Hardware Implemented Fault-Tolerance and Safety for Programmable Automation Systems

A.E. Weinert

Siemens AG, Systemtechnische Entwicklung
Rheinbrückenstrasse 50
D-7500 Karlsruhe 21, BRD

Delft Progr. Rep., 11 (1986-1987) pp. 255-265
Received: May 1987

Also caused historically, the split-up between fail-safe-technology and the other automation-systems -- including the fault-tolerant ones -- seems to be less and less justifiable regarding the growing complexity of both the processor-based automation-systems and the controlled processes. Reasons for this statement are given in the treatise.

A widely used, serialy produced industrial automation system providing a common base for non-fault-tolerant standard as well as for fail-safe and highly available configurations fully compatible to each other, would overcome this split-up and all disadvantages due to it. One exemplary benefit also in regard to safety, of such general a compatibility is the feature, that the user's sight of the safety-relevant systems in his plant (concerning operating, controlling, observation and maintenance) is the same as of his (many) other systems he is accustomed to.

The modular TELEPERM M-System is an applied example for reaching these aims while fulfilling very strict real-time requirements by providing hardware-implementable fault-tolerance (HIFT). An indispensable condition for HIFT is a general non-diversity of all redundant configurations. Because of the many drawbacks of diverse solutions, especially for widely used systems, the non-diversity of HIFT-systems is no handicap at all. The exemplary automation system complies with several safety regulations for oil, gas and coal-dust fueled steam boilers.

Certain properties of the exemplary system, such as the use of special bit-slice-processors, did back up a HIFT-solution. Ongoing research and developpement activities showed the feasibility of this service-proved method also for systems utilizing VLSI-processors.

1. Introduction

Hardware-implementable fault-tolerance for widely used standard-type industrial automation systems will be discussed. Programmable automation-systems and, in particular, the process-related control-systems can be considered as specialized process-control-computers; there are no fundamental differences in architecture.

Compared to "normal" computers, the characteristics of automation systems can be found, besides others, in the outfit and the preponderance of input and output (IO). Thousands of analog and digital IO-channels are not exceptional. Some particularities in the configuration and operating of automation systems influenced the hardware-oriented fault-tolerance solutions presented in the following.

2. Fault-Tolerance for Automation-Systems

Redundancy is used with automation-systems aiming in two directions:

- o Fail-safe-behaviour, abbreviated "F" and
- o Higher availability, abbreviated "H" .

F : In the presence of faults in the system itself, F-automation-systems are expected to go into a simple stationary "Off-" state. The controlled process has to be tied to the system in such a manner that the off-state of the automation-system corresponds to a safe state of the concerned process. The F-principle is restricted to processes, which always exists a safe state so simply reachable for.

H : On the other, hand an H-automation-system has to maintain its specified process-control services even in the presence of internal faults. For many real-time applications the tolerating of faults must not lead to any degradation in system performance. Often an on-line-repairability of H-systems is called for. On-line-repairable H-systems can maintain uninterrupted services over many years (and faults).

Safety by fail-safe-behaviour (F) and higher availability (H) of the normal process-control functions seem to be distinct or even antagonistic requirements to an automation-system. Consequently, both fields, H and F, are usually covered by different research and development activities and by different products. Partly very strict, this separation is inherited from the non-electronical automation-technology too. In this conventional technology it was possible to produce a high-grade reliable F-behaviour for processes with a safe state always easily reachable, and it could be done at considerably low expense. An essential and consequently applied principle thereby is the correspondence between states of lower energy (hydraulic, magnetic, kinetic, potential etc.) in the automation-system and safe states of the process (generalized closed-circuit operation).

To increase the reliability the use of high-grade components and materials as well as preventive maintenance of non-electronic parts were common practice in the conventional technology, whereas fault-tolerance methods were less adequat and less usual. The preference of pure F-solutions and the restraining of H-configurations in the conventional technology is one historical cause for the split-up between the fail-safe-field and the other automation-systems. Starting from the then state of art and the

corresponding safety-rules led to highly specialized F-systems after the introduction of electronics to automation. It was tried to replicate the compononent-level F-behaviour of conventional F-systems for electronic systems, but, lacking the favourizing conditions of the conventional technology at high expenses. These dedicated F-systems fullfilled the high safety-requirements imposed to them, but were totally incompatible to the other (Non-F-) electronic automation systems. This separation of the F-field implies some disadvantages:

- Produced in small quantities, dedicated F-systems are afflicted with some problematic peculiarities. Development, test, documentation, maintenance, stocking of spare-parts and so on require great expenses; nevertheless, the broad base and quality-assesment possible with great standard-production-line systems will usually not be achieved. Special systems will never gain the ample service-proof of widely used standard-type automation-systems. In one plant the special F-system will be used besides the many "normal" standard-systems. The operating personell will have to learn the handling of both types of systems; and their experience with the many standard-type systems can not be for the benefit of the safety-critical fields.

- Pure F-systems tend to shut down (fail-safe) their controlled processes very often. Of course, this typical "over-reaction" reduces the availability, but, primarily, not the safety. Indeed, the stricter the applied safety-rules, the more likely is the fail-safe shut-down. Therefore, many customers are prepared to accept higher costs for a higher availability (H) to be combined with the requested F-behaviour.

- Such H-extensions are allowed to the user, if the F-behaviour relevant for the approvement is not afflicted. Sometimes, it fails to be noticed, that such an H-behavior, granted as beeing not detrimental to safety, actually enhances the safety. As long as the specified services of the automation-system can be maintained by H-provisions there is no need for safety-directed shut-downs. An automation-system with (theoretically) 100% availability would be 100% save. Besides, this "safety by reliabilty" is independend of the existence of a safe process-state (which would have to be always easily reachable). It can be said, that reliable H-behavior is of higher grade a property than is F-behaviour. The theoretical limit of 100% reliabilty can be met closely by means of adequat redundancy-structures and on-line-repairability.

- Besides economical reasons, even in typical fields of application for pure F-systems, safety reasons can advocate for a supplementary H-behaviour. In these cases, there exists an off-state of the controlled process allways easily reachable. Regarded by itself, this off-state is not critical and completely safe. Nevertheless, from an overall point of view one often can become aware of secondary effects by the shut-down degrading the safety at other places. For example, one transport-system is overloaded -- and hence degraded in safety -- by the fail-

safe shut-down of an other one.

Principally it seems worth while to eliminate the split-up of the safety-field leading to dedicated F-automation-systems, and, so, to overcome all the disadvatages due to it. To achieve that an automation-system should provide a common base for the following solutions, fully compatible to each other:

- o non-fault-tolerant and non-fail-safe standard-systems and
- o higher reliable H-systems,
- o F-systems suitable for safety-critical applications subject to authorization, as well as
- o HF-systems combining F- and H-characteristics in a suitable manner.

3. An Industrial Automation-System with Hardware-Implementable Fault-Tolerance as an Example and Starting-Point

It is to be shown that these properties may well be achieved by providing hardware-implementable fault-tolerance. This has been done successfuly with the automation-system TELEPERM M AS220 E.

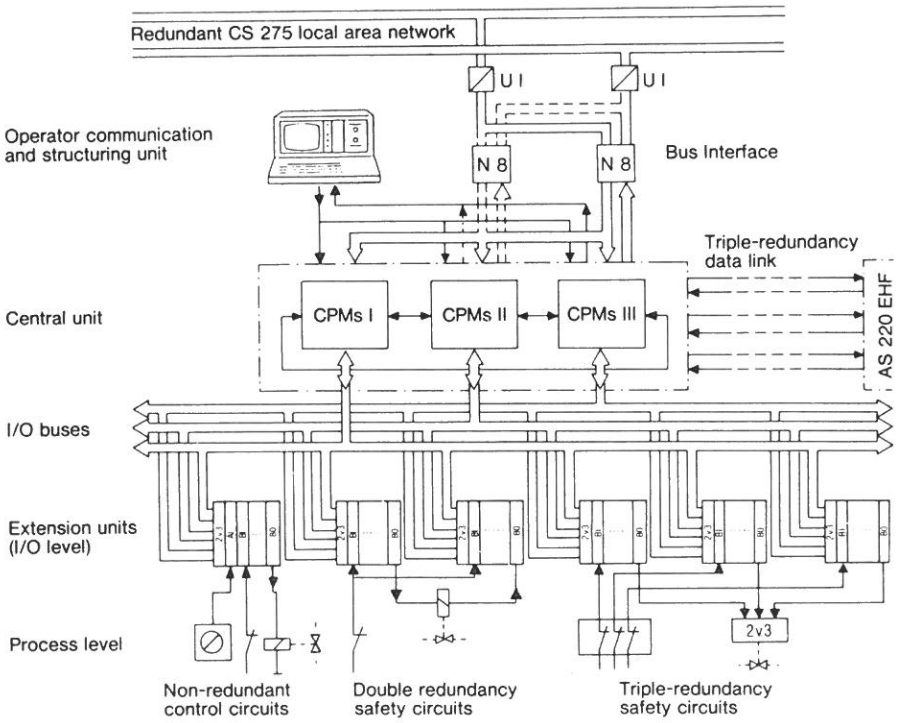
Besides the single-channel standard (1-out-of-1) possible redundancy-schemes are:

F : 2-out-of-2
H : 1-out-of-2 and
HF : 2-out-of-3 .

At the fault-tolerant or fail-safe configurations multiplicated standard-hardware runs the same standard-software in tight (clock-) synchronism; see figure 1 for the HF-configuration. Detecting, localizing and -- if applicable -- masking of faults is done by the only non-standard hardware in the modular system. Of course, these special boards for comparing, voting, synchronizing a.s.o. are redundated too and permanently on-line-tested.

Considered here as an example, the standard-type industrial automation-system realizing a concept of hardware-implementable fault-tolerance has the following properties in the main:

A) Faults in one of three subsystems are tolerated without any time-consuming error-recovery actions. There is no "shock" to the controlled process at the advent of tolerable faults.
B) Tolerated faults do not (not even "gracefully") degrade the performance of the automation-system.
C) Tolerated subsystem-faults can be repaired without disturbing the systems process-control-functions. Of course, this on-line-repair includes the actualization and synchronization of the repaired subsystem.



UI Inductive bus converter
AI Analog input module
BO Binary output module
2v3 Two-out-of-three
N8 Bus interface module
BI Binary input Module
CPMs Central Processing modules

Fig. 1 : Redundany in the AS 220 EHF automation system.

D) The fault-tolerant configurations utilize multiplicated standard-hardware and -software, almost completely unmodified. Hence they are non-diverse throughout.

E) The central processing units of TELEPERM M beeing spezialized bit-slice-processors is exploited for the hardware-implementation of comparing, synchronizing etc.. The processor has relatively few internal states beeing, above all, easily visible and controllable.

Properties A to C, i.e. shock-free operation at the advent of faults, on-line-rapairability etc., can be considered as indispensable for an universally applicable, industrial H/HF-automation-system. No application justifying the expense of an HF-system can be thought of, that does not require at least one of the said properties. Suitable hardware-implementation allowed the realization of all these properties fulfilling very strict real-time requirements. So, this hardware-solution set standards for future systems and especially for software-oriented solutions.

Non-diversity (property D) is the necessary consequence of the fault-tolerance implementation by hardware-means, i.e. CPU-synchronism a.s.o.. The following will demonstrate the possibility to achieve the potential or claimed advantages of diversity by other means and system-properties, while, of course, avoiding all disadvatages and dangers due to design-diversity [5].

The utilizing of un-common bit-slice-processors in the TELEPERM M-CPU did effectively back-up the hadware-implementation of fault-tolerance choosen. Consequences for similar solutions utilizing VLSI-CPUs are discussed in a later chapter.

4. The Non-Diversity of serially produced Automation-Systems

It's not so long ago that design-diversity was commonly considered the one and only instrument against software-faults supposed to be unavoidable. Consequently, the use of diversity was postulated very globally and uncritically. From this point of view the incompatibility of diversity and the hardware-implementability of fault-tolerance seems to be principal a disadvatage of the latter. However, closer consideration shows, that for multiply produced industrial systems the claimed advantages of design-diversity will be almost in-effective, but its drawbacks all the more. Hence, serialy produced systems will be non-diverse in any case and the said incompatibility to diversity is no handicap for hardware-implementable fault-tolerance.

Remark: Software-diversity using different versions of programmes one after another — i.e. so-called time-diversity -- is compatible with hardware-implementable fault-tolerance, as this could be done in synchronism on multiplicated, redundant subsystems too. This may be suitable for special applications but not commonly for systems with strict real-time-requirements.

While running, the occurance of physical faults in a system is not avoidable principally. Therefore, these faults have to be tolerated by fault-tolerance-provisions or mastered by a fail-safe-behaviour of the concerning system. At least for H-systems with strict real-time-reqirements, above demands A to D (undisturbed, "shock-free" service, etc.) lead to the necessity of redundantly and simultaneously processing the same task on multiplicated hardware. On the other hand, systematic faults (and all software-faults are systematic ones) of the design or realization of a system are unchangeable from the beginning and while running. Steps toward their avoidance or their detecting before the commencement of operation are -- in any case -- more meaningful than are expenses for their tolerating while running, i.e. more meaningful than design-diversity. This statement holds in a very general scope. On top of that, for the automation-systems produced in large quantities, one has to pay special attention to the on-servivce expenses due to diversity, which would have to be multiplicated by the large number of sold systems and by the average of their life-times (10 to 40 years). The multiplicated expediture for documentation, handling, maintenance, administration of versions etc. is unbearable for the manufacturer and

above all for the user. Contrary to design-diversity every effort to avoid systematic fault is non-recurring (and hence bearable) development-expense.

Besides their bad compatibility to large-quantity-production, diverse solutions are afflicted with inherent disadvantages, some beeing critical to safety:

- o Even different solutions have to start from a certain level of common task specification. In the field of process-automation these common specifications will have to be very detailed, already. Faults at this level will not be fought against by design-diversity.
- o The tolerability of random physical faults by redundancy and the simultaneous tolerating of systematic faults by design diversity impair each other, at least.
- o By itself, organizing diverse solutions brings restrictions and new sources of error.
- o Proceeding diversely does not at all exclude correlated failures in the different versions ([5]).

On the manufacturers side, design-faults in hardware and software and realization-faults are detected by measures and factors equivalent to design-diversity. Among them is the range of high-grade qualtity-assesement feasible with a great manufacturer and large-quantity-production as well as the many different -- "diverse" -- service-experiences of standard-type systems.

For programming a single project or task by application-engineers or customers the advantages of large quantities are not so effective as they are for the system-design and -programming by the manufacturer. So, in the field of the unique application or project, diversity could still be promising. However, in the exemplary TELEPERM M-automation-system there is no programming (in a traditional sense) of single applications. Solely software-modules supplied with the system are linked and parametrized in a project-language. Future automation-systems will have a software-architecture like this, too. Understood directly by the system, the project-language is at that level of detailed description, that would have the common specification for diverse design. Hence, with automation-systems having that kind of software-architecture the diverse designing of the respective application is senseless and unnecessary.

With the automation-system presented here briefly the non-diversity and the special kind of hardware-implementable fault-tolerance require and support each other, luckily:

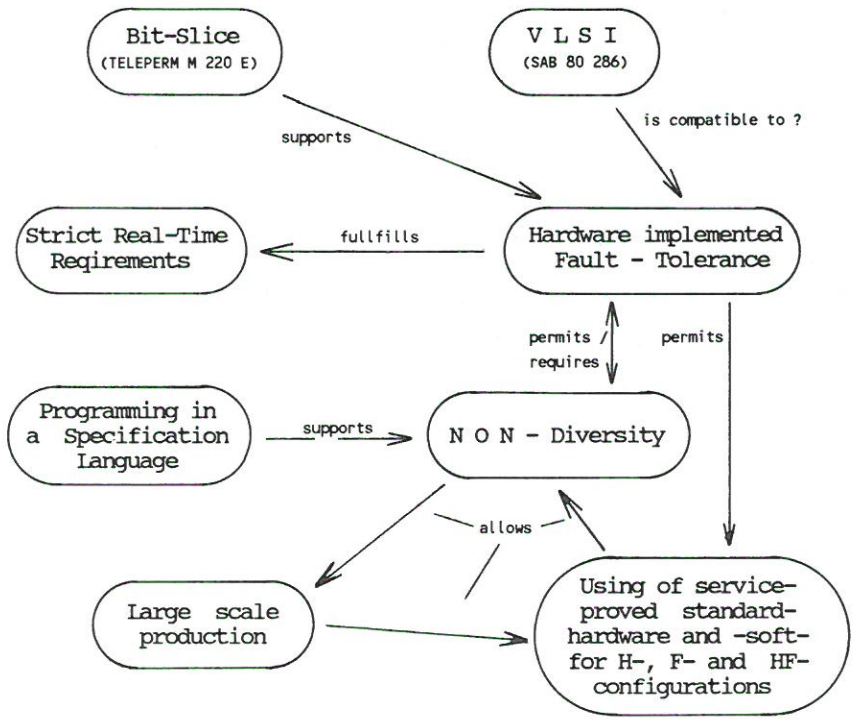


Fig. 2 : The interdependence of hardware implemented fault - tolerance and non - diversitiy.

Non-diversity is, on the one hand, a basic condition for the hardware-implementable fault-tolerance; on the other hand, it is backed not only by the many disadvantages of diverse solutions, but also by the consequent compatibility of the hardware-solution, described above. The H-, HF-, and F-configuration fulfilling high requirements concerning safety and availability essentially equal the wide-spread, non-redundant standard-systems, and, hence, inherit the service-proof at both hardware and software from the latter. Beyond non-diversity, this consequent compatibility between standard-, H-, HF- and F-systems has advantages by itself: All automation-systems in a plant, safety-related as well as "normal" ones don't look alike by artificial user-sufaces, but are intrinsicly equal. This can enhance the safety, especially in exceptional situations:

- o The staff will not be bewildered by differences of their automation-systems.
- o Their experience gained with their (many) standard-systems will be for the benefit of the safety-critical applications, too.

Figure 2 resumes the described relations and conditions between requirements, software-architecture, large quantities, non-diversity and hardware-implementable fault-tolerance. It seems advantageous and desirable to go this way -- service-proved with TELEPERM M -- also with systems utilizing VLSI-processors, e.g. the SAB 80 286-microprocessor and its numerical co-processor SAB 80 287.

5. The Utilizing of VLSI - Processors

The hardware-implementation of fault-tolerance sketched above necessitates :

- o The bit- or clock-synchronism of several processors has to be realized by links free of retroaction, lest defects in one subsystem disturb another subsystem.
- o The synchronization of reset processors to undisturbedly running ones has to be possible. This feature is needed for the completation of on-line-repairs.

With the exemplary TELEPERM M-system, the solution of these and other problems was facilitated by properties of the utilized bit-slice-processors :

- o Their internal states and operations are totally known and clearly arranged.
- o Their internal states are easily visible and controllable.

Contrarily, most VLSI-processors tend to have the following peculiarities:

- o Many loosely coupled queues, bus-, storage-, adress-, arithmetic and logic units, working partly in parallel, are not or not clearly synchronized to each other.
- o There is a large number of states and state-transitions, partly in-visible from outside and partly unknown.
- o Some state-transitions are irreversible, except by hardware-reset, e.g. the adressing-modes of the SAB 80 286 ([7]).

Some obscurities in the first two points may be due to bad or (intentionally) incomplete processor-documentation.

An SAB80286/80287-twin-system utilizing hardware-implemented synchronism, comparing etc. was developped as laboratory-prototype and investigated thoroughly. All conceived difficulties could be overcome. Hence, the realization of hardware-implementable fault-tolerance for systems utilizing that kind of VLSI-processors turned out to be feasible. Of course, this implies certain but not principal restrictions at technical details of the system concerning the backplane-bus, the allotment of functional modules to boards and the like. For example, due to the need for processor-related synchronization and comparing this allotment has to be different from usual single-board-computer solutions.

6. Conclusions

Having historical causes in the conventional automation-technology too, til now automation-systems with fail-safe-behaviour (abbreviated "F") used in safety-critical applications tend to be a separate field having only few connections to the other industrial automation-technology, including the fault-tolerant. The growing complexity of the controlled processes calls for processor-based automation systems also for safety-critical processes. Hence this separation is less and less justifiable. Reasons against this split-up are:

- o The technical or physical preference for pure F-behavior by the conventional technology is completely missing with the electronic and especially processor-based automation-technology.
- o Dedicated F-systems, developped specially and produced only in small quantities, are not likely to gain the the ample quality-assesment and service-proof of serially produced systems, in spite of the immense expenses with regard to the quantity.
- o By special F-systems the operating-personell will be confronted with operating, handling and maintenance procedures different from that of the many non-F-automation systems.
- o Pure F-systems gain safety on cost of availability.
- o The basic condition for the use of pure F-systems is the existence of a safe process-state, that is always easily reachable. Such a state may not exist. And even if it does, it may have secondary effects detrimental to safety.

A multiply produced industrial serial automation-system, which provides a base for standard-, F-, HF- and H- configurations fully compatible to each other, overcomes all these disadvantages. The non-diversity of the safety-directed configurations is both necessitated and justified by this consequent compatibility, as the "diverse" service-proof of the many standard-systems is to the benefit of the fail-safe and fault-tolerant configurations. Regarding the many drawbacks of diverse solutions, this backed-up dispense with diversity is a great advantage.

The suitable implementaribility of fault-tolerance by hardware-means is one way to this consequent compatibility. TELEPERM M 220 E is a servive-proved example for going this way. Some specialities in the architecture of this exemplary system backed-up very effectively hardware-oriented solutions. This can't be expected with VLSI-processor-based systems. Nevertheless, as experimental results show, hardware-orientated solutions are feasible for them, too.

7. References

[1] Algirdas Avižienis and J. Kelly , Fault-Tolerance by Design Diversity : Concepts and Experiments , Computer 17 (1984) , Nr. 8 (August) , pp. 67-80.

[2] M. Euringer and W. Reichert , Hochverfügbares und fehlersicheres Automatisierungssystem AS 220 EHF in 2-von-3-Technik , Siemens Energietechnik 6 (1984) H.5 , pp. 245-249.

[3] - , The AS 220 EHF Fault-Tolerant and Fail-Safe Automation Subsystem with Two-Out-Of-Three Redundancy , Siemens Power Engineering 6 (1984) Vol.6 , pp. 323-327 , (english version of [2]).

[4] U. Kling and E. Schrodi , Redundantes hochverfügbares Automatisierungssystem AS 220 H im dezentralen Prozeßleitsystem Teleperm M , Siemens Energietechnik 5 (1983) , Vol.2.

[5] N.G. Leveson , Correlated failures in Multi-Version Software , Pp. 159-166 in Safety of Computer Control Systems 1985 "SafeComp '85" , Proceedings of the Fourth IFAC Workshop , Como, Italy, 1-3 October 1985.

[6] H.G. Nix , Sichere Mikroprozessorsysteme für Schutzaufgaben , atp 28(1986) , Vol. 3 , Pp. 130-135

[7] Siemens AG , SAB 80 286 , Programmer's Reference Manual including the SAB 80 286 Numeric Supplement , User's Manual 1984.

[8] A. Weinert , Über hardwareimplementierbare Fehlertoleranz bei industriellen Automatisierungssystemen mit sehr hoch integrierten Prozessoren , NTG - Fachberichte Nr. 92, pp. 268-278 , Berlin 1986.