ソフトエラー耐性を強化した高信頼性安全システムの開発

樋口 敏之

ソフトエラー起因でのシステム異常により設備が緊急停止することなく、安全機能を維持しながら設備が稼働し続ける信頼性の高い安全システムを開発したので報告する。

半導体デバイスの高集積化および微細化に伴い、メモリにおける一過性のビットエラー（ソフトエラー）が問題となっている。ソフトエラーにより半導体メモリに保存されているデータが一時的に書き換わることで、システムの一時的な停止やシステムダウンを引き起こす恐れがある。半導体製造工場のように 24 時間稼働して高価な材料を取り扱う設備においては、そのような一時的なシステム停止でも過大な利益損失が発生する。

一方で、半導体製造装置などに多く使用されている安全 PLC（Programable logic controller）は、安全制御に関連するすべての半導体集積回路やメモリ回路に対して自己診断を行い、異常な動作を検出した場合は直ちに装置を停止するように制御する。そのため、ソフトエラーに対してもデータ化けが発生すると、安全 PLC は装置を停止させてしまうことから、安全機能は維持しつつ不要な停止を抑え生産性を継続する対策が求められている。

この課題に対して、安全性を損なうランダムハードウェア故障を検出して安全状態を維持する機能の実現だけではなく、ソフトエラーによるデータ化けを検出して、さらにデータの修復を行う機能も実現する技術を開発した。

本論文は、そのソフトエラーによるデータ化けを検出してデータの修復を行う機能の具体的な方策と、その効果の検証結果についてまとめる。

Development of Highly Reliable Safety System with Enhanced Tolerance against Soft-Error

HIGUCHI Toshiyuki

We report on the development of a highly reliable safety system that ensures that the equipment continues to operate while maintaining the safety function without making an emergency stop due to the system error caused by the soft error.

As semiconductor devices become more highly integrated and miniaturized, transient bit errors (soft errors) in memories are a problem. If data stored in the semiconductor memory is temporarily modified due to a soft error, a short time breakdown or a system down may occur. In equipment that operates 24 hours a day and handles expensive materials in a semiconductor manufacturing factory, temporary stoppage can cause excessive profit loss.

On the other hand, Safety PLCs (Programable logic controller), which are often used in semiconductor manufacturing equipment, perform self-diagnosis on all semiconductor integrated circuits and memory circuits related to safety control, and immediately stop the equipment when abnormal operation is detected. Even if data is garbled due to a soft error, the safety PLC will stop the equipment. Therefore, there is a need for measures to maintain productivity while maintaining safety functions and suppressing unnecessary outages.

In response to this issue, we has realized the technique not only the function of detecting random hardware failures that impair safety and maintaining the safety state, but also the function of detecting data corruption due to soft errors and recovering data.

This paper describes the specific measures for the function of recovering data by detecting data corruption due to the soft error, and the verification results of the effect.

Contact : *HIGUCHI Toshiyuki* toshiyuki.higuchi@omron.com

1. まえがき

工場内における機械設備からの人体保護を目的とした安全システムの構築において、各種の安全 PLC（Programable logic controller）が採用されている。安全 PLC とは、IEC 61508[1)] を代表とする国際安全規格の認証を取得した安全制御のためのコントロールユニットである。作業者の安全を確保するために安全 PLC が利用されることから、機械設備の危険な状態を安全 PLC 自体の故障により検知できず、安全だと判断して機械の稼働を許可させるように動作してはならない。そのため、ハードウェアおよびソフトウェアに冗長性と多様性をもたせ、安全制御に関連する部品を常時自己診断し、安全性を損なうランダムハードウェア故障を検出した場合には機械設備を安全側に停止するような機能が安全PLCには施されている。これにより、一般の PLC に対して安全性と信頼性を格段に高めている。

安全 PLC は、安全回路のソフトウェア化や安全ネットワークによる省配線により、大規模で複雑なアプリケーションにおいてより柔軟な安全システムを実現することを可能としている。そのため、大容量のプログラムを高速に処理する半導体デバイスが安全 PLC には実装されている。安全 PLC は、安全制御に関連するすべての半導体集積回路やメモリ回路に対して自己診断を行い、異常な動作を検出した場合は直ちに装置を停止するように制御する。

近年の半導体デバイスの高集積化および微細化に伴い、メモリにおける一過性のビットエラー（ソフトエラー）が注目されている。ソフトエラーは、例えばα粒子や宇宙線中性子の衝突によって生じる。また、微細な異物によるデータ化けや、外部との入出力制御装置などから単発的に飛び込むノイズによるデータ化けも同様に問題となってきている。これらのソフトエラーにより、半導体メモリに保存されているデータが一時的に書き換わることで、システムの一時的な誤動作やシステムダウンを引き起こす恐れがある。半導体製造工場のように 24 時間稼働して高価な材料を取り扱う設備においては、そのような一時的なシステム停止でも過大な利益損失を発生させるため、その対策が求められている。

ソフトエラーによるビット反転を検出する診断、およびビット反転の発生したデータの修復については、Error Correction Code（以下、“ECC”という）を利用することにより実現できることが一般的に知られている。しかし、そのような ECC 機能による対策を安全 PLC に適用するためには、ECC を生成・チェックする特別なハードウェアが搭載されたメモリや MPU に変更しなければならないためコスト高となる。また、デバイスの変更により安全PLCの故障分析と安全性評価を再度実施しなければならず、機種毎にそのハードウェア開発と評価が別途必要となる。

そこで、ソフトウェアによる対策だけでソフトエラー耐性強化をはかる安全 PLC の開発に取り組んだ。ソフトウェアによる対策を施すことは、追加のハードウェア回路を必要としないためコストアップとならないことや、既存の安全 PLC やその他の安全コンポーネントにも展開できる、という利点がある。

2. ソフトウェアによる対策の課題

安全 PLC を含む多くの既存の安全コンポーネントは MPU を冗長化している。データ化けを検出する方策として、図 1 に示すように MPU 間でデータをチェックする方法を採用している。これは、データの比較によりデータ化けを含む MPU の異常状態の検出が可能であるが、データ化けの修復までは行ってはいないことが現状である。

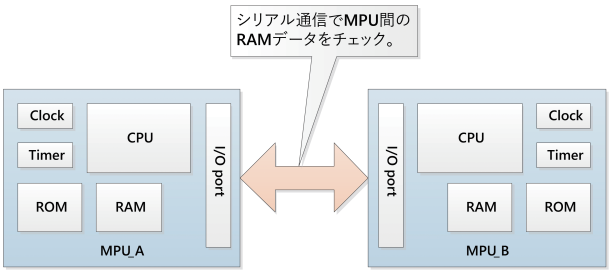


図 1　MPU 間でのデータチェック

そこで、従来技術として各 MPU で変数データを 3 重化してチェックする方法が挙げられる。3 重化によるチェックは、データ化けを検出して、さらにデータの修復を行う方策として高い効果が見込まれる。しかし、3 重化による変数データのチェックは、予めプログラム上で変数データが定義されている必要があり、スタック領域といった一時的にしか使われないデータに対しては保護できないリスクがある。また、3 重化によるチェックは変数データの読み書き時に行われるため、使用頻度の低い変数データに対してはチェック間隔が長くなり、複数の変数データでデータ化けが発生し修復されない可能性がある。

3. 対策

前述の課題に対して、各 MPU にてデータ化けを検出してデータの修復を行う機能の実現のため、以下の 3 つのソフトウェアによる対策の検討を行った。

① 変数データの 3 重化
② スタック領域の保護
③ 周期検査によるエラー累積予防

3.1 変数データの 3 重化

プログラムのソースコードにて、メモリで扱うデータを読み書きする記憶域として変数が定義される。代表的なプログラム言語としては C 言語があるが、今回の開発では C++ 言語を使用した。その C++ 言語で変数として宣言され

る Static 変数および Auto 変数に対して、プログラム上で冗長化された変数として扱うように宣言することとした。そして、対象となる変数を演算処理で使用する前に多数決判定を行う。これにより、変数のデータ化けを検出し、データの修復を行うことができる。図 2 に変数データの 3 重化処理の詳細を示す。

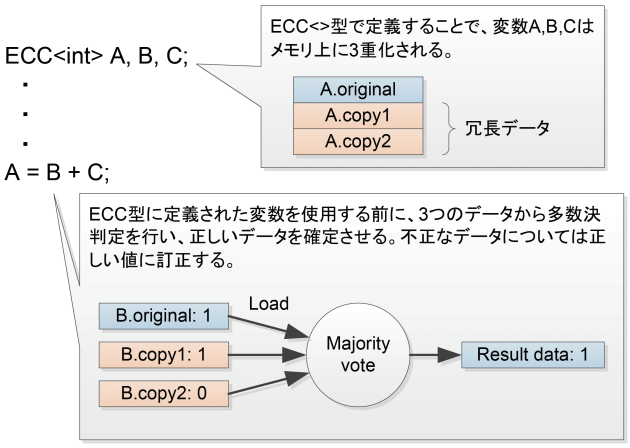


図 2　変数データの 3 重化処理

3 つのデータのうち、いずれのデータも一致しない場合はエラーとする。図 3 に多数決処理の詳細を示す。

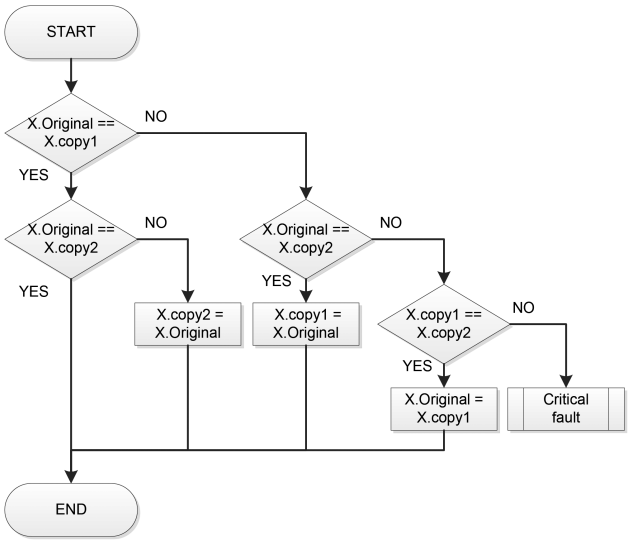


図 3　変数データの 3 重化処理

3.2 スタック領域の保護

メモリを用いて演算処理を実行するプロセッサは、レジスタを内蔵する。レジスタは、計算やメモリの特定アドレスを指すポインタ用途などに使用される。メモリ上にはそのレジスタのデータを一時的に保存するスタック領域がある。プログラムのメインルーチンを実行中に関数処理（サブルーチン）が発生すると、メインルーチンを中断させるためにスタック操作が行われる。スタック操作には以下の 2 つの処理がある。

（1）サブルーチン処理開始時に、プロセッサに内蔵されているレジスタに保持されたデータを RAM（Random Access Memory）のスタック領域に一時退避させる操作（Push 操作）。
（2）サブルーチン処理終了時に、スタック領域に退避させたデータをレジスタに復帰させる操作（Pop 操作）。

スタック操作によってデータが一時退避されるスタック領域についてもソフトエラーが起こり得る。しかしながら、サブルーチン処理に伴うスタック操作のコードは、コンパイラによって自動生成される。そのため、汎用のコンパイラを用いる場合、スタック操作に対して 3.1 項に記載の変数データの 3 重化処理を適用することができない。

そのため、サブルーチン処理の先頭でスタック領域に退避させたデータの複製を RAM 上に作成することで冗長化し、サブルーチン処理終了時に冗長化されたデータをチェックすることとした。

図 4、5 にスタック領域の 3 重化処理の詳細を示す。なお、3 重化されたデータをチェックする多数決処理は図 4 に示す方法と同様である。

- funcY の functionEntry 処理において、funcY 呼び出し前のスタックポインタ（=prevSP）と funcY 呼び出し後のスタックポインタ（=nowSP）を比較し、増加したデータのコピーを RAM 上に作成する。
- funcY の実行を終了する直前に functionExit 処理にて増加したデータとコピーしたデータとを用いて多数決判定を行う。

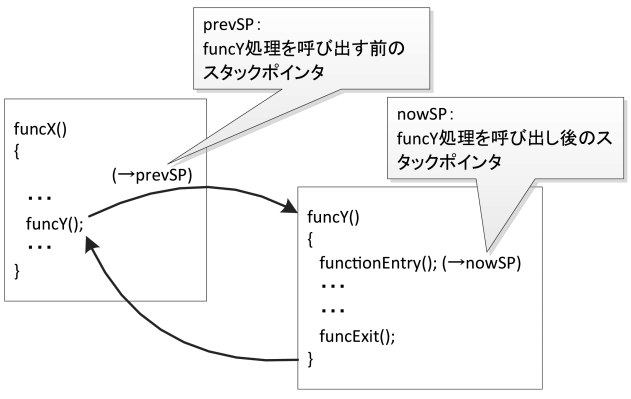


図 4　スタックポインタの読み出し

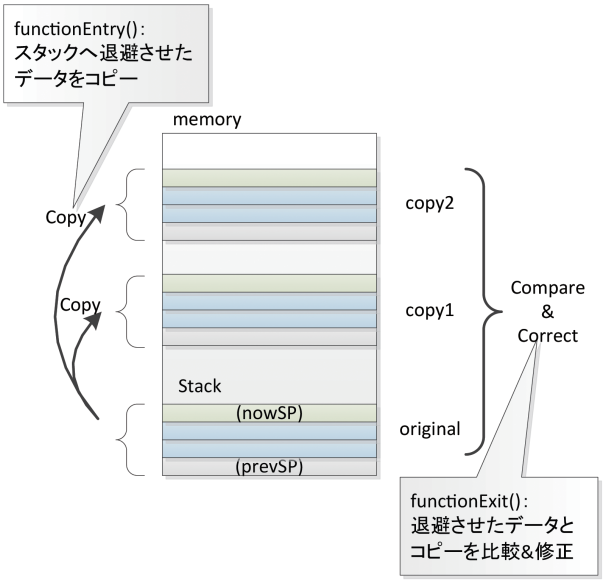


図 5　スタックの 3 重化

3.3 周期検査によるエラー累積予防

3.1 項に記載の変数データの 3 重化処理は、プログラムの実行によって変数を読み出す際に、その読み出しに該当するアドレスのデータは修復される。ただし、各変数が読み出される頻度は、プログラムに依存する。読み出される頻度の高い変数については、データの修復される機会が多くなる。しかし、読み出される頻度の低い変数については、データの修復される機会が少なくなり、エラーが累積することで RAM 上の 3 重化された変数のアドレスのうち複数のアドレスに対してソフトエラーが起こり得る。

そのため、メインルーチンおよびサブルーチン処理で実施される演算処理などのタスクの実行とは別に、ダミーの処理部を設け、予め定められた周期毎に変数を読み出すダミー処理を実行することとした。

図 6 にタスク実行部とダミー処理部と書込読出処理部を示す。書込読出処理部の第 1 ～第 3 アドレスはメインメモリのアドレスを示し、第 1 アドレスはオリジナルの変数データ、第 2 と第 3 アドレスはコピーの変数データを示す。

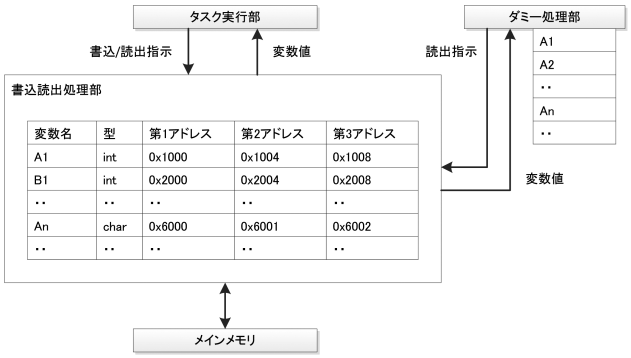


図 6　タスク実行部とダミー処理部と書込読出処理部

図 7 にダミー処理部によるデータ化け修復の例を示す。変数 A1 が使用されるタスクの発生頻度は低いが、一定周期毎にダミー処理が実行される。そのため、変数 A1 が使用される次のタスクまでにデータ化けが発生してもデータは修復され、エラーの累積予防が可能となる。

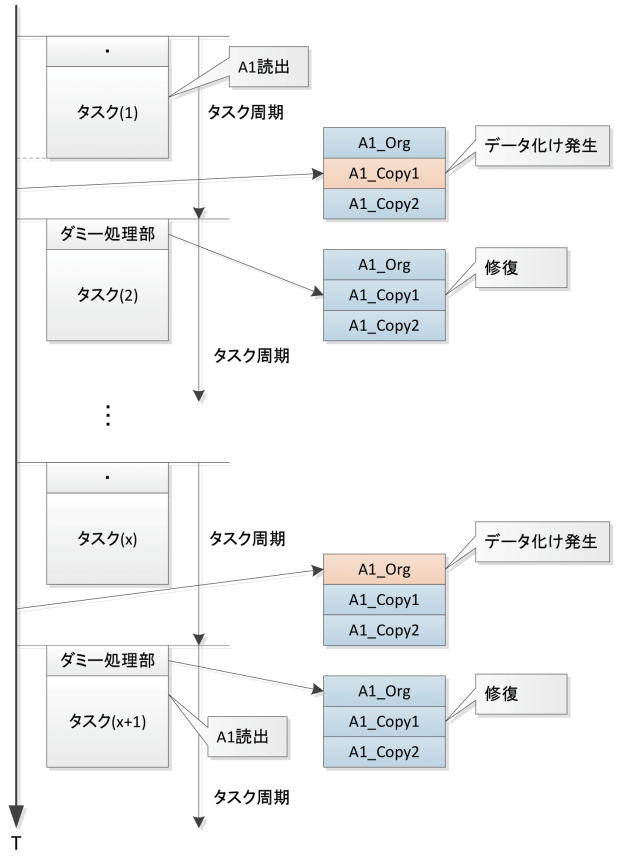


図 7　ダミー処理部によるデータ化け修復

4. 検証

前述した「変数データの 3 重化」「スタック領域の保護」「周期検査によるエラー累積予防」の 3 つの方策によるデータ化け対策の効果を検証するためには、ソフトエラーを発生させてデータ化けが発生したメモリの修復度合いとシステム異常の発生頻度を計測する必要がある。そこで、以下の 2 つの方法により確認を行った。これらにより、短期間で加速的にソフトエラー評価を行うことができる。

- 疑似ソフトエラー処理による評価
- α 線照射試験による評価

4.1　疑似ソフトエラー処理による評価

疑似ソフトエラー処理による評価は、擬似的にビット化けを発生させるソフトウェアバッチ処理をプログラムに実装して行う評価である。RAM エリア全体に対して網羅的にビット化けを発生させ、システム異常発生時のビット化

けが発生した箇所を特定することができるよう、その評価方法と構成の検討を行った。

図 8 に疑似ソフトエラー処理による評価構成を示す。以下の手順により、ビット化けの位置情報と異常情報とが対応付けされてデータベースに蓄積することができる。

① パーソナルコンピュータにて安全 PLC の内のメインメモリ内のビット化けを発生させる 1 ビットの位置を指定する。
② 安全 PLC 内のソフトウェアバッチ処理にてメインメモリの指定された位置の 1 ビットのデータを反転させる。1 ビットのデータを反転させた後の所定時間内に発生した異常に関する異常情報を収集する。
③ パーソナルコンピュータは、安全 PLC から異常情報を取得する。
④ パーソナルコンピュータは、①において指定した位置に関する位置情報と③において取得した異常情報とを対応付けてデータベースに登録する。

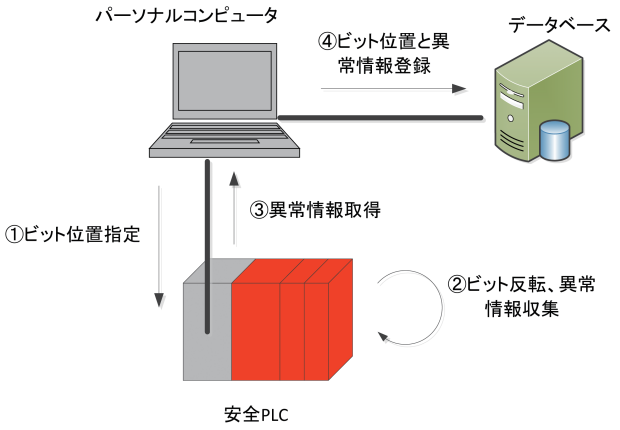


図 8　疑似ソフトエラー処理による評価構成

表 1 は、データベースに登録された位置情報と異常情報の一覧の例である。データベースを参照することで、システム異常発生時の原因分析とソフトエラーが発生したときの影響を容易に確認することができる。

表 1　位置情報と異常情報のデータベース

No.	アドレス	bit 位置	変数名	異常 ID	ソースコード（ファイル名、行番号）
1	0x200010AC	1	val_A	30	file_A.c, 867
2	0x20006000	7	val_B	57	file_B.c, 620
3	0x20007032	3	val_C	42	file_C.c, 500
4	0x20009A00	4	stuck	30	file_B.c, 827
⋮	⋮	⋮	⋮	⋮	
N	0x20002000	5	val_X	77	file_A.c, 712

表 2 に疑似ソフトエラー処理による評価結果を示す。同一のハードウェアを用いて、ソフトエラー対策を実装する前のソフトウェアと対策後のソフトウェアとのそれぞれに擬似的にビット化けを発生させるソフトウェアバッチ処理を実装してランニング試験を行った。結果、対策前のユニットに対するソフトエラー対策を実装したユニットのシステム異常発生率比は 0.00082 となった。

表 2　システム異常発生回数

	対策前のユニット	ソフトエラー対策後のユニット
a. ビット化けを発生させた回数	77,465 回	289,636 回
b. システム異常の発生回数	1,630 回	5 回
c. システム異常の発生割合（b/a）	0.021041	0.000017
d. 対策前後のシステム異常発生率	1	0.00082

4.2 α 線照射試験による評価

JEDEC JESD89[2] には、以下の 3 つのソフトエラー評価試験が規定されている。

JESD89-1：フィールドテスト
JESD89-2：放射性物質を用いた α 線照射試験
JESD89-3：加速器を用いた中性子線照射試験

フィールドテストは、多数のサンプルを長期間かけて評価を行うため、コスト高で時間もかかる。加速器を用いた中性子線照射試験は、中性子線照射が可能な施設が限られており、容易には行うことができない。一方で、放射性物質を用いた α 線照射試験は、線源があれば短時間で実施することが可能である。今回は、実際にソフトエラーを発生させてソフトウェアによる対策効果を早期に確認するため、簡易的に加速試験が行える α 線照射試験を参考にした。

図 9、10 に試験構成を示す。メモリが搭載されている MPU に対して α 線源を直接照射し、製品の動作を観測する。

- 製品に MPU が実装された状態で、MPU のパッケージ表面を開封して内部の IC チップを露出させる。
- IC チップ上面に α 線源である ^{241}Am（アメリシウム）を配置する。
- 製品を動作させ、挙動を観測する。システム異常が発生した回数と時間を計測する。

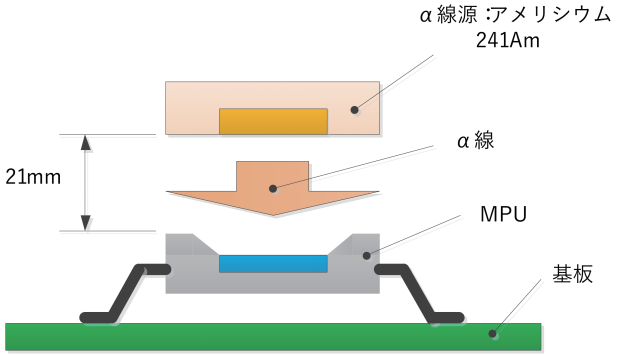
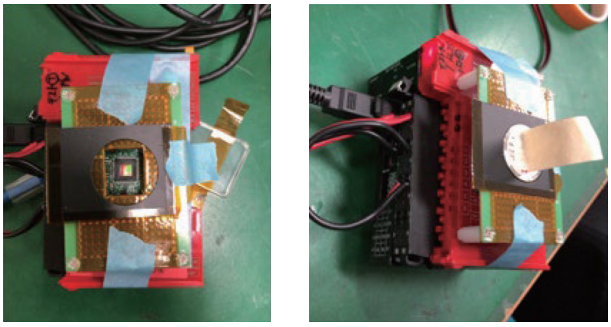


図 9　α線照射試験構成



α線の配置前　　α線の配置後

図 10　疑似ソフトエラー処理による評価構成

表 3 に既存のユニットとソフトエラー対策を実装したユニットのα線照射試験結果を示す。結果、既存のユニットに対するソフトエラー対策を実装したユニットのシステム異常発生率比は 0.0012 となった。

表 3　システム異常発生回数（α線源との距離：21mm）

	既存のユニット	ソフトエラー対策後のユニット
a．累積照射時間	76 分	13,000 分
b．システム異常の発生回数	5 回	1 回
c．システム異常の発生間隔（b/a）	15.2分/回	13,000分/回
d．対策前後のシステム異常発生率	1	0.0012

なお、疑似ソフトエラー処理による評価とα線照射試験による評価とで、システム異常発生率比が異なっている。これは、ソフトウェアを実装して動作するユニットの MPU のパッケージ表面を開封して試験を行うため、試験に用いたユニットを同一のハードウェアで比較試験ができていないことが挙げられる。また、アメリシウムから放射されるα線の流速量（α粒子の数）がアメリシウムと MPU との距離によって変化するため、α線の流速量はα線源を固定する治具の精度に依存する。今回は簡易的な固定方法で試験を行ったため、同一のα線の流速量で比較できていない可能性がある。これらは今後のソフトエラー評価を行う上での課題である。

5. むすび

今回はハードウェアの追加をせずにソフトウェアの変更だけでソフトエラーの耐性強化をはかる安全 PLC の開発に取り組んだ。

「変数データの 3 重化」「スタック領域の保護」「周期検査によるエラー累積予防」の 3 つのソフトウェアによるソフトエラー対策は、メモリに対して網羅的にソフトエラーを発生させる疑似ソフトエラー評価により、対策前のシステム異常発生率に対して 0.0009 以下に低減できることを確認した。また、α線照射試験により、実際の放射線が発生する環境においても、ビット化けを修復して動作し続けることが確認できた。本技術を用いることで、ソフトエラーによるシステムダウンの発生頻度を約 1/1,000 に改善できる効果が見込める。

今回は安全 PLC を対象としたが、ソフトウェアによる方策を用いていることから他の安全コンポーネントへの展開と応用が容易である。国際安全規格で要求される安全性を損なうランダムハードウェア故障を検出して安全状態を維持するといった安全コンポーネントとしての機能の実現だけでなく、この技術を用いることで、ソフトエラーによるデータ化けを検出して、さらにデータの修復を行う機能もソフトウェアにより実現できる。そして、ビット化けによるシステム異常が発生することなく動作し続ける信頼性の高い安全システムの構築が可能となる。

今後の課題としては、データ化けの監視および修復を行う処理時間の短縮と高速化である。この技術は、ソフトエラーによる影響が懸念される大容量のメモリを扱うネットワーク機器を扱ったアプリケーションに対しても同様の効果が期待できる。ネットワークを含めた安全システム全体の応答時間の高速化と合わせて検討していく。そして、安全機能を維持しながら設備が稼働し続ける高信頼性安全システムの更なる創出により、生産性の向上に貢献していきたい。

参考文献

1）IEC61508:2010. Functional safety of electrical / electronic / programmable electronic safety-related systems.
2）JEDEC JESD89:2006. Measurement and reporting of alpha particle and terrestrial cosmicray-induced soft errors in semiconductor devices.

執筆者紹介

樋口 敏之　HIGUCHI Toshiyuki
インダストリアルオートメーション
ビジネスカンパニー
商品事業本部 セーフティ事業部　開発部
専門：安全工学、ソフトウェア工学
