



# Compiler für Eingebettete Systeme

## [CS7506]

Sommersemester 2014

Heiko Falk

Institut für Eingebettete Systeme/Echtzeitsysteme  
Ingenieurwissenschaften und Informatik  
Universität Ulm



# Kapitel 1

## Einordnung & Motivation der Vorlesung

# Eingebettete Systeme

**Definition:** Eingebettete Systeme (ES) sind

- informationsverarbeitende Systeme,
  - die in ein größeres Produkt eingebettet sind.
- 
- Informationsverarbeitung Eingebetteter Systeme nicht ausschlaggebend für Kauf
  - Statt dessen: Nutzen des übergeordneten Produkts beeinflusst Kaufentscheidung

*[P. Marwedel, Eingebettete Systeme, Springer, 2007]*

# Anwendungsbereiche Eingebetteter Systeme

**Konsumgüter**



**Multimedia**



**Transportmittel**



**Telekommunikation**



**Gebäudeautomation, Robotik, ...**

# Bedeutung Eingebetteter Systeme

- **Smartphones**            113 Mio. Geräte 2007  $\rightsquigarrow$  25,6% *Steigerung p.A.*  
365 Mio. Geräte 2012
- **UMTS**                    402 Mio. Kunden weltweit 2008  
30 Mio. Neukunden pro Quartal
- **Energieverbrauch mobiler Breitbandinfrastruktur**  
42,8 Mrd. KWh 2005  $\rightsquigarrow$  124,4 Mrd. KWh 2011
- **Breitband-Internet**  
576 Mio. Kunden 2011  $\rightsquigarrow$  100% *Steigerung i. Vgl. zu 2007*
- **US Konsum-Elektronik**  
 $\emptyset$  Haushalt: 25 Geräte,  $\emptyset$  *Erwachsener: 1.200\$ p.A.*

[[www.itfacts.biz](http://www.itfacts.biz)]

# Anforderungen an Eingebettete Systeme (1)

## Effizienz

– Laufzeit-Effizienz



– Energieverbrauch

– Codegröße



– Physikalische Größe / Gewicht



– Kosten



# Anforderungen an Eingebettete Systeme (2)

## Realzeit-Fähigkeit

- **Definition:** Für eine Eingabe  $x$  berechne ein System  $f(x)$ .
  - Ein ***Nicht-Realzeit-System*** heißt korrekt, wenn  $f(x)$  korrekt berechnet wird.
  - Ein ***Realzeit-System*** heißt korrekt, wenn zusätzlich  $f(x)$  innerhalb von außen vorgegebener Zeit berechnet wird.
- ☞ **Eine zu späte Berechnung von  $f(x)$  durch ein Realzeit-System ist gleich einer falschen Berechnung.**

# Anforderungen an Eingebettete Systeme (3)

## Realzeit-Fähigkeit

- „**Hartes**“ Realzeit-System:

Zu späte Berechnung von  $f(x) \rightsquigarrow$  Katastrophe  
(Verlust menschlichen Lebens, Umweltschäden, ...)

*Beispiel Airbag-Steuerung:*

Befehl zum Zünden der Airbags: 15ms

Zu späte Entscheidung: Verletzungsgefahr für Insassen und/oder Retter.

Daher: Airbags nicht zünden

- „**Weiches**“ Realzeit-System: Keine katastrophalen Folgen

*Beispiel DVD-Player:*

Zu späte Frame-Dekodierung: *Frame-Drop*

Unschön, aber (i.d.R.) nicht katastrophal

# Anforderungen an Eingebettete Systeme (4)

## Zuverlässigkeit / Sicherheit

- Lebensdauer Eingebetteter Systeme: Einige Jahre
- Während der gesamten Lebensdauer: Keine Ausfälle
- *Beispiel Drosselklappen-Steuerung:*
  - Produktionsvolumen: 2 Mio. Einheiten pro Jahr
  - Erlaubte Fehlerquote: 1 Einheit pro Jahr

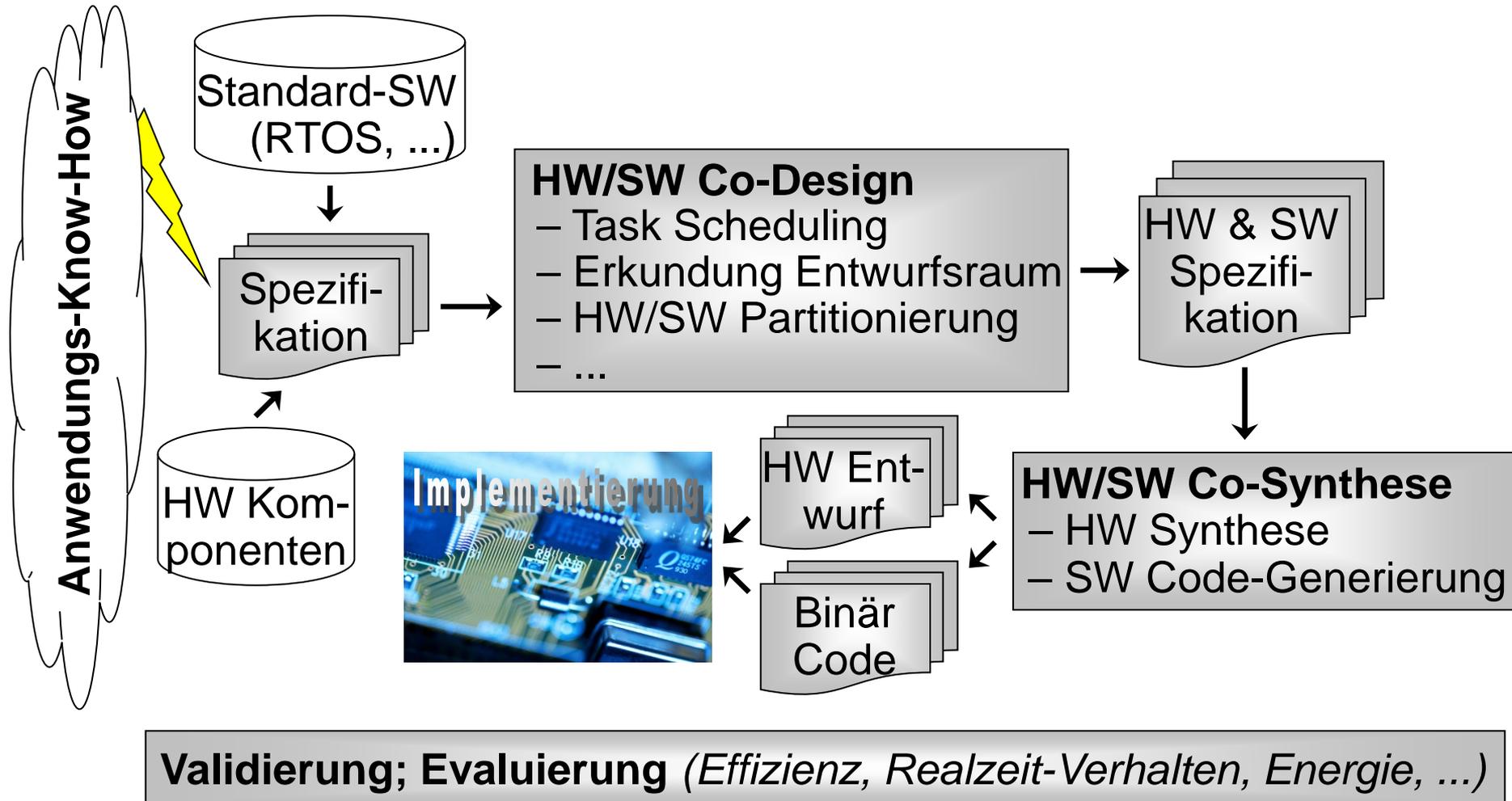
## Wartbarkeit, (begrenzte Erweiterbarkeit)

- Fehlersuche, Diagnose, Rekonfiguration zur Laufzeit

## Unterstützende Entwurfswerkzeuge ( $\leadsto$ Time to Market)

- Spezifikation, Synthese, Code-Generierung

# Entwicklungsprozess Eingebetteter Systeme



# Inhalte der Vorlesung

## Überblick

1. Einordnung & Motivation der Vorlesung
2. Compiler für Eingebettete Systeme – Anforderungen & Abhängigkeiten
3. Interner Aufbau von Compilern
4. Prepass-Optimierungen
5. HIR Optimierungen und Transformationen
6. Instruktionsauswahl
7. LIR Optimierungen und Transformationen
8. Register-Allokation
9. Compiler zur WCET<sub>EST</sub>-Minimierung
10. Ausblick

# Allgemeine Literatur

## Eingebettete Systeme

- Peter Marwedel. *Eingebettete Systeme*. Springer, 2007.  
ISBN 978-3-540-34048-5

## Compilerbau

- Steven S. Muchnick. *Advanced Compiler Design & Implementation*. Morgan Kaufmann, 1997.  
ISBN 978-1-55860-320-2
- Andrew W. Appel. *Modern compiler implementation in C*. Cambridge University Press, 2004.  
ISBN 0-521-60765-5