モデル検査による高レジリエンス・システムの検証

早水 公二　山形 頼之　林 伸行　大崎 人士†　辰野 功††　兼松 順三　白井 大†††

産学官連携の共同研究によって，マイコン・システムに形式手法（モデル検査）を適用した事例を紹介する．研究では，最初に役割分担を明確にすることで，企業が持つ高度な設計技術と，研究所が持つ専門的なモデル化の技術を活用して，検証を成功に導くことができた．

Model checking a highly resilient system

Koji Hayamizu, Yoriyuki Yamagata, Nobuyuki Hayashi, Hitoshi Osaki†　Isao Tatsuno††
Junzo Kanematsu, Hajime Shirai†††

We present our experience of applying formal methods (model checking) to a micro-computer. Making clear separation of the role between designers and verifiers, we successfully verify a micro-computer with a novel design.

1. はじめに

本事例は，高いレジリエンス性を確保することで，「止まらないこと」を追求したマイコン「FUJIMI」を，モデル検査を用いて検証した事例である．これまでの事例と大きく異なるのは，過去に経験した事が無い非常に特殊なモデルを作成したことである．本稿では，そのモデルの特徴についても紹介する．

2. FUJIMIとは

FUJIMI とは「マイコン・システムの中心機能である CPU のみを繰り返しリセットし，リセット後，予め保存しておいた CPU データを CPU に戻して再スタートする事で，システムの継続動作を可能としたシステム」と定義される．CPU のみをリセットすることで，正常な処理を継続しながら，マイコンの基本的な問題である暴走の発生を抑えることができる．

3. 形式手法（モデル検査）の適用

FUJIMI のメリットを以下に列挙する．

- 見かけ上，暴走が見えなくなる．
- 暴走の最長時間を規定できる．
- 初期化無しの再スタートが可能．
- RAM と I/O は初期化しないため，リセットしても処理を継続することが可能．

一方で FUJIMI には以下のデメリットがある．

- アプリケーションの開発が難しい．
- 複雑さ故にシステムの検証が難しい

研究では，2 つ目のデメリットを解消するべく，形式手法（モデル検査）でシステムの検証を行った．モデル検査器は SMV を用いた．

4. 適用の流れと役割分担

モデル検査を適用するにあたっての作業の流れと役割分担を図 1に示す．

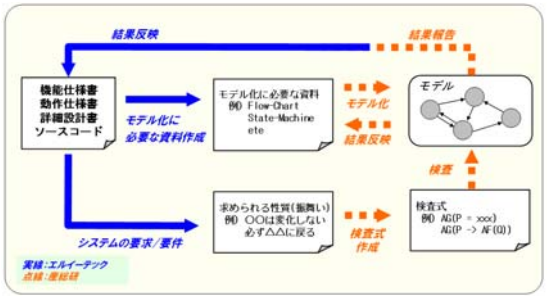

図1　適用の流れと役割分担

実線は企業側が担当した作業であり，点線は研究所が担当した作業である．研究開始当初より，役割分担を明確にすることで，企業が持つ高度な設計技術と，研究所が持つ専門的なモデル化の技術を，互いに最大限発揮することができた．

5. SMVモデル

5.1. モデル化の進め方

FUJIMI は非常に複雑な振る舞いをするシステム

† 独立行政法人産業技術総合研究所
†† 株式会社エルイーテック
††† システム・コンサルタンツ株式会社

であるため，事前に状態爆発の発生が予想されていた．そのため，モデル化にあたっては，詳細モデルから抽象モデル，最終的には最適な規模のモデルへと状態を絞り込んでいく手法を採用した．

ただし，割込みの位置や，発生する頻度などの，FUJIMI にとって重要な要素については，抽象化の対象とせず，極力実機の動きを維持したままモデル化した．

5.2. モデル化の特徴

本研究で作成したモデルの特徴を説明する（図2参照）．統合モデルは，アプリケーションのモデルやFUJIMIのコア部分のモデル，外部環境モデル等のサブモデルを統合したモデルである．統合，サブの両モデルには，モデルの振る舞いを制御する制御変数を定義した．アプリケーションモデルでは制御変数A，FUJIMIのコア部分では制御変数B，統合モデルでは制御変数Xを定義した．本モデルの最大の特徴は「統合モデルの中で，本来操作すべきではない，統合モデル自身の制御変数を操作すること」である．

これまで 40 件以上のシステムにモデル検査を適用して，数百種類のモデルを作成してきたが，このようなモデルは本事例が初めてであり，非常に特殊なモデルである．

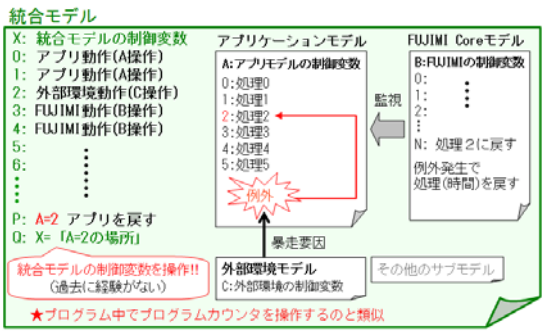

図2　モデルの特徴

6. 検査項目と検査結果

6.1. 検査項目と検査式

本研究で最も重要な検査項目は「CPU の暴走が継続しない」ことである．CTL 式を以下に示す．

CTL 式 ： !EF (EG (20 < CPU))

CPU の状態は数値 1, 2, ･･･, 20→1 の循環遷移を正常状態とし，各値から任意に遷移する数値 21 を暴走状態として定義した．暴走状態では 21, 22, ･･･, 30→21 の循環遷移を行う．したがって EG(20 < CPU)は暴走が継続している状況を示す．

6.2. 検査結果

モデル検査の結果はFALSEであり，反例としてCPUの暴走が継続するパスが出力された．システムへの入力イベントとCPUの状態のみに着目して反例を整理した結果を図 3に示す．

図3　CPU の暴走が継続するパス

上記の反例では，まず，イベントが何も無く CPU 状態も正常な状態から，突然 CPU が暴走する．そして，その直後に割込み(NMI)が発生して，暴走状態を保存してしまう．さらに FUJIMI のリセット(RESET)処理によって，保存した暴走状態で，CPU が復帰される．この後，例外(EXCEPTION)が定周期，かつ，一定の遅れで発生した場合には，CPU の暴走が継続することが判明した．ただし，この現象は，①～⑤の稀なイベントが連続して発生した場合に起こるため，FUJIMI では対応しないこととなった．この反例は，設計の参考情報の位置付けで研究所から企業に提示した．

6.3. 最終動作確認

検査で得られた反例に対して，企業より「例外(EXCEPTION)は暴走直後に必ず発生するはずである」との指摘があったためモデルを改良した．さらに，WatchDogTimer によるリセットや予期せぬリセット，メモリ破壊等の通常発生し得るイベントを全てモデルに追加して，最終的な動作確認を行った．再度「CPU の暴走が継続しない」ことを検査したところ，TRUE が出力されて，モデル化した範囲内では，一旦は CPU が暴走したとしても，それが継続しないことを確認できた．モデル検査器の実行に要した時間は 9000[sec]で，15.8[GB]のメモリを利用した．

7. モデル検査報告書

産総研では，共同研究の成果として，モデル化の方針やモデルの設計内容，検査項目とその結果等をまとめたモデル検査報告書を作成して，企業の開発に役立てて頂いている．