Integrating DFT and DRBD Formalizations in HOL4

Yassmeen Elderhalli, Osman Hasan, and Sofiène Tahar

Department of Electrical and Computer Engineering,
Concordia University, Montréal, QC, Canada

{y_elderh,o_hasan,tahar}@ece.concordia.ca

TECHNICAL REPORT

October 2019

Abstract

Dynamic Fault Trees (DFT) and Dynamic Reliability Block Diagrams (DRBD) are two modeling approaches that capture the dynamic failure behavior of engineering systems for their reliability analysis. Recently, two independent higher-order logic (HOL) formalizations of DFT and DRBD algebras have been developed in the HOL4 theorem prover. In this work, we propose to integrate these two modeling approaches for the efficient formal reliability analysis of complex systems by leveraging upon the advantages of each method. The soundness of this integration is provided through a formal proof of equivalence between the DFT and DRBD algebras. We show the efficiency of the proposed integrated formal reliability analysis on a drive-by-wire system as a case study.

Keywords— Dynamic Reliability Block Diagrams, Dynamic Fault Trees, Integrated Formal Framework, Theorem Proving, HOL4

1 Introduction

Dynamic reliability models, such as dynamic fault trees (DFTs) [1] and dynamic reliability block diagrams (DRBDs) [2], enable modeling the failure dependencies among system components by using DFT gates, such as Functional DEPendency (FDEP) gate, and DRBD constructs, like the spare construct. DRBDs consist of blocks that represent system components and connectors to model the successful paths or multiple paths from the input to the output. These paths determine the required system components to maintain its proper functionality. DFTs, on the other hand, graphically model the faults of system components that lead to the failure of an undesired event, represented by a top event. The required conditions for the occurrence of this top event are captured using DFT gates. An algebra was proposed in [1] for the analysis of DFTs, where inputs and outputs of DFT gates are modeled based on their time of failure. In [3], we developed the higher-order logic (HOL) formalization of this algebra and verified the probability of failure of commonly used DFT gates, which enables conducting a formal DFT analysis within the HOL4 theorem prover [4].

Following the same lines of the DFT algebra, we recently proposed an algebra to analyze DRBDs with spare constructs [5]. We introduced new DRBD operators that allow expressing the structure of a given DRBD to conduct its analysis. We developed the HOL formalization of this algebra using HOL4 to ensure its soundness. It is worth mentioning that the graphical representation of the sources of failure of a system modeled as a DFT cannot be directly obtained using DRBDs. Such a graphical representation is quite helpful in quickly identifying the vulnerabilities in systems. On the other hand, DRBDs identify the required paths and options for the successful behavior that cannot be directly identified using DFTs. However, the DRBD algebra leads to a more efficient reliability analysis since the DRBD algebra is simpler to conduct.

In this work, we propose an integrated framework that enables formally analyzing DFTs and DRBDs based on their algebraic approaches. The proposed framework also allows the formal analysis of DRBDs using the DFT algebra and vice-versa, which requires verifying the formal equivalence of both algebras. The proposed integration provides the possibility to express the failure behavior of a system modeled as a DRBD and the success behavior of a system modeled as a DFT. Moreover, using the integrated framework, a given DFT can be formally modeled using the formalized DFT algebra. Then, based on the formal equivalence of the DFT and DRBD algebras, we can obtain the corresponding DRBD model of the given system in a sound manner and thus use the DRBD model to conduct the formal reliability analysis. As an illustration, we formally analyze the reliability analysis of a drive-by-wire (DBW) system [6] using both reliability models and show that the DRBD algebra based formal analysis results in a shorter proof script and a smaller number of proof goals, and thus a reduction in the time required to conduct the analysis (by 1/24 for the DBW system).

Table 1: Definitions of DFT Temporal Operators

Operator	Mathematical Expression	Formalization

`Before`	$A \lhd B = \begin{cases} A, & A < B \\ +\infty, & A \geq B \end{cases}$	`⊢ ∀ A B. D_BEFORE A B =`
		`(λs. if A s < B s then A s`
		`else PosInf)`

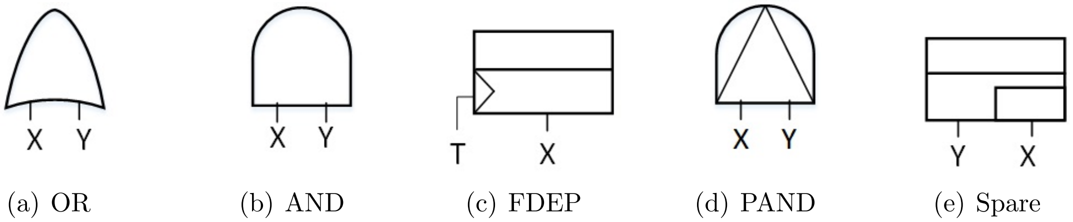
`Simultaneous`	$A \Delta B = \begin{cases} A, & A = B \\ +\infty, & A \neq B \end{cases}$	`⊢ ∀ A B. D_SIMULT A B =`
		`(λs. if A s = B s then A s`
		`else PosInf)`

`Inclusive Before`	$A \unlhd B = \begin{cases} A, & A \leq B \\ +\infty, & A > B \end{cases}$	`⊢ ∀ A B. D_INCLUSIVE_BEFORE A B =`
		`(λs. if A s ≤ B s then A s`
		`else PosInf)`

2 DFT Algebra and its HOL Formalization

The algebraic approach of DFT analysis relies on presenting the basic events, which represent system components, and the output of DFT gates based on their time of failure [1]. Identity elements are defined to express two states of system components. The ALWAYS element represents a component that already failed, i.e., the time of failure equals 0. The NEVER element models a fail safe component, which means that its time of failure equals $+\infty$. Three temporal operators are also introduced, i.e., *Before* ($\lhd$), *Simultaneous* (Δ) and *Inclusive-before* ($\unlhd$), to model the dynamic behavior of one event failing before the other, at the same time and before or at the same time, respectively [1]. In [3], we provided the HOL formalization of these operators (Table 1), where we defined them as lambda abstracted functions that return extended-real numbers (`extreal`), which include real numbers and $\pm\infty$ to model the NEVER element.

In [1], the DFT gates, shown in Figure 1, are modeled based on the time of failure of their output. For instance, the Functional DEPendency (FDEP) gate is used to model failure triggers of system components. The spare gate models spare parts in a system, where the spare (X) replaces a main part (Y) after its failure. In the general case, the failure distribution of the spare is attenuated by a dormancy factor from the active state. Therefore, in the DFT algebra, two variables are used to distinguish the



(a) OR (b) AND (c) FDEP (d) PAND (e) Spare

Figure 1: Fault Tree Gates

Table 2: DFT Gates Expressions and Probability of Failure

Gate	Mathematical Expression	Probability of Failure
AND	$X \cdot Y = max(X, Y)$	$F_X(t) \times F_Y(t)$
OR	$X + Y = min(X, Y)$	$F_X(t) + F_Y(t) - F_X(t) \times F_Y(t)$
PAND	$Q_{PAND} = \begin{cases} Y, & X \leq Y \\ +\infty, & X > Y \end{cases}$	$\int_0^t f_Y(y)\ F_X(y)\ dy$
FDEP	$X + Y = min(X, Y)$	$F_X(t) + F_Y(t) - F_X(t) \times F_Y(t)$
Spare	$Q_{SP} = Y \cdot (X_d \lhd Y) + X_a \cdot (Y \lhd X_a)$ $+Y \Delta X_a + Y \Delta X_d$	$\int_0^t \Big(\int_v^t f_{(X_a \mid Y = v)}(u) du \Big) f_Y(v) dv +$ $\int_0^t f_Y(u) F_{X_d}(u) du$

spare in both its states; active (X_a) and dormant (X_d). Table 2 lists the definitions of these gates. In [3], we provided the HOL formalization of these gates. However, to verify the probability of failure expression given in Table 2, it is required first to define a `DFT_event` to be used in the probabilistic analysis. This is formally defined as [3]:

Definition 1. `⊢ ∀p X t.  DFT_event p X t = {s | X s ≤ Normal t} ∩ p_space p`

where `p` is a probability space. `p_space` is a function that returns the space of `p`. `X` is the time to failure function that can represent inputs and outputs of DFT gates and `t` is the time until which we are interested in finding the probability of failure. The type of `t` is real, while the time to failure functions are of type `extreal` and thus it is required to typecast `t` to `extreal` using the `Normal` function. We verified the probability of failure of all DFT gates based on this event and using their formal definitions, as given in Table 2 [3].

As an example, we provide the details of analyzing the DFT of a drive-by-wire system (DBW) [6], shown in Figure 2, to explain the required steps to use our

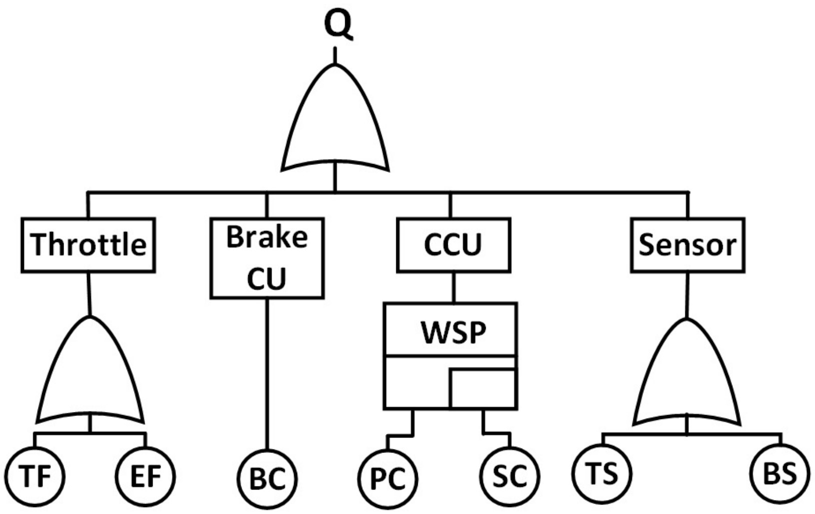


Figure 2: DFT of Drive-by-wire System

formalized algebra. This system is used in modern vehicles to control its functionality using a computerized controller. We provide the reliability model of the brake and throttle subsystems. The throttle system fails due to the failure of the throttle (TF) or the engine (EF). The brake control unit (BCU) failure leads to the failure of this system. A spare gate is used to model the failure of a primary control unit (PC) with a warm spare (SC). Finally, the system can fail due to the failure of the throttle sensor (TS) or the brake sensor (BS).

To formally conduct the analysis using our formalization, it is required first to express the function of the top event algebraically as:

$$\mathtt{Q_{DBW} = (TF + EF) + BCU + WSP\ PC\ SC_a\ SC_d + (TS + BS)}$$

Then, we create a DFT_event for $\mathtt{Q_{DBW}}$ as: DFT_event p $\mathtt{Q_{DBW}}$ t, and verify that it equals the union of the individual DFT events, i.e.:

DFT_event p TF t ∪ DFT_event p EF t ∪ DFT_event p BCU t ∪ DFT_event p (WSP PC $\mathtt{SC_a}$ $\mathtt{SC_d}$) t ∪ DFT_event p TS t ∪ DFT_event p BS t

Thus, we can use the probabilistic principle of inclusion and exclusion (PIE) [1] to verify the probability of failure of $\mathtt{Q_{DBW}}$. The probabilistic PIE expresses the probability of the union of events as the continuous summation and subtraction of the probabilities of combinations of intersection of events. The DBW example is represented as the union of six events, therefore, applying the probabilistic PIE results in having 63 different terms in the final expression. We verify the probability of failure of the DBW as:

Theorem 1.

$$\begin{aligned}
&\vdash \forall \mathtt{BS\ TS\ BCU\ PC\ SC_a\ SC_d\ EF\ TF\ p\ t\ f_{PC}\ f_{(SC_a|PC)}\ f_{SC_aPC}}.\quad \mathtt{0} \leq \mathtt{t} \wedge \\
&\mathtt{dbw\_event\_req\ [BS;\ TS;\ BCU;\ PC;\ SC_a;\ SC_d;\ EF;\ TF]\ p\ t\ f_{PC}\ f_{(SC_a|PC)}\ f_{SC_aPC}} \Rightarrow \\
&\Big(\mathtt{prob\ p\ (DFT\_event\ p\ Q_{DBW}\ t)\ =} \\
&\mathtt{F_{TF}(t)+F_{EF}(t)+F_{BCU}(t)+}\left[\int_0^t \mathtt{f_{PC}(pc)}\times\Big(\int_{pc}^t \mathtt{f_{(SC_a|PC=pc)}(sc_a)}\ d\mathtt{sc_a}\Big)d\mathtt{pc}\right]\mathtt{+F_{BS}(t)+F_{TS}} \\
&\mathtt{-...+...-}\ \mathtt{F_{TF}(t)\times F_{EF}(t)\times F_{BCU}(t)\times F_{BS}(t)\times F_{TS}(t)\times} \\
&\left[\left(\int_0^t \mathtt{f_{PC}(pc)}\times\left(\int_{pc}^t \mathtt{f_{(SC_a|PC=pc)}(sc_a)}d\mathtt{sc_a}\right)d\mathtt{pc}\right)+\int_0^t \mathtt{f_{PC}(pc)\times F_{SC_d}(pc)}\ d\mathtt{pc}\right]\Big)
\end{aligned}$$

where dbw_event_req ensures the required conditions for independence of the events and defines the conditional density functions with their proper conditions [7]. The first six terms in the conclusion of Theorem 1 represent the probabilities of the six individual events of the union of the DBW. Since there are 63 different terms, we are only showing a part of the theorem and the full version is available at [7]. The script of the DBW DFT analysis required around 4850 lines of code and 24 man-hours to be developed.

Table 3: Definitions of DRBD Operators

Operator	Mathematical Expression	Formalization
AND	$X \cdot Y = min(X, Y)$	⊢ ∀X Y. R_AND X Y = (λs. min (X s) (Y s))
OR	$X + Y = max(X, Y)$	⊢ ∀X Y. R_OR X Y = (λs. max (X s) (Y s))
After	$X \rhd Y = \begin{cases} X, & X > Y \\ +\infty, & X \leq Y \end{cases}$	⊢ ∀X Y. R_AFTER X Y = (λs. if Y s < X s then X s else PosInf)
Simultaneous	$X \Delta Y = \begin{cases} X, & X = Y \\ +\infty, & X \neq Y \end{cases}$	⊢ ∀X Y. R_SIMULT X Y = (λs. if X s = Y s then X s else PosInf)
Inclusive After	$X \unrhd Y = \begin{cases} X, & X \geq Y \\ +\infty, & X < Y \end{cases}$	⊢ ∀ X Y. R_INCLUSIVE_AFTER X Y = (λs. if Y s ≤ X s then X s else PosInf)

3 DRBD Algebra and its HOL Formalization

DRBDs capture the dynamic dependencies among system components using DRBD constructs, such as the spare and load sharing constructs. The blocks in a DRBD can be connected in series, parallel, series-parallel and parallel-series. Recently, we proposed an algebra that allows expressing the structure of a given DRBD based on system blocks [5]. The reliability of a given system can be expressed using this DRBD algebra. We defined several operators that enable expressing DRBDs of series and parallel configurations and even more complex structures. Furthermore, the defined operators allow modeling a DRBD spare construct to capture the behavior of spares in a system. We provided the HOL formalization of this algebra to ensure its soundness and enable the formal analysis using HOL4. We first formally define a DRBD event that creates the set of time until which we are interested in finding the reliability [5]:

Definition 2. ⊢ ∀p X t. DRBD_event p X t = {z | Normal t < X s} ∩ p_space p

where X is the time to failure function of a system component and t is the moment of time until which we are interested in finding the reliability of the system. The probability of this event represents the reliability of the system until time t [5]:

Definition 3. ⊢ ∀p X t. Rel p X t = prob p (DRBD_event p X t)

Then, we verify that its probability is related to the CDF [5].

We introduced DRBD identity elements and operators to model both the combinatorial and dynamic behaviors, as listed in Table 3. The idea is similar to the DFT algebra, where the blocks are modeled based on their time of failure. We need to recall that DRBDs are concerned in modeling the successful behavior, i.e., the "not failing" behavior, and thus we can use the time to failure functions to model the behavior of a given DRBD. We defined two identity elements for DRBD that are similar to the DFT

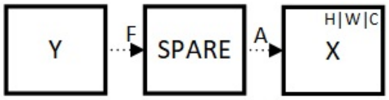


Figure 3: DRBD Spare Construct

Table 4: Mathematical and Reliability Expressions of Spare Constructs

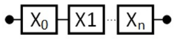
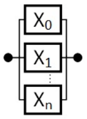
Math. Model	Reliability
$Q_{SP} = (X_a \rhd Y) \cdot (Y \rhd X_d)$	$R_{SP}(t) = 1 - \int_0^t \int_y^t f_{(X_a \mid Y=y)}(x)\ f_Y(y) dx dy - \int_0^t f_Y(y) F_{X_d}(y) dy$

elements, i.e., ALWAYS = 0 and NEVER = $+\infty$. The DRBD operators are listed in Table 3. The AND operator ($\cdot$) models series DRBD blocks, where it is required that all the blocks are working. The output of the AND operator fails with the first failure of any component of its inputs. On the other hand, the OR operator ($+$) models parallel structures, where at least one of the blocks should continue to work to maintain the system functionality. To capture the dynamic behavior, we introduced three temporal operators, i.e., *After*, *Simultaneous* and *Inclusive-after* [5]. The after operator ($\rhd$) models the sequence of events, where the system continues to work as long as one component continues to work after the failure of the other. The simultaneous operator (Δ) is similar to the one of the DFT algebra, where its output fails when both inputs fail at the same time. Finally, the inclusive-after operator ($\unrhd$) combines the behavior of both after and simultaneous operators. We provided the HOL formalization of these elements and operators based on lambda abstracted functions and `extreal` numbers. The mathematical expressions and the HOL formalization are listed in Table 3. The reliability expressions of these operators are available at [5].

A spare construct, shown in Figure 3, is introduced in DRBDs to model spare parts in systems by having spare controllers that activate the spare after the failure of the main part. In Table 4, Y is the main part and after its failure X is activated. We use two variables (X_a, X_d), like the DFT algebra.

DRBD blocks can be connected in series, parallel and more nested structures. We provide here the details of only the series and parallel structures, as listed in Table 5.

Table 5: Mathematical Models and Reliability of Series and Parallel Structures

	Math. Model	Reliability
Series 	$\bigcap_{i=1}^{n}(event\ (X_i,\ t))$	$\prod_{i=1}^{n} R_{X_i}(t)$
Parallel 	$\bigcup_{i=1}^{n}(event\ (X_i,\ t))$	$1 - \prod_{1=1}^{n}(1 - R_{X_i}(t))$

Details about the nested structures can be found in [5]. The series structure, shown in Table 5, continues to work as long as all the blocks are working. Once one of these blocks stops working, then the entire system stops as well. It can be expressed using the AND operator. Its mathematical model is expressed as the intersection of the individual DRBD events [8]. The parallel structure, shown in Table 5, is composed of several blocks that are connected in parallel. Its structure function can be expressed using the OR operator. Its mathematical model is represented using the union of the individual DRBD events. We developed the HOL formalization of these structures and verified their reliability expressions assuming the independence of the individual blocks [5].

We demonstrate the applicability of the DRBD algebra in the formal analysis of the DRBD of the DBW system given in Figure 4. This DRBD is a series sructure with one spare construct to model the main part PC that is replaced by SC after failure. The structure function of the DBW DRBD (F_{DBW}) can be expressed as:

$$F_{DBW} = TF \cdot EF \cdot BCU \cdot (SC_a \rhd PC) \cdot (PC \rhd SC_d) \cdot TS \cdot BS \tag{1}$$

Then, we verify the reliability of the DBW system as:

Theorem 2. $\vdash \forall$p TF EF BCU PC $\mathtt{SC_a}$ $\mathtt{SC_d}$ TS BS t.
DBW_set_req p TF EF BCU PC $\mathtt{SC_a}$ $\mathtt{SC_d}$ TS BS t $\Rightarrow$
(prob p (DRBD_event p $\mathtt{F_{DBW}}$ t) =
Rel p TF t * Rel p EF t * Rel p BCU t * Rel p (R_WSP PC $\mathtt{SC_a}$ $\mathtt{SC_d}$) t *
Rel p TS t * Rel p BS t)

where DBW_set_req ascertains the required conditions for the independence of the DBW system blocks [5]. The reliability of the spare construct can be further rewritten using the reliability expression of the spare using integrals. The script of the reliability analysis of the DBW DRBD is 150 lines long and required only one hour of work.

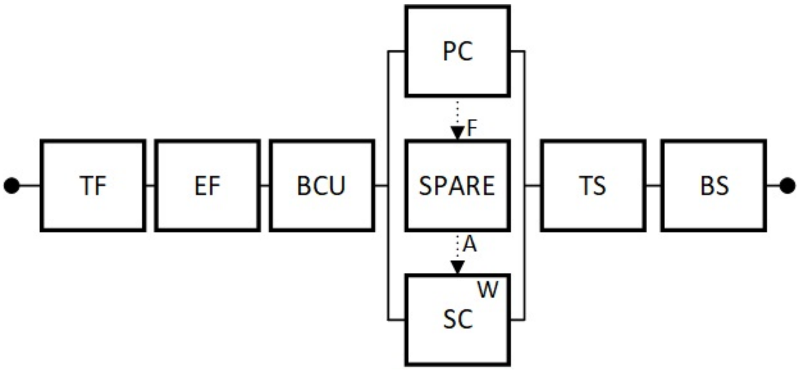


Figure 4: DRBD of Drive-by-wire System

4 Integrated Framework for Formal DFT-DRBD Analysis

The proposed framework integrating DFT and DRBD algebras is depicted in Figure 5. It can be utilized to conduct both DFT and DRBD analyses using the HOL formalized algebras and allows formally converting a DFT model into its corresponding DRBD based on the equivalence of both algebras. The analysis starts by a given system description that can be modeled as a DFT or DRBD. Formal models of the given system can be created based on the HOL formalized algebras. The DRBD model can be analyzed as described in Section 3, where a DRBD event is created and its reliability is verified based on the available verified theorems of DRBD algebra. On the other hand, a DFT model can be analyzed using the formalized DFT algebra, which requires dealing with the probabilistic PIE. Furthermore, the DRBD model can be converted to a DFT to model the failure instead of the success, then this model is analyzed using the DFT algebra. Similarly, the DFT model can be analyzed by converting it to its counterpart DRBD model, which results in an easier process as the PIE is not invoked.

In order to handle the DFT analysis using DRBD algebra and the DRBD analysis using the DFT algebra, it is required to be able to represent the DRBD of the corresponding DFT gates using the DRBD algebra and vice-versa (the equivalence proof in Figure 5). According to [9], the OR, AND and FDEP gates can be represented using series, parallel and series RBDs, respectively. Therefore, they can be modeled using AND and OR operators, while the spare gate corresponds to the spare construct. Finally, the PAND gate can be expressed using the inclusive after operator ($Y \trianglerighteq X$). However, we need to formally verify this equivalence to ensure its correctness. In Table 6, we provide the theorems of equivalence of DFT gates and DRBD operators and constructs, where `D_AND`, `D_OR`, `FDEP`, `P_AND` and `WSP` are the names of the AND, OR, FDEP, PAND and spare DFT gates in our HOL formalization [3]. `R_WSP` is the name of the spare DRBD construct in our formalized DRBD [5] and `ALL_DISTINCT [Y X`$_a$ `X`$_d$`]` ensures that the inputs cannot fail at the same time.

In order to use these verified expressions in Table 6, we need to verify that the `DRBD_event` and the `DFT_event` possess complementary sets in the probability space. We formally verify this as:

Theorem 3. $\vdash \forall$`p X t.  prob_space p` $\wedge$ `(DFT_event p X t)` $\in$ `events p` $\Rightarrow$

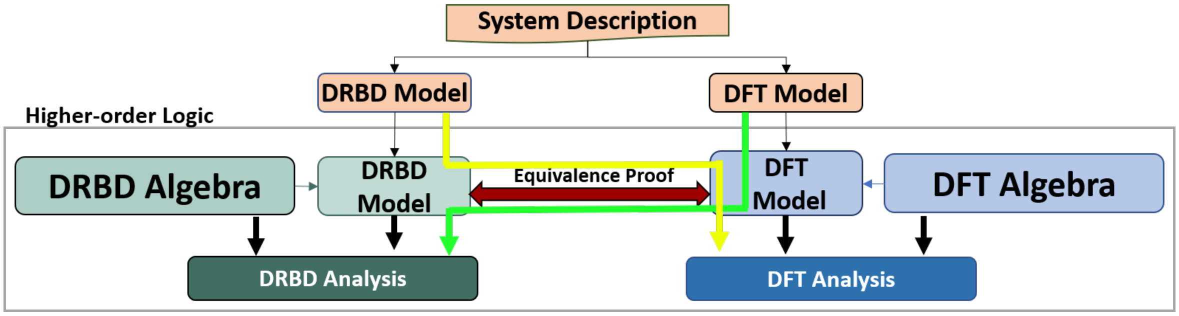


Figure 5: Integrated Framework for Formal DFT-DRBD Analysis using HOL4

Table 6: Verified Equivalence of DFT Gates and DRBD Algebra

DFT Gate	DRBD Operator/Construct	Verified Theorem
AND	OR	$\vdash$ ∀X Y. D_AND X Y = R_OR X Y
OR	AND	$\vdash$ ∀X Y. D_OR X Y = R_AND X Y
FDEP	AND	$\vdash$ ∀X Y. FDEP X Y = R_AND X Y
PAND	Inclusive After	$\vdash$ ∀X Y. P_AND X Y = R_INCLUSIVE_AFTER Y X
Spare	Spare	$\vdash$ ∀X_a X_d Y. (∀s. ALL_DISTINCT [Y s;X_a s;X_d s]) ⇒ (WSP Y X_a X_d = R_WSP Y X_a X_d)

(prob p (DRBD_event p X t) = 1 - prob p (DFT_event p X t))

where the conditions ensure that p is a probability space and that the DFT event belongs to the events of the probability space. This theorem can be verified also if we ensure that the DRBD event belongs to the probability space. This theorem means that for the same time to failure function, the DRBD and DFT events are the complements of each other. This way, we can analyze DFTs using the DRBD algebra and vice-versa.

Based on the verification results obtained in Table 6, DFT gates can be formally represented using DRBDs. We show that the amount of effort required by the verification engineer to formally analyze DFTs by analyzing its counterpart DRBD is less than that of analyzing the original DFT model. In Section 2, a DFT is formally analyzed using the DFT algebra by expressing the DFT event of the structure function as the union of the individual DFT events. Then the probabilistic PIE is utilized to formally verify the probability of failure of the top event. The number of terms in the final result equals $2^n - 1$, where n is the number of individual events in the union of the structure function. Therefore, in the verification process, it is required to verify at least $2^n - 1$ expressions. On the other hand, verifying a DRBD would require verifying a single expression for each nested structure.

As an example, consider the reliability analysis of the DBW system. Analyzing the DFT of this system required verifying 63 subgoals as the top event is composed of the union of six different events. While analyzing the DRBD of the DBW system required verifying only one main subgoal to be manipulated to reach the final goal. Table 7 provides a comparison of the size of the script, the required time to develop it and the number of goals to be verified. Based on these observations, analyzing the reliability of the DBW using the DRBD required 1/24 of the time needed by the DFT. These results show that it is more convenient to analyze the DRBD of a system rather than its DFT if the algebraic approaches are to be used. The only added step will be to formally verify that the DFT and DRBD are the complements of each other, which is straightforward utilizing the theorems in Table 6. Therefore, we verify this as:

Theorem 4. $\vdash$ ∀p TF EF BCU PC SC_a SC_d TS BS t.
prob_space p ∧ DBW_events_p p TF EF BCU PC SC_a SC_d TS BS t ⇒.
(prob p (DRBD_event p F_{DBW} t) = 1- prob p (DFT_event p Q_{DBW} t))

Table 7: Comparison of Formal Analysis Efforts of DBW

	# of subgoals	# of lines in the script	required time
DFT	63	4850	24 hours
DRBD	1	150	1 hour

where DBW_events_p ensures that the DBW DFT events are in the events of the probability space. Thus, we can use the DRBD reliability expression (Theorem 2) to verify the probability of failure of the DFT, which results in a reduction in the analysis efforts.

5 Conclusions

In this report, we proposed an integrated framework to enable the multiway formal algebraic analysis of DFTs and DRBDs within a theorem prover. This framework allows transforming a DFT and DRBD models into their corresponding DBRD and DFT models, respectively, to be either analyzed more effectively using the DRBD algebra or to clearly observe the failure dependencies in the form of a DFT. This requires formally verifying the equivalence of both DFT and DRBD algebras. To illustrate the efficiency and usefulness of the proposed framework, we provided a comparison of the efforts required to analyze a drive-by-wire system and the results showed that using the DRBD in the analysis instead of DFTs required verifying less goals (1:63), smaller script size (150:4850) and less time (1h:24h).

References

[1] G. Merle. *Algebraic Modelling of Dynamic Fault Trees, Contribution to Qualitative and Quantitative Analysis.* PhD thesis, ENS, France, 2010.

[2] H. Xu and L. Xing. Formal Semantics and Verification of Dynamic Reliability Block Diagrams for System Reliability Modeling. In *Software Engineering and Applications*, pages 155–162, 2007.

[3] Yassmeen Elderhalli, Waqar Ahmad, Osman Hasan, and Sofiène Tahar. Probabilistic Analysis of Dynamic Fault Trees using HOL Theorem Proving. *JAL*, 2631(3):469, 2019.

[4] HOL4. https://hol-theorem-prover.org/, 2019.

[5] Y. Elderhalli, O. Hasan, and S. Tahar. A Formally Verified HOL Algebra for Dynamic Reliability Block Diagrams. Tech. rep., Concordia University, Canada., 2019.

[6] A. Altby and D. Majdandzic. Design and Implementation of a Fault-tolerant Drive-by-wire System. Master's thesis, Chalmers University of Technology, Sweden, 2014.

[7] Y. Elderhalli. DFT-DRBD Formal Equivalence: HOL4 Script, Concordia University, Canada, http://hvg.ece.concordia.ca/code/hol/DFT_DRBD_eq.zip, (2019).

[8] O. Hasan, W. Ahmed, S. Tahar, and M. S. Hamdi. Reliability Block Diagrams based Analysis: A Survey. In *Numerical Analysis and Applied Maths*, volume 1648, pages 850129.1–4, 2015.

[9] S. Distefano and A. Puliafito. Dynamic Reliability Block Diagrams vs Dynamic Fault Trees. In *Reliability and Maintainability Symposium*, pages 71–76. IEEE, 2007.