POLARIZED PSEUDO-KRONECKER SYMMETRY AND SYNTHESIS OF 2x2 LATTICE DIAGRAMS

BENJAMIN T. DRUCKER, CRAIG M. FILES,
MAREK A. PERKOWSKI and MALGORZATA CHRZANOWSKA-JESKE
Department of Electrical Engineering, Portland State University
Portland, OR 97207, USA

Layout-driven logic synthesis combines logical and physical design to minimize interconnect length for speed-, noise- and power-critical applications. The lattice diagram synthesis approach constructs, for combinational functions, regular lattices with only local connections and input buses. Lattice diagrams are directly mappable to hardware without additional layout steps. This synthesis approach decomposes functions symmetrically using Shannon and Davio expansions and by repeating decomposition variables, when necessary. It has been proven that lattice diagrams can always be constructed in this manner, independent of input variable ordering and expansion types applied. Lattice size, however, is quite sensitive to these parameters. This paper proposes new types of symmetries of Boolean functions, and shows how they can be used in a new approach to the important problem of finding variable-expansion orders for lattices. Although this research originates from specific technological considerations, a general problem in functional decomposition is considered: how to decompose a function to a regular structure of simpler functions using symmetries. It has also applications in functional decomposition and Machine Learning.

1 Introduction

Traditionally, implementation of circuits has been broken into disjoint tasks of logic synthesis and physical design. First, a synthesizer estimates circuit area and speed independently of target technology. A layout engine then maps synthesis results to hardware. This approach works well with circuits designed with rules of older manufacturing processes. For such circuits and processes, a good area metric is the total number of devices used and a good speed metric is the length of the longest chain of devices. Area estimates based entirely on device count are used because wires are very small compared to device size for older technologies and because wires can pass over regions of die already used to implement transistors. Speed estimates based entirely on device count are also fairly accurate because interconnect RC delay typically contributes negligible amounts to total speed compared to device propagation delay.

As circuit size and connectivity increase and gate dimensions approach 0.1 micron, problems of logic synthesis and physical design become more closely interrelated. For ULSI and deep sub-micron processes, synthesis processes are required which optimize decomposition with respect to implementation resource topology. Synthesis results should map directly to hardware with minimal interconnect delays.

One way to control the complexity of such layout-driven synthesis is to use regular hardware architectures that are tuned to specific types of logic decomposition. An early example of this is Akers' rectangular logic array [1] which is a two-dimensional, planar grid of small cells. A benefit of this structure is that each cell connects to its four neighbors, thereby limiting the distance most signals are required to travel. However, since Akers designed his synthesis method for worst-case problems, it produces very large and sparse arrays for many common circuits.

Akers' approach could benefit from symmetry information of functions to be processed. Symmetry indicates parts of a function that are used more than once. Methods which detect symmetry can save area by implementing symmetrical parts only one time in hardware. Target architectures optimized for this type of subfunction sharing can also minimize interconnect delays.

This paper first describes symmetry as it is typically defined in literature, and then extends this concept to the more general *polarized pseudo-Kronecker symmetry*. Section 3 introduces a planar, two-dimensional topology to illustrate how polarized pseudo-Kronecker symmetry can be applied to a specific architecture, 2x2 lattice. The choice of architecture is not essential; we believe this concept is extensible to any geometry, such as 3x3 lattices and layouts whose nodes have even more neighbors. Section 4 outlines a heuristic, layout-driven synthesis method based on polarized pseudo-Kronecker symmetry, and section 5 gives experimental results.

2 Standard Symmetry and Polarized Pseudo-Kronecker Symmetry

Most research on symmetry defines *symmetry* as a property of a function $f(X)$ with respect to a subset λ of variables in which f is invariant for any permutation of variables in λ. Subset λ is called a *symmetry set*, or if $|\lambda| = 2$, a *symmetry pair*. For the remainder of this paper, symmetry of this restricted type is prefaced with the modifier *standard*, e.g. standard symmetry, standard symmetry pair, etc.

The concept of standard symmetry can be expanded to include several new classifications.[2] Function $f(X)$ is *Non-equivalent symmetric* with respect to symmetry pair λ if Eq. 1 holds, and function $f(X)$ is *Equivalent symmetric* if Eq. 2 holds. If f is invariant when only one variable in λ changes, f exhibits *Single-Variable Symmetry* with respect to λ.

$$f(X - \lambda, 0, 1) = f(X - \lambda, 1, 0) \tag{1}$$

$$f(X - \lambda, 0, 0) = f(X - \lambda, 1, 1) \tag{2}$$

Tsai and Marek-Sadowska[3] introduce *skew symmetries* to describe functions which only change polarity in response to a symmetry pair permutation. For example, if $f(X,1,1)=0$ and $f(X,0,0)=1$,f is said to be *skew equivalent symmetric*.

All symmetry relations described above involve comparing different cofactor functions with respect to a given pair of variables. Polarized pseudo-Kronecker symmetry (PPKS) is more general than standard symmetry in several ways. First, PPKS is a relationship among subfunctions created during arbitrary *nonsingular expansions of functions*[4-9] not just among positive and negative cofactors. This generalization allows for expansions other than Shannon. For instance, the simplest nonsingular expansions are Davio expansions[12] which generate exclusive sums of two cofactors. Polarized pseudo-Kronecker symmetry gets its name, in part, from the observation that this kind of symmetry can describe equivalence among nodes of pseudo-Kronecker functional decision diagram (FDD) representations of functions.[10,11]

Polarized pseudo-Kronecker symmetry further expands standard symmetry by relaxing restrictions on polarities of variables and subfunctions used for expansion. For example, for a function f which is decomposed twice by Shannon expansion – first by a, then by b – standard symmetry holds when $f_{b_1}(f_{a_0}) = f_{b_0}(f_{a_1})$ where f_{a_0} and f_{a_1} are negative and positive cofactors of f with respect to a. If either a or b is negated, standard symmetry becomes equivalent symmetry. If either cofactor with respect to b is negated, standard symmetry becomes skew symmetry.

The power of polarized pseudo-Kronecker symmetry over previous symmetry definitions can be seen when applied to Davio expansions. For example, if function f is decomposed by positive Davio followed by negative Davio expansions – using a and b respectively – one possible polarized pseudo-Kronecker symmetry holds when $f_{\overline{b}_1}(f_{a_0}) = f_{\overline{b}_2}(\overline{f_{a_2}})$ where f_{a_2} is the exclusive sum of negative and positive cofactors of f with respect to a. In this example, subfunction f_{a_2} and expansion variable b are negated.

3 2x2 Lattices and Lattice-Specific PPKS

To illustrate polarized pseudo-Kronecker symmetry, we use a lattice geometry similar to Akers' array. A *2x2 lattice* is planar array of hardware cells in which each cell is adjacent to four other cells. A cell inputs one signal each from two neighboring cells (in_A and in_B) and provides one output to each of the remaining two neighbors (out_A and out_B).

In addition to local interconnects which transmit information between adjacent cells, global interconnects distribute common signals to multiple cells. A third cell input, x, is a tap off a diagonal global wire. All cells on the same

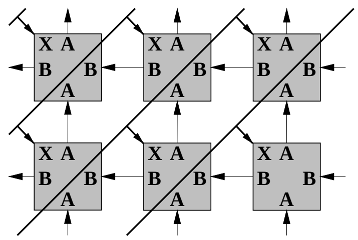

Figure 1: *Lattice Interconnect Pattern*

positive diagonal connect to the same global interconnect and therefore have the same value on their x input pins. (See Fig. 1) In our example, cells of a 2x2 lattice contain a pair of AND gates which feed an EXOR gate which, in turn, produces a signal for outputs out_A and out_B. This structure can be described by three important equations,

$$out_A = out_B = x \cdot in_A \oplus \overline{x} \cdot in_B \tag{3}$$
$$out_A = out_B = x \cdot in_A \oplus in_B \tag{4}$$
$$out_A = out_B = in_A \oplus \overline{x} \cdot in_B \tag{5}$$

Eq. 3 through Eq. 5 are equivalent to Shannon, positive Davio and negative Davio expansions, respectively. Input x corresponds to the variable of decomposition, and in_A and in_B correspond to the logic functions of negative cofactors, positive cofactors, or the exclusive sum of cofactors, depending on which type of expansion is used. In addition, cells can implement various symmetry polarities by using normal or inverted versions of any input.

We make several assumptions to simplify our example. Functions are represented by ordered Kronecker FDDs (OKFDDs) [12] rather than pseudo-Kronecker diagrams. In addition, we consider only symmetry pairs rather than symmetry sets of greater cardinality. Also, only a subset of possible symmetry polarities are inspected. Given these simplifications, there are three general cases in which a polarized pseudo-Kronecker symmetry among OKFDD nodes is useful in mapping functions to 2x2 lattices and other regular layouts. These symmetry types are called *single-variable symmetry*, *double-variable symmetry* and *constant symmetry*.

To describe symmetry types, we introduce an EXOR-ternary diagram (ETDD) [12] of function $f(a, b, \cdots)$, shown in Fig. 2. The ETDD shows all unique subfunctions of f that are generated by Shannon and Davio expansions with respect to variables a and b. Only the first two levels are shown. Subsequent levels of the tree are absent for the sake of clarity.

Each edge is labeled with 0, 1, or 2, representing negative cofactors f_0, positive cofactors f_1, and exclusive sums of cofactors $f_2 = (f_0 \oplus f_1)$, respectively.

In terms of these subfunctions, Eq. 3 through Eq. 5 can be rewritten:

$$f = xf_1 \oplus \overline{x}f_0 \qquad f = xf_2 \oplus f_0 \qquad f = \overline{x}f_2 \oplus f_1$$

Subfunction f_0 is needed for Shannon and negative Davio, f_1 is needed for Shannon and positive Davio, and f_2 is needed for positive and negative Davio expansions. Subfunctions of f are labeled with expansion variable names and with the type of expansion associated with each variable. For example, subfunction $f_{a_0b_2}$ is obtained by expanding f by a using either Shannon or negative Davio to produce f_{a_0}, then expanding this function by b using either negative or positive Davio expansion.

For a *single-variable symmetry pair*, two subfunctions adjacent to the same parent node are equal and can be combined into a single node. Since cells of a 2x2 lattice can invert any input, we can also combine pairs of nodes which represent two subfunctions that differ only in polarity. If nodes are combined after an inversion, the pair of expansion variables associated with such nodes is a *skew single-variable symmetry pair.* Fig. 3a shows a single-variable symmetry pair and its realization in a 2x2 lattice.

Of the nine possible subfunctions of a two-variable expansion, there are three sets of functions that are children of the same parent node. Within each set, there are three different function combinations possible. Therefore, there are nine instances of single-variable symmetric PPKS which use only non-inverting edges and nine instances of skew single-variable symmetric PPKS.

For a *double-variable symmetry pair*, two subfunctions nodes that are not adjacent to the same parent node are combined. Fig. 3b shows a double variable symmetry pair and its realization in a 2x2 lattice. Each subfunction can combine with any of six other functions that do not share its parent node. There are twenty-seven instances of double-variable symmetries which use no inversions and twenty-seven instances of double-variable symmetries in which one subfunction is inverted, called *skew double-variable symmetries.*

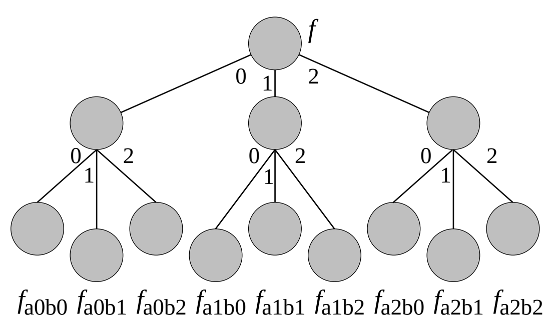

Figure 2: *Nine EXOR Ternary Expansion Subfunctions with Respect to Variables a and b*

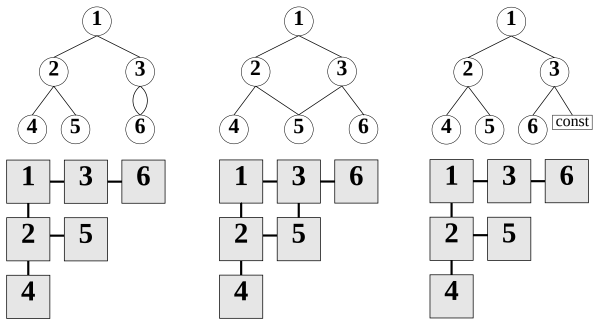

Figure 3: *a) 1-Variable, b) 2-Variable and c) Constant PPKS and their Lattice Realizations*

A special case of single-variable symmetry occurs when a subfunction is constant, called *constant symmetry*. All functions generated from this subfunction using any expansion or variable are constant, equal and therefore single-variable symmetric. The cells of our lattice can generate constants internally, which saves a level of expansion otherwise required for single-variable symmetry to merge equivalent subfunctions. In the example shown in Fig. 3c, node three requires only a single external signal because the constant can be generated within a cell of our lattice.

Relationships among symmetries are used when analyzing functions for symmetry to reduce the number of checks. Positive or negative verification of some of symmetries precludes checking another ones. It may be productive to create various smaller but "minimal" sets of symmetries. We are working on defining minimum sets of independent minimal symmetries to be checked for various binary and multi-valued applications.

3.1 Example of Finding Symmetry Pairs

This section presents an example of finding polarized pseudo-Kronecker symmetry pairs and illustrates the interdependence among such pairs. Given function $f(a,b,c,d,e) = \bar{a}\bar{c}\,\overline{d} \oplus \overline{a}\overline{b} \oplus c\overline{d} \oplus \overline{a}cde \oplus a\overline{b}c$, symmetries between variables a and b are found by first calculating the three possible subfunctions of variable a and the nine possible subfunctions of variable b:

$$
\begin{aligned}
f_{a_0} &= \bar{c}\,\overline{d} \oplus \overline{b} \oplus c\overline{d} \oplus cde \\
f_{a_1} &= c\overline{d} \oplus \overline{b}c \\
f_{a_2} &= \overline{d} \oplus \overline{b}\,\bar{c} \oplus cde \oplus c\overline{d} = \overline{d}\,\bar{c} \oplus \overline{b}\,\bar{c} \oplus cde
\end{aligned}
$$

$$
\begin{array}{lll}
f_{a_0 b_0} = \bar{c}\,\overline{d} \oplus 1 \oplus c\overline{d} \oplus cde & f_{a_0 b_1} = \bar{c}\,\overline{d} \oplus c\overline{d} \oplus cde & f_{a_0 b_2} = 1 \\
f_{a_1 b_0} = cd & f_{a_1 b_1} = c\overline{d} & f_{a_1 b_2} = c \\
f_{a_2 b_0} = \overline{d} \oplus \bar{c} \oplus cde \oplus c\overline{d} & f_{a_2 b_1} = \overline{d} \oplus cde \oplus c\overline{d} & f_{a_2 b_2} = \bar{c}
\end{array}
$$

As this example shows, there are often several types of symmetries for the same variable pair. Subfunctions $f_{a_0b_0}$ and $f_{a_0b_1}$ are mutual negations and can be combined in a lattice using an inverter. Since these subfunctions are children of the same function, f_{a_0}, this relationship is a single-variable skew symmetry between a and b. There is also a constant symmetry between a and b because one of their nine subfunctions, $f_{a_0b_2}$, is constant. The choice of which symmetries between variables are used in building lattices influences which expansions must occur at each level of a diagram.

For example, to construct a 2x2 lattice using constant symmetry for a and b, we need to produce subfunction $f_{a_0b_2}$. We can first expand by a using Shannon or positive Davio to produce f_{a_0} and f_{a_1} or f_{a_0} and f_{a_2}, respectively. If we then expand f_{a_0} by b using positive Davio, we produce $f_{a_0b_2}$ and $f_{a_0b_0}$. Alternately, if we expand by negative Davio, we produce $f_{a_0b_2}$ and $f_{a_0b_1}$. Any combination of these four expansions generates the desired subfunction.

For a 2x2 lattice which uses the single-variable skew symmetry described above, we need subfunctions $f_{a_0b_0}$ and $f_{a_0b_1}$. We first expand by a using Shannon or positive Davio and then by b using Shannon. These expansions produce $f_{a_0b_0}$ and $f_{a_0b_1}$ regardless of how we expand our first level, then $f_{a_1b_0}$ and $f_{a_1b_1}$ or $f_{a_2b_0}$ and $f_{a_2b_1}$ for Shannon or positive Davio in the first level, respectively. We negate one subfunction in order to combine lattice nodes.

4 Lattice Synthesis and Layout using PPKS

The final goal of our synthesis approach is to find a input variable ordering such that a function, when expanded by variables in the given sequence, fits a lattice pattern and occupies as small a region of cells as possible. In practice, our results do not exactly conform to a lattice pattern.

To simplify our synthesis method, we require that the order of variables used to expand a 2x2 lattice is constant for all branches of a tree, i.e. lattices must be ordered Kronecker FDDs. For this kind of representation, there is a relation between function variables and diagram levels. Since every function can be realized in a lattice with a finite number of variable repetitions,[13,14] one possible metric for quality of results in the total number of levels in a decision diagram. Using this gauge, the best variable order is defined as an order with the fewest number of repeated variables.

We use a *symmetry compatibility graph* (G_{SYM}) to record polarized pseudo-Kronecker symmetry pairs. A node of G_{SYM} represents an expansion (Shannon, positive or negative Davio) with respect to an input variable. For an input a, for example, G_{SYM} contains nodes a-S, a-pD, and a-nD, representing a Shannon expansion by a, a positive Davio expansion by a and a negative Davio expansion by a, respectively. Edges between nodes indicate symmetries.

We find PPKS by comparing ETDD subfunctions. For each possible coupling of input variables, the first two levels of an ETDD are constructed using each variable pair at the beginning the diagram's variable ordering. The nine resulting subfunctions are compared to check for PPKS. Although some pseudo-Kronecker symmetries overlap, redundant subfunction comparisons are not performed.

After all variable pairs have been considered and G_{SYM} has been constructed, we analyze the graph for areas of connectivity which suggest good variable orderings. Nodes of cliques and near-cliques, for example, can indicate good partial sequences of variables and expansions. One characteristic of the graph which makes traditional clique identification difficult is the variable and expansion relationships among nodes. If a clique of G_{SYM} contains node a-S, it is not desirable for the clique to also contain a-pD. This would mean that variable a is used twice for expansion, once using Shannon and once using positive Davio.

We use good variable sequences identified from G_{SYM} analysis at the start of the total variable order for the 2x2 lattice. This insures that the top levels of the lattice will conform to a lattice structure without variable repetition. Remaining variables can be repeated, as necessary, to create the lower levels of the lattice. Several methods exist to minimize repetitions.[8,13,14]

We have implemented a variant of the above method for OKFDDs. A diagrams is created with variable and expansion orders found as above, but the resulting OKFDD does not necessarily conform to 2x2 lattice structure. It can, however, map to grids of greater connectivity such as a 3x3 lattice, or some other structure with short and regular connections. (See Fig. 6)

5 Results

We tested our approach with benchmarks taken from the *Wright Laboratory Data Set*. This benchmark set was selected because circuits of this set have only one output and fewer than nine inputs. Table 1 shows numbers of standard symmetry and PPKS relationships between non-vacuous variable pairs for each benchmark. While all standard symmetries were detected by PPKS tests, 26 of 54 circuits had additional symmetries, not detected during standard symmetry tests. The more instances of symmetry, the greater the probability that subfunctions may be combined to form lattice diagrams without costly variable repetition.

Fig. 4 shows a symmetry compatibility graph for benchmark rnd-m10. From this graph, we chose a variable ordering which maximizes the number of symmetries between sequentially ordered variables. A OKFDD lattice built from this ordering is shown in Fig. 5. In Fig. 6, the diagram has been mapped

Table 1: Occurrence of Standard vs. Polarized Pseudo-Kronecker Symmetries

Circuit	Variable Pairs	Stand. Symm.	PPKS	Circuit	Variable Pairs	Stand. Symm.	PPKS
substr2	28	0	8	kdd8	3	1	3
subtraction1	28	4	4	kdd9	15	7	9
subtraction3	1	1	1	kdd10	15	3	11
and_or_chain8	28	2	28	kdd5	28	1	16
ch15f0	15	0	0	kdd6	6	6	6
ch176f0	3	1	3	kdd7	28	4	4
ch177f0	1	1	1	mux8	15	0	14
ch22f0	15	6	6	kdd1	3	3	3
ch30f0	15	1	2	pal	28	4	4
ch47f0	15	0	1	kdd2	10	4	10
ch52f4	28	0	3	kdd3	6	3	6
ch70f3	28	1	18	kdd4	0	0	0
ch74f1	28	0	9	primes	28	0	7
ch83f2	28	0	3	primes8	28	0	7
ch8f0	15	2	5	rnd1	28	0	0
contains_4_ones	28	28	28	rnd2	28	0	0
greater_than	28	4	4	rnd3	28	0	0
interval1	28	0	0	rnd_m1	28	28	28
interval2	28	0	0	rnd_m10	28	0	8
or_and_chain8	28	2	28	rnd_m25	28	0	0
majority_gate	28	28	28	rnd_m5	28	1	22
pal_dbl_output	21	0	0	rnd_m50	28	0	0
pal_output	28	0	0	rndvv36	15	0	0
parity	28	28	28	substr1	28	0	4
monkish1	6	2	6	add0	28	4	4
monkish2	28	8	8	add2	15	3	11
monkish3	10	4	10	add4	1	1	1

to a square grid of hardware cells. All cells which require the same input variable occupy the same grid diagonal. While a few wires span multiple cells, most cells connect to their neighbors by abutment.

Our method for variable and expansion ordering can be used to create lattice diagrams which exactly conform to a 2x2 grid, or OKFDDs which have subfunction sharing but do not necessarily fit a 2x2 pattern. For lattice diagrams, variables must be repeated for typical circuits to obtain correct connectivity. OKFDDs, which can have little or no repetitions, can have fewer levels of logic, but do not pack as well into a regular geometry.

References

1. S.B. Akers, "A Rectangular Logic Array," *IEEE TC.*, v. C-21, pp.848-857, Aug. 1972.
2. C.R. Edwards, S.L. Hurst, "A Digital Synthesis Procedure under Function Symmetries and Mapping Methods," *IEEE Trans. Comp.*, v. 27, No. 11, pp. 985-997, Nov. 1978.
3. Ch. Tsai and M. Marek-Sadowska, "Detecting Symmetric Variables in Boolean Functions using Generalized Reed-Muller Forms," *Proc. ISCAS*, pp. 1,287-1,290, 1994.
4. M. Perkowski, "A Fundamental Theorem for Exor Circuits," *Proc. RM'93*, pp. 52-60.
5. M.A. Perkowski, L. Jozwiak, R. Drechsler, and B. Falkowski, "Ordered and Shared, Linearly-Independent, Variable-Pair Decision Diagrams," *Proc. ICICS'97*, Singapur, Sept. pp. 10-12, 1997.

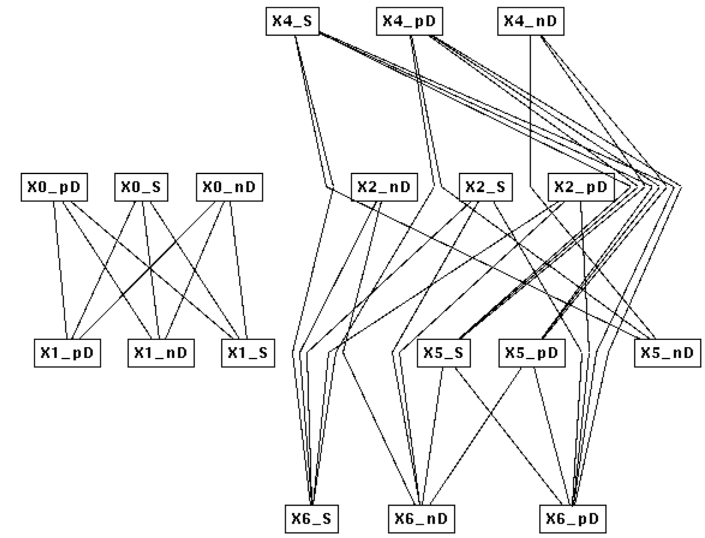

Figure 4: G_{SYM} *Graph for rnd_m10*

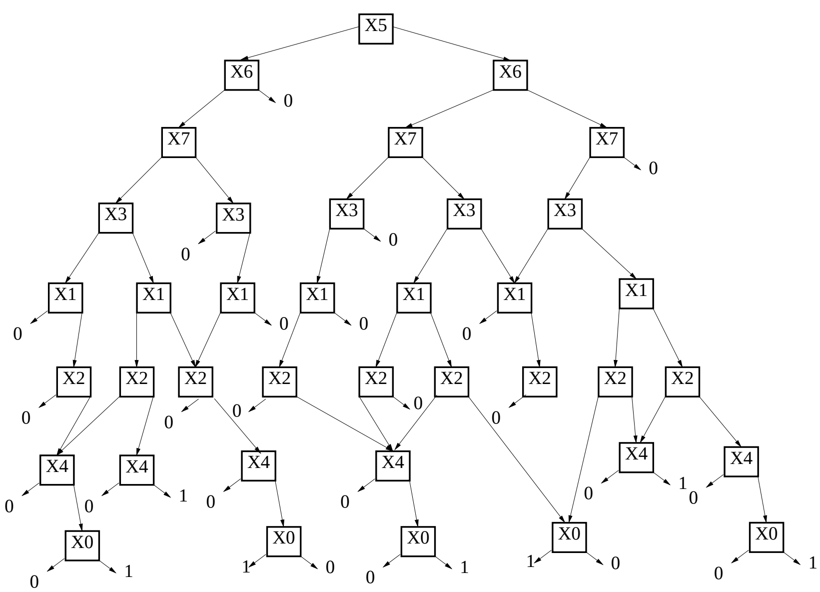

Figure 5: *OKFDD for rnd_m10*

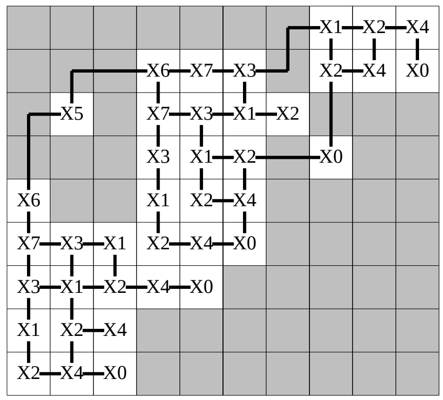

Figure 6: *Layout of rnd_m10 OKFDD*

6. M. Perkowski, A.Sarabi, F. Beyl, "XOR Canonical Forms of Switching Functions," *Proc. RM'93*, pp. 27-32.
7. M. Perkowski, A.Sarabi, F. Beyl, "Fundamental Theorems and Families of Forms for Binary and Multiple-Valued Linearly Independent Logic," *Proc. RM'95*, pp. 288-299.
8. M.A. Perkowski, E. Pierzchala, and R. Drechsler, "Layout-Driven Synthesis for Submicron Technology: Mapping Expansions to Regular Lattices," *Proc. ISIC-97,* Singapur, Sept. pp. 10-12, 1997.
9. M.A. Perkowski, E. Pierzchala, and R. Drechsler, "Ternary and Quaternary Lattice Diagrams for Linearly-Independent Logic, Multiple-Valued Logic, and Analog Synthesis," *Proc. ICICS'97,* Singapur, Sept. pp. 9-12, 1997.
10. R. Drechsler, "Pseudo Kronecker Expressions for Symmetric Functions," *Proc. VLSI Design Conference*, 1997.
11. P. Ho, M. A. Perkowski, "Free Kronecker Decision Diagrams and their Application to ATMEL 6000 FPGA Mapping," *Proc. Euro-DAC'94/VHDL'94*, pp. 8 - 13, Sept. pp. 19-23, 1994, Grenoble France.
12. T. Sasao, "Logic Synthesis and Optimization," *Kluwer Academic Publishers,* 1993.
13. M. Chrzanowska-Jeske, Z. Wang and Y. Xu, "A Regular Representation for Mapping to Fine-Grain, Locally-Connected FPGAs," *Proc. ISCAS'97*.
14. M.A. Perkowski, M. Chrzanowska-Jeske, and Y. Xu, "Lattice Diagrams Using Reed-Muller Logic," *RM'97*, pp. 85-102.