

組み込みデバイスにおけるテスト方法改善の取組み

—ハード/ファーム/ソフトの機能テスト FuST(Function Slice Test)の推進—

† 永田 充 †† 田中 宣好 篠原 範子 板倉 正樹 碓間 優一

日本システムウェア(株) PS 事業本部 PF 事業部 ND 部

〒150-0036 東京都渋谷区南平台町 2-15 9F

E-mail: † mnagata1@gw.nsw.co.jp †† ntanaka2@gw.nsw.co.jp

あらまし 組み込みデバイスにおけるテスト方法改善として FuST(Function Slice Test)を確立した。FuST を使う事で、従来のテスト方法では検出が出来ないバグが検出できた。また、このバグを解決することにより、機能の並列動作時における低消費電力化、組み込みデバイスのリワークを減らす事ができた。FuST は従来のハード/ファーム/ソフトウェアによるそれぞれの分業制テストに、機能テスト、機能動作によるハードウェア特性値劣化、領域境界テスト、をテスト条件に考慮するテスト方法である。従来の分業制テストプロセスにおける課題の解決に FuST が有効であることが判った。

キーワード 組み込みテスト、機能テスト、ハードウェア特性値

Approach of test method improvement in built-in device

—Proposal of FuST(Function Slice Test) by functional test on hard/farm/software.—

NIPPON SYSTEMWARE CO., LTD. Network Design Department Platform Solution Division Product Solution Group

2-15 9F Nanpeidai-chou Shibuya-ku Tokyo 150-0036 Japan

† Mitsuru Nagata, †† Nobuyoshi Tanaka, Noriko Shinohara, Masaki Itakura, Yuichi Hazama

E-mail: † mnagata1@gw.nsw.co.jp †† ntanaka2@gw.nsw.co.jp

Abstract It proposed FuST(Function Slice Test) as a test method improvement in the built-in device. Malfunction not to be able to do in the test of the past was able to be detected by using FuST. The re-design of low power consumption and the built-in device when the function was operated parallel by solving this malfunction was able to be decreased. In the test of past hard/firm/software, the functional test, the characteristic value deterioration because of the function operation, and the area boundary are made a test condition. It has been understood that FuST is effective for the problem solution in a past test process.

Keyword Built-in test(BIT), Functional test, Characteristic value

1. はじめに

近年の組み込みデバイス開発においてハードウェア/ファームウェア/ソフトウェア(以後順に下位レイヤ/中位レイヤ/上位レイヤとする)はそれぞれの特性を生かし高機能と省電力化やコストダウンが行われている。

以前、下位レイヤは CPU、ロジック IC、OSC、トランジスタ、抵抗、コンデンサなどの部品で構成されていた。ASIC や FPGA(以後デバイス)が前述した下位レイヤの部品を集約し 1 チップ SoC (System-On-Chip) 化を行っている。また、中位/上位レイヤも機能の集約が進み膨大な機能を制御可能としている。これによりテストの方法も変化してきている。

1.1. 現状の分業制テストにおける問題点

現状実施されている各レイヤの分業制テスト(図 1 参照)では、『レイヤにおける機能間接続のテスト(以後、レイヤ内テスト)』を重視し、合否判定は各レイヤから

の出力結果にのみにより判定を行っている。(図 1 参照) 『機能におけるレイヤ間接続のテスト(以後、FuST(Function slice Test)』は現状あまり行われていない。

上/中位レイヤから下位レイヤ機能を制御時において出力特性値を網羅できていない場合がある。下位レイヤ制御不良時の出力特性値異常は組み込みデバイスの性能や品質に直接影響する。現状は、機能や特性値にウェイトを置いたテストは考慮されていない場合が多い。レイヤ内テストが完了しても要求される特性値を満たしているかは判らず、全てのレイヤをコミットしなければ特性値のテストが出来ない。

また、レイヤ内テストだけでは機能におけるレイヤ間接続がテストされていない場合が多く、各機能においてのレイヤ間接続テストの微妙な認識のズレや見解の違いが発生しやすく、テスト項目の漏れやテスト項目の偏りなどが発生している。

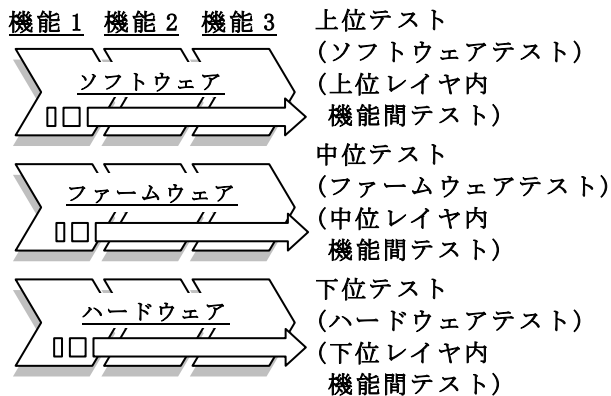


図1 現状のテスト構成

『レイヤにおける機能間接続のテスト(レイヤ内テスト)』

2. FuST の構築

今回は最も開発インパクトの大きい下位レイヤ不具合の早期発見と、リワーク回数の減少、十分性と網羅性向上を目的としてテストを行った。

現状のレイヤ内テスト(図1参照)に加えて取り組んだのが FuST (図2参照)である。

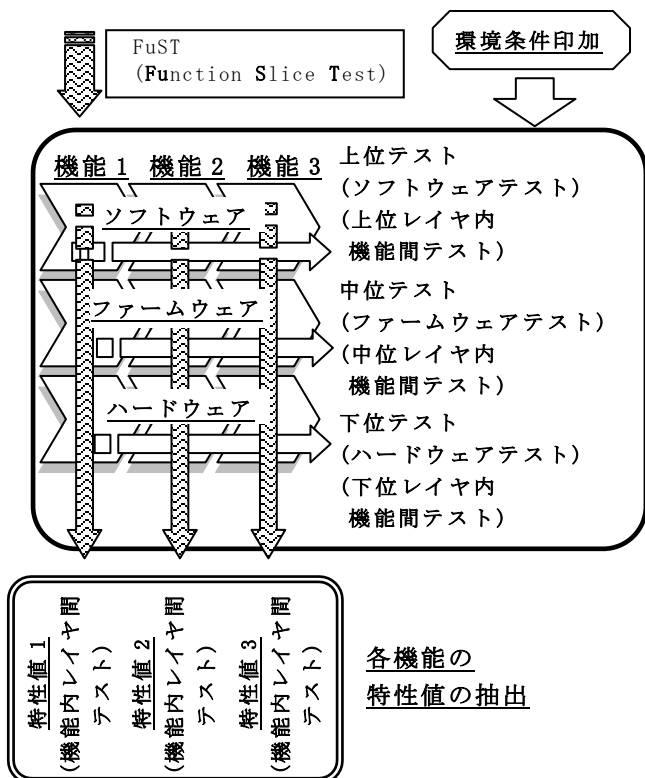


図2 機能内におけるレイヤ間テスト(FuST)構成

通常のレイヤ内テスト実施時は、各レイヤが組み込みデバイスの機能に影響しているかが判断できない。これに対して FuST は、機能ごとにレイヤを切り出して行うテスト方式である。このため機能に対して各レイ

ヤの影響要因が容易に判断できる。

2.1. 実施モデル

FuST 実施モデルは低消費電力、暗号化、ネットワーク接続、屋外設置、画像変換をセールスポイントとしている機器に実装されるデバイス(以後、テストデバイス)である。

上/中位レイヤは下位レイヤの資源を有効かつ効率的に使う事を求められている。また、温湿度対策、放熱対策、EMI といった環境条件も考慮にいれなければならない。

今回の FuST 実施モデルのテストデバイスは大規模かつ短納期であった為、テスト設計にかかる時間が無かった。このため、一部の機能に対して FuST を適用し有効性の確認を行った。

2.2. テスト方法の選択

システム設計書や上/中/下位レイヤの仕様書、設計書などの開発成果物を分析し、テスト項目設計に必要な要素を抽出した。

テスト方法は実験計画法を用いた。合理的に実験の計画すなわち因子を組み合わせる方法を定め、因子間の関係が統計的に直交する性質を持つ直交表を用いているため、各因子の水準がバランスよく(同じ回数)出現するのが特徴であり、影響要因特定の容易性という点で有効であると考えた。直交表は「どの要因が特性値に影響を与えているのか」を効率的に解析することを目的としている。

2.3. テスト項目抽出

2.2 項の要素リストをもとにテスト項目の抽出を行った。上/中/下位レイヤ間に接続される機能、下位レイヤの特性値が影響する機能、シミュレーション実施結果で消費電力が大きい機能を中心に抽出し要素をリストアップした。また、過去のテスト項目を集め、機能に不具合が発生した項目もリストに加えた。

これらの抽出したテスト要素リストをもとに、使用可能な機能と同時動作可能な機能、およびそれらのパラメータを抽出、整理し状態遷移図を作成した。(図2.3)

図2.3の状態遷移図より、制御因子と信号因子のみを抽出した。今回は影響要因の特定をメインに考えているため誤差因子の抽出は行わなかった。

また、FuST 実施時の消費電力値、テストデバイス内マクロの電圧を記載する欄を設けた。これにより上/中位レイヤ動作時の下位レイヤにおける機能と特性値をテストした。上記のテスト項目に環境条件を印加しテストを行った。

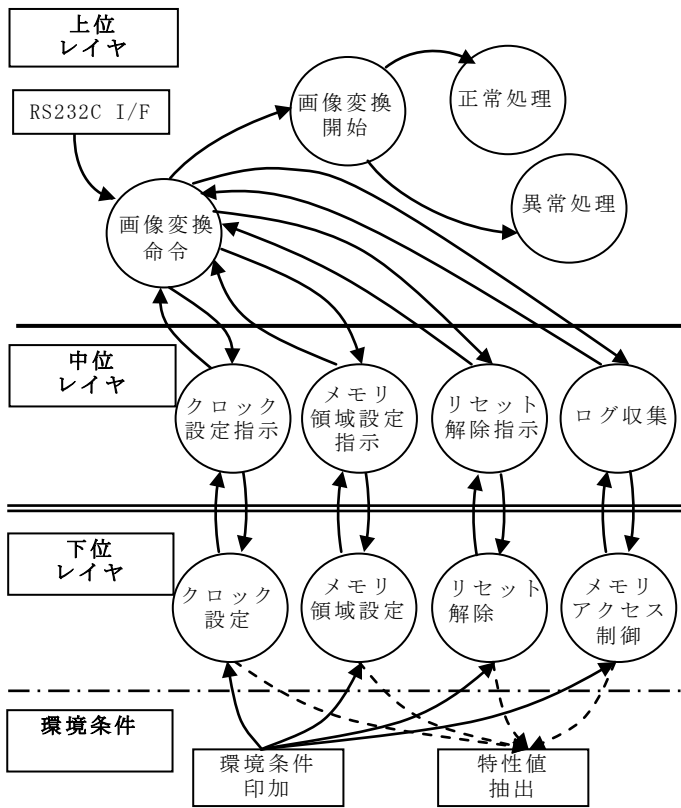


図 2.3 状態遷移図例

表 2.3 直交表によるテスト項目の抽出例

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0:正常/1:不正	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON	0:OFF/1:ON
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	0	0	0	0	0	0	0	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0	0	0	0	0
8	0	1	1	1	0	0	0	0	0	0	0	0	0	0
9	1	0	1	0	0	0	0	0	0	0	0	0	0	0
10	1	0	1	0	0	0	0	0	0	0	0	0	0	0
11	1	0	1	1	0	0	0	0	0	0	0	0	0	0
12	1	0	1	1	0	0	0	0	0	0	0	0	0	0
13	1	0	1	1	0	0	0	0	0	0	0	0	0	0
14	1	1	0	0	0	0	0	0	0	0	0	0	0	0
15	1	1	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	0	1	0	0	0	0	0	0	0	0	0	0

3. FuST 実施

図 3 の構成にてFuSTを実施した。LabVIEW^{*1}システムを使い統合的な評価環境を構築した。

テストは表 2.3 のテスト項目をもとに機能動作時のシナリオを作成した。

テストデバイス内の RS232C(GPIB ポート)インターフェースを使いシナリオをキックし、テストデバイス内の各マクロからの返り値、周辺デバイス(α~γ)からの出力値をオシロスコープ、ロジスティックアナライザと LabVIEW を使用し出力値を確認した。

この出力値を確認することで機能におけるレイヤ間テストの確認を行った。また、上位/中位レイヤ動作時における下位レイヤの電源電圧値、テストデバイスとしての消費電力値も測定した。これらを表 2.3 に記入し機能と特性値の合否判定を行った。合否判定基準は仕様より抽出した。

表 2.3 によるテストで設計条件規格値のマージンが少ないマクロの電圧を規格値の許容範囲まで上下させ機能動作時の特性をテストした。

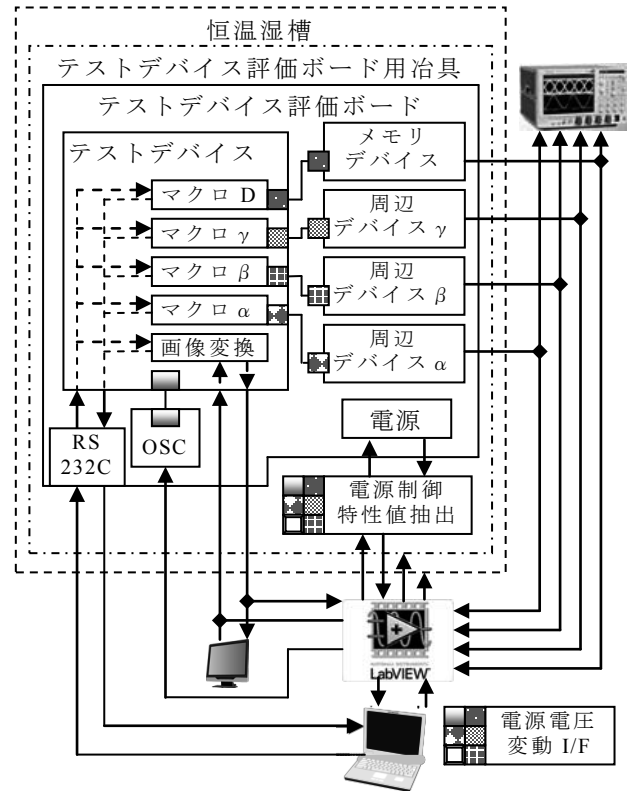


図 3 FuST実施構成

*1 LabVIEWは日本National Instruments®の登録商標です。

3.1. 実施結果

表 3.1.1 に FuST 実施後の結果例を記載した。特性値による判定を行い 3 段階で分けた。これにより特性値の悪いマクロや機能を整理した。Fault Tree を抽出し再テストを行った。

表 3.1.2 には不具合抽出例を記載した。FuST 実施において発見された比較的大きな不具合を記載した。

1 項のクロック回路の不備はレジスタにより振幅を Gain 側と Loss 側へ設定できる予定だったが、テストデバイス内の配線不備により不具合が発生した。機能動作時において消費電力増大すると振幅不良が発生し、テストデバイスが Boot しないなどの間欠不具合が発生した。

2 項は同一電源のマクロがスリープモードからの復

帰時、電源とクロックが安定する前に中/上位がリセットを解除してしまった。これにより下位レイヤが異常データをラッチしてしまい、割込み信号等を誤検出し不具合が発生してしまった。

3項はマクロ仕様検討時には計算上マージンが取れていたが、上位/中位が高負荷並列動作時に、電源電圧が+側、-側に変動した場合、瞬間的にCPUのコアとマクロαの電位が逆転することが判明した。

条件が重なると不具合が発生するため、電源回路の変更を行い、テストデバイスの仕様へリスクとしてフィードバックを行った。

シミュレーションの結果と同じく、マクロα,β,γが動作中は消費電力が高い事が判明した。特にマクロαの影響が大きくマクロαの再検討と再構築を行った。マクロαは機能を更に分割し細かく制御することし低消費電力化を行った。また、マクロα,β,γと画面リサイズ機能は共に消費電力が大きいため並列動作を行わないようにシーケンス上の変更を行った。

今回のテストで最大の消費電力値が算出でき、目標値を上回り消費電力値の対処もできた。

表 3.1.1 直交表によるテスト実施例

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	電圧値	テストデバイス内の電圧値α	テストデバイス内の電圧β	テストデバイス内の電圧γ	テストデバイス内の電圧δ	機能動作判定	特性値判定
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1481	1063	1110	1702	2488	○	Blue
2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2100	1062	1102	1701	2491	○	Red
3	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1690	1061	1202	1737	2503	○	Blue
4	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1680	1063	1202	1737	2503	○	Blue
5	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1980	1108	1087	1602	2350	○	Yellow
6	0	1	1	0	0	1	1	0	0	1	1	0	0	0	1905	1107	1089	1601	2380	○	Yellow
7	0	1	1	1	0	0	0	0	0	1	1	1	0	0	1610	1100	1202	1739	2391	○	Blue
8	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1675	1101	1203	1800	2375	○	Blue
9	0	1	1	0	1	0	1	0	1	0	1	0	1	1	1681	1101	1089	1798	2503	○	Blue
10	1	0	1	0	1	0	1	1	0	1	0	1	0	1	1531	1102	1089	1801	2503	○	Blue
11	1	0	1	1	0	1	0	0	1	1	0	1	0	0	1512	1103	1202	1702	2495	○	Blue
12	1	0	1	1	0	1	0	1	0	0	1	0	1	0	1556	1101	1203	1704	2496	○	Blue
13	1	1	0	0	1	1	0	0	1	0	0	1	1	0	1680	1055	1111	1900	2353	○	Yellow
14	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1693	1058	1112	1798	2398	○	Yellow
15	1	1	0	1	0	1	0	1	0	1	0	0	1	0	1676	1058	1202	1603	2359	○	Yellow
16	1	1	0	1	0	1	0	0	1	0	1	1	0	0	1695	1057	1203	1601	2350	○	Yellow

表 3.1.2 不具合抽出内容例

No	不具合抽出レイヤ	不具合の内容	不具合の原因
1	中位/下位	テストデバイス内のクロック回路と設定	仕様 下位レイヤ (テストデバイス不具合) 中位レイヤ設定不具合
2	上位/中位	クロックスリップとリセットの機能整合	仕様 下位レイヤ (テストデバイス不具合) 上位/中位レイヤ設定不具合
3	下位	中位/上位が高負荷制御時に下位レイヤが電源電圧変動時にテストデバイス内の電源電圧が逆転する。	仕様(テストデバイス)

3.2. 実施時の工夫と課題

今回、下位レイヤ上の特性値抽出がテスト上重要な要因であるため、下位レイヤの担当者が主体となってFuSTを実施した。

また、温湿変動を加える事で、プロダクトテストよりも厳しい環境にてテストができた。

オートマチックにテストが行えるように環境を構築し、シナリオを使いテストを自動化した。温湿変動などは時間を要するため、夜間帯を利用して無人環境でテストを自動実行し作業効率の改善を行った。

テストデバイスの異なるバージョンに対しても全く同じ環境、同じ条件でテストが実施でき、特性値のばらつきが抽出できた。回帰(リグレッション)テストにも有効であった。

このテスト環境を作り出すには、下位レイヤ設計時に特性値を抽出する専用の外部回路システムとそれに連動する環境構築も重要であった。

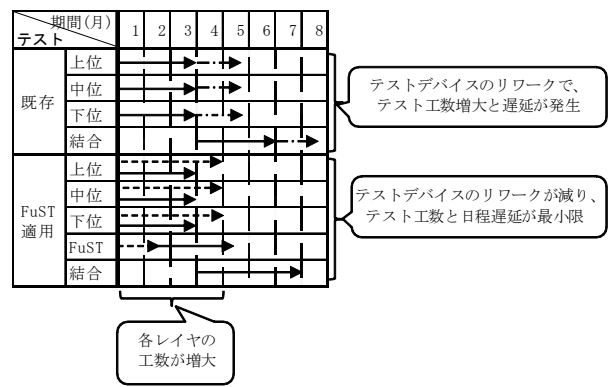
テスト項目を継続して抽出することや、NDA や特許の制約があるため、上/中位レイヤはパッケージによるコードの支給となった。このため問題があるマクロの内部への踏み込んだ解析が出来ず、原因特定に時間が掛かることもあった。

また、LabView の試験構成構築、テスト用シナリオ作成や修正にも時間が必要であり、各レイヤを統合しトータル管理や折衝を行い、テスト設計を効率的に実施する人員も必要であった。

下位レイヤによる不具合の早期発見とリワーク回数を減らすことにより、日程の削減には成功したが、短期的にテスト実施の工数は増大してしまった。(表 3.2 参照)

今後、テスト工数の削減と効率を改善する必要がある。

表 3.2 既存のテストとFuSTの比較



←→ テスト期間
 - - - - -▶ FuST は各レイヤ設計スタート後に検討準備開始
 - · - · - ▶ テストデバイスリワーク時はテスト工数が発生

4. まとめ

今回の FuST 対象はテストデバイス内の部分的な機能ではあるが、各レイヤ動作時における機能不具合検出と特性値のテストが実施できテストは成功した。

状態遷移図(図 2.3 参照)を作成することで、特性値に影響する機能とレイヤ間の繋がりを明確にできた。テスト要素リストを抽出する作業が重要であり、更

に改善する必要があると感じた。平行して行ったシミュレーションと FuST の統合がとれ、有効性を証明することができた。今までのテストには無かった『機能におけるレイヤ間接続のテスト』のプロセスを実現できた。シミュレーションでは検出することが出来ない不具合(表 3.1.2 参照)を発見することができ、不要なリワークが減らせた。

今回、テストデバイスが大規模な為、シミュレーション処理(検証)に時間を要した。実機での確認が高効率であった。

FuST は、機能ごとの切り出すため、影響度の判断が容易であり、今までにはない視点でのテストが実施できた。

5. 今後の取組み

組込みデバイスは更なる高機能化、省電力化が予想され、レイヤ内テストだけでは不十分になる事が予想される。各レイヤの設計者も他レイヤの機能を理解し、シームレスな関係でなければ設計できなくなる。テストする側も設計思想と全レイヤを理解しテスト設計を実施していく必要があると考えられる。

段階的にテスト対象機能をズームイン/アウトを行い、誤差因子も含んだ FuST を確立したい。

また、シミュレーションと連携して工数削減を行い機能や性能をテストし組込みのシステムの品質を向上を行いたい。

文 献

- [1] Lee Copeland 宗雅彦訳, "ソフトウェアのテスト技法", 日経 BP 社, 2009
- [2] 吉澤正孝, 秋山浩一, 仙石太郎, "ソフトウェア HAYST 法", 日科技連, 東京 2008.
- [3] Glenford J. Myers 長尾真監訳, 松尾正信訳, "ソフトウェア・テストの技法", 近代科学社, 東京都, 1981.
- [4] 日経ビジネス 2005 年 4 月 25 日・5 月 2 日号: ソフトが危ない 品質崩壊 クルマも電機も鉄道も, 日経 BP 社(2005).
- [5] 岩崎学, "統計的データ解析入門 実験計画法", 東京図書株式会社, 東京 2006.
- [6] 日経 SYSTEMS 2006 年 8 月号: ソフトウェア改造力, 日経 BP 社(2006)
- [7] 日経 エレクトロニクス 2010 年 11 月 29 日号: 組み込みソフトウェアの未来, 日経 BP 社(2010)