Oliver Bringmann, Walter Lange, Martin Bogdan
Eingebettete Systeme
De Gruyter Studium

Weitere empfehlenswerte Titel

Digitaltechnik und digitale Systeme
Eine Einführung mit VHDL
Jürgen Reichardt, 2021
ISBN 978-3-11-070696-3, e-ISBN 978-3-11-070697-0,
e-ISBN (EPUB) 978-3-11-070706-9

Elektronik für Informatiker
Von den Grundlagen bis zur Mikrocontroller-Applikation
Manfred Rost, Sandro Wefel, 2021
ISBN 978-3-11-060882-3, e-ISBN 978-3-11-060906-6,
e-ISBN (EPUB) 978-3-11-060924-0

VHDL-Simulation und -Synthese
Entwurf digitaler Schaltungen und Systeme
Jürgen Reichardt, Bernd Schwarz, 2020
ISBN 978-3-11-067345-6, e-ISBN 978-3-11-067346-3,
e-ISBN (EPUB) 978-3-11-067350-0

Mikrocontrollertechnik mit AVR
Programmierung in Assembler und C – Schaltungen und Anwendungen
Günter Schmitt, Andreas Riedenauer, 2019
ISBN 978-3-11-040384-8, e-ISBN 978-3-11-040388-6, e-ISBN (EPUB)
978-3-11-063688-8

FPGA Hardware-Entwurf
Frank Kesel, 2018
ISBN 978-3-11-053142-8, e-ISBN 978-3-11-053145-9; e-ISBN (EPUB)
978-3-11-053199-2

Oliver Bringmann, Walter Lange,
Martin Bogdan

Eingebettete Systeme

Entwurf, Synthese und Edge AI

4., überarbeitete Auflage

DE GRUYTER
OLDENBOURG

Autoren

Prof. Dr. Oliver Bringmann
Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Lehrstuhl für Eingebettete Systeme
72076 Tübingen, Germany
oliver.bringmann@uni-tuebingen.de

Dr. Walter Lange
71088 Holzgerlingen
wltrlange@aol.com

Prof. Dr. Martin Bogdan
Universität Leipzig
Fakultät für Mathematik und Informatik
Lehrstuhl für Neuromorphe Informationsverarbeitung
bogdan@informatik.uni-leipzig.de

ISBN 978-3-11-070205-7
e-ISBN (PDF) 978-3-11-070206-4
e-ISBN (EPUB) 978-3-11-070313-9

Library of Congress Control Number: 2021949185

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

© 2022 Walter de Gruyter GmbH, Berlin/Boston
Einbandabbildung: MARHARYTA MARKO / iStock / Getty Images Plus
Druck und Bindung: CPI books GmbH, Leck

www.degruyter.com

Vorwort

Der Siegeszug des Personal Computers in den achtziger Jahren des vergangenen Jahrhunderts wird abgelöst durch die wachsende Verbreitung der Eingebetteten Systeme. Eingebettete Systeme werden immer kleiner, heute bestehen sie aus mikroelektronischen Einheiten, deren Strukturen zu weniger als einem Tausendstel des Durchmessers eines Menschenhaares zusammengeschrumpft sind. Man spricht vom „unsichtbaren Computer“. Die elektronischen Winzlinge drängen nicht nur in alle Bereiche der Technik, der Wirtschaft, des Bauwesens, sondern auch in alle privaten Lebensbereiche wie den Haushalt, die Freizeit usw. Beinahe jeder Teenager trägt ein hoch kompliziertes Mini-System in Form eines Handys in der Tasche, aber auch in Kraftfahrzeugen, Photoapparaten, medizinischen Geräten, Flugzeugen, Haushalsgeräten, Fernsehapparaten usw. findet man Eingebettete Systeme, die dort Steuerungs-, Regelungs- und Datenverarbeitungs-Aufgaben wahrnehmen (Ubiquitous Computer, Ambient Intelligence [deMan03]). Nach heutigen Schätzungen werden bereits weit mehr Prozessoren in Eingebetteten Systemen eingebaut als in PCs. Damit wächst auch ständig der Bedarf an Programmcode für Eingebettete Systeme.

Das vorliegende Buch will der wachsenden Bedeutung der Eingebetteten Systeme Rechnung tragen. Es richtet sich hauptsächlich an Studenten der Informatik und der Naturwissenschaften in den Bachelor- und Masterstudiengängen, die sich für die praxisbezogene Facette der Informatik interessieren sowie an Ingenieure, die sich in das Themengebiet einarbeiten wollen und die Grundkenntnisse auf dem Gebiet der Elektrotechnik, Elektronik, der Rechnerarchitektur, der Technischen und der Theoretischen Informatik mitbringen.

Ziel des Buches ist es, einen Überblick über die Grundlagen, Anwendungen sowie die Entwicklungsmethodik von Eingebetteten Systemen zu vermitteln, wobei versucht wird, den Stoff mit vielen Beispielen aus der Praxis anschaulich und verständlich darzustellen. Für jeden einzelnen Themenbereich ist jedoch eine tiefer gehende Behandlung im Rahmen dieses Buches nicht möglich und der Spezialliteratur vorbehalten. Studenten, aber auch Ingenieure aus anderen Fachbereichen, die sich in das Gebiet Eingebettete Systeme einarbeiten wollen, werden eventuell einige der folgenden Fächer vertiefen müssen: Digitale Signalverarbeitung, Regelungs- und Steuerungstechnik, Rechenarchitektur, Betriebssysteme, Formale Verifikation, Spezifikationssprachen, Hardwarebeschreibungssprachen, Techniken des Niedrig-Leistungsaufnahme (Low Power), Datensicherheit, Zuverlässigkeit, Telekommunikation, drahtlose Kommunikation, Sensoren, Aktoren usw. Hinweise auf einschlägige Literatur werden gegeben.

Ein Lehrbuch stützt sich immer auch auf vorhandene Literatur. Wir haben zum Beispiel folgende Quellen verwendet: D. D. Gajski et al.: Embedded Systems Design, [Ga09], W. Wolf, Computers as Components, [Wolf02], Principles of Embedded Computing System Design [Wolf07], P. Marwedel, Eingebettete Systeme [Mar07], G. Martin und H. Chang,

https://doi.org/10.1515/9783110702064-202

Winning the SoC Revolution [Mart03] und viele andere. Weitere Bücher, wissenschaftliche Aufsätze, Normen usw. sind im Text und in den Bildunterschriften zitiert und in der Literaturliste zusammengefasst.

Eine Anmerkung zur Rechtschreibung: Zusammengesetzte Wörter werden aus Gründen der besseren Lesbarkeit meist in der Bindestrich-Form geschrieben. Zum Beispiel wird anstatt Hardwarebeschreibungssprachen die Bindestrich-Form Hardware-Beschreibungs-Sprachen verwendet. Beide Schreibformen existieren nebeneinander. Die im Buch aufgeführten Firmennamen sind gesetzlich geschützte Bezeichnungen.

Die meiste Fachliteratur ist in englischer Sprache und daher sind fast alle Fachausdrücke englisch. Oft gibt es deutsche Begriffe für diese Fachausdrücke, die in diesem Buch bevorzugt verwendet werden, jedoch wird in einem solchen Fall der englische Ausdruck das erste Mal und oft auch die Abkürzung in Klammern dahinter gesetzt. Zum Beispiel: „System-Verhaltensmodell“ (System Behaviour Model SBM). Viele Fachausdrücke haben sich allerdings bei uns so eingebürgert, dass eine Übersetzung nicht verstanden oder zur Verwirrung führen würde. Ein Beispiel dafür ist der Begriff „Chip“ als Siliziumplättchen, das eine gedruckte elektronische Schaltung enthält.

Dieses Buch ist aus Vorlesungen an den Universitäten Tübingen und Leipzig entstanden. Wir danken Herrn Professor Dr. Wolfgang Rosenstiel für die Unterstützung und Beratung bei den Tübinger Vorlesungen und den Kollegen des Arbeitsbereichs Technische Informatik der Universität Tübingen für viele Diskussionen, Anregungen und Hilfen. Ein herzlicher Dank geht an Herrn Professor Dr. Oliver Bringmann für die fachliche Durchsicht und für viele konstruktive Vorschläge zur Aktualisierung des Inhalts. Des weiteren danken wir Herrn Prof. Dr. Udo Kebschull (ehemals Universität Leipzig) für die Einführung der Vorlesung VHDL an der Universität Leipzig, die den Nährboden für die Vorlesung Eingebettete Systeme bereitet hat. Weiterhin danken wir Herrn Jörn Hoffmann von der Universität Leipzig für die Erstellung des Titelbildes und der Durchsicht eines Kapitels sowie den Herren Wolfgang Fuhl und Benjamin Matthes aus Tübingen für die Durchsicht einzelner Kapitel des Buches und für viele Korrekturvorschläge.

Die Struktur des Buches ist wie folgt: Im ersten Kapitel wird der Begriff „Eingebettete Systeme“ erklärt und abgegrenzt, Beispiele werden aufgeführt und erläutert sowie der typische Aufbau von Eingebetteten Systemen beschrieben und Bauformen und Implementierungsarten werden vorgestellt. Kapitel 2 gibt einen Überblick über die Verwendung von Mikroprozessoren in Eingebetteten Systemen. Kapitel 3 geht auf verteilte Systeme und Verbindungselemente bei Eingebetteten Systemen ein, beispielsweise auf parallele und serielle Busse. In Kapitel 4 wird das weite Gebiet der Entwicklungsmethodik von den Anforderungen über die Spezifikation zur Modellierung und der Realisierung von Eingebetteten Systemen vorgestellt. Kapitel 5 behandelt Beschreibungssprachen für den Systementwurf mit Schwerpunkt auf VHDL (Very High Speed Integrated Circuit Hardware Description Language) und SystemC. Kapitel 6 ist den Sensornetzwerken gewidmet. Kapitel 7 stellt den theoretischen Hintergrund der High-Level-Synthese vor. High-Level-Synthesesysteme sind wichtige Werkzeuge für den Entwickler von Eingebetteten Systemen. Die Kapitel 1, 2, 3 und 5 sind Grundlagenkapitel und für Bachelorstudenten gedacht, während die Kapitel 4, 6 und 7 weiterführende Themen behandeln und eher für Masterstudenten und Fortgeschrittene bestimmt sind.

Für positive Kritik und vor allem Verbesserungsvorschläge sind wir aufgeschlossen und dankbar. Unsere Email-Adressen können Sie beim Verlag erfragen oder auch im Internet auf den Seiten der Universitäten Tübingen und Leipzig finden. Wir wünschen unseren Lesern einen guten Lernerfolg beim Lesen des Buches!

Die Autoren. Tübingen und Leipzig im Dezember 2012.

Vorwort zur zweiten Auflage

Wir freuen uns, dass sich Herr Dr. Thomas Schweizer als dritter Autor unserem Autorenteam angeschlossen hat.

Die Texte und Bilder der zweiten Auflage wurden überarbeitet, etwas komprimiert und aktualisiert. Die kurzen Kapitel 1 und 2 wurden zusammengelegt. Einige Abschnitte aus Kapitel 4 „Entwicklungsmethodik“ wurden in die Kapitel 3 „Kommunikation“ und Kapitel 7 „Software-Synthese und High-Level-Synthese“ übernommen, da sie dort besser zum Kontext passen.

Unser Dank gilt folgenden Personen: Herrn Dr. Stefan Stattelmann für die Genehmigung, seine Materialien im Abschnitt „Performanzabschätzung“ im Kapitel 4 verwenden zu dürfen, Herrn Dipl.-Ing. (FH) Dieter Schweizer für Korrekturlesen, dem Leser Bastian Haetzer vom Institut für Technische Informatik der Universität Stuttgart für Korrekturen und Fragen zum Text, Herrn Dipl.-Inform. Jörn Hoffmann für Anregungen zur Erweiterung des Abschnitts „Betriebssysteme“ im Kapitel 2, dem Lektor und den Mitarbeitern des DeGruyter-Verlags für die sehr gute Zusammenarbeit.

Für positive Kritik, Korrektur- und Verbesserungsvorschläge sind wir aufgeschlossen und dankbar. Schicken sie bitte diese an
Herrn Dr. Thomas Schweizer (tschweiz(at)informatik.uni-tuebingen.de) oder an den Verlag.

Walter Lange, Martin Bogdan und Thomas Schweizer.
Tübingen und Leipzig im März 2015.

Vorwort zur dritten Auflage

Wir freuen uns, dass sich Herr Prof. Dr. Oliver Bringmann unserem Autorenteam angeschlossen hat.

Alle Kapitel des Buches wurden aktualisiert. Die Gliederung wurde verändert. „Entwicklungsmethodik“ und „Modelle“ sind als Kapitel 2 und Kapitel 3 ausgewiesen und wurden überarbeitet. Insbesondere wurde das Kapitel „Hardware-Synthese“ neu gestaltet und durch den Teil „RT-Synthese“ erweitert.

Wir danken Herrn Dustin Peterson vom Lehrstuhl für Eingebettete Systeme der Universität Tübingen für die Mithilfe an der Bearbeitung der Bilder, für wertvolle Diskussionen und Beiträge zu den Themen Hardwaresynthese und Low Power Design (siehe Abschnitt 2.10, Seite 72, und Abschnitt 5.1.19, Seite 223). Ein weiterer Dank gilt den Mitarbeitern des DeGruyter-Verlags für die sehr gute Zusammenarbeit.

Für positive Kritik, Korrektur- und Verbesserungsvorschläge sind wir aufgeschlossen und dankbar. Schicken sie bitte diese an
Herrn Professor Dr. Oliver Bringmann (oliver.bringmann(at)uni-tuebingen.de) oder an den Verlag.

Oliver Bringmann, Walter Lange und Martin Bogdan.
Tübingen und Leipzig im Juni 2018.

Vorwort zur vierten Auflage

Die Kapitel des Buches wurden aktualisiert.

Künstliche Intelligenz (KI) dringt mehr und mehr in viele unserer Lebensbereiche ein. Dem haben wir Rechnung getragen und eine Einführung in Systeme mit künstlicher Intelligenz in Kapitel 1 eingefügt (Seite 11). Eingebetteten KI-Systemen wird ein neues Kapitel gewidmet (Kapitel 4, Seite 107), das in die Grundlagen von KI-Systemen, insbesondere Neuronale Netze einführt, sowie maschinelles Lernen, Hardware für KI-Systeme und weitere Themen behandelt. Auf autonome Fahrzeuge als klassische Anwendung von KI-Systemen gehen wir etwas näher ein.

Der Abschnitt „Bussysteme im Kraftfahrzeug“ wurde durch die aktuellen Entwicklungen und Trends erweitert (Seite 453).

Wir danken Herrn Dr. Dustin Peterson für seine Mithilfe beim Bearbeiten und Verwalten der Dateien für dieses Buch. Ein weiterer Dank gilt Herrn Paul Palomero Bernardo und seinem Team für den Beitrag „*UltraTrail*“ [PaBe20], dessen Text gekürzt und in den Rahmen des Buches eingepasst wurde.

Wie bei jeder neuen Auflage danken wir den Mitarbeitern des DeGruyter/Oldenbourg-Verlags für die sehr gute und freundliche Zusammenarbeit.

Für positive Kritik, Korrektur- und Verbesserungsvorschläge sind wir dankbar. Schicken sie bitte diese an
Herrn Professor Dr. Oliver Bringmann (oliver.bringmann(at)uni-tuebingen.de) oder an den Verlag.

Oliver Bringmann, Walter Lange und Martin Bogdan.
Tübingen und Leipzig im Oktober 2021.

Inhaltsverzeichnis

1 Einführung, Systeme, Bauformen und Technologien

Eingebettete Systeme nehmen dem Menschen vielfältige Routinearbeiten ab. Wir betrachten als Beispiel das Eingebettete System „Temperaturregelung“ (siehe Abbildung 1.3, Seite 5). Zu Beginn des vorigen Jahrhunderts wurde die Temperatur in einem Raum durch die Intensität des Heizens und der Lüftung bestimmt, die eine Person im Raum vornahm. Dadurch waren relativ große Temperaturschwankungen unvermeidlich. Erst mit dem Aufkommen und der Verbreitung der Messgeräte, der Elektronik und der mathematischen Grundlagen wurden Geräte ersonnen, die Regel- und Steueraufgaben übernehmen konnten. Mit dem Eingebetteten System „Temperaturregelung“ ist es möglich, sehr effizient und wirtschaftlich, ohne menschlichen Arbeitsaufwand die Temperatur in einem Raum in engen Grenzen konstant zu halten.

Dieses Kapitel beginnt mit einer Auslegung des Begriffs Eingebettete Systeme und mit einigen Beispielen für Eingebettete Systeme aus unserer Umgebung und der Industrie. Der typische Aufbau von Eingebetteten Systemen wird vorgestellt und anhand der Beispiele Temperaturregelung und „Smartphone“ veranschaulicht. In weiteren Abschnitten gehen wir auf verschiedene Bauformen und auf Technologien von Eingebetteten Systemen ein.

1.1 Begriffsbestimmung und Beispiele

Eingebettete Systeme sind Rechensysteme, die in einen technischen Kontext, bzw. in ein übergeordnetes System eingebunden sind und vordefinierte Aufgaben erfüllen. Sie werden ausschließlich für diese Funktionen entwickelt. In dem vorliegenden Buch werden ausnahmslos elektronische Systeme behandelt. Eingebettete Systeme sind, wenn nicht gerade das Herzstück, so doch meist unverzichtbarer integraler Bestandteil in beispielsweise folgenden übergeordneten Systemen (siehe Abbildung 1.1):

- in Transportsystemen, zum Beispiel in Kraftfahrzeugen als Antriebssteuerung, Klimaregelung, Bremssysteme, Navigationssysteme, Assistenz- und autonome Fahrsysteme.
- in Flugzeugen z. B. als Autopilot, Klimaanlage und Druckregelung in der Kabine,
- in Eisenbahnen,
- in Schiffen,
- in medizinischen Geräten als Tomographen, Mikromessgeräten, zum Beispiel für die Darmspiegelung, in Herzschrittmachern, künstlichen Prothesen usw.
- in der mobilen und Festnetz-Kommunikation,
- in militärischen Geräten und Anlagen,
- im „intelligenten Haus“ bzw. im „intelligenten Büro“ als Klimaregelung, Sicherheitsüberwachung, Anzeige von belegten Räumen usw.,

https://doi.org/10.1515/9783110702064-001

- in optischen Geräten,
- in Produktionsanlagen als Fließbandsteuerung, zur Steuerung von Hochregallagern, als Klimaregelung, in Alarmsystemen usw.,
- in Kraftwerken, als Steuerung und Regelung der Energieumwandlung in Wärme oder Strom und deren Verteilung,
- in der Chemie- und Verfahrenstechnik als Prozess-Steuerungen,
- in Multimedia-Anwendungen, z. B. in der Unterhaltungselektronik und in Spielen,
- in Handel- und Bankenendgeräten als Kassensysteme, in speziellen Druckern, zum Beispiel in Kontoauszugsdruckern, Sparbuchdruckern, in Geldausgabe-Automaten usw.

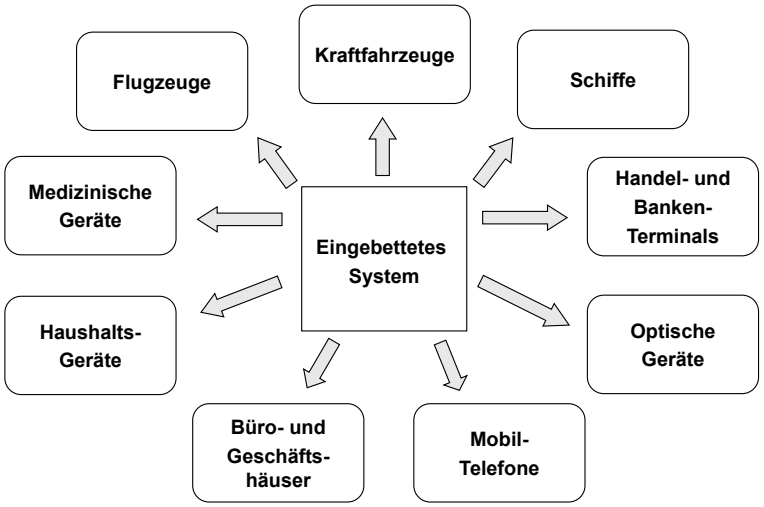

Abbildung 1.1: *Beispiele von übergeordneten Systemen, die Eingebettete Systeme enthalten können.*

Wie bereits erwähnt, gehört zu einem Eingebetteten System immer ein übergeordnetes System, in dem es eine dedizierte Aufgabe ausführt. Damit ist der Begriff „Eingebettetes“ System hinlänglich eingegrenzt. Ist beispielsweise das Platzbuchungssystem einer Fluggesellschaft ein Eingebettetes System? Antwort: Nein, es ist selbst ein eigenständiges Software-System, das zwar in einem oder mehreren Computern läuft, die aber nicht als übergeordnete Systeme betrachtet werden.

Der Trend zur Miniaturisierung Eingebetteter Systeme und zur Durchdringung aller Arbeits- und Lebensbereiche schlägt sich nieder in den folgenden Begriffen: „allgegenwärtiges Computing“ (Ubiquituos Computing [Mar07]), „überalleindringendes Computing“ (Pervasive Computing [Hnsm01]), „Intelligente Umgebung“ (Ambient Intelligence, AmI [deMan03]), Internet of Things (IoT) und Cyber-Physische Systeme. Während „allgegenwärtiges Computing“ die Tendenz „Information überall und zu jeder Zeit“ zum Ausdruck bringt, zielt das „überalleindringende Computing“ mehr in die Richtung praktischer Geräte wie intelligente Assistenten, intelligente Raum- und Gebäudesysteme (Smart Home, Smart Building), Haushaltsmaschinen, die nach und nach von Eingebetteten Systemen gesteuert bzw. geregelt werden und mit den Benutzern kommunizieren.

„Ambient Intelligence“ bedeutet, dass hochintegrierte Chips in unsere nächste Umgebung einziehen werden, in unsere Kleider, Jacken, Handschuhe, beispielsweise als medizinische Helfer, Körpersensoren, Gesundheitsüberwachung, Smart Watches, zur Unterhaltung, im Sport oder als Helfer im Beruf, z. B. als Erinnerungs-Ansage für einen Termin bzw. als Navigationshilfe in einem unübersichtlichen Gebäude. Auf Cyber-Physische Systeme gehen wir in Abschnitt 1.4, Seite 8 näher ein.

1.2 Systemkategorien

Im vorhergehenden Abschnitt werden Beispiele Eingebetteter Systeme in verschiedenen Industriezweigen und Anwendungen aufgeführt. Eine Einteilung in verschiedene Kategorien kann aus verschiedenen Gesichtspunkten vorgenommen werden. A. Jantsch [Jan04] zeigt eine Klassifikation entsprechend der Systemeigenschaften.

Eine Klassifizierung der Eingebetteten Systeme im Hinblick auf die Schnittstellen nach außen und in Bezug auf die Bedienung des jeweils übergeordneten Systems kann für den zivilen Nutzungsbereich – ohne Anspruch auf Vollständigkeit – wie folgt vorgenommen werden:

- Eingebettete Systeme, die Steuerungs- und Regelungsaufgaben wahrnehmen und die keine offenen Bedienungs-Schnittstellen zum Benutzer haben. Dies sind überwiegend sogenannte „versteckte Systeme“ wie z. B. die Heizungsregelung, ABS im Kraftfahrzeug, Autopilot im Flugzeug, Herzschrittmacher, Fokussiersteuerung in der Kamera, Geräte der Unterhaltungs- und Kommunikationselektronik wie z. B. Fernseh-, Video- und Stereo-Geräte usw.
- Geräte, die von der Fachfrau bzw. vom Fachmann bedient werden. In diesem Zusammenhang bedeutet Fachfrau/Fachmann eine Person, die Kenntisse über das Einsatzgebiet des Geräts besitzt oder einer Anlernung bedarf, entweder durch eine Fachperson oder entsprechende Literatur. Beispiele dafür sind: Steuerungen von chemischen Prozessen und Kraftwerken, medizinische Geräte, physikalische Messgeräte usw.
- Datenendgeräte, die von der Fachfrau bzw. vom Fachmann bedient werden. Datenendgeräte sind Geräte, die mit einem zentralen Computer verbunden sind. Oft stehen diese Geräte mit einer Datenbank in Verbindung. Beispiele sind: Supermarkt-Kasse, elektronische Waagen im Supermarkt, Drucker in LAN-Umgebungen usw.
- Selbstbedienungsgeräte, die von Laien benutzt werden: z. B. alle sogenannten Automaten wie Fahrkartenautomat, Geldausgabeautomat, Haushaltsmaschinen usw. Für diese Geräte ist besondere Beachtung auf eine gute Benutzerführung zu legen. Die Bedienung muss einfach sein und die Bedienerführung gut verständlich. Besonders wichtig ist, dass Fehlbedienungen keine katastrophalen Folgen haben, sondern wieder geduldig zum Ausgangspunkt der Bedienung führen und eventuell dem Benutzer den Fehler anzeigen. Selbstbedienungsgeräte, die in der Öffentlichkeit stehen, haben oft Vandalismus zu erleiden, d. h. sie müssen entsprechend mechanisch robust aufgebaut werden.
- Mobile, tragbare Geräte für die private und geschäftliche Nutzung meist mit drahtloser Verbindung zu einer Basisstation bzw. einem Server wie zum Beispiel Smartphones, Laptops und Tablets.

Je nach Systemkategorie werden an die übergeordneten Systeme und an die Eingebetteten Systeme verschiedene Anforderungen gestellt (siehe Abschnitt 2.1, Seite 34).

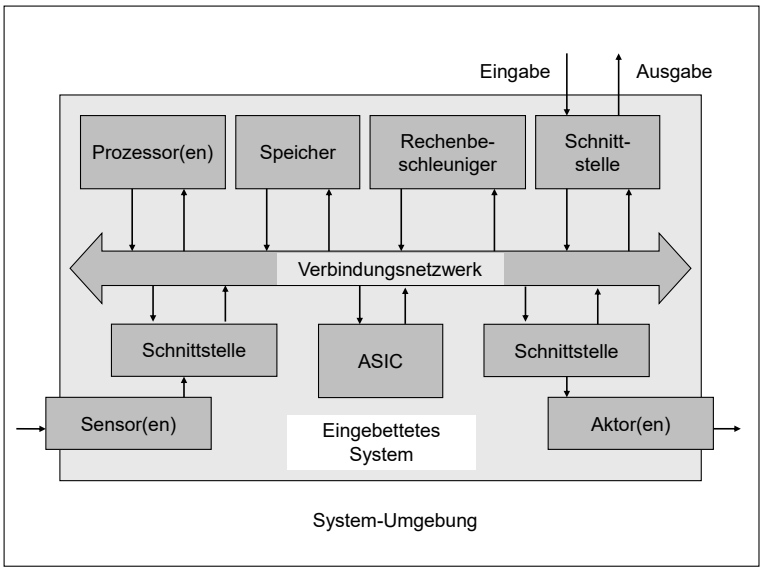

Abbildung 1.2: *Typischer Aufbau eines Eingebetteten Systems.*

1.3 Typischer Aufbau

Der **typische Aufbau** von Eingebetteten Systemen ist in Abbildung 1.2 dargestellt. Ein Eingebettetes System enthält in der Regel einen Prozessor, der ein Mikroprozessor, Mikrocontroller oder ein anwendungsspezifische Prozessor sein kann. Auf die Art und die Anwendungsbereiche der verschiedenen Prozessoren wird intensiver in Kapitel 9 eingegangen. In vielen Fällen ist ein zweiter, dritter usw. Prozessor bzw. ein ASIC (Application Specific Integrated Circuit), also ein anwendungsspezifischer Schaltkreis nötig, um die erforderliche Rechenleistung zu bewältigen, die bei Eingebetteten Systemen oft in „Echtzeit“ ausgeführt werden muss (siehe Abschnitt 2.1.1, Seite 36). Diese zusätzlichen Prozessorsysteme werden „Rechenbeschleuniger“ genannt. Heute werden meist „Mehrprozessorsysteme“ oder Multiprozessor-Systeme (siehe Seiten 408 und 29) eingesetzt. Der Speicher enthält sowohl Daten als auch das Ausführungs-Programm des Prozessors bzw. der Prozessoren und kann sowohl als RAM (Random Access Memory) als auch als ROM (Read Only Memory) bzw. als Flash-Speicher ausgeführt sein. Als zusätzliche Prozessoren können auch FPGAs (siehe Seite 26) zum Einsatz kommen. Beispiele für die Verwendung von Mehrprozessorsystemen sind Smartphones, Videogeräte, Laptops, usw.

Wir unterscheiden Eingabe- und Ausgabe-**Schnittstellen**. Beispiele für Eingaben sind bei einer Temperaturregelung der Sollwert der Temperatur und Parameter der Heizanlage. Ausgaben sind beispielsweise Fehler-Anzeigen. Schnittstellen sind auch nötig für

Sensor-Eingaben und **Aktor**- (oder „Aktuator“)-Ausgaben. Der Wert des Temperatur-Sensors in einer Temperaturregelung ist eine Eingabe. Der Stellwert für das Mischer-Ventil, das beispielsweise in einer Heizungsregelung (siehe Abbildung 1.3) die Mischung des Heizungswassers mit dem Heizwasser-Vorlauf und -Rücklauf regelt und damit die Temperatur der Heizkörper bestimmt, stellt eine Aktor-Ausgabe dar.

Das **Verbindungsnetzwerk** in einem eingebetteten System ist in der Regel ein „Bus“ (siehe Abschnitt 10.2, Seite 424). Ein **ASIC** (siehe Abbildung 1.19) ist ein anwendungsspezifischer Integrierter Schaltkreis. Es ist ein Hardware-System, das speziell für eine dedizierte Anwendung entwickelt und gefertigt wird. Die Vorteile von ASICs sind folgende: Bei sehr hohen Stückzahlen sind sie sehr kostengünstig. Die Ausführungszeit einer Berechnng auf einem ASIC kann relativ kurz sein, ebenso ist die Leistungsaufnahme relativ gering. Die Nachteile von ASICs sind: Da sie nur für die eine Anwendung bestimmt sind, sind sie recht unflexibel, auch kleine Änderungen sind nicht möglich. Der Aufwand für die Entwicklung und Einführung in die Produktion von ASICs ist relativ hoch.

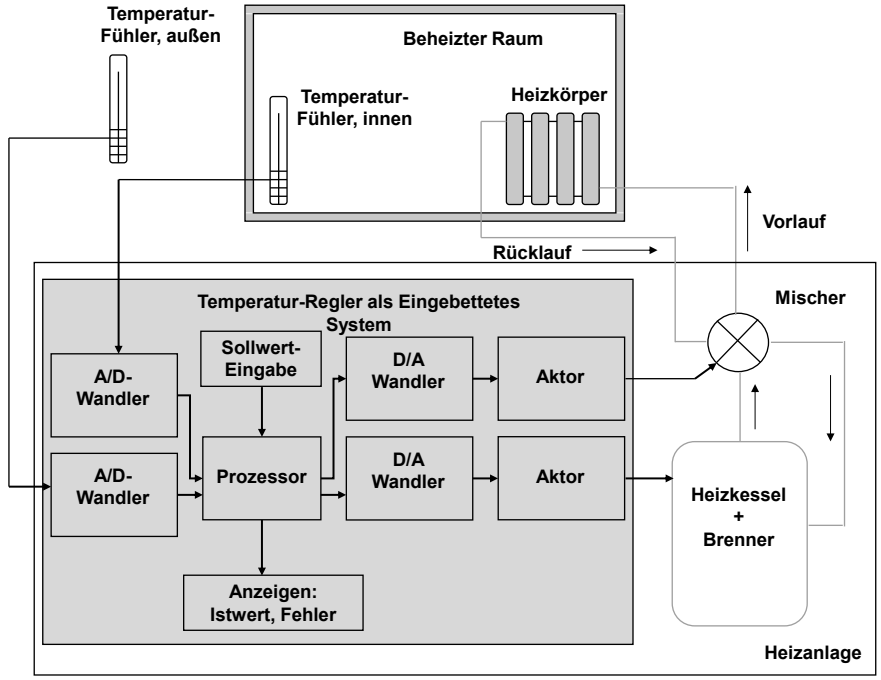

Abbildung 1.3: *Schematische Darstellung einer Temperaturregelung als Beispiel für ein Eingebettetes System. Das übergeordnete System ist die Warmwasser-Heizanlage.*

Abbildung 1.3 zeigt schematisch ein einfaches Beispiel für eine Temperaturregelung als Eingebettetes System. Das übergeordnete System ist die zentrale Warmwasser-Heizungsanlage für ein Haus. Gezeigt wird ein beheizter Raum, in dem sich ein Heizkörper und als Sensor ein Temperaturfühler befindet. Ein weiterer Temperaturfühler ermittelt die Außentemperatur. Das Eingebettete System hat an den Sensoreingängen Analog/Digital-Wandler (A/D-Wandler), in dem die analogen Temperaturfühler-Signale in digitale Sig-

nale umgewandelt werden. Der Prozessor, der als Eingabe den Temperatur-Sollwert und die Parameter der Heizanlage erhält, berechnet aus dem Sollwert und dem Istwert der Temperatur im Raum den Stellwert für den Mischer, der den Heißwasser-Zufluss für den Heizkörper im Raum steuert. Der Stellwert liegt zunächst als digitaler Wert vor und wird in einem Digital/Analog-Wandler (D/A-Wandler) in einen analogen Wert umgewandelt.

Der Außenfühler dient dazu, die Heizleistung zusätzlich zu steuern und bei hohen Aussentemperaturen den Brenner ganz abzuschalten. Je nach Anforderungen an die Regelgenauigkeit muss in diesem Beispiel der Prozessor in der Lage sein, entweder Differenzen zwischen Ist- und Sollwert zu bilden (Proportional P-Regler) oder auch bei höheren Anforderungen die Eingangsfunktion zu integrieren bzw. zu differenzieren (PI- bzw. PID-Regler). Als Parameter werden z. B. das Raumvolumen, die Leistung der Heizanlage, der Zusammenhang zwischen Stellwert und Heizenergiezufuhr eingegeben.

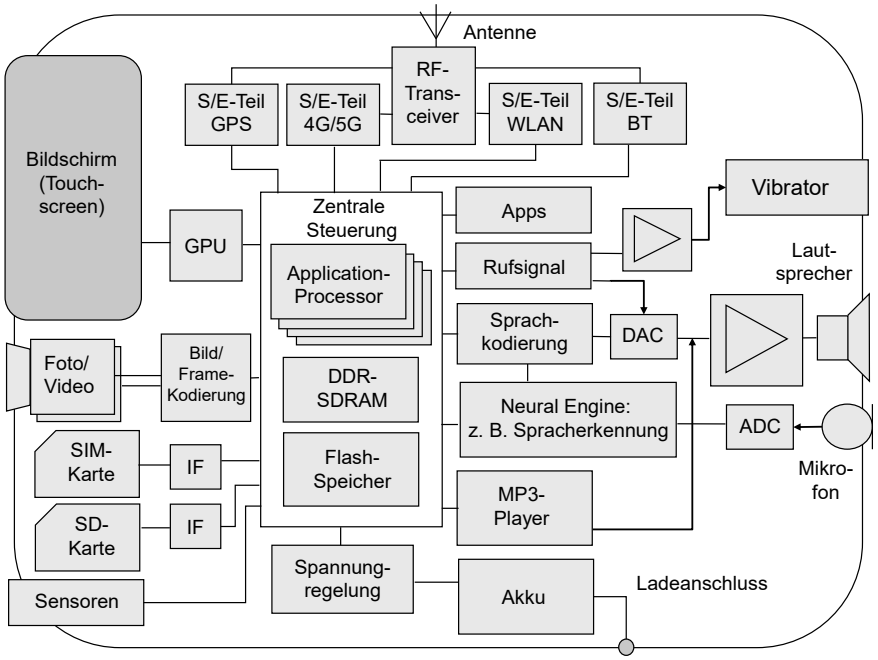

Abbildung 1.4: *Vereinfachte und schematische Darstellung eines „Smartphone“ mit seinen Teilsystemen als Beispiel für Eingebettete Systeme. IF heißt „Interface“ (Schnittstelle). S/E-Teil heißt Sende/Empfangsteil.*

Ein etwas populäreres Beispiel für Eingebettete Systeme ist ein **Smartphone**, dessen Aufbau schematisch vereinfacht in Abbildung 1.4 gezeigt wird. Ein Smartphone ist ein mobiles Telefon mit vielen, teils hochkomplizierten Zusatzfunktionen bzw. Teilsystemen. Es enthält auf kleinstem Raum zum Beispiel folgende Systeme:

– Eine zentrale Steuerung mit einem „Applikations-Prozessor“, der aus 4-8 Prozessorkernen bestehen kann. Zum Beispiel wird das „Snapdragon 845“-Modul der Firma Qualcomm mit dem „Kryo 385“-Prozessorbaustein verwendet [Qual18], der vier ARM-Cortex A75 und vier ARM-Cortex A55 Prozessorkerne integriert hat (siehe

Seite 410). An diesen Prozessoren sind Speichereinheiten angeschlossen, wobei der DDR-SDRAM (Double Data Rate Synchronous Dynamic RAM) als Arbeitsspreicher dient, während der Flash-Speicher als Langzeitspeicher (wie eine Festplatte) verwendet wird.
- Ein Energiespeicher (Akku) mit Ladeanschluss-Buchse und Spannungsregelung.
- Ein berührungsempfindlicher Bildschirm (Touch Screen) für Anzeige und Eingabe. Eine Tastatur für alphanumerische/numerische Zeichen wird bei Bedarf angezeigt. Videos und Computerspiele können dargestellt werden. Dafür ist ein Graphikprozessor nötig (GPU, Graphic Processor Unit).
- Eine SIM-Karte mit Schnittstelle (Interface IF).
- Eine SD-Karte (Secure Digital Memory Card) als Speichererweiterung.
- Sende-Empfangsteile (S/E-Teil) für ein mobiles Telefon mit 4G oder 5G-Technologie (4. oder 5. Generation), für GPS (Global Positioning System), für WLAN (Wireless Local Area Network) und für Bluetooth (BT). Der RF-Transceiver (Radio-Frequency Transmit/Receive) besteht aus einem HF-Teil und einem Basisband-Prozessor. Für den 4G/5G-Sende/Empfangsteil wird der Rechenaufwand für die Kodierung/Dekodierung der Übertragungsprotokolle mit bis zu 16 applikationsspezifischen Prozessorkernen bewerkstelligt.
- Mikrofon, Sprachcodierung und -Decodierung. Vor dem Lautsprecher liegt ein Audioverstärker.
- In der Regel zwei Fotoapparate, davon einer mit Videoaufnahmefähigkeit.
- Eine „Neural Engine“ z. B. für Spracherkennung (siehe Seite 419).
- MP3-Player und andere Anwendungen, die ladbar sind (Apps).
- Sensoren, z. B. Lage- und Beschleunigungssensor in MEMS-Technologie (Seite 20).

	Sensoren	Sensor-Schnittstelle	Aktor-Schnittstelle	Aktoren
Temparatur-regelung	Temparatur-fühler	A/D-Wandler	D/A-Wandler, Komparatoren, Pulsformer	Mischer-Motor Heizungs-Schalter
Antiblockier-System (ABS)	Schlupfanzeige Drehzahl	A/D-Wandler	D/A-Wandler Pulsformer	Hydraulik-Ventile
Laserdrucker	Temparatur-fühler, Tasten, Schalter	A/D-Wandler, Parallel-schnittstelle	D/A-Wandler, Pulsformer	Lasersteuerung, Leistungs-Elektronik für Motoren
Verbrennungs-Motor-Brennstoff-Einspritzung	Motor-Drehzahl, Position des Gaspedals und der Kurbelwelle	A/D-Wandler, Komparatoren	D/A-Wandler, Pulsformer	Zündung, Brennstoff-Einspritzung

Abbildung 1.5: *Beispiele für Sensoren/Aktoren und deren Schnittstellen bei verschiedenen Eingebetteten Systemen.*

1.3.1 Beispiele für Sensoren und Aktoren

Tabelle Abbildung 1.5 zeigt Beispiele verschiedener Eingebetteter Systeme, deren Sensoren und Aktoren mit den jeweiligen dazugehörigen Schnittstellen. Aufgeführt ist eine Temperaturregelung, ein Antiblockiersystem (ABS), ein Laserdrucker und eine

Verbrennungs-Motoreinspritzung im Automobil. Sensoren und viele Aktoren werden heute hauptsächlich in Mikrosystemtechnik hergestellt [Elw01] (siehe Abschnitt 1.7, Seite 20).

1.4 Cyber-Physische Systeme (Cyber Physical Systems)

Eine grundlegende Definition von Cyber-Physischen Systemen (CPS) wurde 2006 von Edward A. Lee eingeführt (übersetzt nach [Lee06]): *„Cyber-Physische Systeme sind die Kombination aus Rechenprozessen (Computation) und physischen Prozessen. Eingebettete Rechner und Netzwerke beobachten und kontrollieren physische Prozesse, üblicherweise mit Rückkopplungen, wobei der physische Prozess den eingebetteten Rechner beeinflußt und umgekehrt.“*

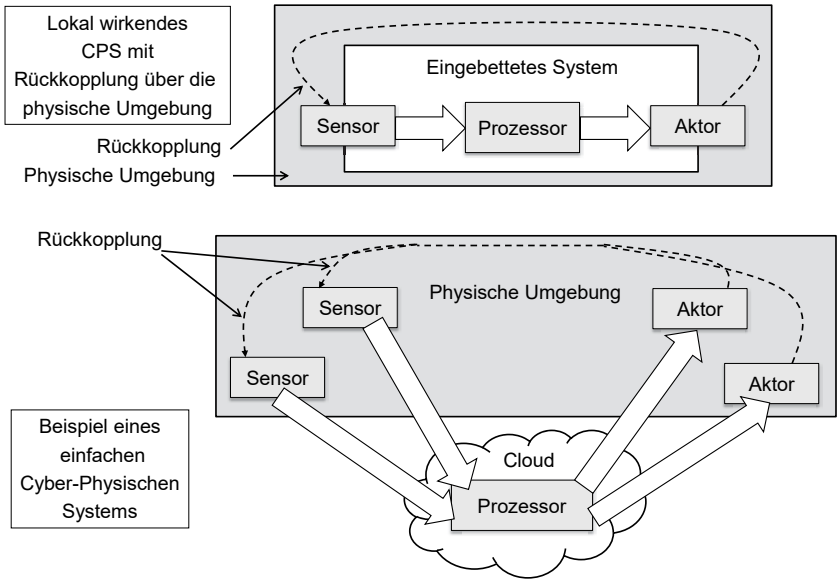

Abbildung 1.6: *Oben: Schematischer Aufbau eines lokal wirkenden CPS, es entspricht einem Eingebetteten System. Unten: Beispiel eines Cyber-Physischen Systems, die Verarbeitung wird in ein entferntes System verlegt, z. B. in die Cloud.*

Der ursprünglich von Edward A. Lee eingeführte Begriff des CPS erweitert die Definition von Eingebetteten Systemen um die einbettende Umgebung. Darunter ist zu verstehen, dass das Eingebettetes System mit der Umgebung interagiert und z. B. ein geschlossener Regelkreis mit Eingebettetem System und der physischen Umgebung in einem ganzheitlichen Modell beschrieben werden kann. Im Jahr 2011 wurde von der Deutschen Akademie der Technikwissenschaften [AKT18] die Bezeichnung CPS folgendermaßen erweitert: Bei Cyber-Physischen Systemen handelt es sich um verteilte Eingebettete Systeme (siehe Seite 17), die über ein Netzwerk von Sensoren und Aktoren auf die reale Welt zugreifen

und damit Steuerungs- und Regelungsaufgaben über das Netzwerk koordiniert ausführen können. Die reale Welt wird durch CPS mit der virtuellen Welt der Informationstechnik zu einem Internet der Dinge (IoT), Daten und Dienste verknüpft.

Dabei sind verschiedene Spielarten vorstellbar: So können verteilte Sensoren/Aktoren über die Cloud in ein CPS eingebunden sein oder der Steuerungs- oder Regelungsdienst kann selbst in der Cloud ausgeführt werden.

Abbildung 1.6 Seite 8 oben zeigt ein einfaches, lokal wirkendes CPS mit Rückkopplung über eine physische Umgebung. Das entspricht einem Eingebetteten System (hier einem Regelsystem) wie es in Abbildung 1.2, Seite 4 dargestellt wird. In Abbildung 1.6 unten wird das Beispiel eines allgemeinen CPS gezeigt, dessen Sensoren und Aktoren in einer physischen Umgebung verteilt sind und die Datenverarbeitung in einem Prozessor in der Cloud geschieht.

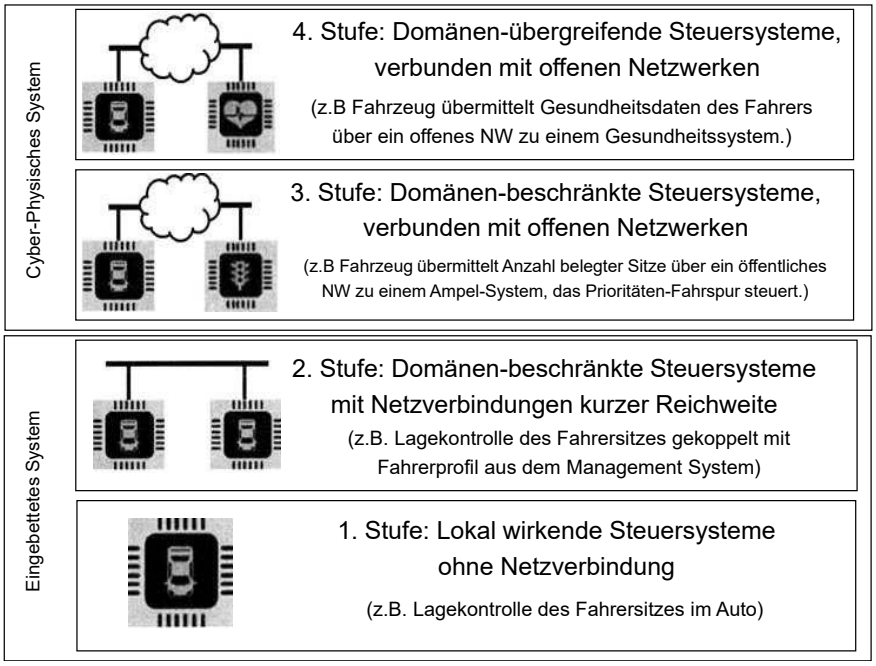

Abbildung 1.7: *Die Klassifikationsstufen von Eingebetteten und Cyber-Physischen Systemen nach Kowalewski [Kow14]. Texte ins Deutsche übersetzt. NW bedeutet Netzwerk.*

Die 4 Klassifikationsstufen für Eingebettete und Cyber-Physische Systeme

Kowalewski et al. [Kow14] unterscheiden vier Klassifikationsstufen von Eingebetteten und Cyber-Physischen Systemen. Am Beispiel der Automobilbranche werden diese vier Stufen in Abbildung 1.7 gezeigt. Stufe 1 zeigt ein Eingebettetes System am Beispiel eines lokal wirkenden Steuersystems, das die Lagekontrolle eines Fahrersitzes im Automobil bewerkstelligt (Non-networked local acting control systems). In der Stufe 2 sind Eingebettete Systeme an ein lokales Netzwerk, z. B. an ein WLAN angeschlossen (Domain-restricted close-networked control systems). Als Beispiel wird die Lagekon-

trolle des Fahrersitzes, gekoppelt mit dem Fahrerprofil aus einem Management System erwähnt. Die dritte und vierte Stufe zählen zu den Cyber-Physischen Systemen.

In der 3. Stufe bewegt sich beispielsweise ein Fahrzeug im öffentlichen Raum, deren Domäne eingeschränkt ist (Domain-restricted open-networked control systems). Im Beispiel übermittelt ein Fahrzeug die Anzahl der belegten Sitze über ein öffentliches Netzwerk zu einem Ampelsystem, das eine Prioritäten-Fahrspur steuert: Voll besetzte Autos haben Vorfahrt.

In der 4. Stufe tauschen Kontrollsysteme Daten über Domänen-übergreifende Netzwerke aus (Domin-crossing open-networked control systems). Im Beispiel überträgt ein Fahrzeug die Gesundheitsdaten des Fahrers (vital Parameter) über ein offenes Netzwerk an ein Gesundheitssystem, an das beispielsweise auch ein Krankenwagen angeschlossen ist.

Anwendungsgebiete von CPS

Cyber-Physische Systeme dringen nach und nach in viele Gebiete der Industrie, der Landwirtschaft, des häuslichen Lebens, des Militärs usw. ein. Weiter unten wird auf einige Bereiche näher eingegangen. CPS haben z. B. Schnittstellen mit Sensornetzwerken (siehe Abschnitt 10.7, Seite 471) und dem „Internet der Dinge“ (**Internet of Things IoT**). Unter dem Internet der Dinge versteht man die Vernetzung von physischen und virtuellen Gegenständen durch ein globales, „internetähnliches“ Netz (siehe auch [IoT17]). Das IoT soll die Aktivitäten des Menschen durch Information(en) unterstützen. Genauso wie die Cyber-Physischen Systeme, steht das IoT noch am Anfang der Entwicklung.

CPS im Gesundheitswesen
Gesundheitsdaten, z. B. Blutdruck, Blutwerte, Herzaktivität, von Personen mit labilem Gesundheitszustand, die sich nicht in einer Krankenanstalt befinden, lassen sich mit CPS überwachen. Dafür müssen Sensoren unter die Haut implantiert werden.

CPS im Verkehr
Fahrerassistenzsysteme und Fahrzeuge, die sich autonom (eigenständig, ohne menschlichen Fahrer) im öffentlichen Straßenverkehr bewegen, verwenden CPS, um sich dem Verkehrsgeschehen anzupassen.

CPS in der modernen Fabrik
Cyber-Physische Systeme haben in Fabrikationsanlagen und bei der Automatisierung sehr viele Anwendungsmöglichkeiten. Beispielsweise bei der Steuerung und Überwachung von Robotern, beim Rohmaterial-, Werkzeug-, Halbzeuge-, Bauteile-, Lacke-, Schmierstoff-, und sonstigem Teilenachschub, bei der Montage, beim Testen, beim Abstransport von Fertigteilen usw. Maschinen und Roboter werden miteinander kommunizieren, die Weitergabe von zu bearbeitenden Teilen organisieren, Fehler melden und korrigieren.

Das ist die Herausforderung von Industrie 4.0 [Ind40-21]. Produktionsbetriebe sollen mit modernen Informations- und Kommunikationssystemen ausgerüstet werden. CPS helfen mit, Industrie 4.0 schrittweise zu verwirklichen. Dabei ist die „Daten-, Informations- und Kommunikationssicherheit (Security) der kritischste Erfolgsfaktor für die Realisierung und Einführung von Cyber-Physischen Produktions-Systemen (CPPS)“ [VDI-CPS13]. Durch die nötige massive zusätzliche Vernetzung in den Produktionsanlagen treten vermehrt Security-Gefährdungen auf, die abgewehrt werden müssen.

CPS in der Energiewirtschaft
Cyber-Physische Systeme können eingesetzt werden z. B. in der Energieumwandlung, Energiespeicherung und Energieverteilung über „intelligente Netze“ („Smart Grids“ [SmGr21]). CPS helfen mit bei der Regulierung des Transports von elektrischer Energie.

1.5 Systeme mit Künstlicher Intelligenz (KI-Systeme)

Der Begriff **„Künstliche Intelligenz“** (KI, Artificial Intelligence, AI) wurde 1955 von John McCarthy geprägt. McCarthy (1927 - 2011) war vielfach ausgezeichneter Professor für Computer Science (Informatik) und bekannter Pionier auf dem Feld für künstliche Intelligenz an der Stanford University in Kalifornien (siehe [WikJMC20]).

Was versteht man unter Künstlicher Intelligenz? Im Allgemeinen bezeichnet Künstliche Intelligenz, den Versuch, menschliche Intelligenz nachzubilden, das heißt, einen Computer so zu programmieren, dass dieser eigenständig Probleme bearbeiten kann. Oftmals wird damit aber auch eine effektvoll nachgeahmte, vorgetäuschte Intelligenz bezeichnet, insbesondere bei Computerspielen, die durch meist einfache Algorithmen ein intelligentes Verhalten simulieren soll.

In dem grundlegenden Lehrbuch *Künstliche Intelligenz* von Stuart Russel und Peter Norwig [RuNo12] wird der **Turing-Intelligenz-Test**, entwickelt von Alan Turing (1950), aufgeführt, der folgendes aussagt: „Ein Computer besteht den Test, wenn ein menschlicher Fragesteller, der fünf Minuten lang schriftliche Fragen stellt, nicht erkennen kann, ob die Antworten von einem Menschen stammen oder nicht ... Das Programm besteht den Test, wenn es den Gesprächspartner in 30% der Fälle täuschen kann.“ Man unterscheidet den „eingeschränkten Turing-Intelligenz-Test“, wobei das Gesprächsthema sich auf ein bestimmtes Wissensgebiet (zum Beispiel Medizin) beschränkt und Fragen nur aus diesem Gebiet gestellt werden dürfen und den „uneingeschränkten Turing-Intelligenz-Test“, bei dem das Gesprächsthema unbeschränkt ist.

Es wird zwischen „symbolischer KI“ und „neuronaler KI“ unterschieden. Die **symbolische KI** oder regelbasierte KI basiert auf strukturiertem Wissen (z.B. semantische Netze, Graphenalgorithmen, logisches Schlussfolgern), sie wird oft als „klassische“ KI bezeichnet und geht davon aus, dass menschliches Denken von einer logisch-begrifflichen Ebene aus rekonstruiert werden kann.

Die **neuronale KI** nähert sich aus einer anderen Richtung: Sie versucht, die Lernfähigkeit des menschlichen Gehirns nachzubilden und Muster in Daten zu erkennen. Wir werden uns in Kapitel 4 hauptsächlich mit neuronaler KI beschäftigen. Die neuronale KI oder datenbasierte KI basiert auf statistischen Methoden, z.B. künstliche Neuronale Netze, Data Mining, Entscheidungsbäume usw.

Bei Wikipedia [WikKI20] findet man folgende Erklärung: „Künstliche Intelligenz ist ein Teilgebiet der Informatik, das sich mit der Automatisierung intelligenten Verhaltens und maschinellem Lernen befasst.“ Abgesehen davon, dass der Begriff „Intelligenz“ nicht eindeutig definiert ist, trifft diese Erklärung eher das heutige Verständnis von Künstlicher Intelligenz.

Aus philosphischer Sicht wird die *„starke“* und die *„schwache“* KI-Hypothese unterschieden [RuNo12]. Die starke KI-Hypothese behauptet, dass Maschinen mit Künstlicher Intelligenz wirklich denken können (und nicht nur denken simulieren), die schwache KI-Hypothese hingegen geht davon aus, dass Maschinen mit KI agieren, „als ob sie denken könnten“. Wir wollen dieser Frage nicht weiter auf den Grund gehen und betrachten im Folgenden KI-Ssteme aus der Sicht der Technischen Informatik und der Eingebetteten Systeme.

Was sind KI-Systeme? Eine kurze Charakterisierung wäre zum Beispiel: *„KI-Systeme sind Computersysteme, die spezielle, komplexe Aufgaben ausführen und sich anpassen können, das heißt lernfähig sind, um ihre Aufgabe optimal zu bewältigen.*“ Auf Maschinelles Lernen gehen wir in Kapitel 4 näher ein.

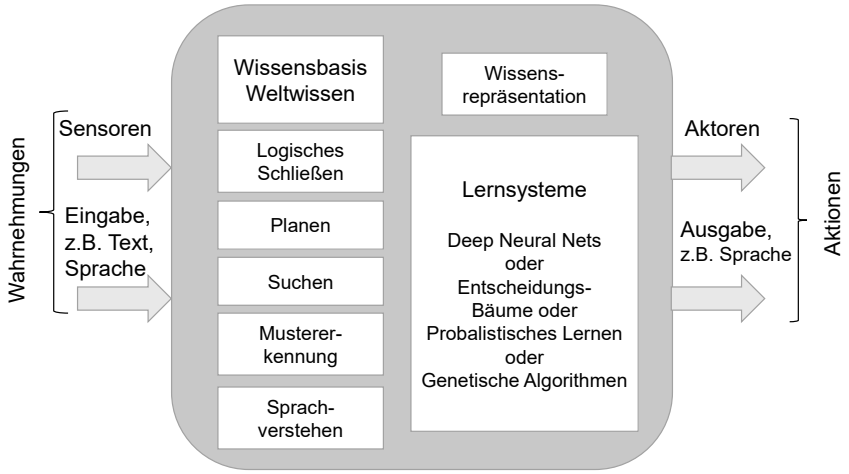

Abbildung 1.8: *Besipiel: Schematischer Aufbau eines intelligenten Agenten (erweitert aus [RuNo12]).*

Der intelligente Agent

Ein „intelligenter Agent“ beinhaltet ein KI-System, das auf Grund von Wahrnehmungen (als Eingabe) rationale (intelligente) Aktionen durchführt, um ein vorgegebenes Ziel zu erreichen. Allen Newell, John Laird und Paul Rosenbloom haben 1987 die Architektur eines „vollständigen Agenten” bei ihren Arbeiten an der „Soar-Architektur“ entwickelt (zitiert in [RuNo12]). Soar steht für „State, Operate, Apply Result“ und ist ein Projekt an der Carnegie Mellon Universität in den USA. Ein vollständiger Agent kann komplexe Aufgaben meistern, ähnlich den menschlichen Fähigkeiten.

Abbildung 1.8 zeigt den schematischen Aufbau eines **intelligen Agenten** der ein KI-System beinhaltet. Ein intelligenter Agent kann zum Beispiel ein Humanoider Roboter sein (siehe unten). Agenten erhalten Informationen über Sensoren und einen zusätzlichen Eingabeapparat, die Eingangsinformationen werden „Wahrnehmungen“ (Perceptions) genannt. Sie agieren über Aktoren und einem Ausgabeapparat mit ihrer Umgebung. Die Ausgaben werden „Aktionen“ (Actions) genannt. Sensoren und der Eingabeapparat können zum Beispiel sein:

– Mikrofon(e),
– Kamera, Videokamera,
– Empfangsantenne(n) für Funk, GPS, etc.,
– Radar-Empfangsantenne, Laser-Empfangssensor.

Aktoren (auch Aktuatoren, engl. effectors) und die Ausgabe können zum Beispiel sein:

– Lautsprecher,
– Bildschirm,
– Sendeantenne für Funk, GPS, Radar, Laser etc.,
– Mechanische Stellglieder zum Beispiel für Greifarme usw.,
– Antrieb für einen Bewegungsapparat.

Ein intelligenter Agent ist zum Beispiel der „Spurhalte-Agent“ oder das „Spurhalte-System“ in einem autonom fahrenden Automobil. Der Spurhalte-Agent muss das Fahrzeug auf einer Straße „in der Spur“ halten und den Abstand zu anderen Fahrzeugen gewährleisten. Ein anderes Beispiel wäre der „Überhol-Agent“, der für ein Überholmanöver des Fahrzeugs zuständig ist. Das Internet ist eine beliebte Umgebung für intelligente Agenten. Der Suffix „-bot“ charakterisiert diese webbasierten KI-Systeme [RuNo12].

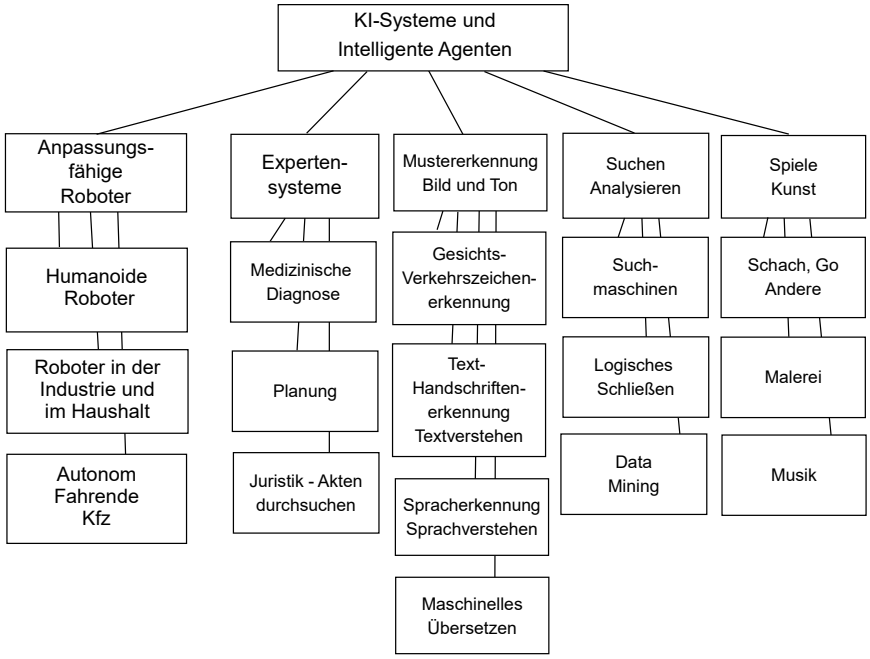

Abbildung 1.9: *Anwendungen von KI-Systemen bzw. Intelligenten Agenten*

1.5.1 Anwendungen von KI-Systemen

Es gibt ein große Menge von Anwendungen für KI-Systeme oder für intelligente Agenten. In Abbildung 1.9 sind einige Beispiele aufgeführt und in „Anpassungsfähige Roboter“, „Expertensysteme“, “Mustererkennungs-Systeme“, „Such- und Analysesysteme“, „Spiele und Kunst“ aufgeteilt.

Anpassungsfähige Roboter

Roboter gibt es schon längere Zeit, hauptsächlich in der Industrie. Dort agieren sie stationär, führen einige, immer wiederkehrende Handgriffe zum Beispiel an Fließbändern der Automobilindustrie aus. Mobile Roboter kommen hinzu, die beispielsweise Nachschub von Teilen an die Fließbänder bringen. Heute werden die Roboter dank KI-Systemen anpassungs- und lernfähig. Beipiele sind der **Humanoide Roboter** und das autonom fahrende Automobil. Humanoide Roboter sind dem Menschen in der Form und der Mechanik, zum Beispiel durch künstliche Gliedmaßen, nachgebaut.

Ein Humanoider Roboter, der auch den uneingeschränkten Turing Test bestehen könnte, müsste folgende Fähigkeiten aufweisen [MaDa19]:

- Eine Wissensbasis an „Weltwissen“. Darunter versteht man:
  - Verständnis für Raum und Zeit.
  - Verständnis der Schwerkraft. (Dinge fallen nach unten).
  - Verständnis von Ursache und Wirkung (Kausalität).
  - Grundlegende Kenntnisse von physikalischen Objekten, ihren Eigenschaften und Interaktionen. (Kugeln rollen (nach unten), Blei ist schwerer als Styropor).
  - Grundlegende Kenntnisse von Menschen und anderen Lebewesen, ihrem Verhalten und Interaktionen.
- Lernfähigkeit (siehe Abschnitt Maschinelles Lernen, Seite 132)
- Text- und Sprachverständnis.
- Logisches Schließen (Schlussfolgern, *engl. Reasoning*)

Diese Fähigkeiten müssen in eine Architektur eingebettet werden, die erweitert werden kann durch zusätzliche Wissensgebiete, die komplex und „unscharf“ sein können. Unter unscharfem Wissen versteht man Wissen, das unsicher ist und das mit der „Wahrscheinlichkeitslogik“ („Fuzzy-Logik“) behandelt werden kann. Mit Hilfe der oben genannten Fähigkeiten muss das intelligente System ein Modell der Welt in Zeit und Raum konstruieren können und sich selbst in diesem Modell verorten.

Humanoide Roboter könnten im Pflegedienst oder im Haushalt eingesetzt werden, vorausgesetzt, sie wären verfügbar und erschwinglich. Dabei denken wir nicht an den Staubsauger-Roboter „Roomba“ der Firma iRobot [IRob20], der bereits millionenfach verkauft wurde, der jedoch nicht wirklich „intelligent“ im Sinne des Turing-Intelligenz-Tests ist (siehe oben). Wir denken vielmehr an Gehilfen, die putzen, kochen und pflegen können. Dazu schreiben Gary Marcus et al. in [MaDa19], Seite 98 (frei übersetzt): *„.. aber die Aussicht, dass ein Robotorgehilfe, der kochen, putzen und die Windeln eines Babys wechseln kann und vor dem Jahr 2025 verfügbar sein wird, ist praktisch Null.“*

Im Jahr 2011 ereignete sich nach einem Tsunami, ausgelöst von einem unerwartet starken Erdbeben vor der japanischen Küste eine Kernreaktor-Katastrophe im japanischen Fukushima. Mehrere Reaktorblöcke wurden durch Kernschmelze und der folgenden Explosionen zerstört. In einer für Menschen gefährlichen, verstrahlten Umgebung sind Roboter mit Künstlicher Intelligenz außerordentlich nützlich. Da diese damals nicht zur Verfügung standen, wurden ferngesteuerte Roboter der Firma iRobot aus Bedford, Massachusets, USA [IRob20] nach dem Unglück eingesetzt um die Lage in den verstrahlten Räumen zu untersuchen und Aufräumarbeiten durchzuführen. Zu den Robotern zählen im weitesten Sinne auch die selbstfahrenden Autos, auch autonom fahrende Automobile genannt. Bei einem sich autonom bewegenden System zeigen Sensoren an, wo es sich befindet und wo es sich hinbewegt. Das System selbst beinhaltet meist mehrere intelligente Agenten. Auf diese Fahrzeuge gehen wir in Kapitel 4 näher ein.

Roboterfahrzeuge agieren in *„Offenen Umgebungen“*. Es gelten eine gewisse Anzahl von Regeln, an die sich eine künstliche Intelligenz halten muss, jedoch ist die Anzahl von Akteuren meist „nach oben offen“ und damit unbegrenzt. Damit sind auch die Ereignisse, die eintreten können und auf die das System reagieren muss, unbegrenzt. Im Gegensatz dazu gehen wir auf „Geschlossene Umgebungen“ auf Seite 16 näher ein.

Zum Beispiel gelten für ein Roboterfahrzeug in der Bundesrepublik Deutschland die 27 Paragraphen der Straßenverkehrsordnung (StVo), die allerdings nicht immer einfach zu interpretieren sind. Zum Beispiel Paragraph 1 der StVo Punkt (2):

„Wer am Verkehr teilnimmt, hat sich so zu verhalten, dass kein anderer geschädigt, gefährdet oder, mehr als nach den Umständen unvermeidbar, behindert oder belästigt wird.“ Dieser Punkt muss in vielen Situationen im Straßenverkehr angemessen interpretiert werden. Das fahren auf Autobahnen ist im Gegensatz zur Fahrt auf Landstraßen oder in der Stadt noch relativ einfach, trotzdem können auch hier unvorhergesehene Ereignisse eintreten. Zum Beispiel fliegt ein Vogel gegen die Windschutzscheibe, verdeckt die Sicht und auch den einen oder anderen Sensor. Diese oder ähnliche Situationen muss ein autonom fahrendes Fahrzeug angemessen meistern können.

Expertensysteme

Expertensysteme sind KI-Systeme, die Fachleuten bei der Analyse bestimmter Aufgaben helfen und bei der Suche von Problemlösungen aus einer Wissensbasis. Beipiele für den Einsatz von Expertensystemen sind:

- Medizinische Diagnose. Es werden die Symptome eines Patienten eingegeben und das System findet die Ursachen und Behandlungsmöglichkeiten.
- In der Juristik: durchsuchen und analysieren von Akteninhalten.
- Bei der Planung. Beispiel: Große Bauvorhaben wie große Flugplätze oder unterirdische Bahnhöfe.

Mustererkennung

KI-Systeme werden häufig zur Erkennung von Bild- und Tonmustern eingesetzt. Bildmuster können beispielsweise sein:

- Gesichtserkennung,
- Handschriftenerkennung,
- Texterkennung,
- Mustererkennung in der medizinischen Diagnose, zum Beispiel bei der Krebserkennung.

Aus der Texterkennung folgt das *Textverständnis*, das ein umfangreiches Wissen voraussetzt, gepaart mit der Fähigkeiten Ableitungen und Schlüsse zu ziehen. Das funktioniert bisher mit künstlicher Intelligenz nur sehr eingeschränkt. Ein weiteres Folgegebiet der Texterkennung ist das *maschinelle Übersetzen* in andere Sprachen.

Bei der Spracherkennung wird das Frequenzgemisch einer Sprachaufnahme analysiert. Bei jedem Menschen ist dieses Frequenzgemisch charakteristisch. Solche Spracherkennungs-Systeme können beispielsweise eingesetzt werden in Automobilen, die nur auf den Befehl „Öffne die Tür“ des Eigentümers (oder Fahrers) das Fahrzeug aufschließen. Auf das Thema Mustererkennung und Spracherkennung gehen wir im Kapitel 4 näher ein.

Suchmaschinen und Data Mining

Schnelle und zuverlässige Suchmaschinen im Internet sind zu wichtigen Werkzeugen nicht nur für Forscher und Entwickler geworden, sondern sind auch im täglichen Gebrauch sehr nützlich. Suchmaschinen finden dem Suchbegriff ähnliche oder unvollständig angegebene Ausdrücke und können so den Sucher korrigieren. Dazu ist nötig, dass die Suchalgorithmen Ähnlichkeiten in verschiedenen Wörtern finden und logisch auf andere Zusammenhänge schließen können.

Mit *Data Mining* bezeichnet man die Auswertung großer Datenmengen mit Hilfe von statistischen und KI-Methoden, um darin bestimmte Trends und Muster zu erkennen. Beipiel: Wie ändert sich das Geräusch einer elektrisch angetriebenen Ölpumpe, wenn eine neue Wartung oder Schmierung nötig ist? Hier werden Geräuschanlysen des Pumpenantriebs durchgeführt. Tritt eine Frequenzverschiebung zu höheren Frequenzen auf (z. B. ein „Quietschen“, das dem menschlichen Ohr zunächst verborgen bleibt), wird eine Wartung fällig.

Spiele und Kunst

Im Jahr 1966 gelang es das erste Mal in der Schach-Geschichte, dass ein KI-System, der Schachcomputer **„Deep Blue“** der Fa. IBM den damaligen Schachweltmeister Garry Kasparov regulär schlagen konnte.

Ein anderes Beispiel für ein erfolgreiches Computerspiel ist **„AlphaGo“** und „AlphaGo Zero“ des Unternehmens Google „DeepMind“ aus den Jahren 2015 bzw. 2017. Go ist ein sehr altes Brettspiel, das im fernen Osten (Japan, China, Korea etc.) sehr populär ist. AlphaGo und AlphaGo Zero bekamen lediglich die Spielregeln des Go-Spiels einprogrammiert und wurden anhand von Spielen gegen sich selbst trainiert (eingelernt). AlphaGo sowie AlphaGo Zero waren danach in der Lage professionelle Go-Spieler zu schlagen, zum Beispiel im Jahr 2016 schlug AlphaGo den südkoreanichen Go-Großmeister Lee Sedol. Auffällig war, dass AlphaGo Spielzüge machte, die erfahrene Go-Spieler angeblich nie ausgeführt hätten, die aber zum Erfolg führten [Alpha17].

Computerspiele agieren in *„geschlossenen Umgebungen“*. Das heißt es gibt eine feste, begrenzte Anzahl (einfacher) Regeln, ein festegelegtes Aktionsziel (Spielziel), eine begrenzte Anzahl von Akteuren, (Spielern und Spielfiguren) und Spielzügen. Damit ist die Anzahl von möglichen Ereignissen in diese Umgebung begrenzt und meist vorhersehbar, es gibt eine vorausberechenbare, optimale Aktion als Antwort auf jede Aktion des Spielgegners und zur Erreichung des Aktionsziels. Im Gegensatz dazu stehen die „Offenen Umgebungen“, siehe Seite 15. Geschlossene Umgebungen sind jede Art von Gesellschaftsspielen, zum Beispiel Poker, Brettspiele wie Schach, Go usw.

Computer mit KI-Systemen können in geschlossenen Umgebungen sehr erfolgreich sein, das zeigt das oben genannte Beispiel des Schachcomputers „Deep Blue“. Es gibt keine unvorhersehbaren Ereignisse, die mit „vernünftiger Intelligenz“ gemeistert werden müssen.

In der Kunst können KI-Systeme zum Beispiel angelernt werden expressionistische Bilder zu erstellen oder in einer bestimmten Stilrichtung zu malen. Beispielsweise wurde einem KI-System beigebracht ein bestimmtes Motiv (etwa eine Mühle) im Stile Vincent Van Goghs (1853-1890) zu malen. Selbst Kunstkenner hatten Schwierigkeiten zu erkennen, dass das entstandene Bild *nicht* von Van Gogh stammte. KI-Systemen kann auch

beigebracht werden, beispielsweie einen Schlager zu komponieren. Allerdings konnte einem KI-System bisher „echte“ (menschliche) Kreativität nicht nachgewiesen werden.

In KI-Systeme sind große Erwartungen gesetzt worden, vor allen Dingen, was die zeitliche Verfügbarkeit betrifft. Es gab viele Vorhersagen, beispielsweise wie diese von dem Nobelpreisträger, Sozialwissenschaftler und KI-Pioner Herbert A. Simon im Jahr 1965: „*Machines will be capable, within twenty years, of doing any work a man can do.*“ [MaDa19]. Diese Vorhersage ist bis heute so nicht eingetroffen. Dennoch haben KI-Systeme etwa nach dem Jahr 2015 große Fortschritte verzeichnet, dank Tiefer Neuronaler Netzwerke (Deep Neural Nets), hohen Rechenleistungen und energiesparender Lernmethoden. Wir gehen im Kapitel 4: „Eingebettete KI-Systeme“ näher darauf ein.

1.6 Verteilte Systeme

Verteilte Systeme (Distributed Systems) sind eine Ansammlung von Eingebetteten Systemen, die in einem übergeordneten System – räumlich voneinander getrennt – Messungen, Steuerungs- und Regelungsaufgaben wahrnehmen. Die einzelnen Eingebetteten Systeme sind durch ein Netzwerk miteinander und mit einem zentralen Prozessor verbunden und können Daten austauschen. Verteilte Systeme findet man z. B. im Automobil, in Flugzeugen, Bahnen, Schiffen, in Produktionsstätten, in Sensornetzwerken usw. Die Vorteile von Verteilten Systemen sind:

- Die einzelnen Eingebetteten Systeme sind näher am „Ort des Geschehens“ und dadurch können Reaktionen durch kürzere Leitungswege schneller erfolgen.
- Es müssen weniger Daten zum zentralen Prozessor transportiert werden.
- Die einzelnen Systeme bilden autonome Einheiten und das bedeutet Entkopplung von anderen Prozessorelementen und vom zentralen Prozessor.
- Die einzelnen Eingebetteten Systeme können separat gefertigt und getestet werden.

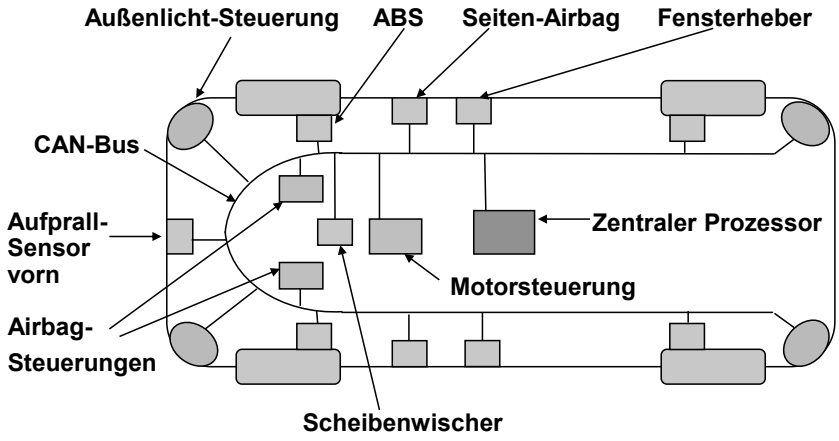

Abbildung 1.10: *Beispiel eines Verteilten Systems in einem Automobil, schematisch dargestellt. Die einzelnen Eingebetteten Systeme sind über den CAN-Bus miteinander und mit dem zentralen Prozessor verbunden.*

Das klassische Beispiel für ein Verteiltes System findet man im Automobil (siehe Abbildung 1.10). In einem heutigen Mittelklasse- und Oberklassewagen sind beispielsweise folgende Eingebetteten Systeme („Electronic Control Units (ECUs)“ genannt), als Steuer- und Regelsysteme installiert:

- Automatisches Bremssystem oder Antiblockiersystem (ABS),
- Airbagsteuerungen,
- Elektronisches Stabilitätsprogramm ESP,
- Elektronische Differentialsperre EDS,
- Lenkhilfe (Servolenkung, meist hydraulisch),
- Klimaanlage (AC: Air Condition),
- ferngesteuerte Wagen-Schließanlage,
- Diebstahlsicherung,
- elektrische Fensterheber,
- Scheibenwischanlage mit Regensensor,
- automatische Scheinwerfer-Winkeleinstellung (abhängig von der Beladung),
- eine „Antischlupfregelung“ (Traction Control) die das Drehmoment auf die Antriebsräder regelt, um ein „Durchdrehen“ der Räder zu vermeiden usw.
- Anti Collision Control (oder Adaptive Cruise Control ACC): Anti-Kollisions-Kontrolle: Warn- und Aktionssystem um Kollisionen zwischen Fahrzeugen zu vermeiden. Diese Systeme sind noch nicht allgemein verfügbar.

Im Automobil werden verschiedene Verbindungsnetzwerke eingesetzt, der „CAN-Bus“ (Controller Area Network, siehe Abschnitt 10.5.1, Seite 455) sowie der „FlexRay“- der „MOST“- und der „LIN“-Bus (siehe Abschnitt 10.5.2, Seite 461). Auf Grund der steigenden Anzahl von Einzelsystemen, und den Anforderungen an Echtzeit-Steuerungen wird mehr und mehr die Kommunikationsstruktur „Automotive Ethernet“ in Kraftfahrzeugen verwendet (siehe Seite 464). Die Ethernet-Technologie wird bereits zum Beispiel für die Diagnose, für die interne Kommunikation, für Fahrerassistenzsysteme (ADAS: Advanced Driver Assistance System), im Infotainment-Bereich, für die Verständigung zwischen Elektrofahrzeugen und Ladestationen usw. zum Einsatz gebracht.

1.7 Bauformen von Eingebetteten Systemen

Eingebettete Systeme werden oft als „unsichtbare Computer“ bezeichnet. In der Tat bauen Eingebettete Systeme zu einem großen Teil auf kostengünstiger Computer-Hardware mit anwendungsspezifischen Schnittstellen auf und führen vordefinierte Aufgaben oft mit anwendungsspezifischer Software aus. Neue Rechenarchitekturen und Technologien sowie die Chipfertigung werden meist mit Zeitverzögerung übernommen und den Anforderungen von Eingebetteten Systemen angepasst. Bauformen von Eingebetteten Systemen sind:

- Ein-Chip-Systeme (System-on-a-Chip, SoC), ASICs und Programmierbare Ein-Chip-Systeme (PSoC),
- Multi-Chip-Module und 3D-ICs,
- Mikrosysteme und
- Platinen-Systeme.

Bei **Ein-Chip-Systemen** (System on a Chip, SoC) ist das gesamte System auf einem Chip integriert. Der Entwicklungsaufwand ist wie auch bei den „Application Specific Integrated Systems“ (ASICs) sehr hoch (siehe Tabelle Abbildung 1.11), dafür ist die Zuverlässigkeit bei diesen Systemen ebenfalls sehr hoch; denn ein Chip ist relativ klein, leicht und in einem Gehäuse eingebaut. Es ist im Vergleich zu Platinensystemen sehr robust und ziemlich unempfindlich gegen Erschütterungen und Stöße. Die Leistungsaufnahme ist niedrig, denn Leitungstreiber und Ausgabe-Puffer entfallen bzw. sind nur an der Peripherie des Chips nötig. Die Entwicklung von Ein-Chip-Systemen lohnt sich nur, wenn die Stückzahlen sehr hoch sind, d. h. abhängig von System und Anwendung etwa in der Größenordnung von einigen zehn- bis hunderttausend Stück und darüber.

Auf programmierbare Ein-Chip-Systeme (PSoCs) und Multiprozessor-Systeme auf einem Chip (MPSoCs) wird auf Seite 29 näher eingegangen.

	Mikrosysteme	ASIC oder SoC	Multi-Chip-Mo-Module (MCM) oder SIP	Platinen-Systeme (Board-Syst.)
Größe	Sehr klein (1)	Sehr klein (2)	Klein (3)	Sehr groß (10)
Leistungs-aufnahme	Sehr niedrig (1)	Sehr niedrig (1)	Sehr niedrig (1-2)	Sehr hoch (10)
Zuverlässigkeit	Sehr hoch (10)	Sehr hoch (10)	Sehr hoch (9)	Gering (1)
Entwicklungs-kosten- und -zeit	Sehr hoch (10)	Sehr hoch (10)	Hoch (8)	Niedrig (3)
Stückkosten Bei hoher Stückzahl	Sehr niedrig (1)	Sehr niedrig (2)	Niedrig (3)	Sehr hoch (10)

Abbildung 1.11: *Vergleich von Bauformen Eingebetteter Systeme. Bewertungsziffern sind von 1 (sehr klein) bis 10 (sehr groß).*

Ein **Multi-Chip-Modul** (MCM, oder Multi Chip Package MCP) besteht aus mehreren separaten Chips (englisch: Dies), die meist auf einem Keramik- oder Kunststoffplättchen (dem Verdrahtungsträger) mit aufgedampfter oder aufgedruckter Verdrahtung planar aufgebracht sind. Die Verdrahtung geschieht durch „bonden“, d. h. hauchdünne Drähte (25 bis 500 μm) aus einer Gold- oder Aluminium-Legierung werden mit den Anschlüssen des Chips und dem Verdrahtungsträger mit speziellen Verfahren verschweißt. Der Verdrahtungsträger wird mit den Chips wie ein Einzelchip in ein Gehäuse eingeschweißt und ist von außen nicht als MCM erkennbar. Die Bezeichnung MCM wird auch auf Module angewendet, die neben Halbleiter-Chips diskrete passive Bauelemente wie z. B. Kondensatoren oder Widerstände in SMD-Bauformen (Surface Mounted Device) beinhalten. Die Einzelchips können speziell für die Integration in einem MCM und auf der Basis unterschiedlicher Technologien entworfen werden, deren Integration auf einem Einzelchip schwierig wäre. Beispiele sind hier Mikrocontroller und ihre analogen Peripheriebausteine und/oder Flash-, oder SRAM-Speicher, Mikroprozessorkerne und Cache-Bausteine oder in Handys die Kombination von Prozessoren mit SRAM-Speichern, Flash-Speichern, Mikrosystemen und anderen Funktionseinheiten.

Werden mehrere Chips übereinander gestapelt, spricht man von einem **System-in-Package (SIP)**, auch „Die-Stacking“ genannt. Damit das komplette Bauteil bei gestapelten Chips nicht zu hoch wird, werden die Chips vorher oft mit einigem Aufwand dünn geschliffen. Die Verdrahtung wird entweder mittels Durchkontaktierung durch die Chips oder durch Dünnschichten an den Seitenkanten mit aufgedruckten Leiterbahnen ausgeführt. Anschließend wird das Chip-Paket mit einer Kunstoffmasse vergossen, die Anschlüsse zu den externen Pins werden gebondet (verschweißt) und in ein Gehäuse verpackt. Die Anzahl der gestapelten Chips ist durch die nötige Wärmeabfuhr begrenzt. Ein MCM oder ein SIP lässt sich meist schneller produzieren als ein SoC. Die 3D-ICs der Firma Xilinx^{TM} [Xi21] sind ähnlich aufgebaut wie SIPs. Mehrere Kintex- und Virtex-Chips (siehe Tabelle 1.17, Seite 28) werden übereinander gestapelt und miteinander verbunden. In der Bewertung der Eigenschaften liegen MCMs und SIPs zwischen den Ein-Chip-Systemen und den Platinen-Systemen (siehe Tabelle Abbildung 1.11, Seite 19).

Mikrosysteme, die im englischen Sprachraum *Micro Electronic Mechanical Systems*, abgekürzt **MEMS** genannt werden, sind Elektromechanische Systeme, die mechanische und elektronische Strukturen im Mikrometerbereich (10^{-6} m) (meist) auf einem Chip kombinieren. Bei einer weiteren Verkleinerung der Mikrosysteme spricht man von *Nanosystemen*. Die Mikrosystemtechnik schöpft aus den Erfahrungen der Chipfertigung, die Schaltkreisstrukturen im Mikrometer- bzw. Nanometerbereich herstellen können [Leo06]. Mikrosysteme werden aus verschiedenen Materialien hergestellt, allen voran wird das aus der Halbleitertechnik bekannte Silizium und Siliziumdioxyd verwendet, aber auch z. B. bestimmte Polymere, Metalle und Keramik kommen zum Einsatz. Metalle wie z. B. Gold, Nickel, Aluminium, Kupfer, Platin, Silber usw. werden auf z. B. Silizium aufgedampft oder durch eine galvanische Methode „elektroplattiert“. Silizium weist gute mechanische Eigenschaften auf. Es ist relativ hart aber auch spröde. Ein längliches, dünnes Siliziumplättchen kann bis zu einem gewissen Grad wie eine mechanische Feder verformt werden und kehrt nach der Auslenkung wieder vollständig in die Ausgangsform zurück („Hookean Material“). Diese Eigenschaft von Silizium wird bei einigen Mikrosensoren ausgenutzt [Leo06] [Elw01]. Eine typische Anwendung eines Mikrosystems ist die Kombination eines Sensorelements mit einem ASIC oder mit einem SIP. Beispiele von Mikrosystemen sind:

- Druckköpfe für Tintenstrahldrucker,
- Beschleunigungssensoren,
- Gyroskope (Kreiselkompass) in der Navigation und zur Lageregelung,
- Drucksensoren,
- Sensoren für Durchflussmessungen von Flüssigkeiten,
- Mikrofone und Lautsprecher,
- Hörgeräte,
- Sehhilfen,
- Medikament-Dosierungvorrichtungen in der Medizin usw.

Die Kombination von elektronischen, mechanischen und optischen Mikrosystemen nennt man *Micro Optical Electrical Mechanical Systems* **MOEMS**. MOEMS können beispielsweise Laserlicht-Signale in Glasfaser-Lichtleitern manipulieren. Anwendungsbeispiele von MOEM-Systemen sind [Leo06]:

- Laser-Scanner,
- Optische Schalter (Optical Switches) z. B. für Glasfaser-Kommunikationssysteme,

– Mikrospiegel-Arrays in dynamischen Mikrospiegel-Anzeigen (Dynamic Micromirror Displays DMD),
– Oberflächenemitter-Laser (Vertical Cavitiy Surface Emitting Laser VCSEL) usw.

Die Bereiche Forschung, Entwicklung und Produktion von Mikro-, Nanosystemen wie MEMS und MOEMS wachsen ständig, da diese Systeme leicht, langlebig und bei großen Stückzahlen sehr kostengünstig sind. Wir gehen auf Mikrosysteme nicht weiter ein.

Platinen-Systeme bestehen aus Leiterplatten, auch „gedruckte Schaltungen“ (Printed Circuits Boards PCB) genannt. Leiterplatten sind die Träger elektronischer Bauteile und bestehen aus faserverstärktem, isolierendem Kunststoff, auf denen die elektronischen Bauteile, meist Module mit eingegossenen Chips, aufgebracht sind. Die Verbindungsleitungen zwischen den Bauteilen und den Steckverbindungen bestehen aus einer dünnen, aufgedruckten Kupferschicht (etwa $35\mu m$). In der Regel gibt es zwei Leiterbahnen-Schichten, eine längs der Plattenebene (oft auf der Oberseite der Leiterplatte), eine quer dazu, (auf der Unterseite der Platte). Bei komplizierten Schaltungen kann es bis zu 12 Leiterebenen geben. Die Modulanschlüsse, es sind in der Regel vergoldete Kupferstifte, werden durch Löcher in der Leiterplatte gesteckt und an der Unteseite der Platte an „Lötaugen“ (Pads) an die Leiterbahnen angelötet. Bei der „Oberflächenmontagetechnik“ (Surface mounted Technology SMT) werden die Bauteile (die Surface mounted Devices SMD) direkt auf die Oberseite der Leiterplatte an die Leiterbahnen aufgelötet. Die SMT ist platzsparend, benötigt keine Bohrlöcher für die Modulanschlüsse und kommt mit dünneren Leiterplatten aus.

Bei Prototypen-Boards kauft der Entwickler die einzelnen Teile des Systems bzw. entwirft fehlende Teile selbst, lässt es auf ein „Board“ oder eine Platine aufbringen und testet das gesamte System. Der Entwicklungsaufwand ist relativ gering, dafür sind die Produktions- und Stückkosten sehr hoch und die Zuverlässigkeit wegen des relativ großen Verdrahtungsnetzes und der Verbindungsstecker gering. Platinen-Systeme können bei höheren Beanspruchungen versagen, z. B. bei Erschütterungen, weil Lötstellen zwischen Modulanschlüssen und gedruckten Leiterbahnen den Kontakt verlieren, oder weil Leiterbahnen oder Steckverbindungen brechen.

1.7.1 Prozessorarten

Die Auswahl des datenverarbeitenden Elements, das heißt des Prozessors eines Eingebetteten Systems hängt ab von den Anwendungsanforderungen, von der Stückzahl, von der Vorgeschichte der Entwicklung, von der geforderten Leistungsaufnahme, vom Zeitplan der Fertigstellung usw.

Abbildung 1.12 zeigt schematisch verschiedene Prozessorarten. Auf der linken Seite von Abbildung 1.12 sind Vielzweck-, Mikro-, Spezial-, Digitale Signal- und anwendungsspezifische Befehlssatz-Prozessoren dargestellt. Für diese Prozessoren muss Software entwickelt werden. Auf der rechten Seite von Abbildung 1.12 ist Hardware als Implementierungsart von Eingebetteten Systemen gezeigt, die programmierbare (bzw. konfigurierbare) Hardware, das FPGA (siehe Seite 23), das PLD (siehe Seite 25), das programmierbare System auf einem Chip (PSoC) und die anwendungsspezifische Hardware (ASIC).

Vielzweck-Prozessoren (General Purpose Processor) sind für Systeme mit hohen Leistungsanforderungen und vielseitiger Programmierbarkeit geeignet. Sie verfügen meist

über mehrere Prozessorkerne, sind relativ teuer, benötigen viel elektrische Energie und werden in Personal Computern (PC) eingesetzt. Sie sind ungeeignet für Eingebettete Systeme, insbesondere für Echtzeitsysteme (siehe Abschnitt 6.1.7, Seite 254). Spezialprozessoren und Mikroprozessoren sind im Vergleich zu Vielzweck-Prozessoren für eine spezielle Anwendung ausgelegt und führen diese Anwendung effizient aus bei relativ geringem Energiebedarf und relativ geringen Kosten.

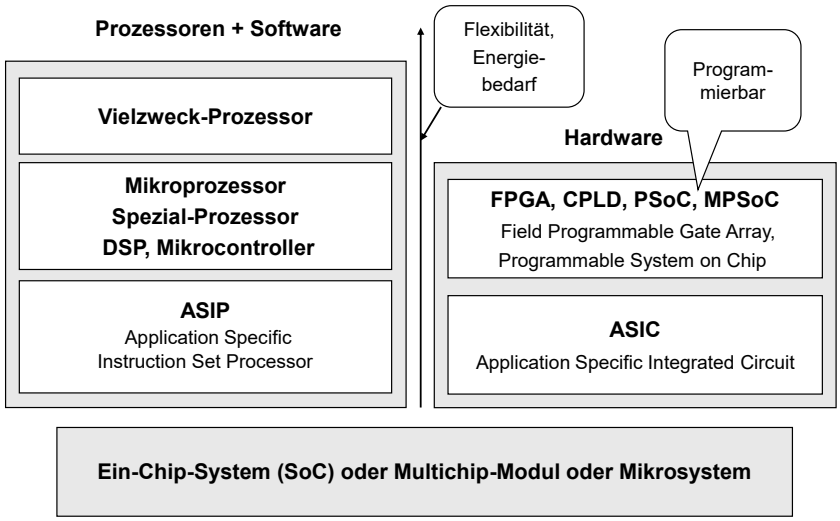

Abbildung 1.12: *Prozessorarten von Eingebetteten Systemen.*

Mikroprozessoren und **Mikrocontroller** werden speziell für Eingebettete Systeme entwickelt. Es gibt sie in großer Vielfalt als „Mikroprozessor/Mikrocontroller-Familien“ auf dem Markt (siehe Abschnitt 9.5, Seite 410). Mikroprozessoren und -kontroller haben meist eine Menge von Zusatz- und Peripheriebausteinen auf dem Chip und können beinahe für jede Anwendung passend ausgewählt werden. Ein **Spezialprozessor** ist ein Prozessor für besondere Aufgaben. Kann mit diesem auch eine ganze Klasse von Anwendungen ausgeführt werden, so spricht man von einem „domänenspezifischen“ Prozessor. Ein Beispiel dafür ist der Graphikprozessor auf einer Graphikkarte, der oft Funktionseinheiten für Pixel-Operationen und mehrere Speicherbänke mit parallelem Zugriff für die Komprimierung/Dekomprimierung von Video/Audio-Daten enthält. **Digitale Signal-Prozessoren** (DSP) werden beispielsweise für die Verarbeitung von Audio- und Video-Datenströmen eingesetzt.

Mikrocontroller sind Prozessoren, denen hauptsächlich spezielle Steuerungsaufgaben zugewiesen werden (siehe Abschnitt 9.3, Seite 407) und die meist nicht für große Rechenleistungen ausgelegt sind. Die Grenze zwischen Mikroprozessoren und Mikrocontrollern ist unscharf. Prozessoren mit anwendungsspezifischem Befehlssatz (Application Specific Instruction Set Processors **ASIP**) verfügen über einen speziellen Befehlssatz mit angepassten Funktionseinheiten und einer eigenen Speicherarchitektur. Ein Beispiel für einen speziellen Befehl ist die Operationsverkettung „Multiply-Accumulate“ (MAC), die bei Matrizen-Rechnungen und bei Digitalen Signal-Prozessoren (DSPs) eingesetzt wird.

Field Programmable Gate Arrays (FPGA) sind Hardware-Komponenten, die vom Benutzer konfigurierbar sind. FPGAs sind praktisch „programmierbare Hardware“, auf sie wird im Abschnitt 1.8.2, Seite 26 näher eingegangen. Unten rechts in der Abbildung 1.12 ist das vollkundenspezifische **ASIC** aufgeführt. Es ist spezielle Hardware für eine spezifische Anwendung, es ist nicht flexibel, weist aber die beste Leistung (Performanz) bei geringster Leistungsaufnahme auf (siehe Abschnitt 1.8).

Ein-Chip-Systeme (SoC) oder Multichip-Module können Mikroprozessoren, Speicherbausteine für die erforderliche Software und Daten, Spezialprozessoren, ASIPs, sowie FPGAs und ASICs enthalten. Mikrosysteme, in denen Mikroprozessoren integriert sind, werden die dazugehörigen Programme normalerweise in einem ROM (Read only Memory) speichern. Mikrosysteme werden in der Regel keine FPGAs enthalten. Für das SoC gilt das Gleiche wie für das ASIC bezüglich Kosten, Entwicklungszeit und Energiebedarf.

1.8 Schaltkreise für Eingebettete Systeme

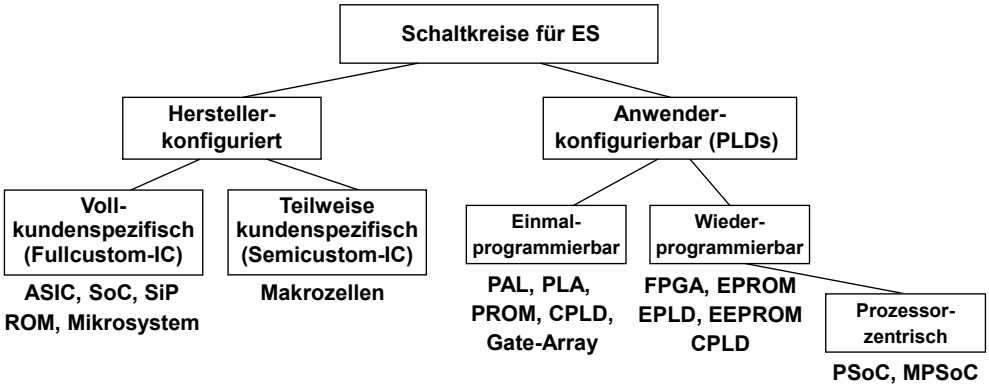

Abbildung 1.13: *Schaltkreise für Eingebettete Systeme. FPGA bedeutet Field programmable Gate Array, PSoC bedeutet Programmable System on Chip.*

Wie Abbildung 1.13 zeigt, können Integrierte Schaltkreise (ICs) für Eingebettete Systeme in „Hersteller-konfiguriert“ und „Anwender-konfigurierbar“ eingeteilt werden.

Bei den herstellerkonfigurierten Schaltkreisen unterscheidet man zwischen vollkundenspezifische (anwendungsspezifischen) und teilweise kundenspenzifischen Schaltkreisen. Die anwenderkonfigurierbaren ICs können als einmalprogrammierbare und wiederprogrammierbare Bauteile kategorisiert werden.

1.8.1 Herstellerkonfigurierte Schaltkreise

Zu herstellerkonfigurierten Integrierten Schaltungen gehören die vollkundenspezifischen (Full custom) und die teilweise kundenspezifischen Schaltkreise (Semi-Custom Integratet Circuits, siehe Abbildung 1.13 oben). vollkundenspezifischen Schaltkreise sind anwendungsspezifische Integrierte Schaltkreis (ASIC, siehe Seite 5), das Read-only-Memory

(ROM), das Ein-Chip-System (SoC), das System-in-Package (SiP, Seite 20) und das Mikrosystem (Seite 20). Wie in den vorhergehen Abschnitten bereits erwähnt, ist die Entwicklung dieser Systeme teuer und aufwändig. Man spricht hier von „Maskenprogrammierten Schaltkreisen“. Beispielsweise werden bei der Fertigung eines ASIC etwa dreißig fotolithograhische Masken verwendet. Ein ganzer Maskensatz ist sehr teuer. ASICs haben aber auch große Vorteile in Bezug auf Zuverlässigkeit, Größe und Stückkosten bei großen Stückzahlen (siehe Tabelle Abbildung 1.19, Seite 31). Das ROM ist ein Programmspeicher, der nur gelesen werden kann und der nicht flüchtig ist, auch wenn die Versorgungsspannung ausgeschaltet ist. Er ist gedacht für Prozessoren, bei denen Teile der Software unverändert bleibt (zum Beispiel beim „Hochfahren“ des Prozessors).

Zu den teilweise kundenspezifischen Schaltkreis-Technologienn (Semi-custom), gehören die „Makrozellen“ (siehe Abbildung 1.14), die auch „Standardzellen“ genannt werden. Darunter versteht man vorgefertigte Maskensätze fester Größe, bei denen die Anordnung der Standard- und Makrozellen und die Verdrahtung durch den Kunden konfiguriert wird. Die Standard- und Makrozellen findet man in vorgegebenen Bibliotheken. Neben einfachen Zellen (z. B. NAND, NOR, Inverter), die zu Blöcken zusammengefasst werden können, werden komplexe Strukturen wie Prozessorkerne, Speicher, Standardzellen-Blöcke, Analog-Digital-Converter (ADCs), DACs usw. kundenspezifisch zusammengestellt und verdrahtet. Die Kombination der Makroszellen geschieht am Bildschirm werkzeugunterstützt und möglichst flächenoptimiert. Die Einsparung basiert auf den teilweise vorgefertigten fotolithograhische Masken. Die Chip-Flächenausnutzung kann bei diesem Verfahren jedoch nicht so optimal durchgeführt werden wie beim ASIC.

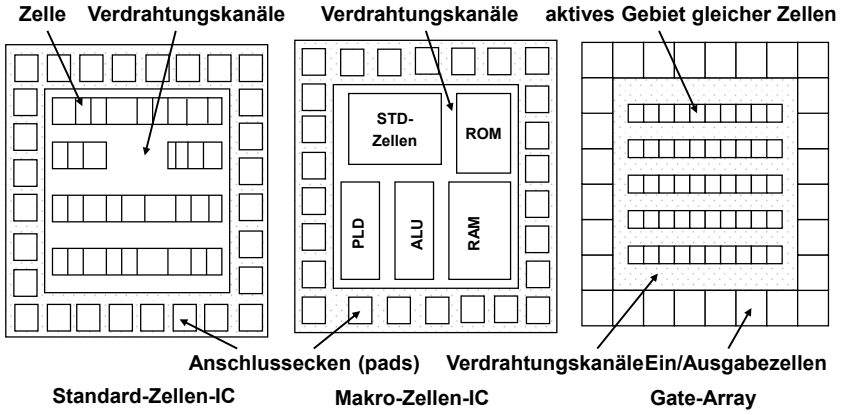

Abbildung 1.14: *Zellen-basierte, teilweise kundenspezifische Technologie. Die linke Seite zeigt schematisch einen Standard-Zellen-Schaltkreis, in der Mitte ist schematisch ein Makro-Zellen-Schaltkreis abgebildet. Rechts: Array-basierter kundenspezifische Technolgie: Gate Array.*

1.8.2 Anwenderkonfigurierbare Schaltkreise

Unter dem Oberbegriff „Programmable Logic Devices“ (PLDs) werden die klassischen programmierbaren Logik-Bausteine oder anwenderkonfigurierbare Schaltkreise, das sind einmalprogrammierbare und wiederprogrammierbare Schaltkreise (siehe Bild 1.13, Sei-

te 23) zusammengefasst. PLDs sind etwa seit dem Jahre 1976 auf dem Markt verfügbar, werden heute jedoch praktisch von den CPLDs und FPGAs verdrängt.

Unter dem Oberbegriff PLD und **SPLD** (Simple PLD) existierten verschiedene Arten von Bausteinen, mit leider nicht immer eindeutigen Bezeichnungen.

Zu den **Einmalprogrammierbaren Schaltkreisen** gehören z. B.
- PLDs mit „Programmable Array Logic“ (PAL)-Struktur, und mit „Programmable Logic Array“ (PLA)-Struktur,
- „Eraseable Programmable Logic Devices“ (EPLDs),
- bestimmte „Complex Programmable Logic Devices“ (CPLDs),
- „Programmierbare Read Only Memories“ (PROMs),
- Gate-Arrays.

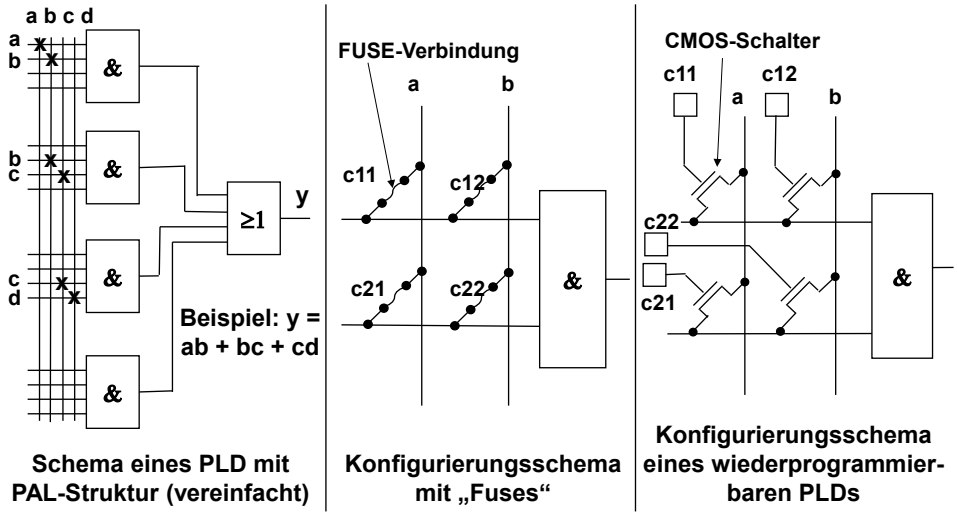

Abbildung 1.15: *Links: Vereinfachte Struktur eines PLD mit PAL-Struktur. Mitte: Konfigurierungsschema mit „Fuses“. Rechts: Konfigurierungsschema eines wiederprogrammierbaren PLDs.*

Das Schema eines **PLD** (Programmable Logic Device) mit Programmable Array Logic (PAL)-Struktur zeigt Abbildung 1.15, linke Seite. Die zu implementierende Logik muss in der konjunktiven Normalform bzw. als „Sum of Products“, beispielsweise in der Form $y = ab + bc + cd$ vorliegen, wobei der Produktterm ab das logische UND und $a + b$ das logische ODER bedeutet. Dies ist realisierbar durch eine Reihe von UND-Gattern, deren Ausgänge in einem ODER-Gatter zusammengefasst werden, (beim PLA sind es mehrere ODER-Gatter, siehe unten). Die Eingangsvariablen, im Beispiel: a, b, c, d, meist auch die invertierten Variablen $\overline{a}$, $\overline{b}$, $\overline{c}$, $\overline{d}$, werden als Reihe paralleler Leitungen senkrecht zu den Eingängen der UND-Gatter geführt und sind durch eine Isolierschicht von den Eingangsleitungen getrennt. Soll eine bestimmte Verbindung einer Eingangsvariablen, (z. B. im Bild die Variable a) mit einem Eingang hergestellt werden, so wird durch eine kurzzeitig angelegte „Programmierspannung“ (relativ hohe Spannung), zwischen der Variablenleitung und der Eingangsleitung, aus einer isolierenden Halbleiterschicht (mit hohem Widerstand) eine leitende Schicht (mit niedrigem Widerstand) erzeugt.

Diese Methode nennt man **Anti-Fuse-Verfahren**. Beim **Fuse-Verfahren**, (Fuse bedeutet Schmelzsicherung), sind zunächst alle Verbindungen in der Eingangs-Matrix fest vorhanden (siehe Abbildung 1.15, Mitte). Die nicht erwünschten Verbindungen werden durch einen Stromstoß aufgeschmolzen und dadurch getrennt. Sowohl das Fuse- als auch das Anti-Fuse-Verfahren erzeugen sogenannte **„strahlungsharte“** (Radiation Hard) Verbindungen, die in strahlungsbelasteter Umgebung, z. B. in der Raumfahrt, in Kernkraftwerken, Kernforschungsanlagen (z. B. CERN, DESY), usw. eingesetzt werden. In Abbildung 1.15, rechte Seite ist das Schema eines wiederprogrammierbaren PLDs dargestellt. An den Kreuzungspunkten der Eingangsleitungen mit den Variablen-Leitungen (a und b) liegen CMOS-Schalter, die über die Konfigurierungs-Anschlüsse c11 bis c22 durch einen positiven Spannungspegel aktiviert werden können.

Beim Programmable Logic Array **PLA** sind im Gegensatz zum PAL mehrere programmierbare ODER-Gatter am Ausgang verfügbar. Sie enthalten oft auch programmierbare Speicherelemente, zum Beispiel D-FFs und Tri-State-Ausgänge (siehe Abschnitt 10.2.1, Seite 425), sodass nicht nur Grundschaltungen mit kombinatorischer Logik (Schaltnetze), sondern auch Schaltwerke einfach konfiguriert werden können.

Die Firma Lattice hat eine Modifikation der PALs unter der Bezeichnung „Generic Array Logic“ (GAL) auf den Markt gebracht. GALs bestehen aus einem programmierbaren Array aus UND-Bausteinen und fest verdrahteten ODER-Bausteinen. GALs sind meist wiederprogrammierbar, unter der Bezeichnung EEPLD (Electrically Eraseable PLD) sind sie elektrisch löschbar. EPLDs (Eraseable PLD) können durch UV-Licht gelöscht werden.

Bei den **CPLDs**, die die PLDs inzwischen im Wesentlichen ersetzt haben, werden mehrere programmierbare PLA-Blöcke über ein „Koppelfeld“ miteinander verbunden. Das programmierbare Koppelfeld enthält „Rückkopplungen“ von den Ausgängen zu den Eingängen. An den Ein- und Ausgängen liegen meist schnelle Speicher, wie Latches, Flipflops oder Register. Eingesetzt werden CPLDs z. B. für Analog/Digital Konverter (ADC, DAC), Digitale Signal Umsetzer (DSP), Read only Speicher (ROM) usw.

Beim **„Gate-Array“** (siehe Abbildung 1.14, Seite 24), liegt die Grundstruktur und damit die Chipgröße sowie die Anzahl der Ein/Ausgabeanschlüsse fest. Das Verbindungsnetz zwischen den Zellen wird kundenspezifisch ausgeführt. Das heißt, in vielen Fällen wird der Baustein nicht optimal genutzt werden können. Gate-Arrays werden eingesetzt für kleinere Serien und bei spezielle Anwendungen, z. B. wenn „strahlungsharte“ Schaltkreise benötigt werden (siehe oben). Sie werden mehr und mehr von den FPGA's (Field programmable Gate Arrays) abgelöst (siehe unten).

Wiederprogrammierbare Schaltkreise

Die Bedeutung der **wiederprogrammierbaren** (auch rekonfigurierbaren) oder **Feldprogrammierbaren** Schaltkreise (Field Programmable Gate Arrays, **FPGAs**) hat im letzten Jahrzehnt sehr stark zugenommen. Zu ihnen gehören auch die rekonfigurierbaren CPLDs (Complex Programmable Logic Devices), sowie die **programmierbaren Systeme auf einem Chip** (Programmable System on a Chip, **PSoC**) und die **Multiprozessor Systeme auf eine Chip** (MPSoC).

FPGAs, anfangs **LCA**s (Logic Cell Arrays) genannt, werden auch als programmierbare bzw. **rekonfigurierbare Hardware** bezeichnet. Sie wurden 1984 von der Firma **Xilinx**

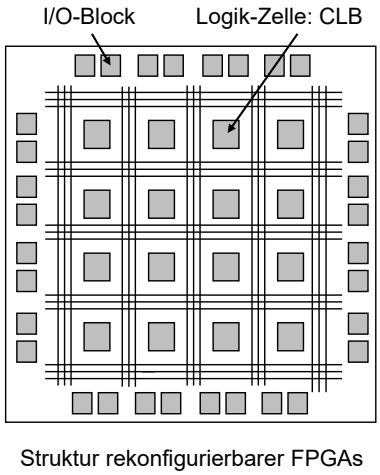

Abbildung 1.16: *Vereinfachte Struktur eines rekonfigurierbaren FPGAs.*

[Xi21] eingeführt. Die Firma Xilinx ist Marktführer auf diesem Gebiet und produziert eine große Vielzahl von FPGAs, programmierbaren Ein-Chip-Systemen (PSoCs) und programmierbaren Multiprocessor-Chip-Systemen (MPSoCs).
Anwendungen für FPGAs sind z. B. :

- Prototypen, d. h. für die Emulation beliebiger Hardware, beispielsweise für Automobil-Fahrerassistenzsysteme, die noch in der Erprobung sind (Automotive Driver Assistance System ADAS).
- Für Eingebettete Systeme, bei denen Flexibilität gefordert ist und Änderungen im Betrieb möglich sind.
- Für Rechenbeschleuniger (siehe Abschnitt 1.3, Seite 4).
- Für kleinere Stückzahlen, ein vollkundenspezifisches ASIC lohnt sich nicht.
- Für Steuergeräte in der Raumfahrt und beim Militär.

Dasselbe FPGA kann innerhalb einer Anwendung mehrfach verwendet werden. Dafür muss es während der Laufzeit **dynamisch rekonfiguriert** werden, d. h. die Konfiguration muss bei Bedarf neu in das FPGA geladen werden. Dies kann, je nach Ausführung, zwischen Sekunden und Bruchteilen von Millisekunden geschehen. Obwohl die Taktfrequenzen in der Regel kleiner sind als bei vollkundenspezifischen ASICs, werden FPGAs oft als Hardware-Beschleuniger eingesetzt, da hier die Parallelität der Ausführung – eine wesentliche Grundeigenschaft der Hardware – die geringere Taktfrequenz ausgleichen kann. Der Unterschied der Taktfrequenzen zwischen ASICs und FPGAs schrumpft allerdings im Laufe der Zeit. Vor einem Jahrzehnt war die Taktfrequenz bei den meisten FPGAs etwa zehn mal niedriger als bei ASICs. Heute nähern sich die Taktfrequenzen der Hochleistungs-FPGAs von Firmen wie Xilinx, Altera, usw. und Prozessoren die im gleichen Jahr gefertigt werden, mehr und mehr an.

Abbildung 1.16 zeigt die vereinfachte Struktur eines rekonfigurierbaren FPGAs. Sie ist ein Feld aus Logik-Blöcken, auch Logik-Zellen oder „Configurable Logic Blocks (CLBs)“ genannt, zwischen denen Verbindungsleitungen angebracht sind. An der Peripherie des

Bausteins liegen die Eingabe/Ausgabe (I/O)-Blöcke. Auf den heutigen FPGAs befinden sich außer den Logik-Zellen beispielsweise auch RAM-Speicherblöcke (sogenannte Block-RAM), DSP-"Slices", Transceiver usw. (siehe Tabelle 1.17, Seite 28). Ein wichtiger Begriff ist die „Granularität" der Logik-Blöcke. Sie kann folgende Blockgrößen annehmen: Gatter, Tabellen mit Speichern (Look Up Tables LUTs mit FlipFlops, siehe unten) und Funktionsblöcke (**CLB** Configurable Logic Blocks). CLBs sind zum Beispiel Arithmetisch-Logische Komponenten (ALUs), Kommunikationseinheiten und/oder Prozessorkerne. Während Gatter für ein FPGA eine geringe Granularität darstellen, sind LUTs, und CLBs „grob-granular".

Die **Look Up Tables (LUTs)** haben jeweils 4 bis 6 Eingänge und Speicherbausteine (Flipflops) an den Ausgängen. Sie sind durch konfigurierbare Selektoren (Multiplexer) miteinander verbunden. Look Up Tables sind im Wesentlichen Wahrheitstabellen, die je nach Konfigurierung verschiedene logische Funktionen in der disjunktiven Normalform (DNF) oder der konjunktiven Normalform (KNF) realisieren können. Die CLBs werden heute durch die Bezeichnung **„Slices"** ersetzt, wobei flächenmäßig etwa vier Slices einen CLB abdecken. Die Konfigurierung kann bei einigen FPGAs über die „JTAG-Schnittstelle" (siehe Abschnitt 8.5.4, Seite 378) durchgeführt werden.

Auswahl von Xilinx-FPGAs	Spartan-7	Artix-7	Kintex UltraScale+	Virtex UltraScale+
Max. Logic-Zellen	102 000	215 000	1 843 000	8 938 000
Max. Speicher (Mb)	4,2	13	142	455
Max. DSP-Slices	160	740	3 528	12 288
Max. Transceiver-speed (Gb/s)	--	6,6	32,75	58
Max. I/O-Pins	400	500	572	1 976
Strukturbreite (nm)	45	28	20	16

Abbildung 1.17: *Vergleich einiger handelsüblicher Xilinx-FPGAs aus der Spartan-Reihe, der Artix-, der Kintex- und der Virtex-Reihe. Die Zahlen geben jeweils die Eigenschaften des Typs mit den Höchstwerten der genannten Familie wieder (Quelle: [Xi21], Stand: Juli 2021).*

Tabelle Abbildung 1.17 zeigt einige Eigenschaften der wichtigsten Xilinx-FPGA-Familien. Die Zahlen in der Tabelle geben jeweils die Charakteristika des Typs mit den Höchstwerten der genannten Familie wieder. Am unteren Ende der Skala steht die Spartan-Familie als „Niedrigpreis-Familie" (Spartan-6 und Spartan-7) mit relativ niedrigem Energiebedarf, gedacht für den Massenmarkt, z. B. für die Anwendung im Automobilbau bei einfachen Funktionen, für drahtlose Kommunikations-Geräte (Mobiltelefone), für Flachbildschirme usw. Die Virtex-Linie ist die Hochleistungs-Familie für hohe Anforderungen an Performanz und Datenübertragungsgeschwindigkeit. Zwischen der Spartan- und der Virtex-Linie liegen die Artix- und die Kintex-Familien. Sie finden Anwendungen beispielsweise in Eingebetteten Systemen von Fahrer-Assistenzsystemen in Autombilen, in der Raumfahrt, der Industrie, der Medizin und der Wissenschaft (IMS) sowie in militärischen Geräten.

In der Tabelle Abbildung 1.17 sind als Eigenschaften die maximale Anzahl Logik-Zellen der Xilinx-Module, die maximale Speichergröße, die maximale Anzahl Digitale Signal-Prozessor (DSP)-Slices, die Tranceiver-Speed, d. h. die Übertragungsgeschwindigkeit der Daten-Übertrager-Einheiten in Gigabit pro Sekunde (Gb/s), die maximalen Ein/Ausgabe-Pins und die Strukturbreite aufgeführt. Bemerkenswert ist, dass die Strukturbreiten bei den aufgeführten Xilinx-FPGA-Familien im Wesentlichen mindestens seit dem Jahr 2014 gleich geblieben sind. Die **Logik-Zellen** (bzw. Logik-Blöcke, CLBs, siehe oben) sind die kleinsten Logik-Einheiten der FPGA-Module, die durch die Konfigurierung ihre spezifische Funktion erhalten und miteinander verbunden werden (nicht zu verwechseln mit den „Standard-Zellen“ aus Abbildung 1.14, Seite 24). Jede Logik-Zelle enthält einige Tausend Transistoren. Dadurch wird eine massive Parallelverarbeitung möglich. Die **DSP-Slices** sind die kleinsten Einheiten für den Aufbau eines Digitalen Signal-Prozessors. Ein DSP-Slice enthält je einen Multiplizierer, Addierer und Akkumulator (Speicherbaustein mit einem oder mehreren Registern), um schnelle Multiply-Accumulate (MAC)-Funktionen durchzuführen.

Ein ausgefeiltes Sicherheits-System, das die Verschlüsselung nach dem Advanced Encryption Standards AES-256 einschließt, schützt die Konfigurations-Datei, die sozusagen die gesamte Schaltungsbeschreibung enthält. Die Typen Virtex 4QV und 5QV entsprechen der Spezifikation die für Systeme der Raumfahrt gelten (Space Grade). Das heißt, sie funktionieren zum Beispiel in einem erweiterten Temperaturbereich, in strahlungsbelasteter Umgebung usw. Die SoC-Typen Spartan, Artix, Kintex, Virtex XQ, z. B. Virtex Zynq 7000Q, sind für militärische Geräte zugelassen (Defense Grade). Auch hier gilt ein erweiterter Temperaturbereich, sicheres Funktionieren bei Erschütterungen usw.

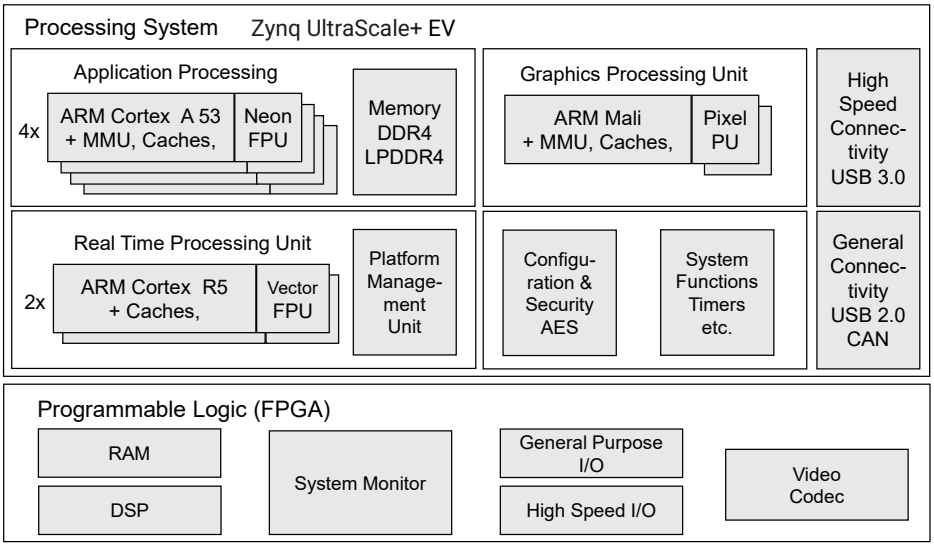

Abbildung 1.18: *Vereinfachtes Blockschaltbild eines Xilinx-Mikroprozessor-Programmierbaren System-on-a-Chip (MPSoC) aus der „Zynq-Familie“: Zynq UltraScale+ EV. EV steht für Enbedded Vision (vereinfacht aus [XiSoC21]).*

Programmierbare Ein-Chip-Systeme (PSoCs und MPSoCs)

Programmierbare Ein-Chip-Systeme (PSoCs) und Multiprocessor Systems on a Chip (MPSoCs) der Firma Xilinx [XiSoC21] sind sogenannte Prozessor-zentrische Entwicklungsplattformen (siehe Abbildungen 1.13 und 1.18), die auf einem Chip-System Software-, Hardware- und Eingabe/Ausgabe-Modellierung ermöglichen. Den Entwickler erhält im MPSoC durch die Kombination von mehreren Prozessorkernen und konfigurierbarer Hardware eine sehr gute Rechenleistung und ein Optimum an Flexibilität. Bei der Firma Xilinx kann der Entwickler z. B. zwischen verschiedenen Zynq^{TM}-MPSoCs wählen [XiSoC21]:

- „Cost Optimized"-Familie: „Dual-core Entry Point to Heterogeneous Processing": Zynq UltraScale+ CG mit Dual ARM Cortex^{TM}-A53, Dual ARM Cortex-R5F, 16nm FinFET+ Programmable Logic (4 Prozessorkerne).
- Zynq UltraScale+ EG: „Broadest Device Range for Next-Generation Applications" mit Quad ARM Cortex-A53, Dual ARM Cortex-R5F, GPU (Graphic Processor Unit): ARM MAli^{TM}-400MP2, 16nm FinFET+ Programmable Logic (7 Prozessorkerne).
- Abbildung 1.18 zeigt das vereinfachte Blockschaltbild des Zynq UltraScale+ EV: „Video Codec Enabled for Multimedia and Embedded Vision" mit Quad ARM Cortex-A53, Dual ARM Cortex-R5F, GPU ARM Mali-400MP2 16nm FinFET+ Programmable Logic (7 Prozessorkerne).

Die Anwendungsbereiche für Programmierbare Ein-Chip-Systeme und MPSoCs sind vielfältig, zum Beispiel [Xi21]:

- Automotiver Bereich: Für Prototypen von Fahrerassistenz-Systemen,
- Militärische Geräte,
- Digitales Radio, Digitales Fernsehen: HDTV (High Definition Television),
- Videokonferenz-Systeme,
- Monitore und Projektionsgeräte,
- Professionelle Kameras,
- Encoder und Decoder für Digitale Unterschriften,
- Medizinische Anwendungen z. B. Endoscopie,
- Drahtgebundene und drahtlose Kommunikation der 5. Generation, usw.

Programmierbare Systeme auf einem Chip sind effektiv einsetzbar für Prototypen von Eingebetteten Systemen, die noch in der Entwicklung sind oder für kleine Stückzahlen, zum Beispiel bei medizinischen oder militärischen Geräten. Für die Entwicklung und Analyse dieser Systeme stellt Xilinx Entwicklungshilfen zu Verfügung z. B. Entwicklungs-Platinen (Development Boards) [Xi21].

Vergleich der Schaltkreis-Technologien für Eingebettete Systeme

Tabelle Abbildung 1.19 zeigt einen ungefähren qualitativen Vergleich der Technologien. Bei einem ASIC, einem Ein-Chip-System (SoC) oder einem Mikrosystem ist jeweils die Transistordichte sehr hoch, ebenso die Taktrate und damit die Ausführungsgeschwindigkeit. Ein Mikrosystem ist etwa gleich zu bewerten wie ein ASIC. Die Entwicklungs- und die Einführung in die Produktion wird bei diesen Systemen relativ lang sein, es fallen dadurch hohe Entwicklungskosten und hohe Prototypen-Kosten an. Hohe Kosten für die Einführung in die Produktion bedeutet, dass für eine geringe Anzahl von Versuchsmustern die Kosten ebenfalls sehr hoch ausfallen, dafür sind aber die Stückkosten bei hoher Stückzahl optimal niedrig. Das heißt: Ein ASIC, bzw. ein SoC oder ein Mikrosystem eignet sich am besten für ein ausgereiftes Produkt, das mit sehr hohen Stückzahlen in Serie

	ASIC, SoC Mikrosysteme	Makro-zellen	Standard-Zellen	Gate-Array	FPGA, PSoC, MPSoC
Anzahl Transistoren	Sehr hoch bis 10^9	Sehr hoch bis 10^9	Hoch bis 10^7	Hoch bis 10^7	Sehr hoch (10^7-10^9)
Chip-Fläche	Sehr klein (1)	Klein (3)	Mittel (5)	Mittel (5)	Sehr groß (10)
Taktrate	Sehr hoch (10)	Sehr hoch (9)	Hoch (8)	Hoch (8)	Mittel bis Hoch (5-9)
Entwicklungszeit	Sehr lang (10)	Lang (8)	Mittel (5)	Mittel (5)	Kurz (1-3)
Einführung in die Produktion	Sehr lang (10)	Lang (8)	Lang (8)	Relatv kurz (2-3)	Sofort verfügbar (1-2)
Prototypen-Kosten	Sehr hoch (10)	Sehr hoch (8-9)	Hoch (8)	Mittel (6)	Sehr niedrig (1)
Stückkosten bei hoher Stückzahl (> 10^6)	Sehr niedrig (1)	Sehr niedrig (2)	Niedrig (3)	Mittel (6)	Sehr hoch (10)

Abbildung 1.19: *Vergleich der Schaltkreis-Technologien für Eingebettete Systeme. Die Bewertungsziffern sind sehr grob und reichen von (1): sehr klein bis (10): sehr groß. Das ASIC, das SoC und das Mikrosystem gehören zu den vollkundenspezifischen Systemen.*

geht. Die Fertigung von Standardzellen-Schaltungen nimmt normalerweise weniger Zeit in Anspruch als bei einem ASIC, die Kosten für wenige Prototypen werden auch hier sehr hoch sein, jedoch werden die Stückkosten bei sehr hoher Stückzahl höher sein als bei ASICs, da die Chip-Fläche bei Zellen-basierten Schaltungen größer ausfallen wird.

Einige weitere FPGA-produzierende Firmen und deren Produkte

Ohne für ein Unternehmen werben zu wollen, führen wir im Folgenden einige bekannte Firmen auf, die FPGAs und CPLDs produzieren, um die Vielfalt der FPGA- und CPLD-Produktion aufzuzeigen. Dem Entwickler, der FPGAs oder CPLDs einsetzen möchte, sei empfohlen, mehrere Produzenten bzw. Lieferanten zu analysieren, um das passende Produkt zu finden.

- Die Firma Intel übernahm Ende 2015 die Firma Altera (www.altera.com). Die Produkte sind: CPLDs, FPGA-Bausteine und SoCs mit Prozessorkernen beispielsweise für Rechenbeschleuniger in Multiprozessor-Systemen.
- Die Firma Microchip (www.microchip.com) übernahm 2016 die Firma Atmel. Microchip bietet eine breite Palette von digitalen und analogen Bausteinen an. Beispiele sind 8-, 16- und 32-Bit-Mikrocontroller in allen Preisklassen, Touch-Screen-Steuerungen, analoge Verstärker, alle Arten von Reglern (Spannungs-, Akku-Laderegler, usw.), strahlungsharte FPGAs, High Speed Networking-, Video Bausteine usw.
- Die Firma Microsemi (www.microsemi.com) übernahm im Jahre 2010 die Firma Actel. Microsemi hat mit der Firma Microchip fusioniert. Microsemi hat ihre Produktpalette mit Actel-FPGAs erweitert und liefert beispielsweise strahlungsharte FPGAs auf der Basis der „Antifuse"-Technologie (siehe Seite 26), „Low-Power"- und „Mixed-Signal"-FPGAs, SoCs mit integriertem Prozessorkern ARM Cortex-M3, usw.
- Die Firma Lattice Semiconductor (www.latticesemi.com) liefert eine Vielzahl von Low-power-FPGAs und CPLDs z. B. für die mobile Kommunikation, Video- und

Fernseh-Übertragungs-Steuerungen, Steuerungen für Virtual-Reality-Brillen, integrierte Schaltungen für effektives Energie-Management (Power Management) usw.

Alle oben genannten Firmen bieten zudem Werkzeuge für die Entwicklungsunterstützung, zum Beispiel Debug-Hilfen und Entwicklungsplatinen an.

1.9 Zusammenfassung

Eingebettete Systeme sind bereits integraler Bestandteil unserer Umgebung, sei es in unserem Büro, im Wohnhaus, im Kraftfahrzeug, in medizinischen Geräten, in Industrieanlagen usw. Ein typisches Eingebettetes System enthält ein oder mehrere Prozessorelemente, Verbindungselemente, Schnittstellen zu Sensoren und Aktoren sowie zu Eingabe- und Ausgabevorrichtungen. Sensoren nehmen bestimmte physikalische Größen der Umgebung auf zum Beispiel Temperatur, Wege, Schaltereinstellungen und berechnen mit Hilfe eines Prozessors oder mehrerer Prozessorelemente den Zeitverlauf eines Aktorsignals. Das Aktorsignal muss oft in „Echtzeit“ erfolgen, das heißt es muss als Reaktion zum Sensorsignal bestimmte Zeitgrenzen einhalten.

Die Bauformen von Eingebetteten Systemen sind Ein-Chip-Systeme (System-on-a-Chip, SoC), Multi-Chip-Module, Mikrosysteme und Platinen-Systeme. Die Wahl der Bauform hängt stark von der Stückzahl der zu fertigenden Systeme ab. Die Wahl der Prozessorart, die für das zu entwickelnde Eingebettete System verwendet wird, ist eine wichtige Entscheidung des Entwicklers. Sie hängt von vielen Faktoren ab, z. B. von den Anforderungen, der erwarteten Stückzahl, des geforderten Termins der Markteinführung, von den Funktionen usw. Die Prozessorfunktion des Eingebetteten Systems kann demnach mit Hilfe von Mikroprozessoren, Mikrocontrollern, Spezial-Prozessoren, FPGAs, voll kundenspezifischen (ASICs), programmierbaren Systemen auf einem Chip, teilweise kundenspezifischen Schaltkreisen oder Ein-Chip-Systemen implementiert werden.

Der Trend der Technologieentwicklung von Eingebetteten Systemen geht bei großen Stückzahlen in Richtung voll kundenspezifischer Integrierter Schaltkreise bzw. Ein-Chip-Systeme mit mehreren Prozessoren und geringem Energiebedarf. Bei kleineren Stückzahlen werden häufig rekonfigurierbare FPGAs oder programmierbare Systeme auf einem Chip (PSoCs und MPSoCs) eingesetzt. Der Trend geht in Richtung programmierbarer Systeme. Damit steigt der Bedarf an Software.

2 Entwicklungsmethodik

Ausschlaggebend für die Entwicklung bzw. den Entwurf eines neuen Produkts ist ein Bedarf oder eine Marktchance, d. h. die Aussicht für den Verkaufserfolg eines Produkts. Aus dem Bedarf werden die Anforderungen an das Produkt formuliert. Entwicklungsmethodik bedeutet *die Vorgehensweise, aus den Anforderungen schrittweise ein qualitativ hochwertiges Produkt zu entwickeln mit dem Ziel, dieses Produkt zu produzieren und zu verkaufen.*

Der Anstoß für eine Produktentwicklung gibt meist ein Auftraggeber, der ein Unternehmer sein kann oder der Auftraggeber ist die Marketing- oder Planungs-Abteilung eines Unternehmens. Der Auftraggeber sucht sich einen Auftragnehmer der in der Lage ist, die Entwicklung durchzuführen, es kann ein Entwicklungshaus sein, oder die Entwicklungsabteilung im selben Unternehmen. Wir haben es in der Regel mit zwei Parteien zu tun: Einem Auftraggeber (einem Kunden), der seine Vorstellungen von dem neuen System in einem Dokument festhalten wird, einem **Lastenheft** (siehe Abschnitt 2.1.2, Seite 37) oder bereits in einer *Spezifikation* (siehe Abschnitt 2.1.4, Seite 40). Der Auftragnehmer formuliert aus den Anforderungen des Lastenhefts ein **Pflichtenheft** (siehe Abschnitt 2.1.3, Seite 39) als Angebot.

Das zu entwickelnde Produkt kann ein umfassendes System sein, beispielsweise ein Flugzeug oder ein Geldausgabeautomat, dessen grundlegender Aufbau schematisch in Abbildung 2.1, Seite 43 gezeigt wird. Das Produkt kann auch ein Teilsystem sein, ein Eingebettetes System, beispielsweise die Steuerung eines Geldausgabeautomaten oder die Steuerung des Kartenlesers eines Geldausgabeautomaten. Ein Eingebettetes System besteht in der Regel aus zwei Entwicklungsbereichen, der Softwareentwicklung (siehe Abschnitt 2.4, Seite 44) und der Hardwareentwicklung (siehe Abschnitt 2.5, Seite 53), die idealerweise miteinander koordiniert ablaufen sollten.

Die folgenden Abschnitte behandeln die heute relevanten Grundlagen für den Entwurf und die Modellierung Eingebetteter Systeme. Zunächst wird auf die Kundenanforderungen, auf das Lastenheft, eingegangen, danach auf das Pflichtenheft und die Spezifikation. Die Softwareentwicklung und die Hardwareentwicklung werden in eigenen Abschnitten behandelt. Der Hauptteil des Abschnitts Hardwareentwicklung beschäftigt sich mit den aktuellen Hardware-Entwurfsmethodem „Plattformbasierten Entwurf“ und dem „Modellbasierten Entwurf“.

https://doi.org/10.1515/9783110702064-002

2.1 Kundenanforderungen und Spezifikation

2.1.1 Nichtfunktionale Anforderungen

Eingebettete Systeme sind in den übergeordneten Systemen oft nur schwer zugänglich, bestimmen aber meist entscheidend das Verhalten dieses Systems. Unterschieden wird grundsätzlich zwischen „funktionalen“ und „nichtfunktionalen“ Anforderungen an das System bzw. an das Eingebettete System. Während die funktionalen Anforderungen vom Auftraggeber des Eingebetteten Systems in einem Anforderungskatalog (Requirements Document RD) oder im „Lastenheft“ (siehe Abschnitt 2.1.2) formuliert werden, muss der Entwickler auf die nichtfunktionalen Anforderungen selbst achten. In guten Entwicklungsbüros werden die nichtfunktionalen Anforderungen als Richtlinien ausgegeben und selbstverständlich in einer Entwicklung eingeplant. Im Folgenden wird eine Unterscheidung zwischen allgemein gültigen und besonderen nichtfunktionalen Anforderungen vorgenommen. Möglicherweise sind die Listen nicht vollständig. Die folgenden **nichtfunktionalen Anforderungen** gelten allgemein für Eingebettete Systeme:

- Zuverlässigkeit, Verfügbarkeit,
- energiesparend,
- Effizienz,
- Größe und Gewicht sollen angemessen sein,
- Physikalische und elektrische Robustheit,
- Heterogenität, Verarbeitung unterschiedlicher Datenarten,
- Entwicklungszeit *(Time-to-Market)* soll möglichst kurz sein,
- Hardwareaufwand und Teilekosten sollen möglichst gering sein,

Die **Zuverlässigkeit** und Verfügbarkeit stehen als sehr wichtige Eigenschaften obenan. Insbesondere in Kraftfahrzeugen und Flugzeugen hängt von der Zuverlässigkeit von Eingebetteten Systemen oft die Sicherheit von Menschenleben ab. Beispielsweise darf in einem Flugzeug der Autopilot oder in einem Kraftfahrzeug das Bremssystem nicht versagen. **Verfügbarkeit** ist eng mit der Zuverlässigkeit verknüpft und bedeutet, dass das Eingebettete System in der Ausführung seiner Funktion nicht aussetzen darf, es sollte möglichst 100 % verfügbar sein. In Eingebetteten Systemen mit programmierbaren Prozessoren dürfen zum Beispiel Programme nicht in Endlosschleifen fallen oder endlos auf eine Eingabe warten.

Energiesparend müssen alle Eingebetteten Systeme sein, die ihre Energie aus Akkus beziehen, also alle tragbaren Eingebettete Systeme. Jedoch besteht diese Forderung auch zunehmend für alle Computersysteme, da Energie allgemein teurer wird. Ein Eingebettetes System spart Energie, wenn die Verlustleistung möglichst gering gehalten wird. Dies kann beispielsweise durch die Wahl geeigneter Mikroprozessoren bzw. durch Mehrprozessorsysteme erreicht werden.

Eingebettete Systeme sollen **effizient**, also leistungsfähig und wirtschaftlich funktionieren. Die Stromversorgung bzw. der Akku soll das Gerät möglichst lange funktionsfähig halten. Das heißt, zur Effizienz gehört in vielen Fällen nicht nur möglichst geringe Größe und Gewicht, sondern auch Sparsamkeit im Energiebedarf, möglichst geringe Programmgröße usw.

Größe und Gewicht, **physikalische und elektrische Robustheit** spielen besonders bei mobilen Systemen wie Kraftfahrzeugen, Flugzeugen, tragbaren Geräten und militärischen Geräten eine Rolle. Beispielsweise darf ein Handy nicht zu groß und zu schwer

sein, sonst ist es unhandlich und wird nicht gekauft. Eingebettete Systeme in Kraftfahrzeugen, Flugzeugen und Geräten müssen unempfindlich gegenüber Erschütterungen, Feuchtigkeit, elektromagnetischen Feldern, Temperaturschwankungen usw. sein. Erschütterungen werden gemessen in „Erdbeschleunigungen g“ z. B. müssen Handys zwischen „4 g“ und „8 g“, also vier bis achtfache Erdbeschleunigung aushalten, ohne ihre Funktionstüchtigkeit einzubüßen. „Elektromagnetische Verträglichkeit“ abgekürzt EMV (Electro Magnetic Compatibility EMC) bedeutet, dass elektrische bzw. magnetische Felder bis zu einer spezifizierten Stärke die Funktion des Geräts nicht stören dürfen.

Eingebettete Systeme sind oft **heterogen** aufgebaut, d. h. bestehen aus unterschiedlichen Hardware- und Software-Komponenten. Beispielsweise kann eine Temperatur-Regelung aus dem analogen Temperaturfühler, einem digitalen Verarbeitungsteil das die „Stellgröße“ in Abhängigkeit der Zeit berechnet und wiederum aus einem analogen Stellglied bestehen. In diesem Beispiel werden **unterschiedliche Datenarten** verarbeitet. Die Temperatur wird als analoges Datum gemessen. Die Temperatur-Abweichung wird als digitales Datum algebraisch ermittelt und daraus wird die Stellgröße bestimmt, die wieder als analoge Größe ausgegeben wird (siehe Abbildung 1.3, Seite 5). Die Temperaturänderung ist ein **asynchrones Ereignis**, d. h. ein Ereignis, das normalerweise nicht zeitgleich mit einem Taktsignal auftritt. Bei anderen Eingebetteten Systemen, zum Beispiel bei Videodekodern, werden die Daten als **kontinuierliche Datenströme** eingegeben.

Die **Entwicklungszeit** ist in Konkurrenzsituationen oft von Wichtigkeit. In vielen Fällen entscheidet der Zeitpunkt der Markteinführung über den Verkaufserfolg eines Produkts. Diese Forderung gilt allgemein für Entwicklungsabteilungen. Im Besonderen gilt dies z. B. für den Spielzeugmarkt. Vor dem Weihnachtsgeschäft müssen neue Spielzeuge in den Läden sein. Hier spielt auch der Preis eine überragende Rolle. Bei Spielzeugen sind weitere Randbedingungen die Ungefährlichkeit für Kinder, Ungiftigkeit der Materialien, scharfe Kanten dürfen nicht auftreten usw.

Hardwareaufwand und Optimierung der Teilekosten gilt ebenfalls allgemein für Entwicklungsabteilungen. In unserem Fall bedeutet es, die Hardware für das Eingebettete System so zu wählen, dass die Teilekosten minimal werden. Dies kann bei der Selektion der Mikroprozessoren eine Rolle spielen oder bei der Entscheidung zwischen einem FPGA oder einem ASIC (siehe Abschnitt 1.7, Seite 18). Bei Spielzeug spielen beispielsweise die Hardwarekosten und damit der Preis eine große Rolle.

Besondere Anforderungen an Eingebettete Systeme

Die folgende Liste der Anforderungen trifft für einen großen Teil der Eingebetteten Systeme zu, jedoch grundsätzlich nicht für alle.

- Echtzeitverhalten, Verarbeitungsgeschwindigkeit,
- Wartbarkeit,
- Wiederverwendbarkeit und Skalierbarkeit (z. B. bei Teilsystemen und Komponenten),
- verteilte Implementierung (verteilte Systeme),
- Geringe Verlustleistung (z. B. bei mobilen Geräten: Handys) Erweiterter Temperaturbereich (z. B. bei militärischen Geräten),
- Manipulationssicherheit, Datensicherheit, Sicherheit gegen Vandalismus (zum Beispiel bei Selbstbedienungsgeräten).
- Einfachheit der Bedienung (zum Beispiel bei Selbstbedienungsgeräten).

Viele dieser Kriterien scheinen dem Kunden selbstverständlich oder sind ihm nicht bewusst, müssen aber für den Entwickler dennoch formuliert werden.

Echtzeitverhalten (Real Time Behaviour) wird vielfach zu den funktionalen Eigenschaften gerechnet. Echtzeitverhalten bedeutet, dass Informationen unter Einhaltung von Zeitschranken verarbeitet werden müssen. Man unterscheidet **harte Echtzeitschranken** (Hard Real Time Constraints) und **weiche Echtzeitschranken** (Soft Real Time Constraints). Die harten Echtzeitschranken müssen auf jeden Fall vom entsprechenden Eingebetteten System eingehalten werden, bei Nichteinhaltung ist möglicherweise eine Katastrophe die Folge. Ein Beispiel dafür ist das ABS-Bremssystem im Kraftfahrzeug. Beim Überschreiten einer weichen Echtzeitschranke leidet schlimmstenfalls die Qualität des Systems, es ist jedoch keine Katastrophe zu erwarten. Falls zum Beispiel das Dekodieren eines einzelnen Teilbildes in einem Videogerät länger dauert als spezifiziert, kommt es eventuell zu einer kurzen ruckartigen Bildfolge, aber der Videofilm läuft weiter. Echtzeitverhalten spielt eine zentrale Rolle in der Entwicklung von Eingebetteten Systemen und wird im Abschnitt „Modellbasierte Entwicklungsmethode“, Seite 69 näher betrachtet. In diesem Zusammenhang wird der Begriff **„Reaktivität“** angeführt. Typisch für viele Eingebettete Systeme ist, dass sie auf Stimuli, d. h. Anregungen bzw. Zustandsänderungen von außen in bestimmten Zeitgrenzen reagieren, also Reaktivität zeigen.

Wartbarkeit bedeutet, dass fehlerhafte Teile leicht diagnostiziert, repariert bzw. ausgetauscht werden können. Im weiteren Sinne versteht man unter Wartbarkeit auch, dass mögliche Fehlerquellen bereits vor dem Versagen des gesamten Systems erkannt werden, z. B. dadurch, dass Fehlermeldungen und grenzwertige Ereignisse innerhalb des Systems, die zu Fehlern führen können, frühzeitig aufgezeichnet und angezeigt werden. Heutzutage wird bei den meisten Eingebetteten Systemen nicht mehr repariert, sondern fehlerhafte Teile bzw. das gesamte Eingebettete System werden ausgetauscht.

Bei **verteilten Systemen** besteht das Gesamtsystem aus einigen Teilsystemen oder Komponenten, die lokal im übergeordneten System **verteilt** sind. Die Komponenten kommunizieren über ein gemeinsames Kommunikationssystem zum Beispiel über einen seriellen Bus (siehe Abschnitt 1.6, Seite 17).

Wiederverwendbarkeit und **Skalierbarkeit** von Komponenten oder Systemteilen ist eine wichtige Forderung an den Entwickler. Dies gilt sowohl für den Hardware- als auch für den Software-Bereich. Teile, die entwickelt und getestet wurden sind kostengünstiger, wenn sie erneut verwendet werden können. Die Voraussetzungen dafür sind, dass die Schnittstellen passen oder skalierbar sind z. B. durch generische Datenbreiten, die bei der Implementierung angepasst werden. „Entwickeln für Wiederverwendung“ (Design for Reuse) bedeutet von Anfang an, das zu entwickelnde Teil „universeller“ zu gestalten, damit es auch bei der nächsten Version des Systems oder bei ähnlichen Systemen passt.

Ein **erweiterter Temperaturbereich** gilt für viele Eingebettete Systeme, die außerhalb von Gebäuden und in Umgebungen mit großen Temperaturschwankungen funktionieren müssen. Beispiele dafür sind Geräte auf Baustellen, Geldausgabeautomaten die im Freien stehen oder militärische Geräte. Hier gelten im Allgemeinen größere Temperaturbereiche, in denen die einwandfreie Funktion des Geräts sichergestellt sein muss, zum Beispiel -60^0 C bis $+80^0$ C.

Manipulationssicherheit, **Datensicherheit**, Sicherheit gegen Vandalismus und **Einfachheit der Bedienung** wird bei Selbstbedienungsgeräten und allen sogenannten Automaten gefordert, die in der Öffentlichkeit stehen und deren Steuerung in den Bereich Eingebettete Systeme fallen. Beispiele sind Geldausgabeautomaten, Fahrkartenautomaten, Getränke- und Lebensmittelautomaten usw. Manipulationssicherheit bedeutet, dass die Steuersoftware und auch beispielsweise die Konfiguration des FPGA eines Systems nicht von einem Außenstehenden (einem „Hacker“) veränderbar sein darf. Aus diesem Grund sind zum Beispiel die Konfigurationsdateien des Xilinx MPSoC: Zynq UltraScale verschlüsselbar (Abschnitt „Programmierbare Ein-Chip-Systeme“, Seite 29 und [Xi21]). Datensicherheit bedeutet, dass beispielsweise die Daten, die in einen Geldausgabeautomaten eingegeben werden, geschützt bleiben. Es handelt sich dabei zum Beispiel um die Geheimzahl oder die Daten, die von einer EC-Karte eingelesen werden. Sie dürfen von Außenstehenden in keinem Fall erreichbar sein.

Je nach Verwendungszweck sind mehrere der oben genannten nichtfunktionalen Anforderungen wichtig, eventuell in unterschiedlicher Priorität. Militärischen Geräten wird beispielsweise der Forderung nach Robustheit eine höhere Priorität zugewiesen als die Effizienz. Bei kommerziellen Geräten, z. B. bei Haushaltsgeräten, steht die Forderung nach Effizienz an höherer Stelle als die Robustheit. Bei Handys wird die Reihenfolge der Prioritäten sein: Größe, Gewicht, Verlustleistung, Einfachheit der Bedienung, Robustheit.

2.1.2 Lastenheft

Ein Kunde sieht einen Bedarf für ein System, das nicht handelsüblich auf dem Markt verfügbar ist und wendet sich mit einem Anforderungsdokument (Requirements Document oder Product Requirements Document PRD) bzw. einem „Lastenheft“ an die Marktabteilung eines Entwicklungshauses. In vielen Fällen hängt die Auftragserteilung von einem Angebot ab, dem ein „Pflichtenheft“, als Antwort auf ein Lastenheft, zugrunde liegt. Dies gilt in gleichem Maße auch für Softwaresysteme. Die Form des Lasten- und Pflichtenhefts ist im deutschsprachigen Raum unter [DIN69905] genormt. Das Lastenheft ist damit sozusagen ein im deutschsprachigen Raum genormter Anforderungskatalog, der oft bei (öffentlichen) Ausschreibungen für ein Projekt verwendet wird. Das Lastenheft enthält die *„Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers“*. Im juristischen Sinne sind Lastenheft und Pflichtenheft Vertragsdokumente für eine Projektentwicklung. Das Lastenheft enthält folgende Punkte:

1. Zielsetzung
2. Produkteinsatz
3. Produktübersicht
4. Produktfunktionen
5. Produktdaten
6. Produktleistungen
7. Qualitätsanforderungen
8. Ergänzungen

Wichtig ist eine gründliche „Machbarkeits-Analyse“ der Kundenanforderungen (Requiremets Analysis). Kunden sind meist keine Ingenieure bzw. Informatiker und haben nur verschwommene Vorstellungen von den technischen Möglichkeiten für die Realisierung eines Systems und der damit verbundenen Kosten.

In Bezug auf die Kundenanforderungen werden wir uns nicht an die Punkte des Lastenhefts nach DIN 69905 halten, sondern für unsere Beispiele die kürzere Version aus dem Buch von Wolf [Wolf02] wählen, die auch mehr die Anforderungen an Eingebettete Systeme berücksichtigt. Im Anforderungskatalog müssen zumindest folgende Punkte aufgeführt werden:

- Name: Kennzeichnende Namensgebung des (Eingebetteten) Systems.
- Aufgabe: Kurze Beschreibung des System-Verhaltens.
- Eingabe/Ausgabe: Datenfolge: z. B. periodisch, sporadisch. Datentypen: z. B. analog, digital.
- Funktionen: Beschreibt die gewünschten Funktionen in Hinsicht auf: Ausgabe = Funktion(Eingabe).
- Geschwindigkeit: Zeitgrenze oder andere Randbedingung.
- Kosten: Meist obere Kostengrenze.
- Energieversorgung: Festnetz oder Akkumulator.
- Größe/Gewicht:Beispiel: tragbar oder stationär. Beispiel: Handgröße.

Als Beispiel werden im Folgenden die gekürzten Kundenanforderungen für einen fiktiven Geldausgabe-Automaten dargestellt.

- Name: Geldausgabeautomat.
- Aufgabe: Gibt anstelle eines Kassenangestellten automatisch Geld aus.
- Eingabe: EC-Karte, PIN (Persönliche Identifikations-Nummer), Wahl der Funktion, Wahl der gewünschten Geldmenge.
- Ausgabe: Kontostand, gewünschte Geldmenge.
- Funktionen: Selbstbedienungsgerät. Soll verschiedene Geldscheine ausgeben können. Prüft: EC-Karte und Geheimzahl (PIN). Auswahl der Transaktionen: Geldabhebung oder Kontostandsabfrage. Anfrage der Karten-Gültigkeit in der Bank-Zentrale. Auswahl des Betrags der Geldabhebung soll über einen Bildschirm mit seitlichen Tasten möglich sein. Der Abhebungsbetrag soll in der Bank-Zentrale autorisiert werden. Das Datum der Transaktion soll auf der EC-Karte vermerkt werden. Betrug und Diebstahl sollen praktisch unmöglich sein. Der Automat soll sicher sein gegen Vandalismus.
- Geschwindigkeit: Soll schneller sein als ein Kassierer.
- Energiebedarf: Festnetz-Anschluss.
- Kommunikation: Anschluss an die Bank-Zentrale über das Telefon-Netz.
- Größe/Gewicht: Nach Bedarf.
- Kosten: kleiner als das Jahresgehalt eines Kassierers.

Ein Geldausgabeautomat ist ein übergeordnetes System, die Steuerung desselben wird einem zentralen Eingebetteten System anvertraut, das wiederum einige untergeordnete Eingebettete Systeme kontrolliert. Es handelt sich hier um ein System von hierarchisch angeordneten, verteilten Eingebetteten Systemen. Die oben aufgeführten Kundenanforderungen betreffen wie üblich, hauptsächlich die funktionalen Anforderungen, in unserem Geldausgabeautomaten-Beispiel sind die Wahl der EC-Karte und des PIN als Eingabe eindeutige funktionale Anforderungen, während die Kosten, der Energiebedarf sowie Größe und Gewicht nichtfunktionale Anforderungen sind.

Das Beispiel ist stark gekürzt; denn das Management einer Bank wird anstatt „Betrug und Diebstahl sollen praktisch unmöglich sein“, sich etwas präziser ausdrücken, wie zum Beispiel: die Geldscheine im Geldausgabeautomat sollen vor Diebstahl in einem Safe in einer bestimmten genormten Sicherheitsklasse geschützt werden. Auch die