

Compiler für Eingebettete Systeme (CfES)

Sommersemester 2009

Dr. Heiko Falk

Technische Universität Dortmund

Lehrstuhl Informatik 12

Entwurfsautomatisierung für Eingebettete Systeme

Organisatorisches (1)

■ Vorlesung (3V)

- Montags, 14.15 – 15.45 Uhr
OH14 / E23 (Hörsaal)
wöchentlich

Freitags, 08.30 – 10.00 Uhr
OH14 / E23 (Hörsaal)
nicht wöchentlich! (☞ Folie 5)

■ Übungen (1Ü)

- Blockübung
- Praktische Implementierung eines (einfachen) ANSI-C Compilers am Rechner

■ Materialien

- Foliensätze online:
<http://ls12-www.cs.tu-dortmund.de/teaching/courses/ss09/cfes>
- Bücher & Originalartikel

Organisatorisches (2)

- **Wünschenswerte Voraussetzungen**
 - Eingebettete Systeme
 - Übersetzerbau
- **Prüfungen**
 - Mündliche Prüfung (30 min, 6 Leistungspunkte)
 - Erfolgreiche Übungsteilnahme für unbenoteten Schein
- **Closed Notebook Policy**
 - Private Laptops bleiben in der Vorlesung bitte geschlossen
 - ☞ *Deutliche Verbesserung der Arbeitsatmosphäre!*
- **Vorlesungsausfall**
 - 24. April und 19. Juni 2009 ist keine Vorlesung!

Übungsbetrieb



■ Ablauf

- Lernen von Compilerbau: Hände schmutzig machen!
- Wöchentliche Übungsgruppe & Übungszettel ungeeignet
- *Blockübung* erlaubt konzentriertes Arbeiten, Herumprobieren und Lernen, direkt am Rechner

■ Terminabsprache

- 1. Übung: KW 21 – 18.-22. Mai? *Prepass / High-Level*
- 2. Übung: KW 26 – 22.-26. Juni? *Code-Selektion*
- 3. Übung: KW 29 – 13.-17. Juli? *Low-Level*

■ Alternativ

- 2 Termine in vorlesungsfreier Zeit (ab 3. August)?

Termine *(in Absprache mit Studierenden)*



- **Freitags-Vorlesung, 08.30 – 10.00 Uhr**
 - 8. Mai
 - 15. Mai
 - 5. Juni
 - 26. Juni
 - 17. Juli
- **Übungen**
 - 1. Übung: Freitag 15. Mai, 12.30 – 16.00 Uhr
 - 2. Übung: Freitag 26. Juni, 12.30 – 16.00 Uhr
 - 3. Übung: Freitag 17. Juli, 12.30 – 16.00 Uhr

Kapitel 0

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

[www.itfacts.biz]

Anforderungen an Eingebettete Systeme (1)

■ Effizienz

- Laufzeit-Effizienz
- Energieverbrauch
- Codegröße
- Physikalische Größe / Gewicht
- Kosten



Anforderungen an Eingebettete Systeme (2)

■ Realzeit-Fähigkeit

Für 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)

- **“Hartes” Realzeit-System:**

Zu späte Berechnung von $f(x)$ → 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-Decodierung: 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

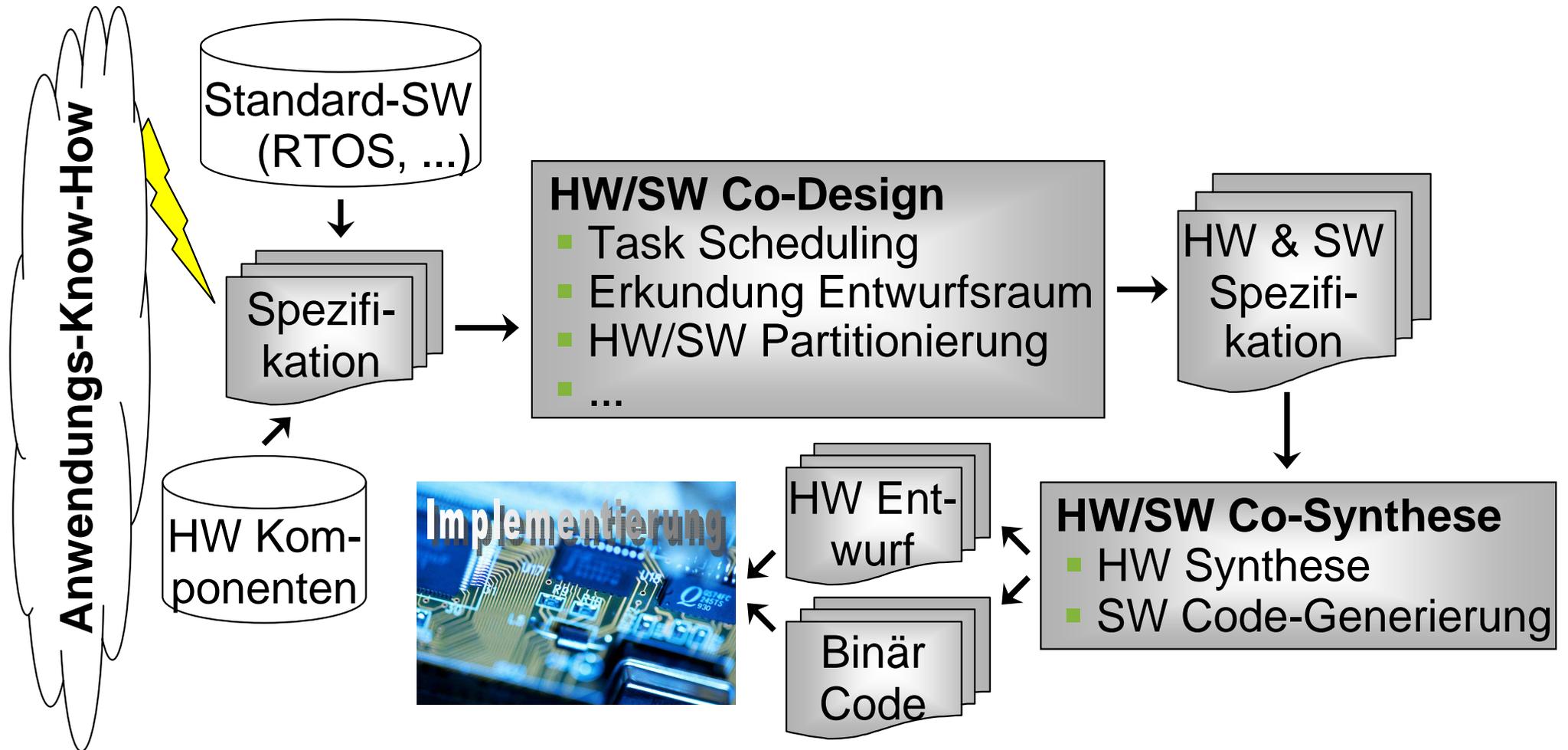
■ Wartbarkeit, (begrenzte Erweiterbarkeit)

- Fehlersuche, Diagnose, Rekonfiguration zur Laufzeit, ...

■ Unterstützende Entwurfswerkzeuge (*→ Time-to-Market*)

- Spezifikation, Synthese, Code-Generierung, ...

Entwicklungsprozess Eingebetteter Systeme



Validierung; Evaluierung (Effizienz, Realzeit-Verhalten, Energie, ...)

Gliederung der Vorlesung

- **Kapitel 1: Compiler für Eingebettete Systeme**
- **Kapitel 2: Interner Aufbau von Compilern**
- **Kapitel 3: Prepass-Optimierungen**
- **Kapitel 4: HIR Optimierungen und Transformationen**
- **Kapitel 5: Instruktionsauswahl**
- **Kapitel 6: LIR Optimierungen und Transformationen**
- **Kapitel 7: Register-Allokation**
- **Kapitel 8: Compiler zur WCET_{EST}-Minimierung**
- **Kapitel 9: 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 1-55860-320-4
- Andrew W. Appel, *Modern compiler implementation in C*, Cambridge University Press, 1998.
ISBN 0-521-58390-X