

Reihe 10

Informatik/
Kommunikation

Nr. 856

Stefan Widmann, M.Sc.,
Freudenberg

Eine Datenspezifikations- architektur



FernUniversität in Hagen
Schriften zur Informations-
und Kommunikationstechnik

Fortschritt-Berichte VDI

Reihe 10

Informatik/
Kommunikation

Stefan Widmann, M.Sc.,
Freudenberg

Nr. 856

Eine Datenspezifikations-
architektur



FernUniversität in Hagen
Schriften zur Informations-
und Kommunikationstechnik

Widmann, Stefan

Eine Datenspezifikationsarchitektur

Fortschr.-Ber. VDI Reihe 10 Nr. 856. Düsseldorf: VDI Verlag 2017.

328 Seiten, 91 Bilder, 42 Tabellen.

ISBN 978-3-18-385610-7, ISSN 0178-9627,

€ 104,00/VDI-Mitgliederpreis € 93,60.

Für die Dokumentation: Echtzeitsysteme – funktionale Sicherheit – sicherheitsgerichtete Echtzeitsysteme – IEC 61508 – Mikroprozessorarchitekturen – Prozessorarchitekturen – Datentyparchitekturen – Befähigungsarchitekturen – Datenfluss – Datenflussüberwachung

Die vorliegende Arbeit richtet sich an Ingenieure und Wissenschaftler in den Bereichen Mikroprozessorarchitektur und sicherheitsgerichtete Echtzeitsysteme. Sie beginnt mit der Identifikation von 20 datenflussbezogenen Fehler- und Angriffsarten und evaluiert anhand dieser den Stand von Wissenschaft und Technik. Anschließend wird eine neue Prozessorarchitektur, die Datenspezifikationsarchitektur, vorgestellt, welche die in Vergessenheit geratenen Merkmale von Datentyparchitekturen stark erweitert und alle Dateneigenschaften in Form zusätzlicher Kennungen untrennbar mit dem Datenwert verknüpft, überträgt, speichert und verarbeitet. Dies ermöglicht es der neuen Architektur, alle 20 Fehler- und Angriffsarten zu erkennen. Die schlussendliche Gegenüberstellung des Stands von Wissenschaft und Technik und der Datenspezifikationsarchitektur zeigt die Überlegenheit der neuen Architektur und deren hervorragende Eignung für die Realisierung sicherheitsgerichteter Anwendungen.

Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter <http://dnb.ddb.de> abrufbar.

Bibliographic information published by the Deutsche Bibliothek

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at <http://dnb.ddb.de>.

Schriften zur Informations- und Kommunikationstechnik

Herausgeber:

Wolfgang A. Halang, Lehrstuhl für Informationstechnik

Herwig Unger, Lehrstuhl für Kommunikationstechnik

FernUniversität in Hagen

© VDI Verlag GmbH · Düsseldorf 2017

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9627

ISBN 978-3-18-385610-7

Vorwort

Der Bedarf an sicherheitsgerichteten, programmgesteuerten (eingebetteten) Systemen aller Art ist hoch und steigt durch die zunehmende Automatisierung von Prozessen kontinuierlich weiter an. Im Einklang damit wächst auch das gesellschaftliche Sicherheitsbewusstsein. Weil die auf dem Markt vorherrschenden Prozessorarchitekturen kaum Schutz gegen typische Programmierfehler und Malware-Attacken bieten, hatte der Autor des vorliegenden Buches sich zur Aufgabe gemacht, aufbauend auf den allgemeinen Sicherheitsanforderungen gemäß der Norm IEC 61508 auf Maschinenebene eine völlig neue, von Grund auf auf Sicherheit hin ausgelegte Rechnerarchitektur zu entwerfen, die eine Fülle von Programmierfehlern ohne Software-Hilfe erkennen kann und daraufhin die Programmausführung abbricht.

Der Autor des 2014 unter dem Titel „Verfahren zur Kontrollflussüberwachung in sicherheitsgerichteten Rechensystemen“ in dieser Buchreihe mit der Nummer 832 erschienenen Bandes hatte sich bereits sehr eingehend mit der Sicherung des Kontrollflusses in sicherheitsgerichteten Echtzeitsystemen beschäftigt und beeindruckende Ergebnisse vorgelegt. Allerdings stellen Kontrollflussfehler nur einen geringen Anteil aller Programmfehler dar. Der weitaus größere Teil aller Fehler, die in programmgesteuerten Digitalrechnern auftreten, sind Datenflussfehler, deren Überwachung und Verhinderung sich deshalb Herr Widmann an dieser Stelle annimmt.

Datenflussfehler sind in höchstem Maße gefährlich, da sie insbesondere in der von Neumann-Architektur, die die völlig beliebige Interpretation jedes Bitmusters erlaubt, eine Vielzahl der technisch möglichen Reaktionen solcher Rechner auszulösen vermögen. Durch während des Entwurfs eines Programms gemachte oder zur Laufzeit auftretende Datenflussfehler kann sich auch der Kontrollfluss in unerwarteter Weise verändern, so dass Befehle in unvorhergesehener und falscher Reihenfolge, aber auch Daten, in denen keine Befehle codiert sind, als Befehle interpretiert und ausgeführt werden. Durch Fehler in der Gerätetechnik, transiente Störungen, intermittierende Fehler oder permanente Ausfälle kann es dazu kommen, dass die Bitmuster von Daten verändert werden. Sehr oft reicht eine einzige fehlerhafte Bitposition aus, um eine völlig verschiedene Aktion auszuführen.

Die von Herrn Widmann verfolgte Zielstellung ergibt sich unmittelbar aus dem unzureichenden Stand der Technik, und zwar gerätetechnische Fehlervermeidungs- und -erkennungsmöglichkeiten zu schaffen, um damit datenflussbezogene Fehler und Angriffe erkennen und die Einhaltung von Echtzeitbedingungen überwachen zu können. Die vorgestellten Ergebnisse sind in allen Bereichen der elektronischen Datenverarbeitung anwendbar und dort wegen deren geringer Zuverlässigkeit auch dringend erforderlich. Trotzdem ist das Werk aus Sicht der Automatisierungstechnik geschrieben, weil Digitalrechner trotz der Unmöglichkeit, wirklich vertrauenswürdige Sicherheitsnachweise für programmgesteuerte Systeme zu führen, mehr und mehr auch für sicherheitsgerichtete Anwendungen eingesetzt werden und dabei bewährte, oft inhärent sichere gerätetechnische Lösungen ersetzen. Weiterhin sind im Anwendungsgebiet sicherheitskritischer Echtzeitsysteme weder nicht zeitdeterministisch arbeitende Verfahren hinnehmbar, noch dürfen Fehler erst nachträglich korrigiert werden.

Bemerkenswert an Herrn Widmanns wissenschaftlich-technischen Beiträgen ist eine Reihe von Aspekten. In ganzheitlicher Betrachtung von Hardware und Software verfolgt er konzeptionell ein neuartiges Entwurfsparadigma, dass nämlich alle Deskriptoren eines Datenspeicherelementes in untrennbarer Verknüpfung mit diesem gespeichert, verarbeitet und übertragen sowie gerätetechnisch überprüfbar dargestellt werden sollen. Mit Hilfe solcher selbstbeschreibenden Daten können dann in Hardware implementierte Überprüfungen die meisten datenflussbezogenen Fehler in der Datenverarbeitung auch über Grenzen zwischen Systemkomponenten hinweg aufdecken. Er liefert theoretische Beiträge, indem er die datenflussbezogenen Fehler- und Angriffsarten analysiert und identifiziert, die Eigenschaften in sicherheitsgerichteten Echtzeitsystemen gehaltener Daten zusammenstellt und darauf aufbauend seine Datenspezifikationsarchitektur entwirft, die die Dateneigenschaften mit bisher unerreichter Aussagekraft abbildet. Und schließlich arbeitet er konstruktiv-ingenieurmäßig, indem er für jedes angegebene Verfahren geeignete Implementierungsmöglichkeiten vorschlägt und ihre jeweiligen Vor- und Nachteile diskutiert.

Der Preis, den Herr Widmann für die Sicherung des Datenflusses in Digitalrechnern bezahlt, ist erheblich erhöhter Speicherbedarf und dementsprechend größerer Übertragungsaufwand, wohingegen der Umfang zusätzlicher Hardware für die Verarbeitung der Datenkennungen gering ist. In Zeiten enormer Speicherkapazitäten ist deutlich erhöhte Sicherheit diesen Preis jedoch unbestreitbar wert.

Hagen, im August 2017

Wolfgang A. Halang

Inhaltsverzeichnis

1	Einleitung	1
1.1	Beispiele für Auswirkungen von Fehlern	3
1.1.1	Selbstzerstörung der Ariane 5	3
1.1.2	Verlust der NASA-Sonde Mars Climate Orbiter	4
1.1.3	Bestrahlungsgerät Therac-25	4
1.1.4	Sicherheitslücke Heartbleed	5
1.2	Der Stand von Wissenschaft und Technik und dessen Nachteile	6
1.2.1	Stand von Wissenschaft und Technik	6
1.2.2	Nachteile des Stands von Wissenschaft und Technik	7
1.3	Ziel der Arbeit	8
1.4	Ergebnisse der Arbeit	9
1.5	Aufbau der Arbeit	11
1.6	Darstellung von Zahlen und Speichergrößen in der Arbeit	13
2	Fehlerarten, -ursachen, -auswirkungen und -behandlung	14
2.1	Fehlerkategorien	14
2.2	Fehlerquellen in Soft- und Hardware	15
2.3	Fehlerdichte in Software	19
2.4	Datenflussbezogene Fehler- und Angriffsarten	20
2.4.1	Inkompatibilität von Operanden	21
2.4.2	Wertebereichsverletzungen und Genauigkeitsprobleme	21
2.4.3	Fehlerhafte Operationen	22
2.4.4	Verletzung von Echtzeitbedingungen	23
2.4.5	Allgemeine Datenflussfehler	23
2.4.6	Datenverfälschung durch Fehler oder Störungen	25
2.4.7	Fehlerhafter Zugriff auf Daten	25
2.4.8	Hackerangriffe	26
2.4.9	Zusammenfassung der identifizierten datenflussbezogenen Fehler- und Angriffsarten	27
2.5	Auswirkungen von Fehlern	27
2.6	Fehlererkennung und -behandlung	30

2.6.1	Einnehmen und Halten eines sicheren Zustands	31
2.6.2	Anwendung von Redundanzmaßnahmen	31
2.6.3	Allmähliche Leistungsabsenkung	31
3	Stand von Wissenschaft und Technik	33
3.1	Konventionelle Architekturen	34
3.1.1	Die x86-Architektur	34
3.1.2	Die ARM-Architektur	40
3.1.3	Integritätsprüfung durch ECC	41
3.1.4	Evaluation konventioneller Architekturen	42
3.2	Prozessoren für sicherheitsgerichtete Anwendungen	45
3.2.1	Aufbau der Prozessoren für sicherheitsgerichtete Anwendungen	45
3.2.2	Evaluation der Prozessoren für sicherheitsgerichtete Anwen- dungen	46
3.3	Datentyparchitekturen	49
3.3.1	Beispiele von Datentyparchitekturen	50
3.3.2	Evaluation der Datentyparchitekturen	52
3.4	Datenstruktur- bzw. Deskriptorarchitekturen	54
3.4.1	Beispiele von Datenstruktur- bzw. Deskriptorarchitekturen .	54
3.4.2	Evaluation der Datenstrukturarchitekturen	55
3.5	Befähigungsarchitekturen	55
3.5.1	Beispiele historischer Befähigungsarchitekturen	58
3.5.2	Beispiele moderner Befähigungsarchitekturen	60
3.5.3	Evaluation der Befähigungsarchitekturen	66
3.6	Datenflussarchitekturen	66
3.6.1	Funktionsweise von Datenflussarchitekturen	68
3.6.2	Evaluation von Datenflussarchitekturen	69
3.7	Die inhärent sichere Mikroprozessorarchitektur ISMA	71
3.7.1	Aufbau der Datenspeicherelemente in ISMA	71
3.7.2	Evaluation von ISMA	75
3.8	Application Data Integrity ADI bzw. Silicon Secured Memory SSM	77
3.8.1	Funktion von ADI bzw. SSM	77
3.8.2	Evaluation von ADI bzw. SSM	77
3.9	Dynamic Dataflow Verification DDFV	79
3.9.1	Funktion der dynamischen Datenflussprüfung	79
3.9.2	Evaluation der dynamischen Datenflussprüfung	80
3.10	Fehlererkennung durch AN(BD)-Kodierung	80
3.10.1	AN-Kodierung zur Integritätsprüfung von Datenspeicherele- menten und arithmetischen Operationen	82

3.10.2	ANB-Kodierung: Hinzufügen der Adressprüfung B	84
3.10.3	ANBD-Kodierung: Hinzufügen der Aktualitätsprüfung D	86
3.10.4	Realisierung der AN(BD)-Kodierung	87
3.10.5	Evaluation der AN(BD)-Kodierung	87
3.11	Datenflussüberwachung in Netzwerken und sicherheitsgerichteten Feldbussen	91
3.11.1	Netzwerkprotokolle TCP/IP	91
3.11.2	Sicherheitsgerichtete Feldbusprotokolle	95
3.11.3	Evaluation der Datenflussüberwachung in Netzwerken und sicherheitsgerichteten Feldbussen	101
3.12	Zusammenfassung des Stands von Wissenschaft und Technik	104
3.12.1	Zusammenfassung der Fehlererkennungsmöglichkeiten	104
3.12.2	Zusammenfassende Kritik am Stand von Wissenschaft und Technik	108
4	Eine Datenspezifikationsarchitektur	111
4.1	Systemaufbau und Fehlerbehandlung	112
4.1.1	Grundlegender Systemaufbau technischer Prozesse	112
4.1.2	Aufbau eines auf einer Datenspezifikationsarchitektur basierenden Systems	113
4.1.3	Fehlerbehandlung in einer Datenspezifikationsarchitektur	116
4.2	Sammlung relevanter Dateneigenschaften	119
4.3	Realisierung der Datenflussüberwachung	122
4.3.1	Einleitende Erläuterungen	122
4.3.2	Datenwert und dessen Genauigkeit	128
4.3.3	Wertebereich	140
4.3.4	Datentyp	148
4.3.5	Einheit	161
4.3.6	Zugriffsrechte und Initialisierungsstatus	175
4.3.7	Quelle, Verarbeitungsweg und Ziel	184
4.3.8	Zeitschritt	203
4.3.9	Frist	220
4.3.10	Zykluszeit	226
4.3.11	Integritätsprüfung und Adresse	239
4.3.12	Signatur und Adresse	244
4.3.13	Redundante diversitäre arithmetisch-logische Einheit	253
4.4	Übersicht der Kennungen in Daten- und Befehlsspeicherelementen	257
4.5	Übersicht der speziellen Register	261
4.6	Pseudocode einer Instruktion	261

4.7	Anforderungen an die Systemkomponenten	272
4.7.1	Schnittstellen zu konventionellen Systemkomponenten	272
4.7.2	Hochpräzise synchronisierte Uhren	273
4.8	Konfiguration der Systemkomponenten	273
4.8.1	Konfiguration der Datenquellen	274
4.8.2	Konfiguration der Datenverarbeitungseinheiten	274
4.8.3	Konfiguration der Datensenzen	276
4.8.4	Konfiguration der Systemüberwachungseinheit	277
4.8.5	Erkennung konfigurationsbezogener Inkonsistenzen	277
4.9	Anforderungen an Begutachtungen und Audits	278
4.10	Realisierung der Datenspezifikationsarchitektur als Datenflussarchitektur	279
4.10.1	Erweiterung der Funktionsblöcke um Lebenszeichen und Diagnose	280
4.10.2	Verbesserung der Fehlererkennung durch zusätzliche Erweiterungen	283
4.10.3	Weiterhin bestehende Einschränkungen	286
5	Evaluation der Datenspezifikationsarchitektur	287
5.1	Evaluation der Datenabbildung der DSA	287
5.2	Einordnung der entstandenen Architektur	290
5.3	Evaluation anhand der Fehlererkennungsmöglichkeiten	291
5.4	Evaluation anhand der Fehlerbeispiele	295
5.4.1	Selbstzerstörung der Ariane 5	295
5.4.2	Verlust der NASA-Sonde Mars Climate Orbiter	296
5.4.3	Bestrahlungsgerät Therac 25	296
5.4.4	Sicherheitslücke Heartbleed	296
5.5	Evaluation der Speicherausnutzung	298
5.5.1	Speicherausnutzung der Datenspeicherelemente	298
5.5.2	Speicherausnutzung der Befehlsspeicherelemente	301
5.5.3	Evaluation der Speicherausnutzung	304
6	Zusammenfassung und Weiterführungsmöglichkeiten	306
6.1	Zusammenfassung der Ergebnisse der Arbeit	306
6.2	Weiterführungsmöglichkeiten	308
	Literaturverzeichnis	310