Preface

This thesis is a summary of my work as a research assistant in the Computer Engineering Research Group of the Computer Science Department at Brandenburg Technical University, Cottbus, Germany. It embraces the concepts, approaches, implementations, and experiments of my work on dependable processor-based embedded systems.

I would like to thank all those, who were directly or indirectly involved in the completion of this thesis, for their support.

Especially, I am grateful to my supervisor and mentor Prof. Dr. Heinrich-Theodor Vierhaus for the motivation regarding this interesting and versatile topic, for the excellent support of my work, for the permanent technical discussion, and for multitudinous ideas and suggestions.

I would like to thank my consultants Prof. Dr.-habil. M. Gössel from the University of Potsdam, Germany and Prof. Dr. Matteo Sonza Reorda from the Politecnico di Torino, Italy for their helpful remarks and hints.

My colleagues T. Mohaupt, O. Kluge, and U. Berger, I would like to thank for giving me the benefit of their experiences and critiques. I would like to acknowledge the support of our secretary, Kathleen Lück, with organizational and administrative tasks. Due to her assistence, many bureaucratic hurdles were removed.

A special thanks goes to the student members of our research group, Andreas Behling, Christian Galke, Falk Pompsch, Christian Rousselle, Thomas Schwanzara-Bennoit, and Karsten Walther, for their tireless co-operation during various investigations.

Finally, I would like to thank my wife Kathrin Pflanz for her infinite patience and support during the preparation of this thesis.

September 2001 Matthias Pflanz

Abstract

This thesis summarizes investigations and experiments on on-line observation and concurrent checking of processors. The objective was to detect single and/or multiple errors within one clock cycle.

First, refined techniques for data-path observation were investigated. Based on an approach for an observation of an ALU by Berger code prediction (BCP), the principle was extended to observe complete data-path structures to detect unidirectional errors.

The applicability of BCP to more complex data-paths with floating-point units was shown with the help of single and double-precision addition/subtraction floating-point-units. Therefore, prediction formulas were developed, which consider the operation in multi-stage units.

The cross-parity observation technique was developed especially for the on-line observation of large register-files or control-registers. By checking row, column, and diagonal-parities, single and multiple register errors can be detected. Cross-parity vectors have a potential diagnosis capability.

Due to the critical character of the processor control-logic, different techniques were developed and investigated to detect single or multiple control-signal errors within the clock-cycle of occurrence. As a simple alternative for a fault-secure controller, a duplicated control-logic was implemented. The identification of control-word differences can be used for error-weighting a subsequent control and, finally, for further recovery strategies. As a practical solution for small processors, a triplicated structure was investigated. With it, a fault-tolerant generation of control-signals to compensate transient errors until the first permanent error was possible. An application-driven reduction (ADR) of control-logic was proposed to decrease the overhead, especially for embedded systems with standard CISCs and a limited number of applications.

To detect control-signal errors, a new approach was taken by the processor state machine. To solve the problem with the complexity of state-spaces of common microprocessors, active control-signals were considered as a definitive representation of a current processor activity – the processor state. Access to all control-signals being assured and transitions being neglected, a combinatorial observation could be realized. Control-signals were encoded to a state-code, which represents the current (legal) state of the processor. With an access to control-signal conditions (instruction, time, flag-variables), a controller-independent generation (prediction) of the same code was realized. A comparison of both identifies an illegal state-code. To manage more complex state machines, an application-driven reduced state-encoder or a state-space partitioning was proposed. For pipeline structures, a partitioned observation of states was implemented as an example.

As a consequence of a successful error detection within the same clock cycle, fast recovery techniques of the processor state were investigated. Starting from the positive

oriented assumption that an error has a transient character, a fast repetition (rollback) of erroneous cycle(s) can deliver correct results. Time-intervals of many thousands of cycles in classical roll-back techniques can not satisfy demands for safety-critical applications. Therefore, a shorter time (rollback distance) for recovery was implemented by micro-rollback strategies. Recent approaches to micro-rollback can recover the corresponding structure in case of a transient error. But this technique fails in the case of permanent errors. Therefore, a double-processor architecture was investigated. The master-trailer structure turns out to be a suitable solution for small processors. The trailer is delayed for one cycle. With this plus on-line checked master, a fast repair (2 cycles) of transient errors can be executed by a backup of all master-registers by their counterparts in the trailer. The advantage is the function-takeover (3 cycles) in the case of a permanent-error occurrence.

For pipeline processors, a further-developed rollback technique considers on one hand dynamical execution lengths for different stages, and on the other hand different error weightings. Therefore, a priority control was proposed to manage different rollback-actions (necessary rollback distances) for the recovery of the pipeline. Possible are one-cycle micro-rollback, a pipeline stage-rollback, and a macro-rollback by re-filling the whole pipeline. In the worst case (lost all stored return points), a program re-execution is realized.

Proposed on-line error detection and fast recovery techniques should be a supplement to other methods. In combination with other on-line observation principles, and/or with a combined hardware-software (self-)test, these techniques are used to fulfill a complete self-check scheme for an embedded processor. Strategies for a static or dynamic (micro-) rollback are a useful solution for processor errors due to transient faults of non-recurring characteristics. Then an executed program can be continued as quickly as the implemented structure allows.

The overall approach for efficient on-line checking and fast recovery techniques enhances processor availability and improves the dependability of an embedded system at very reasonable additional costs.

Zusammenfassung

In dieser Arbeit werden Entwürfe und Implementierungen vorgestellt, die eine on-line Fehler-Erkennung und -Behandlung in eingebetteten Prozessoren realisieren. Ziel war die Entwicklung von Techniken zur Detektion im selben Maschinenzyklus.

Zuerst werden verbesserte und weiterentwickelte Techniken zur Überwachung von Prozessor-Datenpfaden vorgestellt. Ausgehend von einem Verfahren zur Überwachung einer arithmetisch-logischen Einheit (ALU) durch Berger-Code Vorhersage (BCP) wurde das Prinzip dahingehend erweitert, dass eine Erkennung von unidirektionalen Fehlern im gesamten Daten-Pfad eines Prozessors möglich ist.

Die Anwendbarkeit der Berger-Code-Vorhersage auf komplexere Daten-Pfade mit Fließkomma-Arithmetik wurde mit Hilfe von Additions/Subtraktions-Einheiten mit einfacher und doppelter Genauigkeit gezeigt. Hierfür wurden spezielle Prediktionsformeln entwickelt, die die Operation dieser mehrstufigen Komponenten berücksichtigen.

Für die parallele Überwachung von größeren Register-Files oder von Kontroll-Registern wurde die Cross-Parity-(Kreuz-Paritäten-)Überwachungstechnik entwickelt. Durch den Check von Zeilen-, Spalten- und Diagonal-Parität können Einzel- und Mehrfachfehler in Registern aufgespürt werden. Die Cross-Parity-Technik besitzt weiterhin das Potential zur Fehlerdiagnose.

Auf Grund des kritischen Charakters einer Prozessor-Kontroll-Logik wurden verschiedene Techniken entwickelt und untersucht, die eine Erkennung von einfachen und mehrfachen Kontroll-Signal-Fehlern im selben Maschinen-Zyklus ermöglichen. Als eine einfache Variante einer fehlersicheren Auslegung der Kontrol-Logik wurde eine Verdopplung vorgenommen. Die Identifizierung von Kontroll-Wort-Differenzen konnte für eine Fehler-Gewichtung bzw. für eine Prioritäten-Steuerung und letztendlich für die Auswahl der Fehlerbehandlungsmaßnahme verwendet werden.
Als eine praktische Lösung für kleinere Prozessoren wurde eine verdreifachte Struktur untersucht. Damit wird die fehler-tolerante Generierung von Kontroll-Signalen bis hin zum ersten permanten Fehler möglich. Insbesondere für eingebettete Systeme mit Standard-CISC-Prozessoren mit einer begrenzten Anzahl von Anwendungen wurde ein Verfahren entwickelt, mit dem Kontroll-Logiken anwendungs-spezifisch reduziert werden könnten. Damit konnte der Overhead verringert werden.

Ein weiteres Verfahren behandelt die on-line Erkennung von Kontroll-Signal-Fehlern durch Überwachung der Prozssor-Zustandsmaschine. Um das Problem der Komplexität von Prozessor-Zustandsmaschinen zu umgehen, wurden aktuelle Kontroll-Signale als definitive Repräsentation der aktuellen Prozessor-Aktivität bzw. des aktuellen Prozessor-Zustandes betrachtet. Unter Voraussetzung der Zugriffsmöglichkeit auf alle Kontroll-Signale und der Vernachlässigung von

Transitionen wurde eine kombinatorische Überwachung realisiert. Kontroll-Signale wurden in ein Zustands-Codewort kodiert, der den momentanen (legalen) Zustand des Prozessors repräsentiert. Mit einem Zugriff auf die entsprechenden Kontroll-Signal-Bedingungen (Befehlscode, Zeit- und Flag-Variablen) wurde eine Kontroller-unabhängige Generierung (Vorhersage) des selben Codes realisiert. Ein Unterschied im Vergleich beider bedeutet die Erkennung eines illegalen Zustandscodes. Um komplexere Zustandsmaschinen beherrschen zu können, werden zum Einen anwendungsspezifisch reduzierte Zustandskodierer und zum Anderen eine Zustandsraum-Aufteilung vorgeschlagen. Exemplarisch zur Überwachung einer Pipeline-Struktur wurde eine aufgeteilte Zustandsraum-Überwachung implementiert.

Als eine Konsequenz der schnellen Fehlererkennung im selben Takt wurden schnelle Wiederherstellungstechniken untersucht. Ausgehend von einer positiv-orientierten Annahme, dass jeder aufgetretene Fehler transienter Natur ist, sollte eine schnelle Wiederholung (Rollback) des fehlerhaften Zyklus korrekte Daten liefern. Zeitintervalle von mehreren tausend Zyklen in klassischen Rollback-Verfahren sind entsprechend der Anforderungen von sicherheits-kritischen Systemen ungeeignet. Deswegen wurde eine kürzere Rollback-Distanz mit Hilfe der Micro-Rollback-Strategie implementiert. Bisherige Rollback-Verfahren sind auf die Behandlung von transienten Fehlern ausgerichtet, versagen jedoch bei permanenten Fehlern. Hierfür wurde eine Doppel-Prozessor-Struktur untersucht. Die Master-Trailer-Architektur stellte sich dabei als exzellente Lösung für kleinere Prozessoren heraus. Ein Trailer ist gegenüber dem Master um einen Takt in der Programm-Abarbeitung verzögert. Damit und mit einer on-line Fehlererkennung kann eine schnelle (2 Zyklen) Reparatur ausgeführt werden, indem alle Master-Register mit den entsprechenden Trailer-Registern überschrieben werden. Der Vorteil besteht aber in der Möglichkeit der vollständigen Funktionsübernahme innerhalb von 3 Zyklen im Falle eines permanenten Fehlers im Master.

Für Pipeline-Prozessoren wurde die Rollback-Technik verfeinert und weiter-entwickelt. Zum Einen mußte eine dynamische Befehlsausführungslänge in unterschiedlichen Pipeline-Stufen und zum Anderen eine differenzierte Fehler-gewichtung berücksichtigt werden. Deswegen wurde eine Prioritäten-Steuerung entwickelt, die verschiedene Rollback-Aktionen mit unterschiedlichen Rollback-Distanzen steuert. Möglich sind Ein-Zyklus-Micro-Rollback, Pipeline-Stufenrollback und ein Macro-Rollback, bei dem die gesamte Pipeline neu gefüllt wird. Für den Fall, daß alle Rückkehrpunkte verloren bzw. fehlerhaft sind, wird eine Programm-Wiederholung ausgeführt.

Die vorgestellten on-line Fehlererkennungs- und -Behandlungs-Techniken sollen eine Ergänzung zu anderen Methoden darstellen. In Kombination mit anderen on-line Observierungstechniken und/oder mit einem kombinierten Hardware/Software-(Selbst-) Test können diese Techniken genutzt werden, um einen eingebetteten Prozessor vollständig selbst-überprüfend auszulegen. Strategien für ein statisches oder dynamisches (Micro-) Rollback sind nützliche Maßnahmen, um durch transiente Effekte verursachte Prozessor-Fehler auf effektive Art und Weise zu beheben. Ein ausgeführtes Programm kann so schnell fortgesetzt werden. Die vorgestellten Methoden verbessern die Verfügbarkeit eines eingebetteten Prozessors und damit auch dessen Zuverlässigkeit bei gleichzeitig moderaten zusätzlichen Kosten.