Diss. ETH No. 8152

A METHOD FOR THE DESIGN OF EMBEDDED FAULT-TOLERANT COMPUTERS FOR PROCESS CONTROL AND A DESIGN EXAMPLE

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH
for the degree of
Doctor of the Technical Sciences

presented by
HUBERT DANIEL KIRRMANN
Dipl. El. Ing. ETH
born 3rd January 1948
citizen of France

accepted on the recommendation of
Prof. Dr. A. Kündig, ETH-Z, referee
Prof. Dr. J.D. Nicoud, EPF-L, co-referee

1986

A method for the design of embedded, fault-tolerant computers for process control and a design example.

Hubert Kirrmann, Dipl. El. Ing ETH-Z

Industrial plants are controlled by a distributed computing system, which consists of a number of computing nodes, long-distance, short distance and backplane buses, input and output devices, etc. These components are configured differently for each particular plant. While configuration tools exist to achieve a certain functionality, redundancy is still taken care of by ad-hoc solutions.

This thesis considers the systematic introduction of redundancy in a process control system, while maintaining its overall structure. A model is developed based on a data-flow analysis of the control system and on the notion of "dependable blocks". A dependable block generates an output vector based on its input vector and internal state. It is a self-checking unit (SCU) capable of detecting a large number of its internal faults and signal them.

The blocks can be used alone as fail-stop units or connected in parallel to achieve redundancy. Standard components can be used in most cases, provided their self-checking is good enough. Only three special components are introduced for the sake of redundancy: the voter, the spreader and the link used for maintaining redundancy with another block. This concept should therefore pave the way for a cost-effective configuration of dependability.

The operating mode of redundancy, retry, standby or workby, is investigated. It is concluded that the workby approach (maintaining of redundancy through synchronous operation) is the prefered approach. Standby should be used where independence of failure modes is primordial.

The architecture of a redundant process control system is then described. It is shown that a total meshing is not desirable, and that it is sufficient to make the existing data paths redundant. A concept for the I/O devices is developped which makes use of the data-driven approach.

The initial assumption, that self-checking can be implemented for about half the complexity of the functional logic, is investigated at the example of a processor board. Guidelines are developped for the design of self-checking units. The

conclusion is that the initial assumption is correct, but that the figure will be improved when circuits are introduced which avoid duplication and comparison and use coding instead.

The design of a duplicated multiprocessor is then described as an implementation for the workby node. The redundancy is maintained by a special unit called the Update and Synchronization Unit, which is responsible for all redundancy functions. The function of this unit is described in much detail. A result of this design is that it is difficult to mask redundancy from the application, since the handling of redundant data requires a knowledge of their meaning.

Eine Methode zum Entwurf fehlertoleranter Rechnersysteme für die Proßessleittechnik, mit Anwendungsbeispiel

Hubert Kirrmann, Dipl. El. Ing ETH-Z

Industrielle Anlagen werden heute von einer rechnergesteuerten, verteilten Leitanlage geleitet. Diese besteht aus einer Vielzahl von Standardbausteinen (Rechnerknoten, Fern- Nah- und Rückwandbussen, Ein- und Ausgabegeräten, etc.) die für eine bestimmte Anwendung konfiguriert werden. Der Aspekt der Verläßlichkeit wird durch ad-hoc Lösungen berücksichtigt wenn die Zuverläßigkeit der bestehenden Anlageteile nicht ausreicht.

Diese Dissertation befaßt sich mit dem gezielten Einfügen von Redundanz in eine Leitanlage, ohne deren Struktur zu verändern. Es wird ein Modell entwickelt, auf dessen Grund einkanalige, zweikanalige oder mehrkanalige Strecken zusammengeschaltet werden. Als Grundlage dient eine Datenflussanalyse der Anlage. Die einzelnen Komponenten werden als "verläßliche Blöcke" dargestellt, die einen Ausgangsvektor auf Grund eines Eingangsvektors und deren internen Zustand generieren und im Falle eines Ausfalles diesen mit einer grossen Wahrscheinlichkeit entdecken und signalisieren. Diese Blöcke sollen als selbstprüfende Einheiten (SCU) ausgelegt sein.

Die Blöcke können einzeln als "fail-stop" Einheiten verwendet oder können zum Erreichen einer bestimmten Redundanz parallelgeschaltet werden. Dabei können herkömmliche, nichtredundante Blöcke verwendet werden, und es sollen zum Zweck der Parallelschaltung nur drei neue Komponenten eingeführt werden: der Wähler, der Verteiler und eine Verbindung zum Aufrechterhaltung der Redundanz zwichen Prozessoren. Damit ist der Weg offen für eine kostengünstige Konfiguration von Leitanlagen.

Die Arbeitsweise der Redundanz, ob Wiederanlauf, Standby oder Nebenlauf wird untersucht. Es wird empfohlen, soweit wie möglich die Redundanz durch Synchronlauf zu unterhalten und Standby dort einzusetzen, wo eine möglichst grosse Fehlerunabhängigkeit erzielt werden soll.

Die Architektur einer redundanten, verteilten Leitanlage nach diesem Modell wird aufgestellt. Es wird gezeigt, dass eine totale Vermaschung keine nennenswerte Verbesserung der Zuverläßigkeit mit sich bringt und dass es

genügt, die Datenpfade einer nicht-redundanten Anlage nach Bedarf zu verdoppeln. Es wird ein Konzept für die Ansteuerung der Ein- und Ausgabebausteine entwickelt, der datenflussorientiert ist.

Die Ausführbarkeit der selbstprüfenden Blöcke wird am Beispiel einer Prozessorkarte untersucht. Es werden dabei Richtlinien für den Entwurf einer selbstprüfenden Leiterplatte aufgestellt. Es zeigt sich, daß die selbstprüfende Logik etwa die Hälfte der Komplexität der funktionalen Logik aufweist. "Verdoppelung und Vergleich" ist nur bedingt brauchbar, es sollen vielmehr auf Bausteine gewartet werden, die Selbstprüfung durch Kodierung erreichen.

Der Aufbau eines redundanten Multiprozessors (C–Knoten) wird untersucht. Die Redundanz wird unterhalten durch eine spezielle Kommunikationseinheit zwischen den Parallelbussen dieser Knoten, USU genannt (Zustandsübergabe- und Synchronisierungseinheit). Es zeigt sich, dass es schwierig ist, die Redundanz vor der Anwendersoftware zu verstecken, da redundante Daten nur dann sinnvoll gehandelt werden können, wenn ihre Bedeutung bekannt ist.