# Path Delay Fault Testable KFDD Circuits with Polynomial Test Pattern Generation

Martha Schnieber
University of Bremen
Bremen, Germany
schnieber@uni-bremen.de

Rolf Drechsler
University of Bremen/DFKI
Bremen, Germany
drechsler@uni-bremen.de

***Abstract*—Nowadays, *Integrated Circuits* (ICs) are omnipresent in our everyday life, including in safety-critical systems demanding a high test coverage. However, general circuit designs require cost-intensive exponential *Automatic Test Pattern Generation* (ATPG) algorithms, such as SAT-based ATPG. Additionally, general circuit designs are not always fully testable, therefore preventing 100% test coverage. Previous research has shown that circuits derived from *Binary Decision Diagrams* (BDDs) are fully testable under the *Cellular Fault Model* (CFM), the *Stuck-At Fault Model* (SAFM) and the *Path Delay Fault Model* (PDFM), where the test pattern generation can be carried out within polynomial resources. Nevertheless, for various functions, *Kronecker Functional Decision Diagrams* (KFDDs) can be exponentially smaller than BDDs, thus providing significantly smaller circuits. However, the original KFDD circuits are not fully testable and require exponential ATPG. In this paper, we therefore present novel approaches to derive circuits from KFDDs that are fully testable under CFM, SAFM and PDFM, while also guaranteeing polynomial upper bounds for the test pattern generation. The full coverage and efficient test pattern generation are further supported by our experimental evaluation.**

## I. INTRODUCTION

In safety-critical systems, the correctness of ICs is crucial. However, manufacturing faults can result in erroneous chips, leading to the need for thorough testing after production. Circuits which are specifically designed for testability can ensure full testability under different fault models to guarantee correctness. However, the test pattern generation is generally an exponential problem, as circuits require exponential algorithms for the ATPG, e.g. using SAT. Therefore, the generation of a full test set is cost-intensive.

A circuit type for which *Polynomial Test Pattern Generation* (PTPG) and full testability under CFM, SAFM and PDFM have previously been shown is circuits derived from *Binary Decision Diagrams* (BDDs) [1], [2]. In this context, it has been researched that BDD circuits are fully testable under SAFM and PDFM, where the test pattern generation can be carried out within polynomial time regarding the underlying BDD size. However, as BDDs can have an exponential size given the number of inputs, the representation of functions using BDDs can be infeasible, while it has been proven that KFDDs can be exponentially smaller than BDDs for various functions [3]. Consequently, circuits derived from KFDDs can realize these functions with polynomial area, while the respective BDD-based circuit grows exponentially.

However, the original design of KFDD circuits is not fully testable and requires exponential ATPG [4]. Recent research proposed an approach for the synthesis of KFDD circuits that provides full testability with PTPG under CFM and SAFM by introducing a linear number of toggle inputs [5]. However, the resulting KFDD circuits are not fully testable under PDFM. While the testing of e.g. stuck-at faults is widely adapted, testing static fault models within a safety-critical system isn't sufficient and dynamic fault models such as PDFM have to be considered as well to avoid any delays and timing-related faults. However, the testing of path delay faults is hindered by the high test generation cost [6].

In this paper, we propose three synthesis approaches for KFDD circuits that provide full testability with PTPG under PDFM with strong robust tests, which are not invalidated by any hazards or races. All three designs are additionally fully testable with PTPG under CFM and SAFM and can therefore be implemented for safety-critical modules or subcircuits which require full testability. Our experimental evaluation further strengthens the theoretical results on the testability and test set generation time for the different proposed methods.

## II. PRELIMINARIES

### A. KFDDs

Similar to BDDs [7], KFDDs [8] are defined as directed acyclic graphs, consisting of nonterminal nodes, each representing a Boolean function, and terminal nodes representing the constant functions 0 and 1. While BDDs only use the *Shannon* (S) decomposition for the function representation, KFDDs additionally use the *positive Davio* (pD) and *negative Davio* (nD) decomposition types, where one decomposition type is chosen for each variable. Let $f$ be a function, where the variable $x_i$ is replaced by the constant 0 in the function $f_i^0$ and by 1 in the function $f_i^1$. Furthermore, let $f_i^2 = f_i^0 \oplus f_i^1$. The three decomposition types are defined as follows:

$$\begin{aligned} S: &\quad f = \overline{x}_i f_i^0 + x_i f_i^1 \\ pD: &\quad f = f_i^0 \oplus x_i f_i^2 \\ nD: &\quad f = f_i^1 \oplus \overline{x}_i f_i^2 \end{aligned}$$

---

Parts of this work have been supported by DFG within the Reinhart Koselleck Project PolyVer: Polynomial Verification of Electronic Circuits (DR 287/36-1).

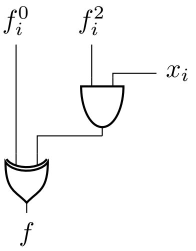


Fig. 1: Subcircuit for pD nodes

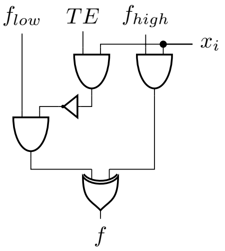


Fig. 2: pD subcircuit with test enable

We denote $f_{low}$ and $f_{high}$ as the children of the node for $f$, i.e. $f_{low} = f_i^0$ for S and pD, $f_{low} = f_i^1$ for nD, $f_{high} = f_i^1$ for S and $f_{high} = f_i^2$ for pD and nD.

KFDDs can be significantly smaller than BDDs for various functions [3]. However, while the Boolean operations AND, OR, NOT and XOR have a polynomial runtime on BDDs, the runtime of the AND and OR operations generally has an exponential complexity on KFDDs.

### *B. KFDD Circuits*

To derive a circuit from a KFDD, the terminal nodes are turned into constant values 0 and 1, the variables $x_i$ are transformed into PIs and the root node is connected to the PO. Each node of the KFDD is realized with a subcircuit implementing the respective decomposition type. For the S decomposition, a multiplexer is used. For pD, an XOR gate and an AND gate are needed, as shown in Figure 1, while a subcircuit for nD additionally requires a NOT gate [9].

In this paper, several libraries are considered for the circuit implementation. We denote $C_{STD}$ as the circuit over the library $STD = \{AND, OR, NOT\}$, $C_{XOR}$ as the circuit over the library $STD \cup \{XOR\}$ and $C_{XORMUX}$ as the circuit over $STD \cup \{XOR, MUX\}$.

## III. FULLY TESTABLE SYNTHESIS

The test pattern generation for KFDD circuits faces multiple problems. Firstly, the test pattern generation for the original KFDD circuits realized with the aforementioned pD and nD subcircuits generally requires exponential effort. In circuit designs derived from BDDs and KFDDs, an AND operation can be used to compute an assignment that applies specific values to the inputs of a subcircuit, e.g. $\overline{f_{low}} \cdot f_{high}$ for the input combination 01. The propagation of the fault is then always possible, as a fault can be propagated through all subcircuits. However, the AND operation is exponential for KFDDs, therefore resulting in exponential runtimes.

Furthermore, not all input combinations are possible at all subcircuits, therefore leading to non-testable faults. For BDD circuits, a single additional toggle input $t$ has been introduced in [2], which inverts all signal values within the circuit if $t = 1$, therefore allowing all input combinations at each subcircuit and leading to full testability. The same concept is however not applicable to KFDD circuits, as a single toggle input doesn't invert all signals within a KFDD circuit.

Lastly, if a pD node is implemented as shown in Figure 1, paths going through $f_i^2$ are not guaranteed to be testable. With a rising or falling flank at $f_i^2$, the off-path input $f_i^0$ is potentially unstable and can feature another rising or falling flank. Thus, a robust path delay fault test for this path only exists if $f_i^0$ can be set to a stable value.

### *A. Test Enable*

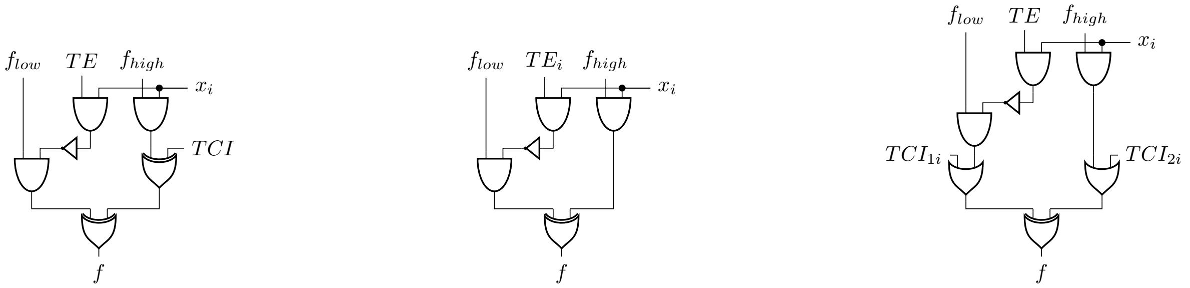
We introduce a test enable input which, during testing, effectively reduces the pD and nD subcircuits into multiplexers and therefore turns the KFDD circuit into a BDD circuit. This allows the testability properties of BDD circuits while utilizing the benefits of KFDDs. The resulting pD subcircuit is shown in Figure 2. If the test enable input $TE$ is set to 0, the subcircuit implements the pD decomposition, while providing a multiplexer if $TE = 1$. Using the proposed method, test patterns satisfying specific input combinations at a subcircuit can be computed by AND operations on BDDs instead of KFDDs, therefore allowing PTPG. Furthermore, a single toggle input, as introduced in [2], is used to ensure that all signals within the circuit can be inverted during testing, therefore enabling all combinations for $f_{low}$ and $f_{high}$ if they are neither equal nor complements of each other. Lastly, the off-path input of the XOR gate is set to a stable 0 while the paths going through $f_i^2$ are tested, therefore enabling full PDFM testability with strong robust tests for $C_{XOR}$.

The test enable $TE$ can be realized as either a single input or a leveled input, i.e. one $TE_i$ is used for each variable $x_i$, which can e.g. be realized with a scan chain. Using a leveled $TE_i$ increases the hardware overhead, but ensures path delay fault testability along the paths starting at the test enable inputs. When using a single $TE$, a rising or falling flank on $TE$ can result in unstable off-path inputs due to reconvergences.

### *B. Test Control Inputs*

However, some problems remain after the introduction of the test enable, which can be solved with test control inputs.

When using a single $TE$ input, the input combination 11 is not possible at the XOR gate if $TE = 1$, as one input is always set to 0. With $TE = 0$ however, the BDD functionality is disabled and thus, the input combination 11 may also not be possible and the test pattern generation has an exponential complexity. When using a leveled $TE_i$ however, the inputs allow to dynamically switch back to the Davio decomposition on each level of the KFDD. Therefore, the input



(a) pD subcircuit for method 1 (b) pD subcircuit for method 2 (c) pD subcircuit for method 3

Fig. 3: pD subcircuits for the proposed designs

TABLE I: Comparison between different KFDD circuits

| Circuit | #inputs | CFM | SAFM | PDFM |
|---|---|---|---|---|
| KFDD circuit | $n$ | None | None | None |
| Toggle circuit [5] | $2n$ | $C_{XORMUX}$ | $C_{STD}$ | None |
| Method 1 | $n+3$ | $C_{XORMUX}$ | $C_{STD}$ | $C_{XOR}$[1] |
| Method 2 | $\leq 3n+1$ | $C_{XORMUX}$ | $C_{STD}$ | $C_{XOR}$ |
| Method 3 | $\leq 3n+2$ | $C_{XORMUX}$ | $C_{STD}$ | $C_{STD}$[1] |

[1] without paths starting at test signals

combination 11 for the XOR gate on level $i$ of the KFDD can be accomplished with $TE_i = 0$, while the BDD functionality remains enabled for the remainder of the KFDD. When using a single test enable $TE$, a single test control input $TCI$ can be used, which sets one of the inputs of the XOR gates to 1 if activated. As 10 and 01 are possible as inputs to the XOR gate, the test control input ensures that 11 is possible as well.

Furthermore, specific input combinations at the inputs of a subcircuit are not possible if it holds that $f_{low} = f_{high}$, which allows only the input combinations 00 and 11, or $f_{low} = \overline{f_{high}}$, allowing only 01 and 10. Thus, neither a single $TE$ nor a leveled $TE_i$ allows full testability under CFM or SAFM in these cases. Therefore, an additional test control input is needed for a variable $x_i$ if such a case exists. To enable all input combinations for the corresponding subcircuit, the test control input inverts one of the subcircuit's inputs if activated.

Ensuring full path delay fault testability for $C_{STD}$ requires that the off-path inputs of the XOR gate can also be set to a stable 1, which is not possible with the design given in Figure 2. Further test control inputs can be added which ensure that each input of the XOR gate within the pD and nD subcircuits can be set to a stable 1, therefore allowing both a stable 0 and a stable 1 at each input of the XOR gate.

## *C. Proposed KFDD Circuit Designs*

We propose multiple specific designs to implement the presented approach which provide trade-offs between hardware overhead and testability properties. Figure 3 shows the pD subcircuits of the proposed designs, while Table I summarizes the differences between the methods and previously known approaches, where $n$ is the number of inputs of the desired function and "None" means that no full testability and no PTPG can be guaranteed under the given fault model.

*1) Method 1:* The first method introduces a single test enable input $TE$ and a single test control input $TCI$, where the resulting pD subcircuit is shown in Figure 3a. With the addition of the single toggle input $t$, a total number of $n+3$ inputs are needed. The test control input enables the input combination 11 at the XOR gate and can also be used as a test control input for subcircuits where $f_{low} = f_{high}$ or $f_{low} = \overline{f_{high}}$ to allow all input combinations. This method provides full testability under CFM for $C_{XORMUX}$ and under SAFM for $C_{STD}$, as well as full testability under PDFM for $C_{XOR}$, excluding the paths starting from $TE$ and $TCI$.

*2) Method 2:* The second method, as shown in Figure 3b, introduces one test enable input $TE_i$ for every variable $x_i$ which is decomposed with pD or nD. Furthermore, an additional test control input is introduced for each variable $x_i$ for which a subcircuit with $f_{low} = f_{high}$ or $f_{low} = \overline{f_{high}}$ exists, therefore resulting in at most $3n+1$ total inputs including the single toggle input $t$. Using this approach, also the paths starting from the test enable signals are path delay testable for $C_{XOR}$ with strong robust tests. The resulting circuit furthermore provides the same CFM and SAFM testability properties as the first proposed method.

*3) Method 3:* The third method introduces one single test enable $TE$ and at most $2n$ test controls $TCI_{1i}$ and $TCI_{2i}$ as shown in Figure 3c. Including the toggle input $t$, this results in at most $3n+2$ inputs for the whole circuit. As a single test enable is used, the paths starting at $TE$ are not testable. The test control inputs however allow to generate both stable 0 and 1 values at the off-path input of the XOR gate, therefore enabling full path delay testability for all other paths in $C_{STD}$. Additionally, $TCI_{1i}$ and $TCI_{2i}$ can be used to enable the input combination 11 at the XOR gate and also for toggling one of the inputs of the subcircuit if both inputs are equal or complements of each other, therefore providing full testability for $C_{XORMUX}$ under CFM and $C_{STD}$ under SAFM.

## IV. EXPERIMENTAL RESULTS

To evaluate our proposed circuit synthesis, we exemplarily derived KFDD circuits from different benchmarks. For the KFDD computations, the FrEDDY tool was used [10]. The evaluation was executed for the ITC'99 benchmarks b11 and

TABLE II: Results number of gates in $C_{STD}$ and $C_{XOR}$

| Benchmark | Optimized | KFDD | Toggle | M1 | M2 | M3 |
|---|---|---|---|---|---|---|
| b11 | 646 / 590 | 2797 / 1181 | 6542 / 2094 | 7037 / 2996 | 4714 / 2417 | 5834 / 3537 |
| b12 | 1167 / 1125 | 5860 / 3139 | 13266 / 5005 | 16250 / 7707 | 11342 / 6489 | 13641 / 8788 |
| r13 | 683 / 380 | 818 / 386 | 2288 / 680 | 1651 / 773 | 1157 / 652 | 1402 / 897 |
| r14 | 8242 / 3990 | 9444 / 3990 | 25983 / 7162 | 20836 / 9056 | 14153 / 7417 | 17473 / 10737 |
| r15 | 16367 / 6141 | 19900 / 6141 | 50524 / 11842 | 42547 / 15059 | 26880 / 11171 | 34692 / 18983 |

TABLE III: Results SAFM coverage and test generation time

| Benchmark | SAFM Coverage | | | | | | Test Generation Time | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Optimized | KFDD | Toggle | M1 | M2 | M3 | Optimized | KFDD | Toggle | M1 | M2 | M3 |
| b11 | 99.92% | 93.53% | 100% | 100% | 100% | 100% | 2.59s | 8.29s | 3.47s | 2.05s | 2.2s | 2.76s |
| b12 | 100% | 94.55% | 100% | 100% | 100% | 100% | 7.21s | 14.78s | 14.24s | 10.81s | 11.39s | 13.22s |
| r13 | 96.72% | 93.96% | 100% | 100% | 100% | 100% | 9.95s | 0.84s | 0.48s | 0.27s | 0.29s | 0.32s |
| r14 | 99.62% | 97.29% | 100% | 100% | 100% | 100% | 431.86s | 327.89s | 15.77s | 13.04s | 14.13s | 14.85s |
| r15 | 99.77% | 98.24% | 100% | 100% | 100% | 100% | 4766.13s | 3367.1s | 65.96s | 38.2s | 40.92s | 42.2s |

TABLE IV: Results PDFM coverage

| Benchmark | KFDD | Toggle | M1 | M2 | M3 |
|---|---|---|---|---|---|
| b11 | 6.17% | 7.27% | 76.83%[1] | 100% | 89.95%[1] |
| b12 | 5.74% | 7.12% | 86.65%[1] | 100% | 93.87%[1] |
| r13 | 4.03% | 4.84% | 86.27%[1] | 100% | 94.1%[1] |
| r14 | 0.66% | 1.03% | 69.6%[1] | 100% | 87.4%[1] |
| r15 | 0.06% | 0.16% | 71.51%[1] | 100% | 88.69%[1] |

[1] 100% without paths starting at test signals

b12, as well as three random KFDD circuits with 13, 14 and 15 inputs, labeled as the circuits r13, r14 and r15.

Table II shows the number of gates for all considered circuits. The results are shown for the original KFDD circuit [4] (KFDD), the KFDD circuits with toggle inputs [5] (Toggle), as well as all three of our proposed circuit designs, i.e. method 1 (M1), method 2 (M2) and method 3 (M3). Furthermore, the results for an optimized circuit (Optimized) are shown, i.e. either the original ITC'99 benchmark or the random KFDD circuit as optimized by yosys-abc. For all circuits, including the benchmarks b11 and b12, the number of gates is shown for the libraries $STD$ and $STD \cup \{XOR\}$.

Table III shows the results in terms of SAFM coverage on $C_{STD}$ and test set generation time. The test sets were computed by the repeated application of exponential AND operations on KFDDs for the original KFDD circuit, polynomial XOR operations on KFDDs for the toggle circuit and polynomial AND operations on BDDs for the proposed circuits. For the optimized circuit, an exponential SAT-based ATPG approach was chosen where random test patterns were generated, followed by SAT solving for the remaining faults.

Lastly, Table IV shows the results concerning testability under PDFM with strong robust tests for different KFDD circuits $C_{XOR}$. For the original KFDD circuit and the toggle circuit, only paths which don't pass through the $f_i^2$ input of a Davio subcircuit are considered to be testable with strong robust tests. Test patterns for a subset of the remaining paths may also be computable through the usage of exponential ATPG, which was however not considered within our evaluation.

The results show that the proposed methods introduce hardware overhead, but result in 100% test coverage under SAFM and PDFM (excluding paths starting from test inputs), while also significantly reducing the test generation time.

## V. CONCLUSION

In this paper, we have proposed multiple synthesis methods for KFDD circuits which are fully testable under CFM, SAFM and PDFM while guaranteeing efficient runtime with PTPG. Even though KFDD circuits can be smaller than BDD circuits, the resulting circuits may still introduce hardware overhead compared to area- or delay-optimized designs. However, our proposed approach can be selectively applied for safety-critical modules where full testability and efficient test pattern generation are required. Future work includes further research on the reduction of hardware overhead to enable the application on larger designs. Additionally, further tradeoffs between testability properties and hardware overhead can be explored.

## REFERENCES

[1] B. Becker, "Synthesis for testability: Binary decision diagrams," in *STACS*, ser. Lecture Notes in Computer Science, vol. 577. Springer, 1992, pp. 501–512.

[2] R. Drechsler, J. Shi, and G. Fey, "Synthesis of fully testable circuits from BDDs," *TCAD*, vol. 23, no. 3, pp. 440–443, 2004.

[3] B. Becker, R. Drechsler, and M. Theobald, "On the expressive power of OKFDDs," *Formal Methods Syst. Des.*, vol. 11, no. 1, pp. 5–21, 1997.

[4] B. Becker and R. Drechsler, "Synthesis for testability: circuits derived from ordered Kronecker functional decision diagrams," in *ED&TC*, 1995, p. 592.

[5] M. Schnieber and R. Drechsler, "Synthesis for testability: Polynomial test pattern generation for KFDD circuits," in *DSD*, 2025, pp. 50–57.

[6] L. Anghel, R. Cantoro, R. Masante, M. Portolan, S. Sartoni, and M. S. Reorda, "Self-test library generation for in-field test of path delay faults," *TCAD*, vol. 42, no. 11, pp. 4246–4259, 2023.

[7] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. 35, no. 8, pp. 677–691, 1986.

[8] R. Drechsler, A. Sarabi, M. Theobald, B. Becker, and M. A. Perkowski, "Efficient representation and manipulation of switching functions based on ordered Kronecker functional decision diagrams," in *DAC*, 1994, pp. 415–419.

[9] H. Hengster, R. Drechsler, B. Becker, S. Eckrich, and T. Pfeiffer, "AND/EXOR-based synthesis of testable KFDD-circuits with small depth," in *ATS*, 1996, pp. 148–154.

[10] R. Krauss, J. Zielasko, and R. Drechsler, "FrEDDY: Modular and efficient framework to engineer decision diagrams yourself," in *DATE*, 2025, pp. 1–2.