

1 Einleitung

Auf Raumsonden zur Planetenbeobachtung befinden sich üblicherweise Kamerainstrumente. Sie dienen z. B. der Erfassung und Kartographierung von Oberflächen oder der Analyse von Atmosphärenphänomenen. Darüber hinaus ermöglichen sie den Zugang der Allgemeinheit zu den Ergebnissen von Weltraummissionen. Daher gibt es kaum noch Missionen zur Planetenbeobachtung, die nicht mit mindestens einer Kamera ausgestattet sind. Solche Missionen und insbesondere solche mit Landeeinheiten erfordern kompakte, leistungsfähige Kameras, die auf die wissenschaftlichen Erfordernisse der Missionen zugeschnitten sind.

Die Schnittstelle zwischen der Raumsonde und den einzelnen Elektroniken der Instrumentensensoren bilden die Instrumentenrechner (*Data Processing Unit*, DPU). Sie übernehmen sehr spezifische Steuerungs- und Konvertierungsaufgaben sowie zunehmend auch komplexe Aufgaben zur Datenaufbereitung an Bord. Klassische Aufgaben einer DPU sind:

- Akquisition von Sensordaten
- Datenspeicherung
- Ausführung der Kommandos von der Raumsonde
- Senden von Telemetrie-Daten (TM) zur Raumsonde
- Erzeugung von Statusdaten (*Housekeeping*, HK) und Instrumentenüberwachung
- Datenkompression zur Reduzierung der Eingangsdatenmenge
- Datenvorverarbeitung, die aufgrund der Datenmengenreduktion nicht nach der Übertragung ausgeführt werden kann
- Ausführung von Skripten des Anwenders (z. B. Messsequenzen).

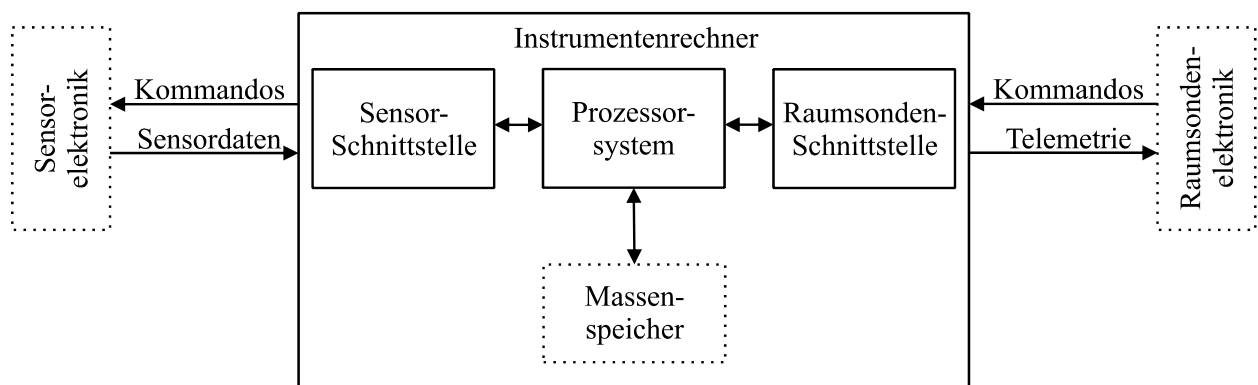


Bild 1.1: Instrumentenrechner

Bild 1.1 zeigt den typischen Aufbau eines Instrumentenrechners, der für unterschiedliche Sensorarten ähnlich ist. Die Instrumente für kompakte Kameras zeichnen sich häufig dadurch aus, dass sie eine hohe Datenmenge produzieren und eine hohe Rechenleistung zur

Bilddatenverarbeitung erfordern. In vielen Fällen gilt dafür das Prinzip, nur die absolut notwendige Bilddatenverarbeitung an Bord der Raumsonde durchzuführen, um Komplexität und Energie zu sparen. Daher reduziert sich die An-Bord-Bilddatenverarbeitung auf die Datenkompression und die dafür notwendigen Vorverarbeitungsschritte. Die benötigte Rechenleistung wird somit i. Allg. durch den Kompressionsalgorithmus bestimmt. Eine Kompression ist wünschenswert, damit die durch Sendeleistung und Entfernung beschränkte Datenrate zur Erde optimal genutzt wird oder um nicht unnötigerweise viel internen Speicher zu beanspruchen. Dafür sind hochwertige und effiziente Kompressionsalgorithmen notwendig. Zur Pufferung der Bilddaten verfügen Instrumentenrechner für Kameras vielfach über eigenen Massenspeicher. Die Bilddatenverarbeitung muss auf diese Weise nicht zwangsweise mit der Erzeugung der Roh-Bilddaten Schritt halten und für die Bedienung der Telemetrestrecke steht mit dem Massenspeicher ein genügend großer Zwischenpuffer zur Verfügung.

Die effiziente Implementierung der Bilddatenverarbeitung in der DPU bekommt damit eine wichtige Rolle bei der Auslegung eines Kamerainstrumentes: durch die Datenkompression wird die wissenschaftliche Ausbeute erhöht, z. B. gemessen an der Zahl der pro Missionszeiteinheit gewonnenen und übertragenen Bilder. Andererseits ist dafür die Implementierung komplexer Algorithmen in der DPU notwendig, für die Designkomplexität und höherer Energiebedarf auf der Aufwandsseite stehen.

Der Kernansatz in dieser Arbeit ist, den Entwurf einer Kamera-DPU unter den Optimierungsgesichtspunkten Ausbeute, Ressourcenbedarf und Fehlertoleranz unter den Randbedingungen einer Weltraummission zu untersuchen.

Anforderungsprofil für eine Kamera-DPU

Am Institut für Datentechnik und Kommunikationsnetze wurden für eine ganze Reihe von Instrumenten-DPUs erstellt. DPUs neueren Typs wurden für die beiden Instrumente Venus Monitoring Camera (VMC) der ESA-Sonde Venus Express und Dawn Framing Camera (DFC) der NASA-Sonde DAWN entwickelt. Typische Kennwerte dieser und auch anderer Kameras zur Planetenbeobachtung sind:

- Bildgröße: $\geq 1.024 \times 1.024$ Pixel
- Radiometrische Auflösung pro Pixel: ≥ 14 Bit
- Grauwertbild
- Gewünschte Akquisitionsrate: alle 1-2 s ein Bild
- Typische Kompressionsfaktoren: 5-8

Die Bearbeitung der Bilddaten erfolgt im Instrument selbst. Die prinzipielle Möglichkeit, die notwendige Rechenleistung durch einen zentralen Rechner auf der Raumsonde zur Verfügung zu stellen, ist aus Gründen der Heterogenität der Instrumente sowie unter dem Aspekt der üblicherweise stark räumlich verteilten Entwicklung der Instrumente nicht praktikabel. Zudem

lassen sich bei dezentraler Verteilung der Rechenleistung die Schnittstellen zur Raumsonde besser einheitlich definieren.

In Abhängigkeit von der Mission werden unterschiedliche Anforderungen an den Instrumentenrechner gestellt. Allgemeine Anforderungen, die für alle Instrumentenrechner gelten, sind:

- Aufgaben angepasste Rechenleistung
- Adäquate Zuverlässigkeit und Verfügbarkeit
- Einhaltung missionsspezifischer Strahlungstoleranzen
- Geringer Energiebedarf
- Platzsparender Aufbau und geringes Gewicht
- Flexibles Design für verkürzte Entwicklungszeiten
- Modulares Design zur Langlebigkeit eines Entwicklungsansatzes

Entwurfs- und Implementierungsaspekte

Beim Entwurf von Elektronik für Raumsonden müssen die im Weltraum auftretenden Umweltbedingungen beachtet werden. Insbesondere hat die dort auftretende Strahlung einen erheblichen Einfluss auf das Verhalten von Elektronik. Sie kann zu temporärem aber auch zu permanentem Fehlverhalten führen. Der traditionelle Entwurf von Elektronik für die Raumfahrt beinhaltet die Verwendung strahlungsfester (*Radiation Hardened*, RH) Bauteile. Diese sind speziell für den Einsatz im Weltraum entwickelt und erhalten ihre Strahlungstoleranz durch entsprechende Chip-Herstellungs-Technologien. Aufgrund des sehr kleinen Marktes ist die Verfügbarkeit von RH Bauteilen sehr stark begrenzt. Auf strahlungsfesten Bauteilen basierende Rechnersysteme sind als Plattform für Steuerrechner von Raumsonden auf dem Markt verfügbar, allerdings sind sie häufig für den sehr speziellen Einsatz in kleinen Instrumentenrechnern wegen ihres Ressourcenbedarfs nahezu ungeeignet. Bei Instrumentenrechnern werden daher oft maßgeschneiderte Designs eingesetzt.

Da wissenschaftliche Instrumente meist nur für kleine Stückzahlen entworfen werden, ist der Entwurf gleichbedeutend mit der Entwicklung eines Prototyps für ein spezielles System. Dieses wird durch die Verwendung programmierbarer Bausteine (*Field Programmable Gate Array*, FPGA) zur freien Logikimplementierung vereinfacht, denen eine wichtige Rolle beim Entwurf eines Instrumentenrechners zukommt. Mit der Verwendung weltraumtauglicher FPGAs kann man teilweise das Problem umgehen, dass die Anzahl an strahlungstoleranten Bausteinen und damit die verfügbare Funktionalität sehr stark begrenzt ist, indem man die benötigte Funktionalität selbst entwirft. Wie Anwendungsspezifische Integrierte Schaltungen (*Application Specific Integrated Circuit*, ASIC) können FPGAs benutzt werden um Masse, Volumen und Gewicht zu sparen. Bei der geringen Stückzahl sind FPGAs wesentlich kostengünstiger als ASICs.

Zudem ist die Erstellung bis hin zur endgültigen Designverifikation in wesentlich kürzerer Zeit möglich und Änderungen in den Spezifikationen können vergleichsweise schnell eingebaut werden. Bei der Entwicklung von Prototypen sind diese Vorteile von hohem Wert. Bei der Benutzung von FPGAs liegt die Verantwortung für die Entwicklung und die Testzyklen üblicherweise innerhalb eines Hauses und eine kosteneffektive Entwicklung und Produktion von kleinen Serien ist möglich.

Die derzeit zur Verfügung stehenden großen weltraumtauglichen FPGAs mit mehreren Millionen Systemgattern können zusätzlich zu den kombinatorischen und sequenziellen Zellen auch fest verdrahtete Module wie internen Speicher oder Multiplikationseinheiten beinhalten. *System-on-Chip*-Entwürfe für die Implementierung von DPU's sind damit möglich. Insbesondere der in den vergangenen Jahren erheblich angestiegene interne Speicher ermöglicht es, zunehmend spezielle Funktionen direkt im FPGA auszuführen und den Arbeitsdatenstrom außerhalb des FPGA stark zu reduzieren.

Die Integration mehrerer Funktionen in einem FPGA hat folgende Vorteile:

- Höhere Systemintegration
- Höhere Performanz und geringerer Energiebedarf durch verringerte Anzahl von *Chip-zu-Chip*-Verbindungen
- Erhöhte Zuverlässigkeit aufgrund reduzierter Bauteilanzahl
- Vereinfachung des Boardlayouts
- Erhöhte Flexibilität durch Mehrfachprogrammierbarkeit (bei SRAM-basierten FPGAs)
- Wiederverwendung von IP Cores und insgesamt kürzere Entwicklungszeiten

Ein Hauptbestandteil der Entwicklung ist der zeitliche Aufwand für den Entwurf der Schaltungsblöcke innerhalb des FPGA. Die Verwendung großer FPGAs in Instrumentenrechnern führt dazu, dass sich ein Großteil der Logik auf wenige Bausteine konzentriert. Eine modulare Designmethode mit eindeutigen Schnittstellendefinitionen vereinfacht die Wiederverwendung solcher Blöcke in nachfolgenden Projekten, erhöht die Langlebigkeit der Entwurfsarbeit und reduziert die Kosten in nachfolgenden Projekten. Erfolgt das Design der Logikimplementierung durch eine Beschreibungssprache wie z. B. VHDL (*Very High Speed Integrated Circuit Hardware Description Language*), kann eine einfache Implementierung auch in zukünftige FPGA-Familien erzielt werden. Bis auf wenige Ausnahmen haben Änderungen innerhalb des FPGA keine Auswirkung auf das Layout der Platine, so dass das Layout schon in einem frühen Stadium festgelegt werden kann. Die ersten Arbeiten an Logikmodulen können zudem häufig bereits auf vergleichbaren Entwurfsplattformen erfolgen. Die Nachteile der Verwendung großer FPGAs sind im Wesentlichen bei der verringerten Zugänglichkeit interner Funktionen und den daraus resultierenden schlechteren Testmöglichkeiten zu sehen. Ein intensiver Einsatz von Simulationssoftware kann aber diesen Nachteil ausgleichen.

Instrumentenrechner werden meistens als nicht missionskritisch eingestuft. Während die Steuerung einer Raumsonde z. B. zum Zeitpunkt eines Landemanövers auch nicht nur für kurze Zeit ausfallen darf, kann beim Bearbeiten von wissenschaftlichen Daten in einer DPU eine niedrigere Verfügbarkeit akzeptiert werden. Beim Aufbau einer DPU mit FPGAs können andere Ansätze als der klassische Weg mit einer strahlungsfesten CPU zur Bearbeitung von Daten gewählt werden. Während sich FPGAs bisher lediglich am Rande eines Designs befanden, so rücken sie mit zunehmender Größe immer mehr ins Zentrum des Designs. Neben dem klassischen Einsatz eines FPGA als Verbindungselement zweier Einheiten (*Glue logic*), können nun FPGAs größere Module wie CPUs und Co-Prozessoren enthalten. Dieses ermöglicht es, allgemeine Funktionen (z. B. mit CPU) sowie spezielle Funktionen (z. B. mit Co-Prozessoren) in demselben Baustein zu vereinen und die Performanz stark anzuheben. Aufgrund der ausreichend zur Verfügung stehenden Ressourcen kann eine Aufteilung der Rechenleistung (z. B. zur Bilddatenverarbeitung) in Software und Hardware erfolgen. In kompakten Kameras wurde bisher die Kompression der Bilddaten hauptsächlich durch Softwareroutinen mit entsprechend niedrigem Durchsatz durchgeführt. Eine Hardwareimplementierung war entweder sehr aufwendig (hoher Ressourcenbedarf und keine Anwendungsmöglichkeit für eine kompakte Kamera) oder aber qualitativ nicht ausreichend. Mit größeren FPGAs kann daher die Designoptimierung durch eine qualitativ hochwertige hardwareunterstützte Kompression mit einem kompakten, Ressourcen schonenden Aufbau einer Kamera vereint werden.

Aktuell sind zwei große FPGA-Familien zur höheren Systemintegration in Instrumentenrechnern von Interesse. Zum einen ist es die RTAX-S Familie der Firma Actel, zum anderen die RT-Virtex-II Familie der Firma Xilinx. Bis auf den internen Speicher ist die RTAX-S Familie weitaus robuster gegenüber der im Weltraum auftretenden Strahlung als die SRAM-basierte RT-Virtex-II Familie. Dieses ist insbesondere auf die verwendete *Antifuse*-Technologie und Implementierung dreifach-redundanter Zellen (*Triple Module Redundancy*, TMR) bei Flipflops zurückzuführen. Bei den SRAM-basierten RT-Virtex-II FPGAs können typische strahlungsbedingte Fehler nicht nur im internen Speicher, sondern auch z. B. in der Kombinatorik oder der Konfiguration des Bausteins auftreten. Hierdurch treten unterschiedliche Fehlertypen auf. Die Bausteine der RT-Virtex-II Familie stellen im Vergleich zur RTAX-S Familie jedoch um den Faktor 1,5 mehr Systemgatter und einen um den Faktor 4,8 größeren internen Speicher zur Verfügung.

Bei der Auswahl eines FPGA können die hohe Performanz und die hohe verfügbare Menge an Ressourcen der RT-Virtex-II Bausteine trotz höherer Empfindlichkeit gegenüber Strahlung ausschlaggebend sein. Zudem ist die Entwicklung mit SRAM-basierten Bausteinen praktikabler, da sie mehrfach programmiert werden können. Insgesamt erhält man mit Verwendung eines großen SRAM-basierten FPGA ein flexibles System während der Entwicklung. Das endgültige FPGA-Design kann kurz vor den abschließenden Tests vor Auslieferung des Instrumentenrechners integriert werden. Diese Vorgehensweise führt zu kürzeren Entwicklungszeiten und damit zu niedrigeren Kosten.

Zielstellung der Arbeit

Der mögliche Einsatz der, bezogen auf die Performanz und der zur Verfügung stehenden Ressourcen, überlegenen größeren SRAM-basierten FPGAs mit Weltraumtauglichkeit führt zu der Frage, inwieweit sie trotz ihrer vergleichsweise hohen Empfindlichkeit gegenüber Strahlung und der dadurch ggf. reduzierten Verfügbarkeit in Instrumentenrechnern innerhalb kompakter Kameras vorteilhaft eingesetzt werden können. Den zu erwartenden höheren Fehlerraten stehen unter Betrachtung der Randbedingungen im Weltraum die hohe Performanz, der relativ geringe Leistungsverbrauch und die höheren Integrationsmöglichkeiten für dedizierte Hardwareprozessoren gegenüber.

Das Ziel bei der Entwicklung eines Instruments ist ein hoher wissenschaftlicher Nutzen, der bei Kamerainstrumenten i. Allg. einem hohen Durchsatz an Bildern entspricht. Um bei gleicher Datenrate mehr Bilder zu übertragen, werden Bilder komprimiert. Eine Kompression wird nur akzeptiert, wenn sie auf einem Algorithmus beruht, der qualitativ hochwertige Kompressionsergebnisse liefert. Derartige Algorithmen benötigen jedoch üblicherweise viel Rechenleistung, so dass der Durchsatz eines Kamerainstruments häufig durch die Kompression begrenzt ist.

Im Folgenden soll gezeigt werden, wie eine hohe Rechenleistung mit akzeptablen Werten für Zuverlässigkeit und Verfügbarkeit und kompakter Bauweise verbunden werden kann. Hiermit entfernt man sich vom traditionellen Ansatz einer strahlungsfesten DPU – strahlungsfeste Bereiche werden in strahlungstolerante Bereiche mit Fehlertoleranz, d. h. Toleranz gegenüber strahlungsbedingten Fehlern, überführt. Analysiert und verglichen werden in dieser Arbeit strahlungsfeste und strahlungstolerante DPU-Implementierungen. Die strahlungstolerante DPU enthält als Kernkomponente einen SRAM-basierten RT-Virtex-II FPGA, der aufgrund des zu erwartenden Vorteils beim Aufbau einer kompakten Kamera mit hoher Performanz gegenüber dem RTAX-S FPGA den Vorrang erhält.

Die Anforderungen an notwendige Rechenleistung werden exemplarisch durch die Bearbeitung des Kompressionsalgorithmus JPEG2000 umgesetzt, der auf der *Wavelet*-Transformation beruht. Dabei ist es nicht das Ziel, ein möglichst hochwertiges Verfahren zu finden. Hingegen soll eine maßgeschneiderte Lösung basierend auf einem hochwertigen *State of the Art* Kompressionsverfahren entwickelt werden, die in einer entsprechenden kompakten DPU-Architektur für Missionen zur Planetenbeobachtung im Weltraum eingesetzt werden kann. Hierbei werden sowohl die Implementierungen des angepassten Algorithmus in Software als auch eine Umsetzung des Verfahrens in dedizierte Hardware als Co-Prozessor-Lösung vorgenommen.

Der Einfluss der Strahlung im All auf die DPUs wird in Anlehnung an die ESA-Missionen Venus Express und ExoMars sowie exemplarisch für einen Planeten mit Magnetfeld für eine Mission im polaren *Low Earth Orbit* (LEO) bestimmt.

In Kapitel 2 werden die Randbedingungen für den Weltraumeinsatz besprochen und strahlungsbedingte Fehlerraten berechnet. Sie verdeutlichen den Unterschied zum Entwurf für erdgebundene Anwendungen und bilden die Grundlage zur Ermittlung der zu erwartenden Fehlerraten für die verschiedenen DPU-Implementierungen.

Kapitel 3 stellt die Besonderheiten des JPEG2000-Standards dar. Es verdeutlicht als Grundlage für die Hardware-Implementierung des Verfahrens, welche Optimierungsmöglichkeiten bei Ausgliederung von Algorithmusteilen aus Software in Hardware vorhanden sind.

In Kapitel 4 werden die Besonderheiten einer Architektur für Instrumentenrechner in Kameras analysiert. Insbesondere ist der Einsatz SRAM-basierter FPGAs im Weltraum als eine Designvariante heraus gestellt.

Kapitel 5 behandelt die Implementierung eines traditionellen Entwurfs einer strahlungsfesten DPU mit einem SPARC-V8-kompatiblen Prozessor in ASIC-Ausführung. Es wird als Variante auch ein Massenspeicher implementiert, der die strahlungsfeste DPU dann in eine überwiegend strahlungsfeste DPU überführt. Die Softwarekompression der Bilddaten bestimmt die Performanz.

In Kapitel 6 wird eine strahlungstolerante DPU mit Verwendung eines SRAM-basierten RT-Virtex-II FPGA und der Implementierung eines Massenspeichers vorgestellt. Aus Vergleichsgründen zu Kapitel 5 beinhaltet der Aufbau ebenfalls einen SPARC-V8-kompatiblen Prozessor, der zusammen mit Logikblöcken wie Schnittstellenlogiken oder Speicherschutz in einem FPGA implementiert ist. Es sind mehrere DPU-Bereiche aus einer strahlungsfesten Version in eine strahlungstolerante Version überführt. Hier bestimmt die Softwarekompression ebenfalls die Performanz.

Mit Kapitel 7 wird eine hardwarebeschleunigte Kompression auf Basis des JPEG2000-Algorithmus vorgestellt. Die Analyse des JPEG2000-Standards in Kapitel 2 zeigt die Möglichkeit, bestimmte Algorithmusanteile zur Geschwindigkeitssteigerung in Hardware ausführen zu lassen. Darauf basierend wurde ein Ressourcen schonender, skalierbarer JPEG2000 Co-Prozessor entwickelt. Dieser ermöglicht eine starke Erhöhung der Performanz und damit des wissenschaftlichen Nutzens des Instruments.

In Kapitel 8 wird die in Kapitel 7 gezeigte hardwarebeschleunigte Kompression in die strahlungsfeste DPU aus Kapitel 5 und in die strahlungstolerante DPU aus Kapitel 6 integriert. Während im strahlungsfesten Design auf eine andere FPGA-Familie ausgewichen und zur Kompression zusätzlicher Zwischenspeicher in Form eines dedizierten Bausteins zur Verfügung gestellt werden muss, kann die Hardwarebeschleunigung bei der strahlungstoleranten DPU zusammen mit dem Prozessor in dem bereits vorhandenen SRAM-basierten RT-Virtex-II FPGA implementiert werden. Abschließend werden in Kapitel 8 Vergleiche bzgl. Performanz, Zuverlässigkeit, Strahlungsfestigkeit, Leistungs- und Energiebedarf, Platzbedarf, Volumen und

Masse sowie Flexibilität zwischen der strahlungsfesten und strahlungstoleranten DPU mit und ohne hardwarebeschleunigter Kompression durchgeführt.

Kapitel 9 gibt eine Zusammenfassung der Ergebnisse dieser Arbeit mit einem Ausblick auf mögliche Erweiterungen.

2 Einsatz im Weltraum

Bei dem Entwurf von Elektronik für den Einsatz in der Raumfahrt sind spezielle Randbedingungen zu berücksichtigen. Nur mit der Kenntnis z. B. der im Weltraum auftretenden Strahlung kann ein gezielter Entwurf erfolgen und können unnötige Überdimensionierungen vermieden werden. Im Folgenden werden daher die Eckpunkte für die richtige Auslegung besprochen.

2.1 Einführung

Ein Instrumentenrechner für die Raumfahrt muss für spezielle Anforderungen entworfen werden:

- Fehlende Wartungsmöglichkeiten
- Hohe Zuverlässigkeit, Datenintegrität und Verfügbarkeit
- Geringe Masse und geringer Energiebedarf
- Besondere Umweltumgebungen im Weltraum wie z. B. Strahlung (siehe Abschnitt 2.2)

Die Anforderungen variieren in Abhängigkeit von der Mission und dem Einsatz des Instruments. Im Allgemeinen sind Instrumente auf Raumsonden für lange Einsätze von einigen Monaten bis zu mehreren Jahren geplant. Bei *Deep Space* Missionen liegen allein die Transferzeiten von der Erde bis zur eigentlichen Zielposition bei mehreren Jahren.

Die **Wartung** eines Instrumentenrechners ist nach dem Start ausschließlich über die installierte Software möglich. Mit dem aktuellen Stand der Technik wird diese Einschränkung durch die Verwendung wiederprogrammierbarer SRAM-basierter FPGAs und damit durch den Hardwareanteil relativiert, der über veränderbare Konfigurationsdaten definiert ist.

Die Zuverlässigkeit eines Systems wird bestimmt durch die Überlebenswahrscheinlichkeit am Ende einer Mission. Sind die Fehlerwahrscheinlichkeiten $P(X_n)$ der N Komponenten unabhängig voneinander, ergibt sich die **Systemzuverlässigkeit** in einer Reihenschaltung bekanntermaßen zu:

$$R = P(X_0) \cdot P(X_1) \cdot \dots \cdot P(X_{N-1}). \quad (2.1)$$

Die Zuverlässigkeit eines gesamten Systems ist somit kleiner als die Zuverlässigkeit der schlechtesten Komponente. Hoch integrierte Bausteine haben in der Regel eine kleinere Ausfallwahrscheinlichkeit als ihre vergleichbare Ausführung aus mehreren geringer integrierten Bausteinen und es lässt sich hiermit in der Regel ein zuverlässigeres System aufbauen. Damit sind zunehmend integrierte Architekturen auch unter dem Gesichtspunkt einer ansteigenden Zuverlässigkeit zu sehen. Die Zuverlässigkeit kann durch die Auswahl spezieller Bauteile oder durch strukturelle Maßnahmen erhöht werden. Die Bauteile werden nach Möglichkeit nach