P.T. Cheung, ``Computer minimization of multiple-valued switching functions,'' IEEE Transactions on Computers, Vol. C-21, pp. 995-1003, 1972.
- ***N. Song, M. Perkowski***, ``Minimization of Exclusive Sum of Products Expressions for Multi-Output Multiple-Valued Input, Incompletely Specified Functions,'' IEEE Transactions on Computer Aided Design, Vol. 15, No. 4, April 1996, pp. 385-395.

- U.C. Irvine, ``Repository of Machine Learning Databases and Domain Theories,'' $ftp://ftp.ics.uci.edu/pub/machine-learning-databases/
- J. Vuillemin, P. Bertin, D. Roncin, M. Shand, H. Touati, and Ph. Boucard, ``Programmable Active Memories: Reconfigurable Systems Come of Age,'' IEEE Trans. on VLSI Systems, Vol. 4, No. 1., pp. 56-69, March 1996
- W. Wan, and M. Perkowski, ``A New Approach to the Decomposition of Incompletely Specified Multi-Output Function Based on Graph Coloring and Local Transformations and Its Application to FPGA Mapping,'' **Proc. Euro-DAC,** pp. 230 - 235, 1992.