

Electronic Design Automation (EDA)

Systementwurf

Systembegriff

Beispiel Antiblockiersystem

Signalverarbeitung

Hardware/Software- Partitionierung

Hardware oder Software?

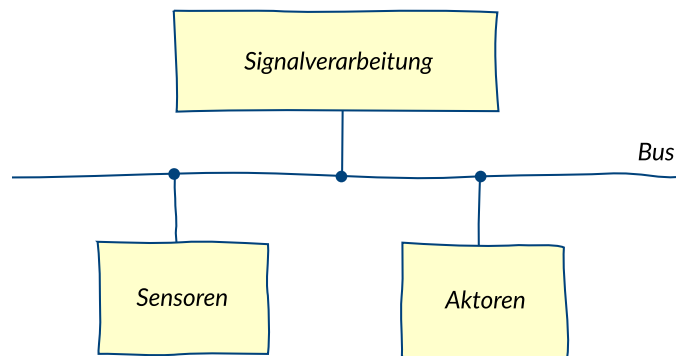
Electronic Design Automation

Systementwurf: Systembegriff

Systembegriff

Technische Systeme...

... umfassen im Wesentlichen Komponenten der Elektrotechnik, der Informatik sowie der mit ihnen wechselwirkenden Umgebungen. Sie bestehen in der Regel aus Sensor-, Aktor- und Signalverarbeitungs-komponenten bzw. -prozessen.



Grob vereinfacht besteht ein (technisches) System aus Sensoren und Aktoren, die den Kontakt mit der Umwelt herstellen, sowie einem Signalverarbeitungsteil. Sensoren nehmen physikalische Signale auf, die im Signalverarbeitungsteil Rechenprozessen unterworfen werden. Als Ergebnis der Verarbeitung werden Aktoren angesteuert, die entsprechende Aktionen an die Umwelt weitergeben. Die Komponenten kommunizieren häufig über einen gemeinsamen, so genannten (System-)Bus miteinander. Sensoren und Aktoren sind elektrischer, mechanischer, optischer, chemischer oder anderer Natur. Der Signalverarbeitungsteil ist heute in der Regel elektronischer Natur.

Systementwurf: Beispiel Antiblockiersystem

Beispiel: Antiblockiersystem



Quelle: Robert Bosch GmbH



Drehzahlsensor



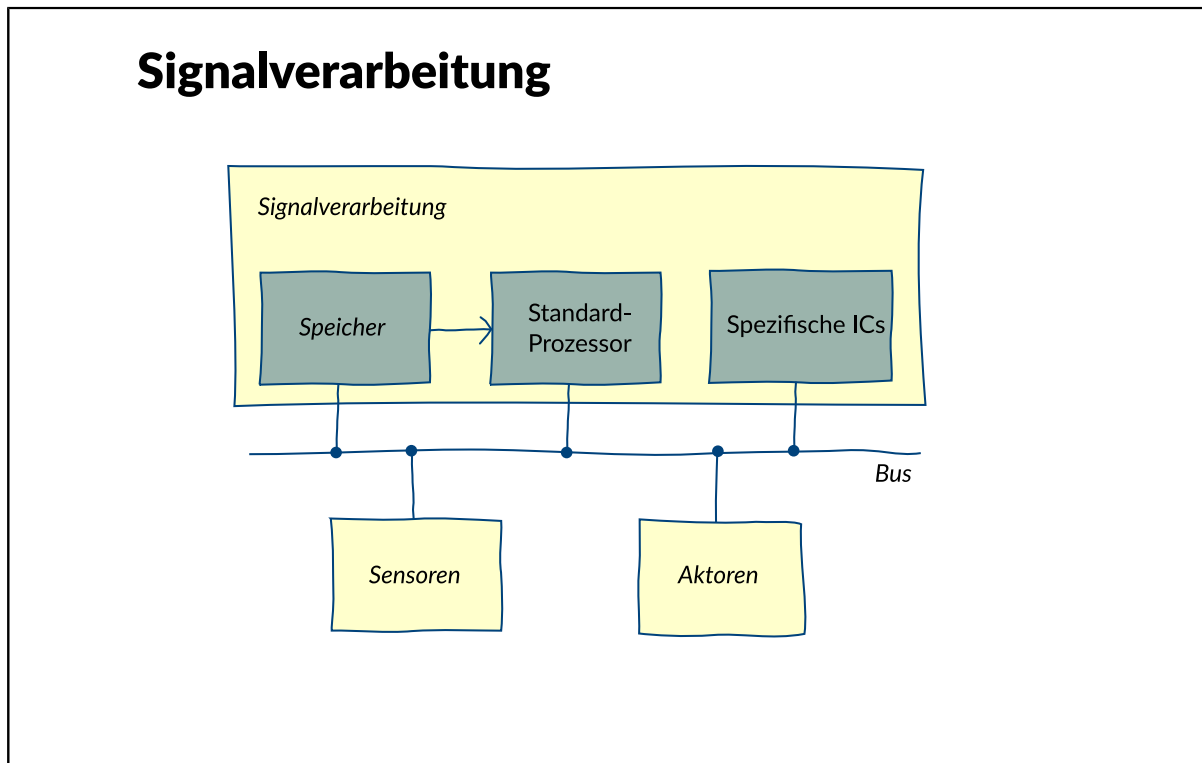
Hydraulik mit Steuergerät

Ein Beispiel für ein technisches System stellt das Antiblockiersystem (ABS) eines Autos dar.

Sensoren erfassen die Umdrehungsgeschwindigkeit der Räder des Fahrzeugs. Die entsprechenden Signale werden im Steuerungsbaustein verarbeitet, der schließlich entsprechende Signale an die hydraulischen Bremszylinder (Aktoren) abgibt.

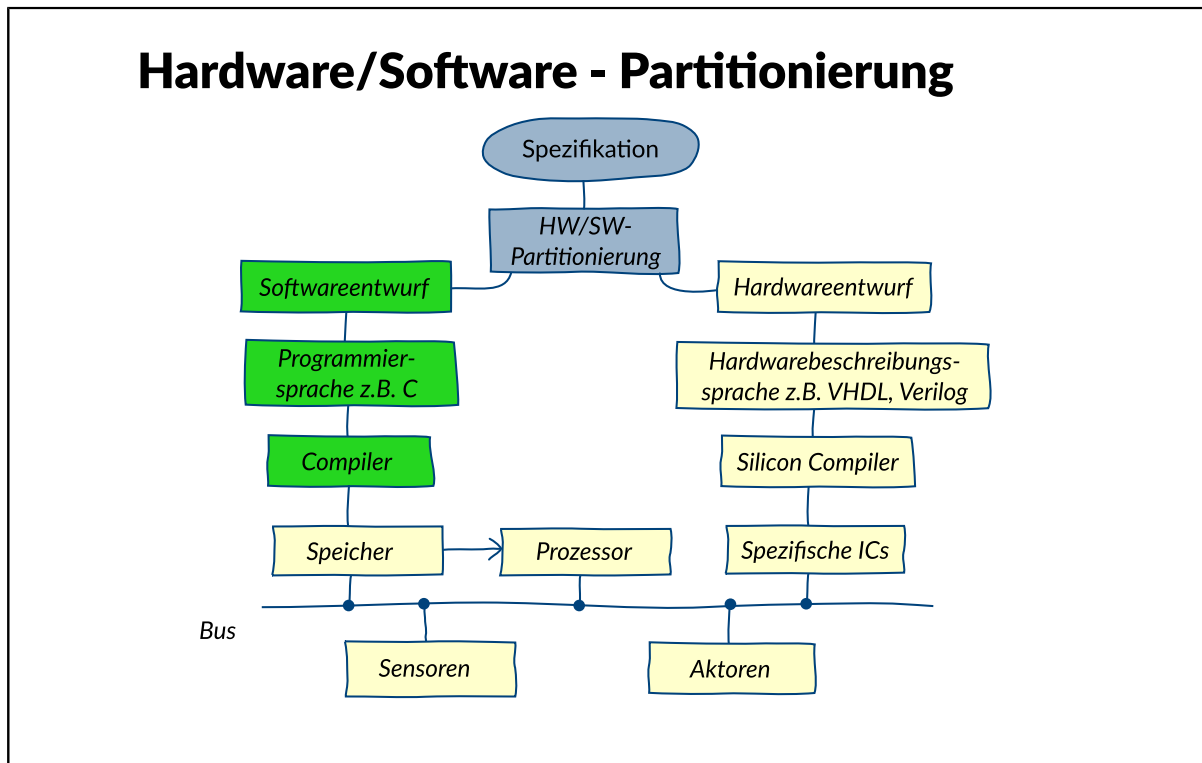
Im Rahmen dieser Vorlesung wollen wir uns auf den elektronischen Signalverarbeitungsteil konzentrieren.

Systementwurf: Signalverarbeitung



Wiederum grob vereinfacht besteht der Signalverarbeitungsteil aus einem universellen Rechenwerk (Standardprozessor), einem Speicher, der das auf dem Prozessor auszuführende Programm sowie die anfallenden Daten enthält, sowie weitere, für Spezialaufgaben zur Verfügung stehende, so genannte spezifische integrierte Schaltungen (Integrated Circuits, ICs). Während ein Speicher in der Regel unverzichtbar ist, können – je nach Aufgabenstellung – entweder der universelle Prozessor oder die spezifischen ICs entfallen.

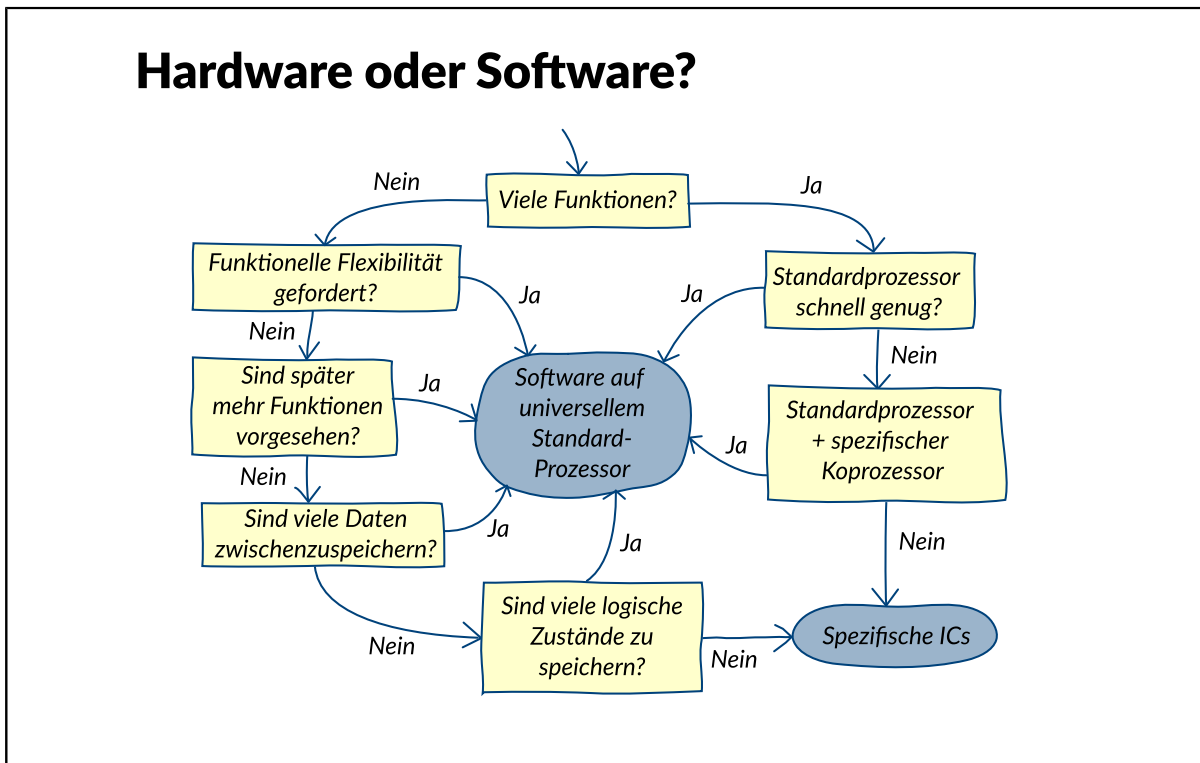
Systementwurf: Hardware/Software- Partitionierung



Ausgangspunkt für den Entwurf des Signalverarbeitungsteils eines Systems ist seine Spezifikation, in der die gewünschten Funktionen, die das System ausführen soll, beschrieben sind. Entsprechend der vorgenommenen Aufteilung in einen universellen Standardprozessor und spezifische ICs besteht der erste Teil der Entwurfsaufgabe darin zu entscheiden, welche der gewünschten Funktionen in – auf dem universellen Prozessor abzulaufende – Software und welche unmittelbar in – durch spezifische ICs zu realisierende – Hardware implementiert werden sollen.

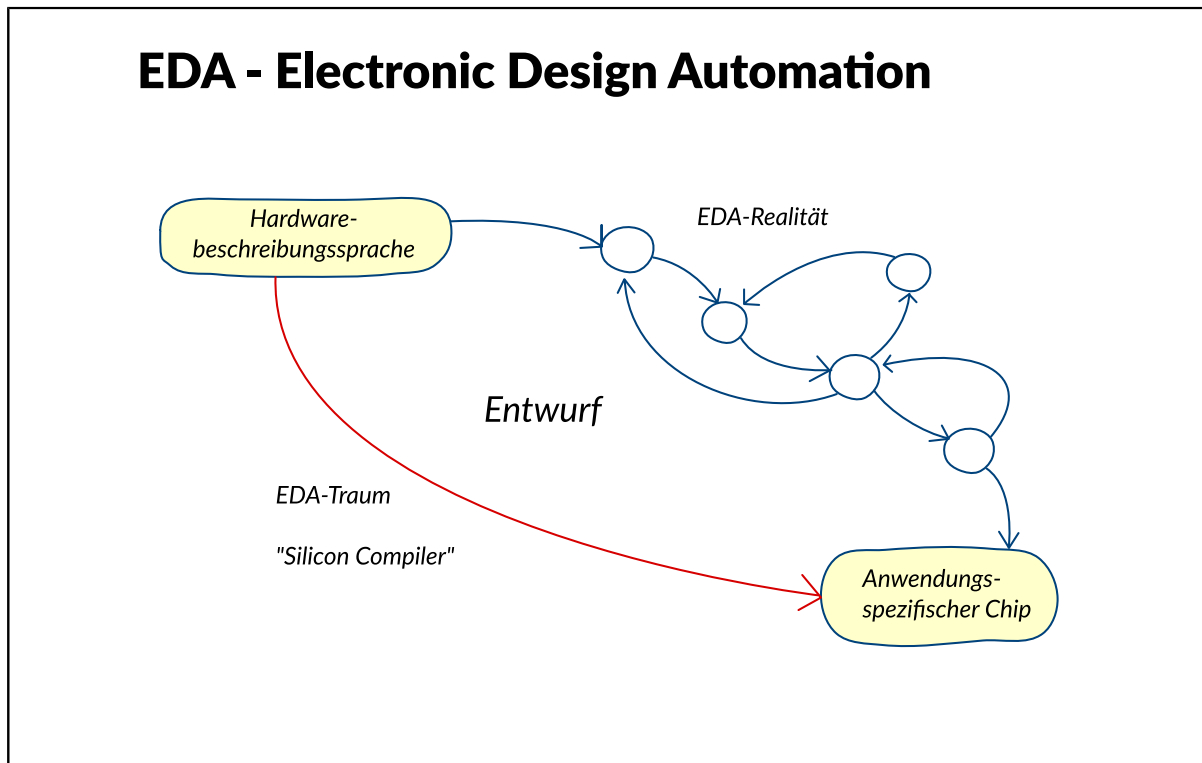
Der Entscheidungsprozess, welche Funktionen in Software bzw. in Hardware ausgeführt werden sollen, ist Gegenstand der sogenannten Hardware/Software-Partitionierung. Sie ist ein aktueller Forschungsgegenstand, jedoch nicht Gegenstand dieser Vorlesung, die sich auf Werkzeuge für den Hardwareentwurf beschränkt. Wie das Bild Hardware/Software - Partitionierung zeigt, weist der Entwurf auf der gewählten, noch sehr abstrakten Betrachtungsebene große Ähnlichkeit mit dem Softwareentwurf auf. Die zu realisierende (Teil-)Funktion muss zunächst in einer geeigneten - maschinenlesbaren - Sprache beschrieben werden. Dies erfolgt beim Softwareentwurf in einer höheren Programmiersprache wie z.B. C, C++ oder Java. Auf der Hardwareseite wird das entsprechende Äquivalent als Hardware-Beschreibungssprache bezeichnet. Die am weitesten verbreiteten Hardwarebeschreibungssprachen sind Verilog und VHDL; dabei handelt es sich um formale, programmiersprachenähnliche Beschreibungen, die in der Lage sind, neben arithmetischen Ausdrücken, Kontrollanweisungen etc. auch Hardwarestrukturen und -eigenschaften zu beschreiben.

Systementwurf: Hardware oder Software?



Diese Frage ist nicht leicht zu beantworten. Einige Kriterien dazu sind beispielhaft dargestellt. Eine Prozessor-Lösung ist sicher stets dann vorzuziehen, wenn viele Daten sequentiell verarbeitet werden müssen und wenn die Lösung flexibel sein soll, da viele Funktionsänderungen durch Modifikation der Software leicht durchgeführt werden können. Die höchste Performance wird dagegen stets durch eine unmittelbare Hardware-Realisierung erzielt. Weitere nicht dargestellte Aspekte sind beispielsweise die Verlustleistung und vor allem die Kosten der gewählten Lösung, die stark von der zu realisierenden Stückzahl abhängig sind.

Systementwurf: Electronic Design Automation



Mit Hilfe eines Compilers wird die in einer höheren Programmiersprache beschriebene Software nun in der Regel vollautomatisch in die Maschinsprache des Prozessors übersetzt, so dass sie dort ausgeführt werden kann. Es liegt deshalb nahe, dieselbe Idee, auch für den Hardwareentwurf zu verwenden, und sich einen "Silicon-Compiler" vorzustellen, mit dessen Hilfe aus dem in der Hardware-Beschreibungssprache vorliegenden Entwurf ebenfalls vollautomatisch die für die Fertigung einer integrierten Schaltung erforderlichen Daten erzeugt werden. Tatsächlich ist dies – und wird es voraussichtlich auch bleiben – ein Traum. Der dazu erforderliche "Übersetzungsprozess" ist derartig komplex und aufwändig, dass er nur als Abfolge von zahlreichen unterschiedlichen Entwurfsschritten mit zunehmendem Detaillierungsgrad beherrscht werden kann. Viele dieser Schritte sind heute automatisiert bzw. teilautomatisiert. Wir sprechen dabei von Entwurfsautomatisierung (Electronic Design Automation, EDA) und sind damit beim eigentlichen Gegenstand dieser Vorlesung angelangt.

EDA als solches ist ein aktueller Forschungsgegenstand, dessen Status sich kontinuierlich verändert. In dieser Vorlesung wird der heutige Stand dargestellt, wobei der Schwerpunkt auf den grundsätzlichen Prinzipien und Algorithmen liegt. Der aktuelle Stand in der Praxis kann davon im Detail abweichen. Wichtig ist deshalb, die grundsätzliche Vorgehensweise und die zugrundeliegenden Zusammenhänge zu verstehen. Dies soll, anhand exemplarischer Beispiele vermittelt werden.