SaberRDの機能安全検証を使用した、DFMEA（設計故障モード影響解析）の自動化

Vishwa Nanjundaiah

Saber AE, Synopsys India

Agenda　アジェンダ

- なぜDFMEA (設計故障モード影響解析) ？
- 従来の検証 vs Saberの機能検証
- SaberRDとISO 26262設計デモ
- 欠陥分析とSaberRDでの安全性メカニズムフロー
- お客様使用事例
- まとめ、Q&A

なぜDFMEA？

- DFMEA設計の主なねらい；
  - 設計リスクの特定、定量化、削減
  - 設計を決定するためのレポート生成
  - 次に取り組むべき設計内容の優先順位付け

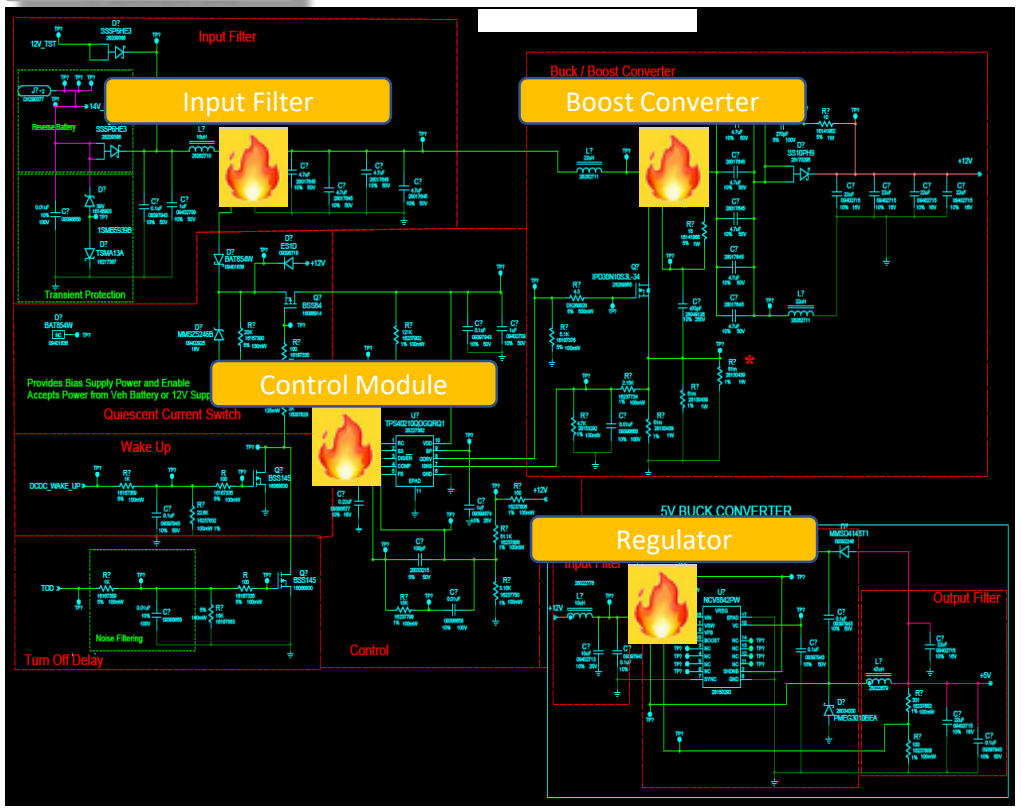
- 何が起こるかを予測；
  - パワーステアリングやABSが故障の場合？
  - エンジン制御が機能しない場合？
  - 熱による電気モジュール不具合の場合？

…

- どの不具合が決定的？　より危険？

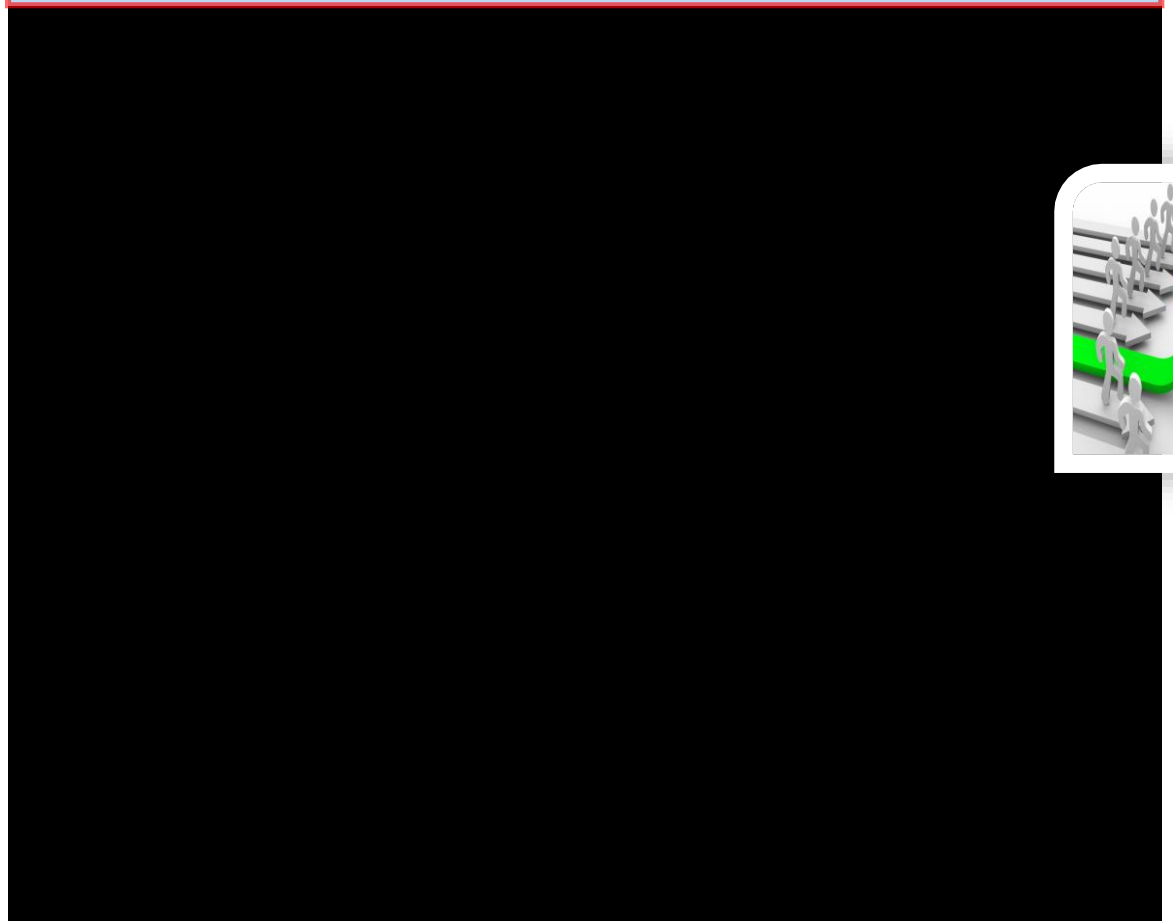
Safety critical automotive systems		
Power Steering	Seatbelt	ABS/EBS
Transmission	Lighting	Engine Control
Airbag	Parking Brake	Start-Stop AL

Failures



従来の検証　vs　Saber の機能安全

従来の方法での課題



Saber機能安全の優位点

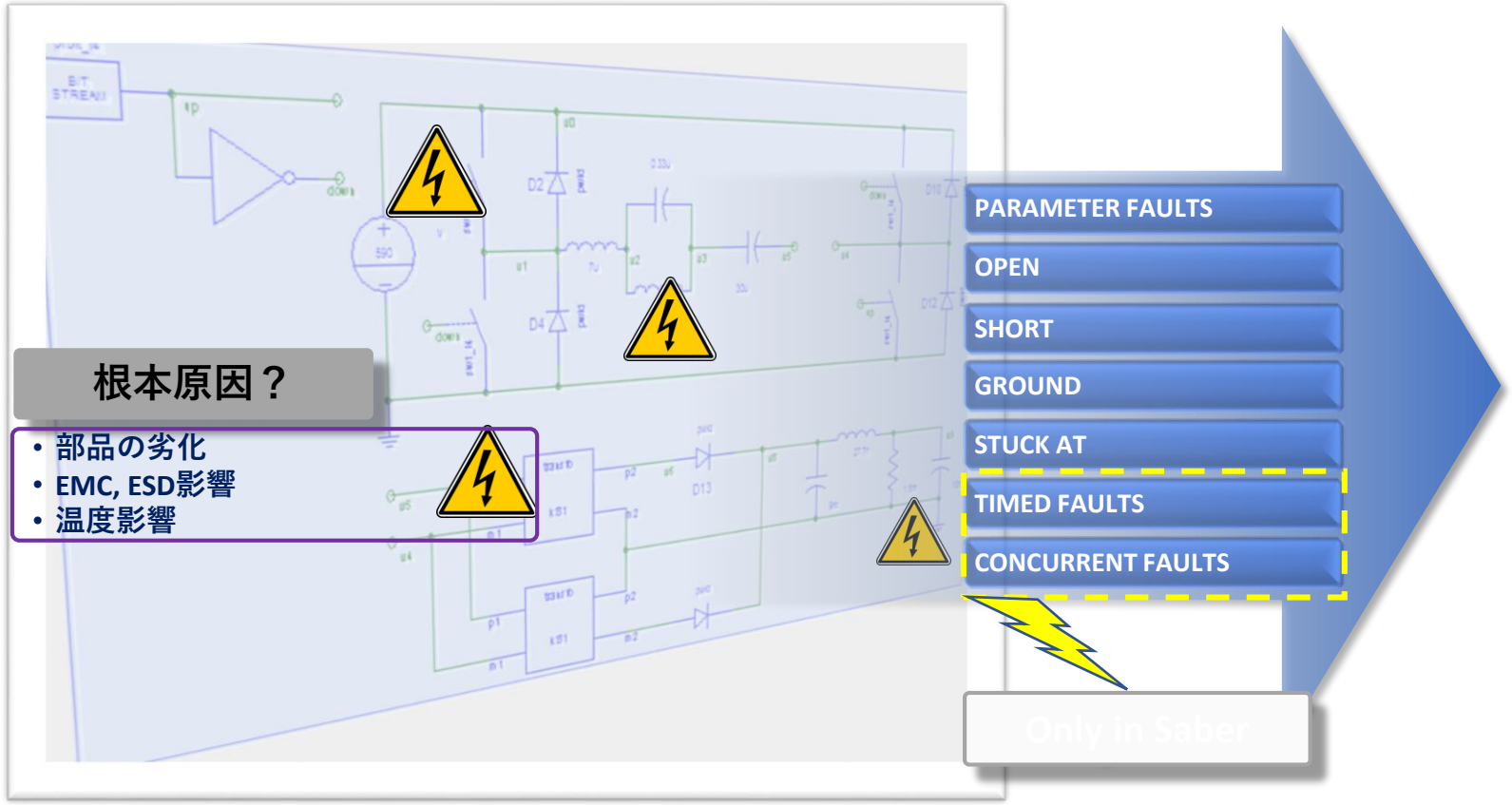
- ホワイトボードでの議論をシミュレーションデータレビューに置き換えることにより、安全検査プロセスを改善する
  - 検証のための会議の時間と頻度を減らします（数ヶ月から数時間に！）
  - 厳密なシミュレーションを使用して複雑な問題を解決することにより、エラーを排除
- システムおよびハードウェアの安全性検証のたえの取るべき手段を追加します
- 考えられるすべての故障モードをシミュレートすることにより、偽陰性となるビルドとテストを排除

複数の技術的安全性のコンセプトをサポート/検証するデータを作り、時間とコストを削減します

Saber機能安全

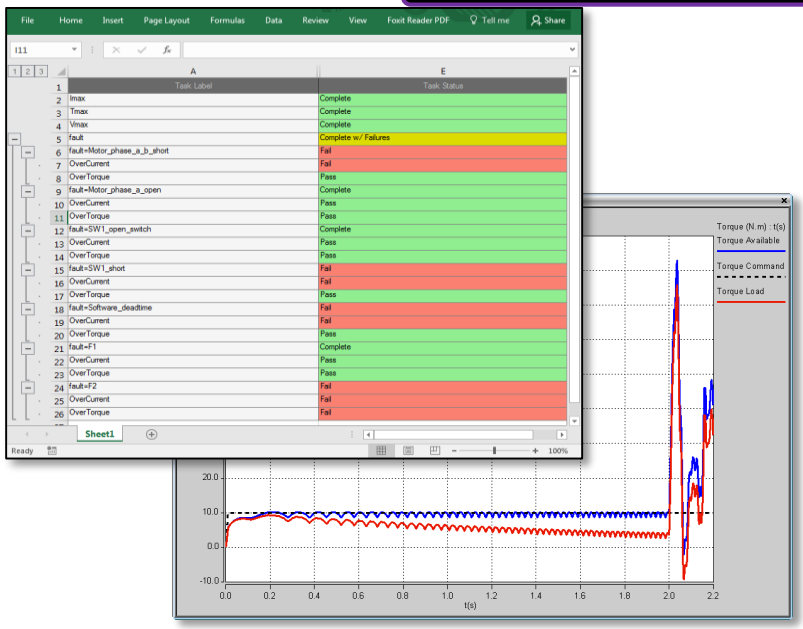
安全要件の厳密な検証

- Saberデザインにハードウェア**FAULT**を注入して影響を解析



セーフティクリティカル

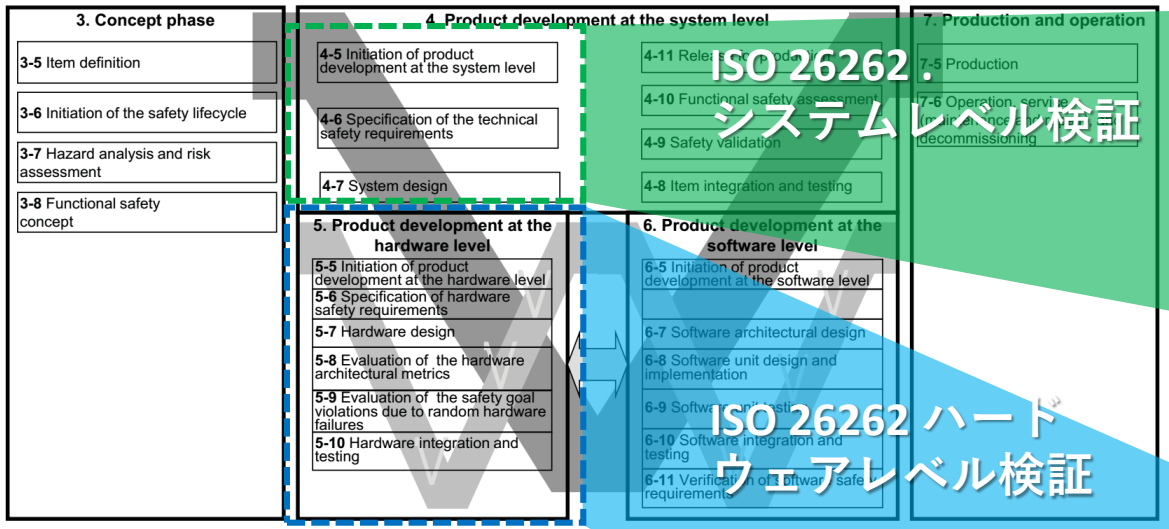
- 安全目標(SG）に違反
- 安全目標(SG)に違反せず



- 解析結果リストのスキャンによってフェイルとなったテストが一目瞭然
- 重要な信号をプロットすることで故障モードの挙動を可視化

機能安全を実現

ISO 26262では、システムおよびハードウェアの技術的安全要件の検証が必要です



ISO 26262：
システムレベル検証

ISO 26262 ハード
ウェアレベル検証

Saber RDはISO26262認証を取得しています！

Table 3 — System design verification

Methods		ASIL A	ASIL B	ASIL C	ASIL D
1a	System design inspection[a]	+	++	++	++
1b	System design walkthrough[a]	++	+	o	o
2a	Simulation[b]	+	+	++	++
2b	System prototyping and vehicle tests[b]	+	+	++	++
3	System design analyses[c]	see Table 1			

[a] Methods 1a and 1b serve as a check of complete and correct implementation of the technical safety requirements.

[b] Methods 2a and 2b can be used advantageously as a fault injection technique.

[c] For conducting safety analyses, see ISO 26262-9:2011, Clause 8.

Table 3 — Hardware design verification

Methods		ASIL A	ASIL B	ASIL C	ASIL D
1a	Hardware design walk-through[a]	++	++	o	o
1b	Hardware design inspection[a]	+	+	++	++
2	Safety analyses	In accordance with 7.4.3			
3a	Simulation[b]	o	+	+	+
3b	Development by hardware prototyping[b]	o	+	+	+

NOTE The scope of this verification review is technical correctness of the hardware design.

[a] Methods 1a and 1b serve as a check of the complete and correct implementation of the hardware safety requirements in the hardware design.

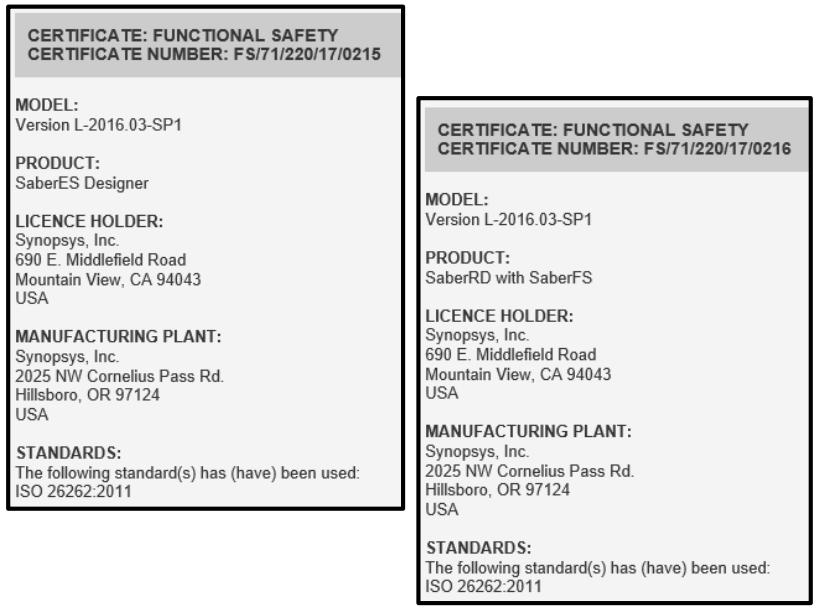
[b] Methods 3a and 3b serve as a check of particular points of the hardware design (e.g. as a fault injection technique) for which analytical methods 1 and 2 are not considered to be sufficient.

*SGS-TuV-SAAR*によるISO26262認証

- 認証されることで
  - Saber機能安全付きSaber RD
  - SaberES Designer

- 達成したこと
  - TCL 1 (Tool Confidence Level/ツール信頼度)
  - ASIL D (Automotive Safety Integrity Level/
    自動車安全インテグリティレベル)

- 以下で確認：
  http://www.sgs-tuev-saar.com/en/certification-database.html



CERTIFICATE: FUNCTIONAL SAFETY
CERTIFICATE NUMBER: FS/71/220/17/0215

MODEL:
Version L-2016.03-SP1

PRODUCT:
SaberES Designer

LICENCE HOLDER:
Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
USA

MANUFACTURING PLANT:
Synopsys, Inc.
2025 NW Cornelius Pass Rd.
Hillsboro, OR 97124
USA

STANDARDS:
The following standard(s) has (have) been used:
ISO 26262:2011

CERTIFICATE: FUNCTIONAL SAFETY
CERTIFICATE NUMBER: FS/71/220/17/0216

MODEL:
Version L-2016.03-SP1

PRODUCT:
SaberRD with SaberFS

LICENCE HOLDER:
Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
USA

MANUFACTURING PLANT:
Synopsys, Inc.
2025 NW Cornelius Pass Rd.
Hillsboro, OR 97124
USA

STANDARDS:
The following standard(s) has (have) been used:
ISO 26262:2011

Saber RD機能安全デモ
モータードライブ設計

機能安全フローでのSaberの役割
例：モータドライブ設計

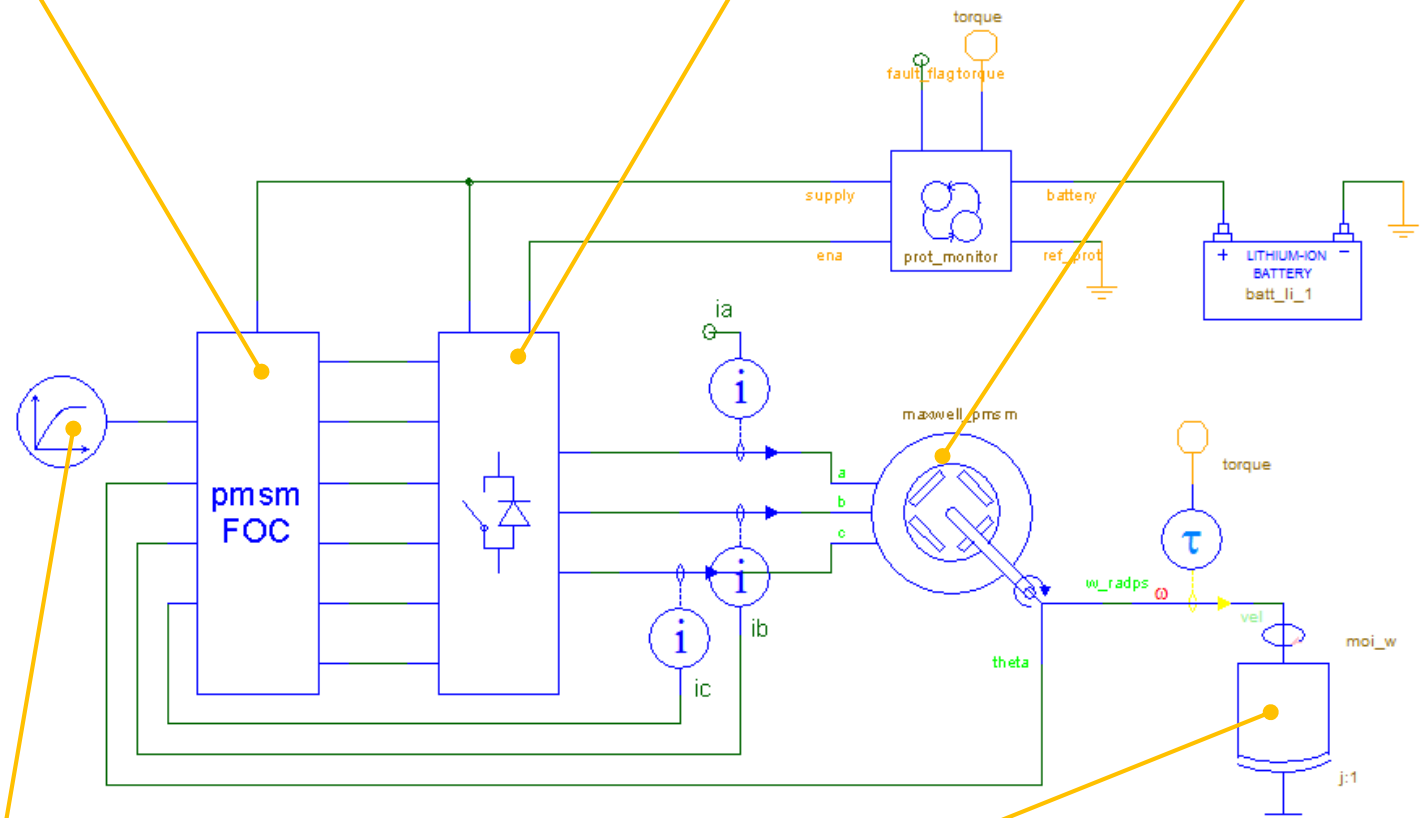
- 欠陥影響のシミュレーションと検証
- 安全メカニズムの開発

Saberを使ったモータ駆動システムのモデリング

PIコントローラ

周波数コンバータ

モータ



トルク指令

慣性荷重

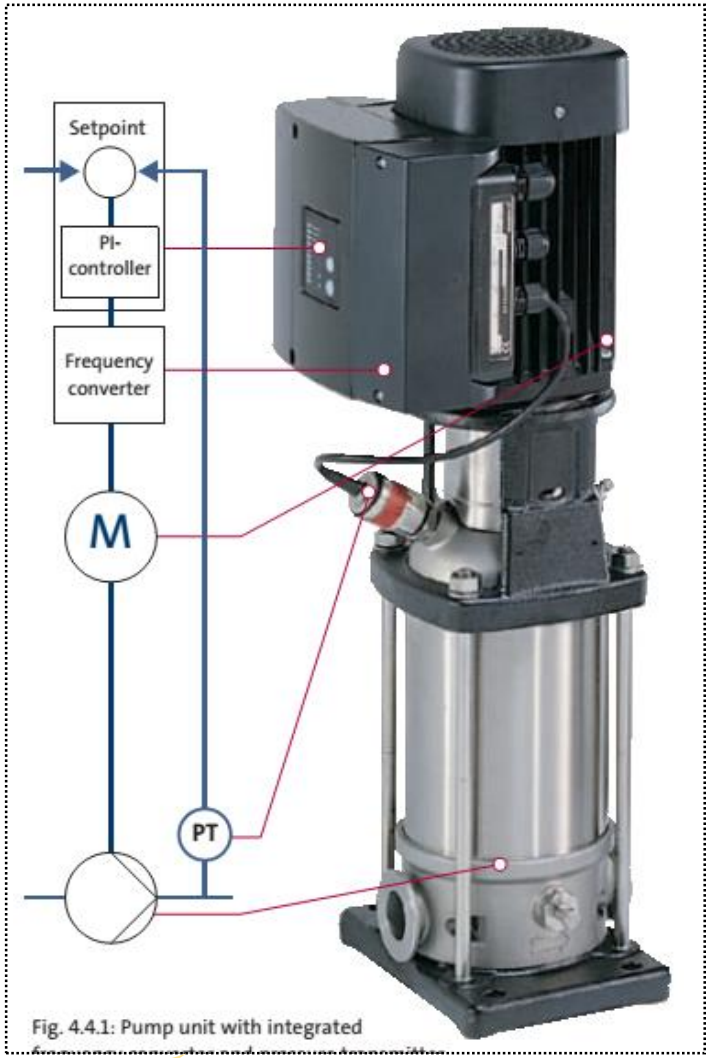
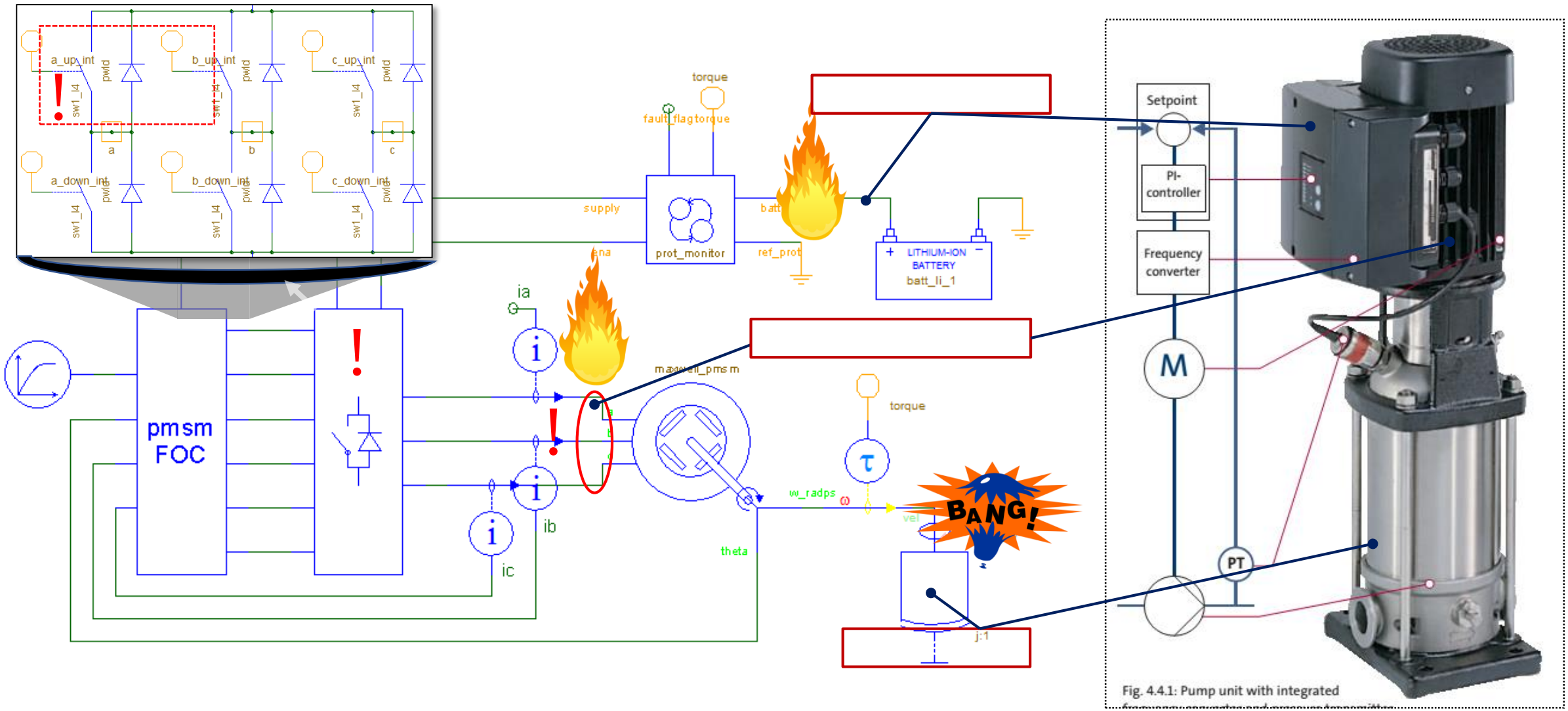


Fig. 4.4.1: Pump unit with integrated

ブロック図はハードウェアと同様にモデル化されます

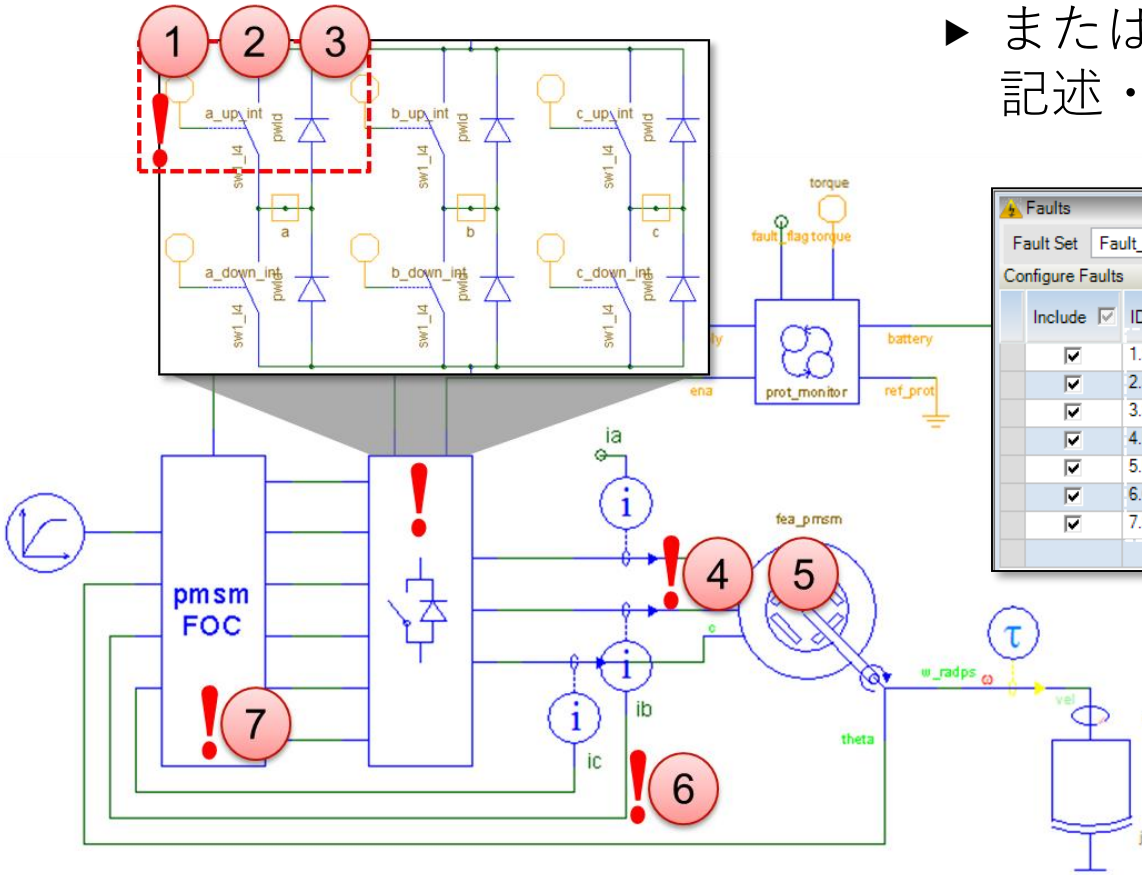
モータ駆動システムの安全要件



主観分析をシミュレーションデータで置き換えることにより、安全性評価時間を短縮

欠陥の記述

- SaberRDの回路図を使用して、コンポーネント欠陥またはネットワーク欠陥を選択します（隣接部分の短絡など）
- または、包括的な検証シミュレーションのために欠陥の記述・定義を自動化します



Faults

Fault Set: Fault_Set

Configure Faults

Faults start at t=2s.

Include	ID	Fault	Type	Parameter	Non-Fault Value	Fault Value	Fault Begin	Fault End	Description
☑	1.SW1_open_diode	/sym1/pwld.pwld1 p open	Analog Pin	rnom	.01m	100meg	2	inf	Open diode in bridge
☑	2.SW1_open_switch	/sym1/sw1_l4.sw1_l4_1 p open	Analog Pin	rnom	.01m	100meg	2	inf	Open switch in bridge
☑	3.SW1_short	/sym1/sw1_l4.sw1_l4_1 p,m short	Analog Pin	rnom	100meg	**100m**	2	inf	Short switch in bridge
☑	4.Motor_phase_a_b_short	/maxwell_pmsm.maxwell_pmsm a,b short	Analog Pin	rnom	100meg	**100m**	2	inf	Short phases of motor
☑	5.Motor_phase_a_open	/fea_pmsm.pmsm a open	Analog Pin	rnom	.01m	100meg	2	inf	Open phase of motor
☑	6.Current_sensor_open	/sense_current_3p.sense_current_3p2 k	Parametric	k	1	**0**	2	inf	Disconnected current sensor
☑	7.Software_deadtime	/foc_pmsm.foc dead_time	Parametric	dead_time	2.5u	**100u**	2	inf	Software failure

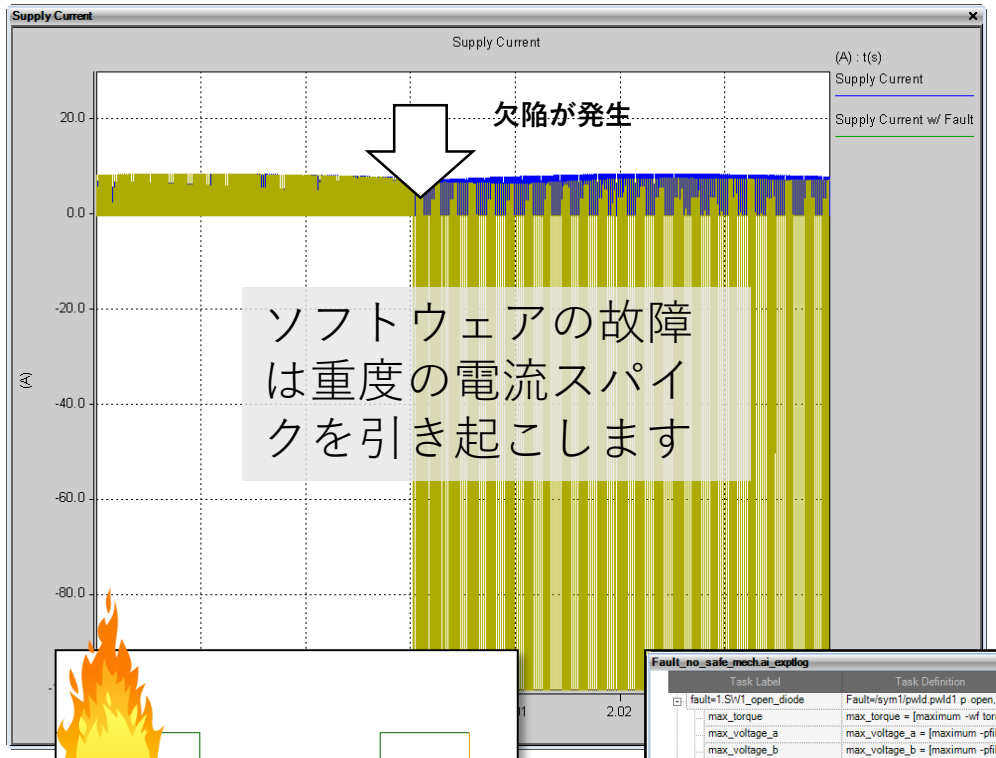
この例では適用される欠陥は次のとおり

1. ダイオード開放回路（ブリッジ）
2. スイッチ融着開放（ブリッジ）
3. 制御線の接地短絡（ブリッジ）
4. アーマチュアコイル間の短絡（モータ）
5. アーマチュアコイルの開放（モータ）
6. 位相電流センサの開放（センサ）
7. ソフトウェアデッドタイム（コントローラ）

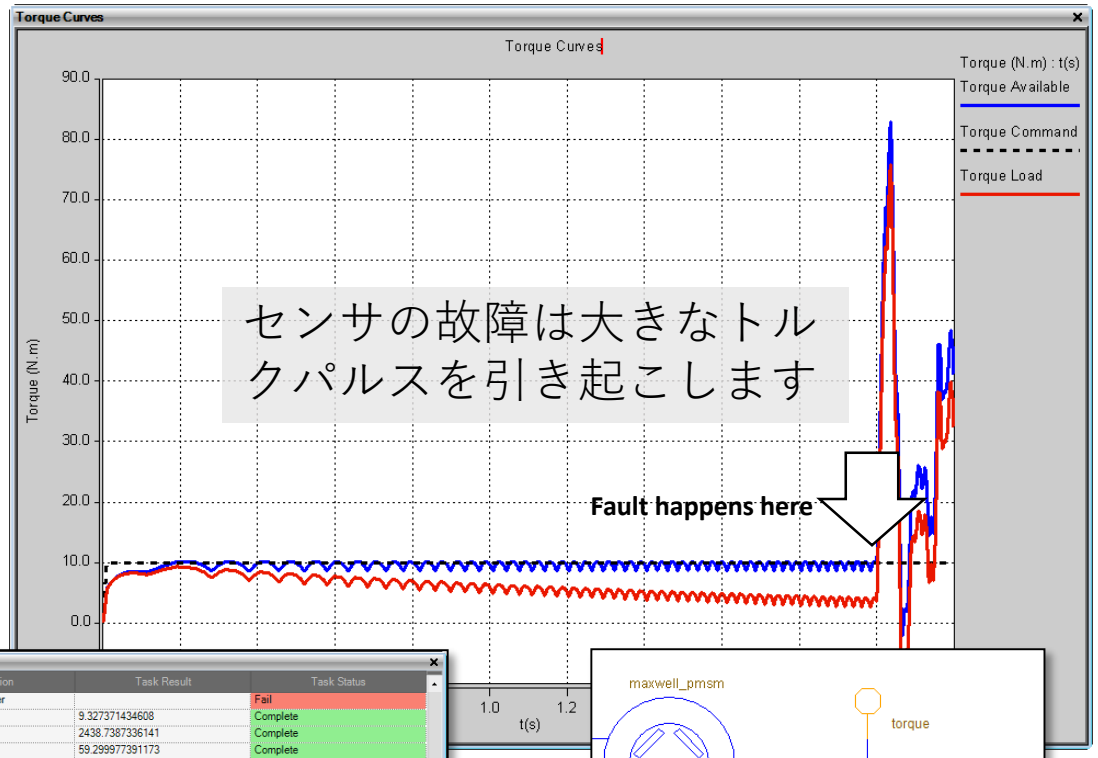
欠陥種類と欠陥タイミングを素早く定義します

システムが故障したとき何が起こる？

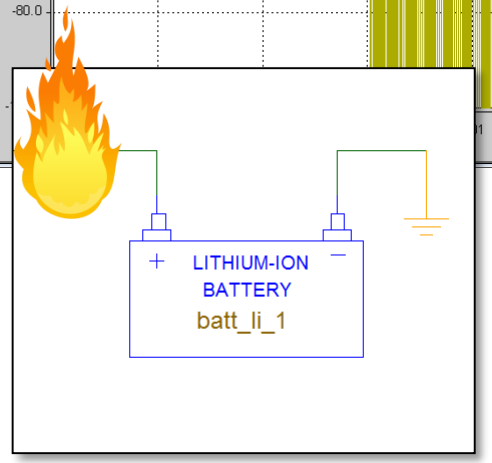
ここまではシステムは正常に機能していた



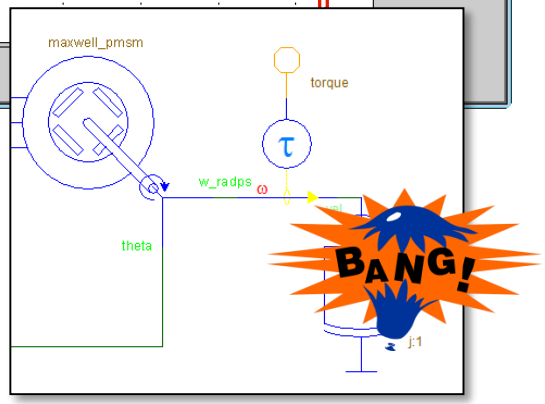
ソフトウェアの故障は重度の電流スパイクを引き起こします



センサの故障は大きなトルクパルスを引き起こします



レポートは信頼性と安全故障を表示します

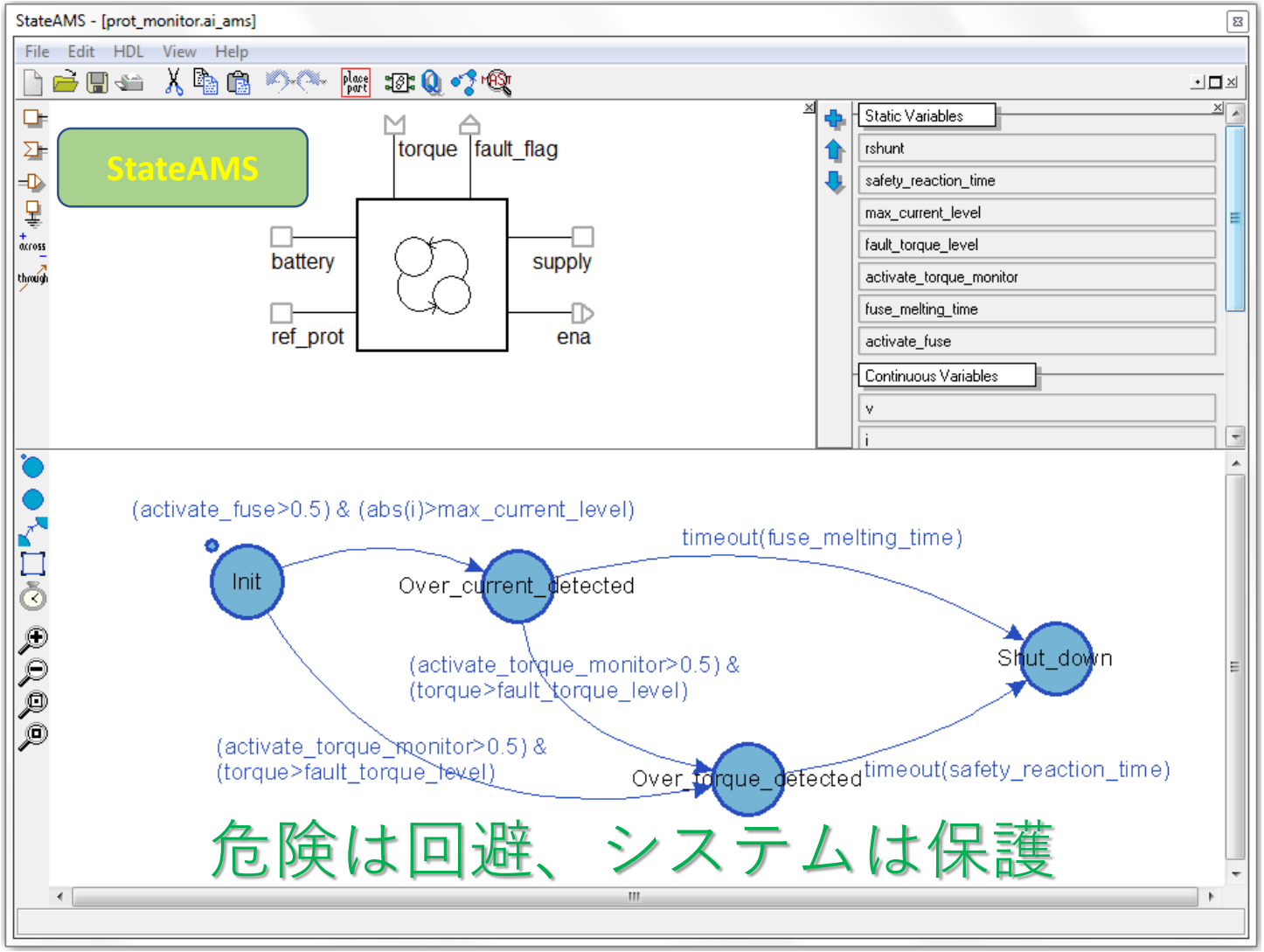


StateAMSによる安全メカニズム

正常にシステム操作

電気機器の消費電流が大きすぎる

ヒューズが飛ぶ



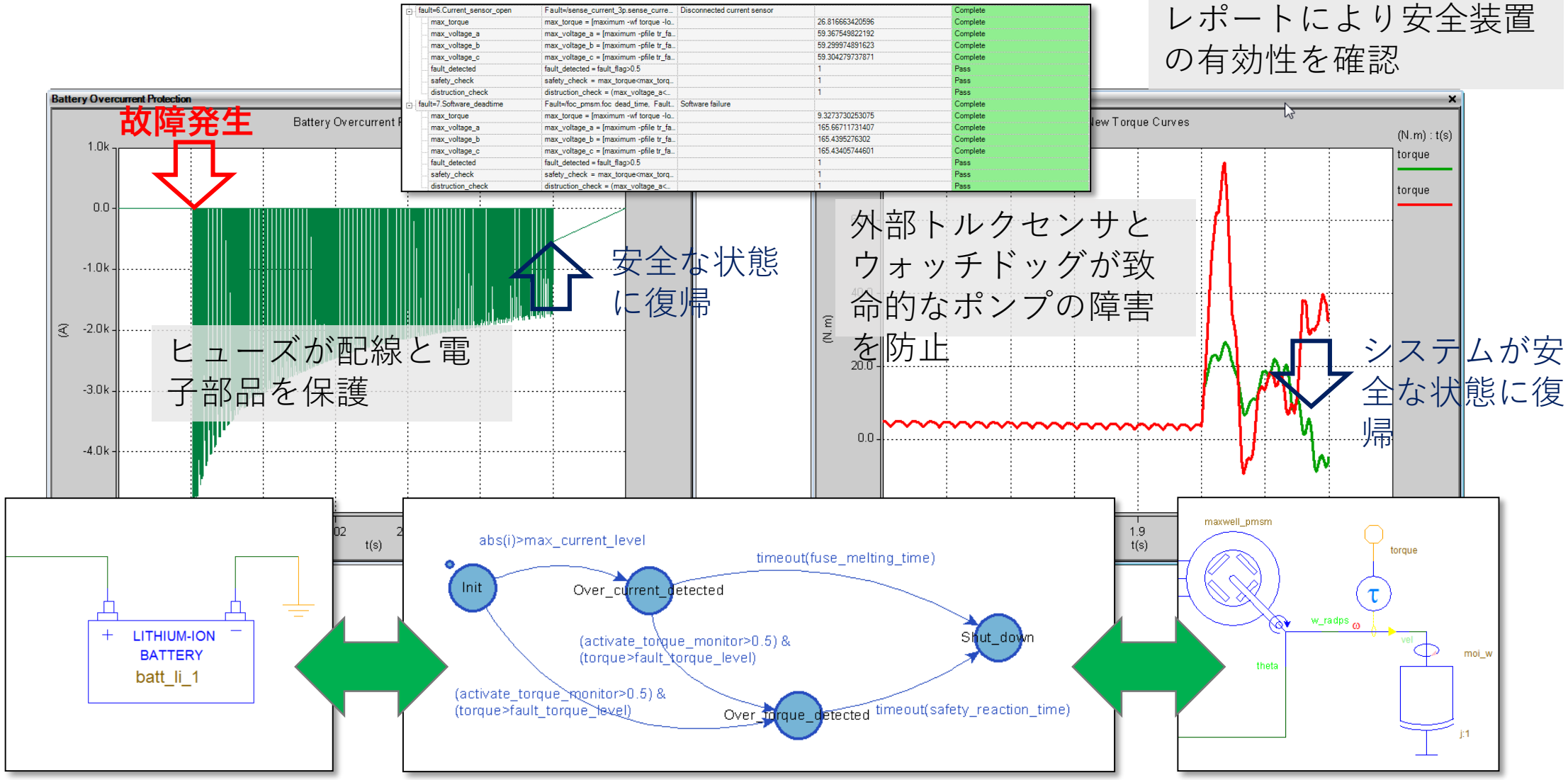
トルク計測

オーバートルク検出

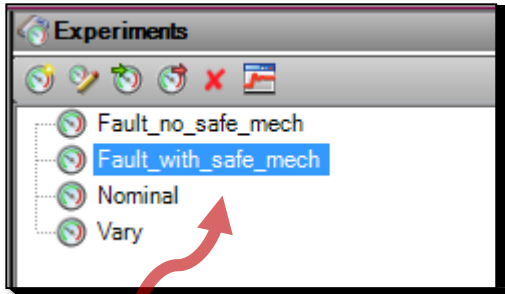
システムが安全状態に遷移

安全機構は、ハードウェア障害/故障を軽減し、システムを安全な状態に維持または戻します

安全メカニズムを適用すると



Saberでの安全性と信頼性テストの自動化



Experiment Analyzerは、迅速で自動化された再現可能な設計検証を可能にします

マルチコアを使用した分散繰り返し解析手法により、より高速なシミュレーションが可能

Loops (queue)	Parallel Runs (parallel)	CPU Time Taken	% Speed Improvement (w.r.t. Sequential run)
Sequential	---	9 min, 25 secs.	---
Multicore	2	5 min, 54 secs.	37%
Multicore	4	3 min, 37 secs.	63%

Note: These results are from performing the simulations in a quad core machine (No of Cores = 4).

```
activate_torque_monitor = 1
max_current_level = 50
fuse_melting_time = 50m
fault_torque_level = 12
safety_reaction_time = 1m
max_torque_level = 30
max_voltage_level = 1000
alter -param :drive3_pmsm:prot_monitor.prot_monitor:activate_torque_monitor -value activate_torque_monitor
alter -param :drive3_pmsm:prot_monitor.prot_monitor:max_current_level -value max_current_level
alter -param :drive3_pmsm:prot_monitor.prot_monitor:fuse_melting_time -value fuse_melting_time
alter -param :drive3_pmsm:prot_monitor.prot_monitor:fault_torque_level -value fault_torque_level
alter -param :drive3_pmsm:prot_monitor.prot_monitor:safety_reaction_time -value safety_reaction_time
fault -progress 500 -ffile Fault_Set
  tr_fault_effects = [tr -tend 2.2 -tstep 1u -siglist :drive3_pmsm:a :drive3_pmsm:b :drive3_pmsm:batt_li_1.batt_li_1:iout :drive3_pmsm:c :drive
  fault_flag = [atx -pfile tr_fault_effects -signal :drive3_pmsm:fault_flag]
  torque = [wf:read -pfile tr_fault_effects -signal :drive3_pmsm:torque]
  max_torque = [maximum -wf torque -log yes]
  max_voltage_a = [maximum -pfile tr_fault_effects -signal :drive3_pmsm:a -log yes]
  max_voltage_b = [maximum -pfile tr_fault_effects -signal :drive3_pmsm:b -log yes]
  max_voltage_c = [maximum -pfile tr_fault_effects -signal :drive3_pmsm:c -log yes]
  fault_detected = fault_flag>0.5
  safety_check = max_torque<max_torque_level
  distruction_check = (max_voltage_a<max_voltage_level) && (max_voltage_b<max_voltage_level) && (max_voltage_c<max_voltage_level)
Graph_Fault_1 = [graph -title "Electrical Behavior"]
  sig1 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:a]
  sig2 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:b -newregion no]
  sig3 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:c -newregion no]
  sig4 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:ia]
  sig5 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:ib -newregion no]
  sig6 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:ic -newregion no]
  sig7 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:batt_li_1.batt_li_1:iout -label "i_battery"]
Graph_Fault_2 = [graph -title "Mechanical Behavior and Fault Detection"]
  sig8 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:rotor]
  sig9 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:torque]
  sig10 = [graph:plot -pfile tr_fault_effects -signal :drive3_pmsm:fault_flag]
```

- 安全メカニズムの初期化（安全上決定的な要件を特定）
- 欠陥をシミュレーション
- 不具合レポートのための測定とテスト
- 出力の可視化

お客様使用事例

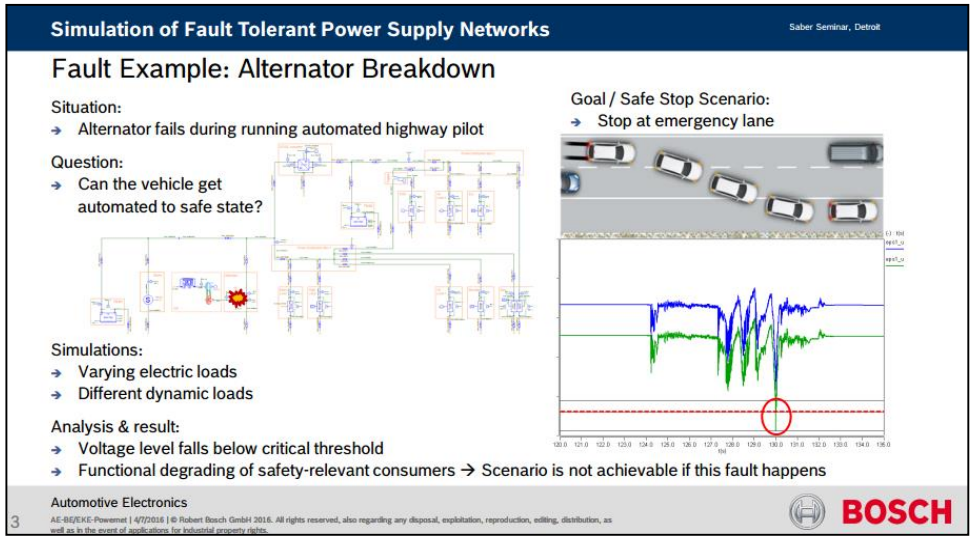
- BOSCH社：電力供給ネットワークの技術的安全
- DELPHI社：セーフティクリティカル回路のDFMEA

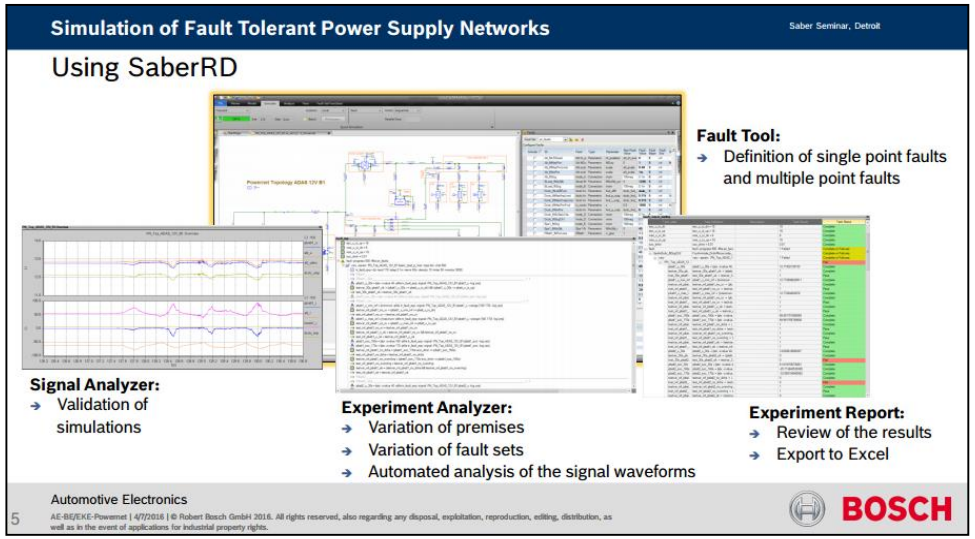
BOSCH

- **課題:**オルタネータおよびパワーネットアプリケーションの複数のシナリオと技術的安全性の概念を評価する

- **解決法:**
  - Saberを使用してモデリング：3つの負荷シナリオ、2つの外気温、10のパワーネット部品、27の単一ポイント欠陥、729の二重ポイント欠陥
  - Simulink制御付きのコシミュレートパワーネット

- **結果:**
  - 安全性検証の時間を数か月から数日に短縮
  - 深い分析から高度な安全についてのコンセプトを作成
  - 実機試験では扱われない安全シナリオを容易に実行
  - 後工程へのレポート用のシミュレーションデータのエクスポート



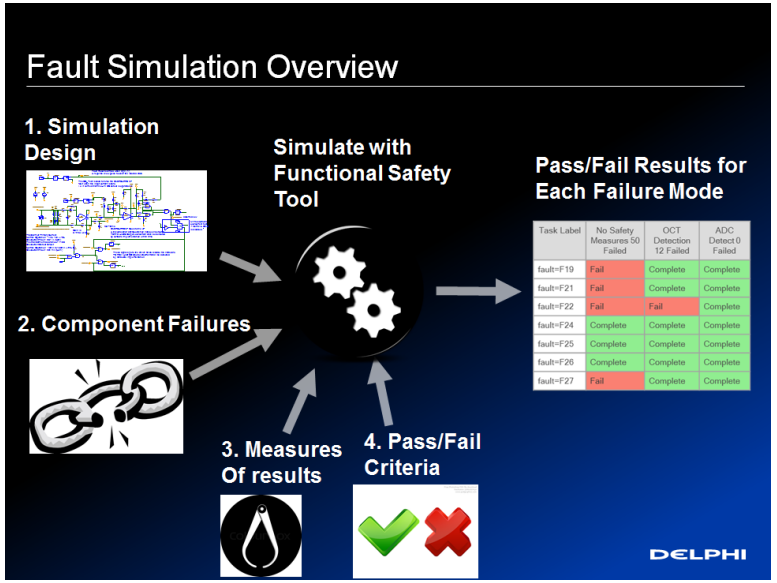


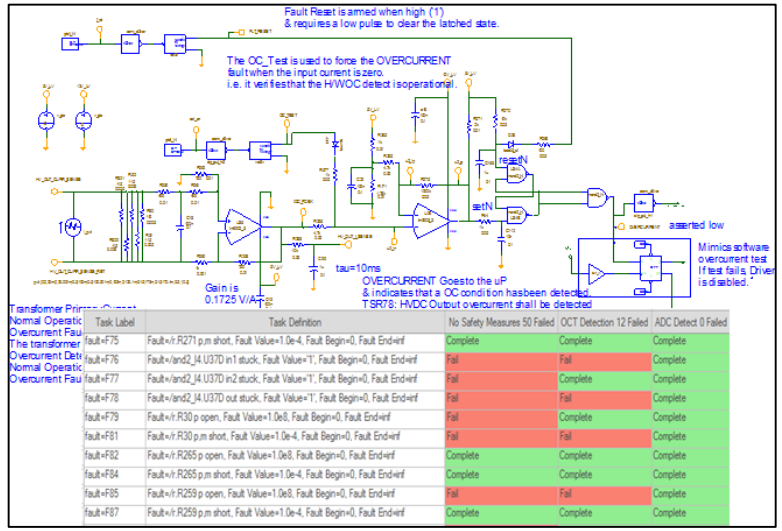
Delphi Technologies

- **課題：**セーフティークリティカル回路図の部品レベルのDFMEA検証

- **解決法:**
  - Saberを使用した過電流保護回路のモデリング:
    - ハードウェア過電流保護
    - ソフトウェア過電流保護
    - 起動時のセルフテスト
  - 全ての部品に対して160の欠陥を自動生成
  - 合否テストのシミュレーションとレポート出力

- **結果:**
  - スループット、欠陥テストの効率と精度を大きく改善
  - 安全レポートを根拠データとともに自動的に生成





自動車以外での機能安全性・信頼性

自動車産業だけでなく、他の産業にも同じ主要部品が見られます

- 航空宇宙
- エネルギー
- 産業オートメーション
- 医療

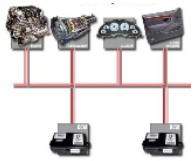
センサ

コンバータ

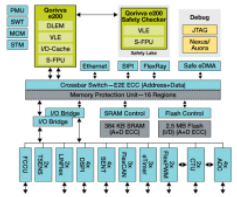
モータ

機械負荷

IVN



ECU



アクチュエータ・モータドライブ

まとめ

- 従来のアプローチでは不可能であったり著しく非効率的であっても、ISO26262認定のSaberRD機能安全ならば実行可能
  - **費用と時間の大幅な削減！**
- 主観的分析をシミュレーションデータに置き換えて、安全性について複数のコンセプトをサポート・検証することにより、安全性の評価時間を短縮
  - **時間でコントロール可能な欠陥や同時欠陥の適用！**
- シミュレーション段階で、安全メカニズムにより、ハードウェア障害/故障を軽減し、システムを安全な状態に維持します
  - **危険を回避し、システムを保護！**
- Experiment Analyzer, Distributive Iterative Analysisにより、完全自動化と高速シミュレーションがそれぞれ可能になります
  - **合否レポートとエクセルへ測定データをエクスポート！**

参照先

Saberオープンフォーラム

https://saberforum.net/discussion/55395/saberrd-design-example-functional-safety-analysis-for-a-motor-drive-design#latest

Saberホームページ

https://www.synopsys.com/verification/virtual-prototyping/saber/methodologies/saber-functional-safety.html

Saber機能安全データシート

https://www.synopsys.com/content/dam/synopsys/verification/datasheets/saber-functional-safety.pdf

SYNOPSYS®
Silicon to Software™

Thank You