

再利用性が高い RAM 診断技術の実現方式

市岡 怜也†

三菱電機株式会社 情報技術総合研究所†

1. はじめに

本研究は、RAM の診断技術に関する研究である。既存の RAM 診断技術は RAM 全体を独自の手順で書き換えてしまいデータを破壊してしまうため、RAM を利用しないようアセンブラで実装するしかない。そのため、マイコンの供給が終了するたびに再開発が必要となり、再開発コストが多くなるという課題がある。

そこで、我々は、上記の問題を解決する再利用性が高い RAM 診断の実現方式を提案する。具体的には、アセンブラでしか実装できなかった既存の RAM 診断技術を、C 言語で実装できるように拡張する方式である。提案方式を適用することで、マイコン変更に伴う再開発時には RAM 診断技術を実装したソースコードのコンパイル作業のみでよく、再開発のコストの削減が可能となる。

以下、本稿では、2 章で既存技術と課題について述べ、3 章で提案方式について説明する。

2. 既存技術と課題

既存の RAM 診断技術として、Abraham[1]や March などが挙げられる。これらの手法は、RAM のセルに対してデータの読み書きを実施し、その後、RAM のデータが期待値かどうかを確認することで RAM の故障を判断する手法である。

これら既存の RAM 診断技術は、C 言語で実装することができない。理由は、RAM 診断を実装したプログラム (RAM 診断プログラム) が利用する RAM データを、自身で破壊してしまうからである。もし、C 言語で実装した場合、RAM 診断プログラムは静的変数やスタックなど RAM のデータを利用して動作する。しかし、既存の RAM 診断プログラムは RAM 全体のデータを独自の手順で読み書きする必要があるため、自身のデータを破壊し正常に動作できない可能性がある。そのため、RAM 領域を利用しないようアセンブラで実装するしかなく、マイコンの供給が終了するたびに再開発が必要となり、再開発コストが多くなるという課題がある。

3. 提案方式

3.1. 目標

本研究の目標は、RAM の診断技術において、マイコン変更に伴う RAM 診断プログラムの再開発時のコストを削減することである。本稿では、この目標を実現する技術として、既存の RAM 診断プログラムを C 言語で実現可能とする方式を提案する。具体的には、RAM 診断プログラムが利用する RAM データを保護する仕組みを C 言語で実現可能とする方式である。提案方式を適用することで、マイコン変更に伴う再開発時には RAM 診断プログラムを実装したソースコードのコンパイル作業のみでよく、再開発のコストの削減が可能となる。

3.2. 要求事項

既存の RAM 診断技術を C 言語で実現でき、かつ、再開発時にはソースコードのコンパイル作業のみとするためには、以下の 2 つ項目の実現が必要と考える。本研究では、これらの項目が実現できれば、目標達成の見込みが得られるとした。

- (1) RAM 診断プログラムが自身の RAM 上のデータ (静的変数やスタック) を破壊しないこと
- (2) マイコンに依存する特殊なマクロや命令を利用しない、つまり、ANSI C[2]に記載の文法のみを利用して(1)の機能を実現すること

3.3. 詳細

本項では、提案方式の詳細について述べる。提案方式では、上記の要求事項の(1)を満たすために、2 つの機能 (機能 1、機能 2) を備える。本項にてこれらの機能について述べた後、次項にて提案方式の実現例を示し、(2)を満たすことを示す。

① 機能 1

提案方式では、RAM の全領域を分割し、分割された RAM 領域に対して RAM の診断を順に実施する。診断とは、既存の RAM 診断プログラムを動作させることを示す。RAM 診断プログラムの動作に

必要となる変数は、診断の対象とならない別の領域（変数保管領域）に保管する。ただし、順に分割された RAM 領域を診断していった場合、変数保管領域が必ず診断の対象となる場合が発生し、RAM 診断プログラムが正常に動作できなくなる。この場合に備え、変数保管領域を複数用意しておき、変数保管領域が診断対象とならないように切り替えて利用する（図 1 参照）。

② 機能 2

ただし、以上の機能を実現する場合、単純に C 言語で複数の変数保管領域を宣言するのみでは、変数保管領域が破壊されてしまうという問題がある。C 言語で定数を定義した場合、その変数が割り当てられる RAM のアドレスは不定である。そのため、単純に図 1 に示したように 2 つの変数保存領域を宣言するのみでは、複数の変数保管領域が診断対象になってしまう場合が発生し、RAM 診断プログラムが正常に動作できなくなるという課題がある（図 2 に記載の課題）。

そこで、提案方式では、変数データ保存領域の宣言方法を工夫することで、これらの問題を解決する。具体的には、変数保管領域を 3 つ以上宣言し、かつ、1 つあたりの変数保管領域のサイズは必ず診断単位となる RAM の単位よりも大きく定義する（図 2 に記載の解決方法）。これにより、RAM 診断の対象がどの領域になっても、必ず診断の対象とならない変数保管領域が存在する。RAM 診断プログラムはこの領域を利用することで、正常に動作することができる。

3.4. 実現例

提案方式を利用して C 言語で RAM 診断技術を実現した場合の例を、図 3 に示す。図 3 が示すように、提案方式を利用することで、マイコンに依存する特殊なマクロや命令を利用せずに RAM 診断技術を実装できることが分かる。

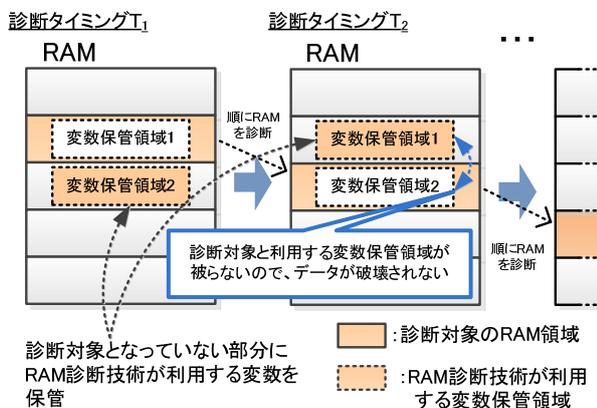


図 1 機能 1

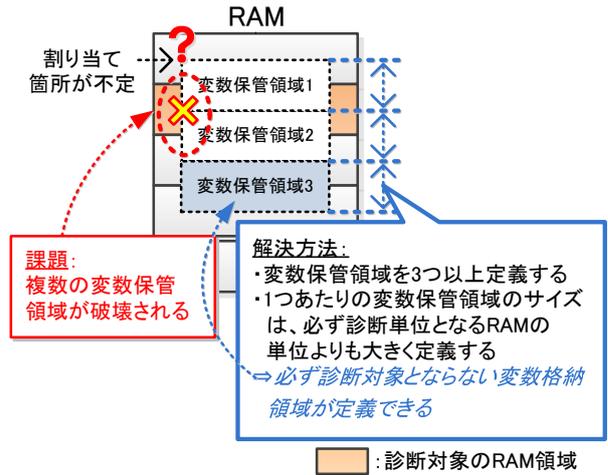


図 2 機能 2

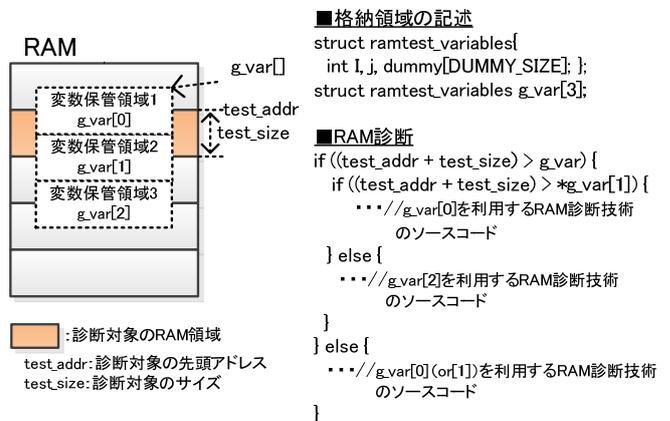


図 3 実現例

以上より、提案手法を適用することで既存の RAM 診断プログラムを C 言語で実現可能となることが分かる。つまり、マイコン変更に伴う再開発時には RAM 診断プログラムを実装したソースコードのコンパイル作業のみでよく、再開発のコストの削減が可能となる。

4. おわりに

本稿では、再利用性の高い RAM 診断技術を提案した。提案方式を適用することで、マイコン変更に伴う再開発時には RAM 診断技術を実装したソースコードのコンパイル作業のみでよく、再開発のコストの削減が可能となる。今後、提案方式を実装し、実現性の確認や処理時間を評価する予定である。

[1] R. Nair, S. M. Thatte and J. C. Abraham.: Efficient algorithms for testing semiconductor random-access memories. IEEE Transactions on Computers, C-27, pp.572-576 (1978).
[2] ISO/IEC9899:1990 Programming languages -- C