

Structurally synthesized binary decision diagrams

A. Jutman A. Peder J. Raik M. Tombak R. Ubar

Tallinn University of Technology, University of Tartu

Binary graph is an oriented acyclic connected graph with root and two leaves (sinks) – **0** and **1**. Every intermediate node v has two successors: $high(v)$ and $low(v)$.

Binary graphs are a skeleton of binary decision diagrams (BDD): BDD is a binary graph in which intermediate nodes are labelled by propositional variables, the sink **1** is labelled by the truth value 1 and the sink **0** by the truth value 0. We denote the label of the node v by $label(v)$.

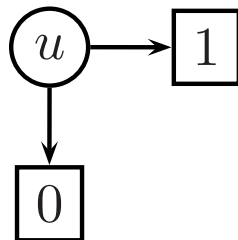
Let D be a decision diagram with variables x_1, \dots, x_n . Every vector $\alpha \in B^n$ activates a path $p(\alpha) = p_0, \dots, p_k$ in D from the root to a terminal node: if $\alpha \vdash \text{label}(v_i)$, then $v_{i+1} = \text{high}(v_i)$ else $v_{i+1} = \text{low}(v_i)$. We call the path an *1-path* (*0-path*), if its last element is the sink $\mathbf{1}$ ($\mathbf{0}$). The Boolean function $f_D(x_1, \dots, x_n)$, represented by D is defined: $f(\alpha) = 1$ iff the path, activated by α , is a 1-path.

Let G and E be two binary graphs and v an intermediate node in G .

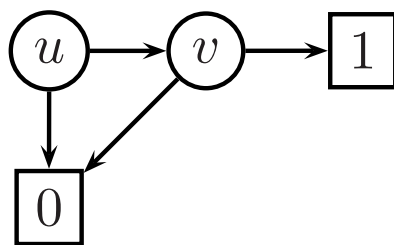
Definition. A *superposition* of E into G instead of v ($G_{v \leftarrow E}$) is a graph, which we receive by deleting v from G and redirecting all edges, pointing to v , to the root of E , all edges of E pointing to sink $\mathbf{1}$ to the node $high(v)$ and all edges pointing to the sink $\mathbf{0}$ to the node $low(v)$.

We define three elementary binary graphs: A, C and D.

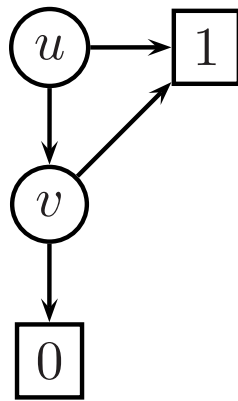
a) A



b) C



c) D



Definition. We define inductively a family of *superpositional graphs*:

1° An elementary graph A , C or D is a superpositional graph.

2° If G and E are superpositional graphs and v is a node in G , then a superposition of E instead of v into G , $G_{v \leftarrow E}$, is an superpositional graph.

Definition. *high-path* (*low-path*) is the path $p = (v_0, \dots, v_k)$, where $v_{i+1} = \mathit{high}(v_i)$ ($v_{i+1} = \mathit{low}(v_i)$) for every $i : 0 \leq i < k$.

Definition. Node v in the superpositional graph is a *final* node if $\mathit{high}(v) = \mathbf{1}$ and $\mathit{low}(v) = \mathbf{0}$.

Theorem 1 *If S is an superpositional graph, then:*

- 1. S is a planar graph.*
- 2. There exists a high-path from every node to the sink **1**.*
- 3. There exists a low-path from every node to the sink **0**.*
- 4. S has exactly one final node.*
- 5. There exists a directed path through all intermediate nodes.*
- 6. For every pair of intermediate nodes u, z there exists a directed path from u to z or from z to u .*

Definition. *Structurally synthesized binary decision diagram* for a formula F , $\mathcal{D}(F)$, is a superpositional graph, defined inductively according to the structure of F :

1° If F is a literal l , then $\mathcal{D}(F)$ is a graph A , where the root is labelled by l .

2° If $F = P \& R$ then $\mathcal{D}(F)$ is a graph $C_{u \leftarrow \mathcal{D}(P), v \leftarrow \mathcal{D}(R)}$

3° If $F = P \vee R$ then $\mathcal{D}(F)$ is a graph $D_{u \leftarrow \mathcal{D}(P), v \leftarrow \mathcal{D}(R)}$

Theorem 2 *Propositional formula F and its SSBDD $\mathcal{D}(F)$ are representing the same Boolean function.*

Not every path of the $\mathcal{D}(F)$'s superposition graph can be activated. Therefore we define a *consistent* path of the SSBDD $\mathcal{D}(F(x_1, \dots, x_n))$ as a path which can be activated by some assignment $(\alpha_1, \dots, \alpha_n) \in B^n$.

The *satisfiability problem* for SSBDD can be formulated: does there exist a consistent path from the root to the sink **1**? It is obvious, that the number of intermediate nodes of the SSBDD $\mathcal{D}(F)$ is the number of occurrences of variables in the formula F . It follows, that the satisfiability problem is NP-complete and counting the number of true assignments is $\#P$ complete for SSBDD representation of Boolean functions.

It is obvious, that the conjunction of literals in a 1-path in $\mathcal{D}(F)$ is a term of the DNF for F . Next theorem (first proved in [?]) says, that we can remove from the terms of the DNF all literals which are false for the corresponding assignment.

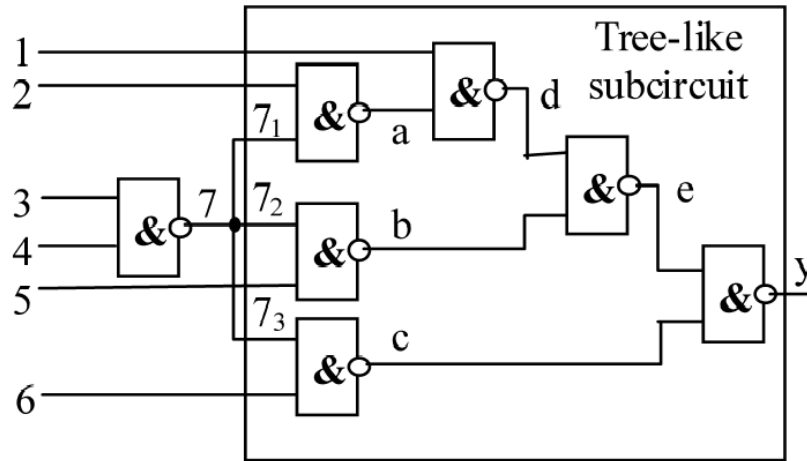
Definition *Positive term* of the path $p = (v_1, \dots, v_l)$ is a conjunction of literals of the nodes v_i , followed by the node $high(v_i)$:

$$\bigwedge_{\substack{0 \leq i < l \\ v_{i+1} = high(v_i)}} label(v_i)$$

.

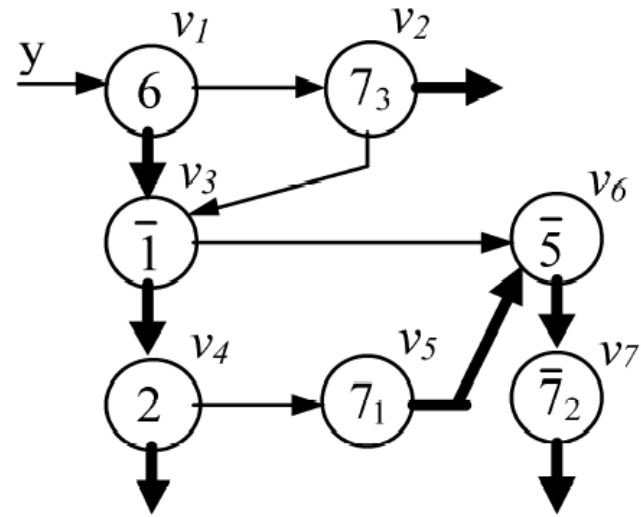
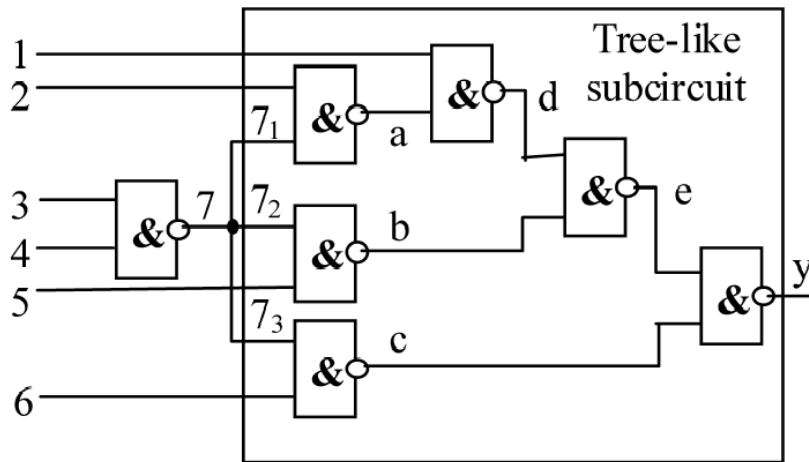
Theorem 3 *Let F be a propositional formula and $\mathcal{D}(F)$ its SSBDD. Disjunction of positive terms of all 1-paths of $\mathcal{D}(F)$ is a DNF for a propositional formula F .*

Logic circuit and its EPF



$$\begin{aligned}
 y &= \overline{c_y e_y} = \overline{c_y} \vee \overline{e_y} = x_{6,c,y} x_{7_3,c,y} \vee d_{e,y} b_{e,y} = \\
 &= x_{6,c,y} x_{7_3,c,y} \vee (\overline{x_{1,d,e,y}} \vee \overline{a_{d,e,y}}) (\overline{x_{5,b,e,y}} \vee \overline{x_{7_2,b,e,y}}) = \\
 &= x_{6,c,y} x_{7_3,c,y} \vee (\overline{x_{1,d,e,y}} \vee x_{2,a,d,e,y} x_{7_1,a,d,e,y}) (\overline{x_{5,b,e,y}}) \vee \overline{x_{7_2,b,e,y}} = \\
 &= \boxed{x_6 x_{7_3} \vee (\overline{x_1} \vee x_2 x_{7_1}) (\overline{x_5}) \vee \overline{x_{7_2}}}
 \end{aligned}$$

Logic circuit and its SSBDD



$$x_6 x_{7_3} \vee (\overline{x_1} \vee x_2 x_{7_1}) (\overline{x_5}) \vee \overline{x_{7_2}}$$

Fault simulation. Stuck-at fault model

Stuck-at fault model:

1. Only one circuit line is faulty.
2. The faulty line is permanently stuck to 0 or 1.

Stuck-at fault simulation for a circuit line l is equivalent to calculating the Boolean derivative for variable x_l .



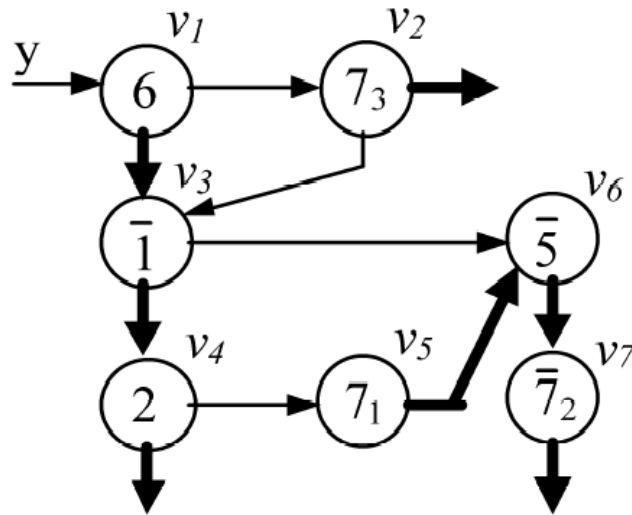
Fault simulation. SSBDD model

We say that fault stuck-at $\neg\alpha_i$ in node v_j , where $label(v_j)$ is x_i , is covered by a vector $\alpha_1, \dots, \alpha_i, \dots, \alpha_n$ iff all of the three conditions are satisfied:

1. There exists an activated path from the root node to v_j .
2. There exists an activated path from $high(v_j)$ to sink **1**.
3. There exists an activated path from $low(v_j)$ to sink **0**.



Fault simulation. SSBDD model



$$\alpha = (1, 0, 0, 1, 0, 0)$$

(i.e. $x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 0, x_6 = 0$)

Fault simulation. SSBDD model

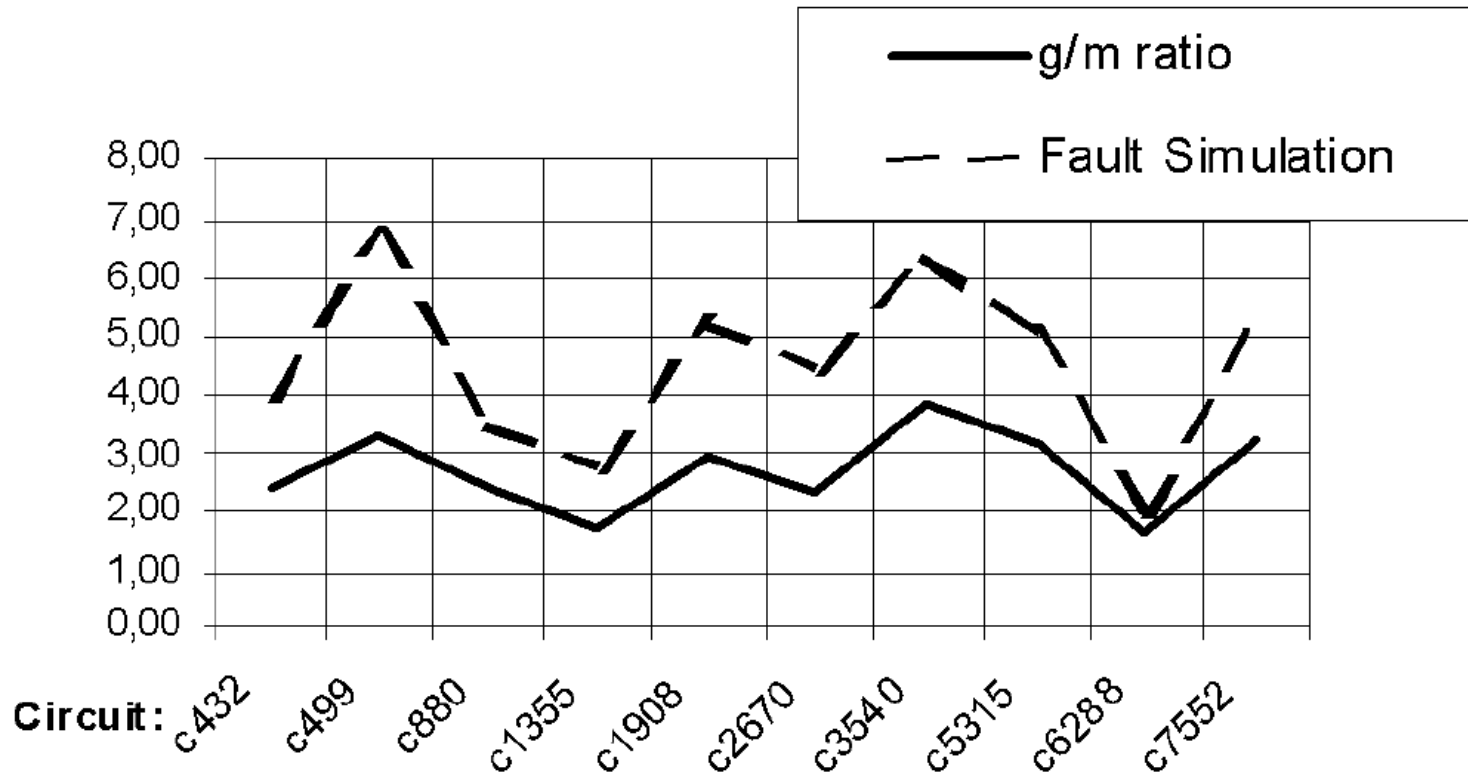
Number of collapsed and SSBDD faults in ISCAS'85

circuit	uncollapsed	collapsed	SSBDD
c880	1550	942	994
c1355	2194	1574	1618
c1908	2788	1879	1732
c2670	4150	2747	2626
c3540	5568	3428	3296
c5315	8638	5350	5424
c6288	9728	7744	7744
c7552	11590	7550	7104



Fault simulation. SSBDD model

Correlation between speed up and gates/subcircuits ratio



Conclusions

- A special case of BDDs called Structurally Synthesized BDDs (SSBDD) was proposed.
- SSBDD representations for both, Boolean formulae and digital logic circuits were discussed.
- The paper showed that the special properties of SSBDDs are highly useful in fault modeling and simulation for digital circuits.
- Experiments showed a speed-up of 2 to 7 times in comparison to traditional gate-level descriptions.

