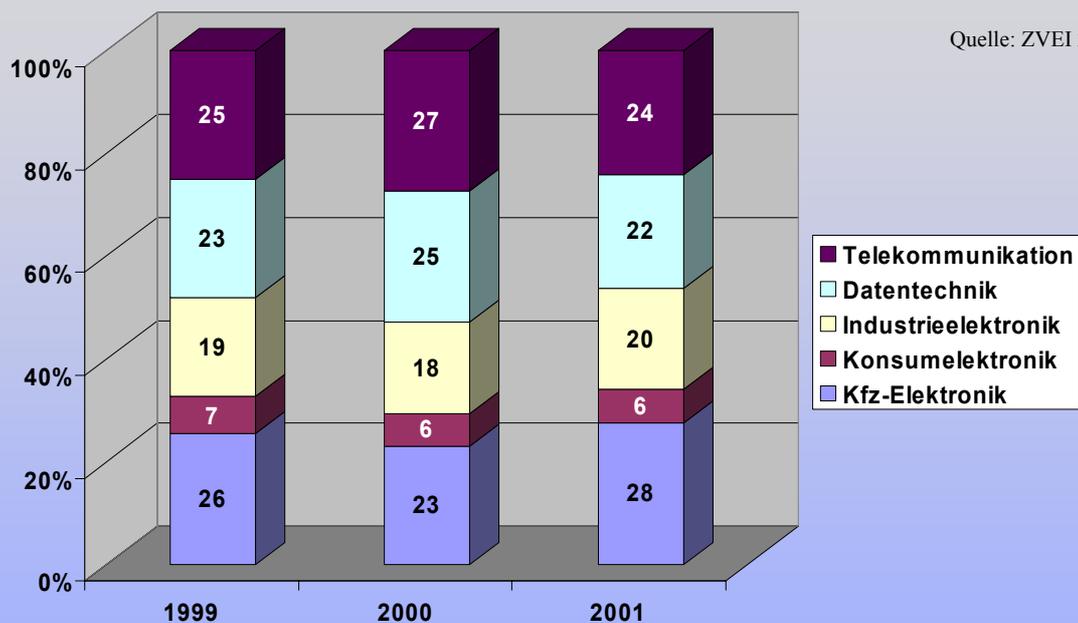


MpSoC, IP und Plattformen - oder wie nutzt man 1 Milliarde Transistoren?

R. Ernst
TU Braunschweig

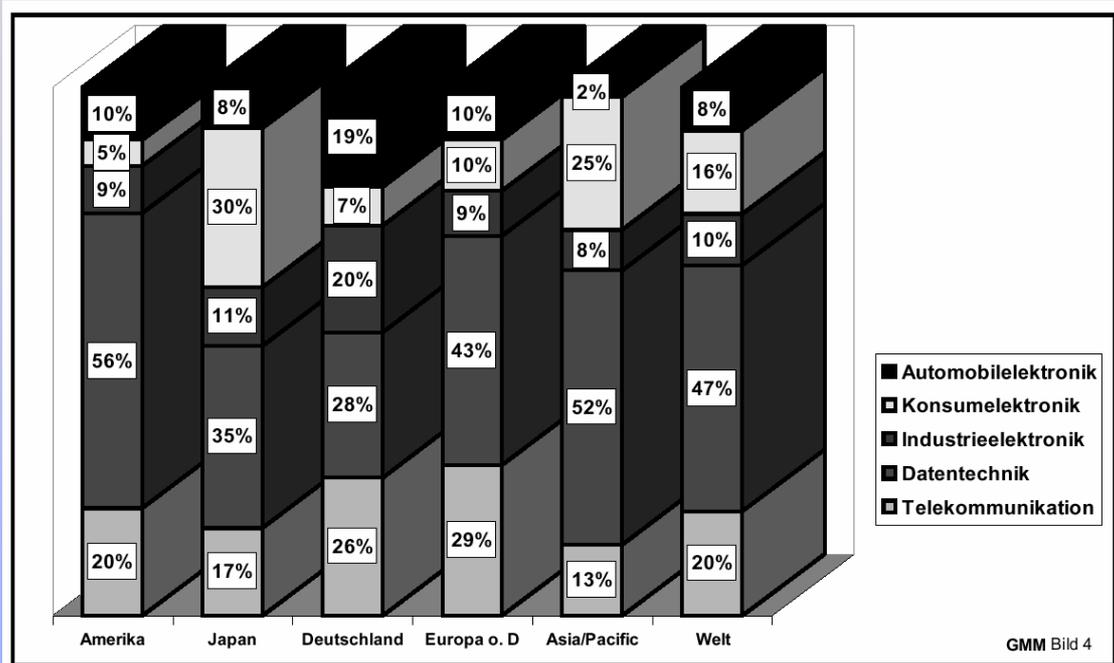
Was wird denn benötigt: *Marktanteile Mikroelektronik Deutschland*



- ⇒ Datentechnik nur ein kleiner Anteil der IT-Anwendungen
- ⇒ der größte Anteil fällt auf die sogenannten *eingebetteten Systeme*

Marktanteile Mikroelektronik international

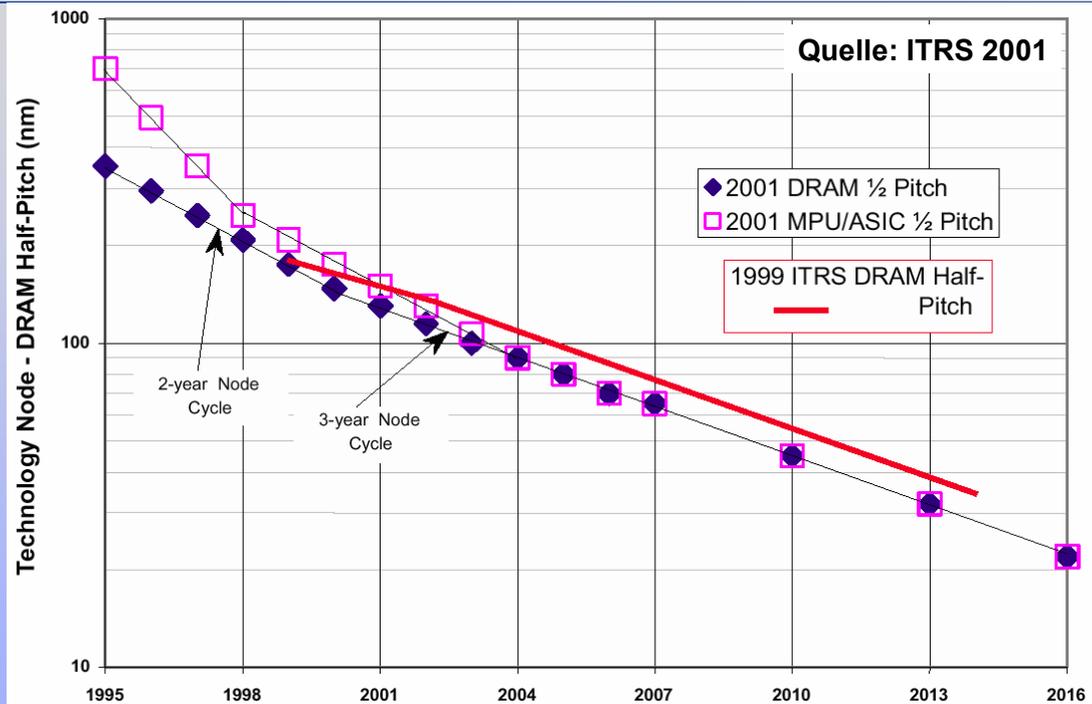
Quelle: VDE Mikroelektronik-Studie 1998 - 2003



Entwicklung der Chips

- Entwicklung der Halbleiterindustrie projiziert in der *International Technology Roadmap for Semiconductors (ITRS)*
 - Fortsetzung des exponentiellen Wachstums der Transistorzahl (Moore's Law) bis mindestens 2016
 - DRAM-Speicher
2001 540MTr/Chip → 2016: 64 GTr/Chip
 - obere Leistungsklasse Schaltungen
2001 193MTr/Chip → 2016: 6,2 GTr/Chip
 - kosteneffiziente Schaltungen
2001 97MTr/Chip → 2016: 3,1 GTr/Chip
 - ökonomische Grenze der Wachstumsgeschwindigkeit erreicht
 - kein Wachstum der Chipgröße über 3cm² erwartet
 - weiteres Wachstum der Wafergröße zur Deckung der wachsenden Herstellungskosten

Abnahme der Strukturgrößen

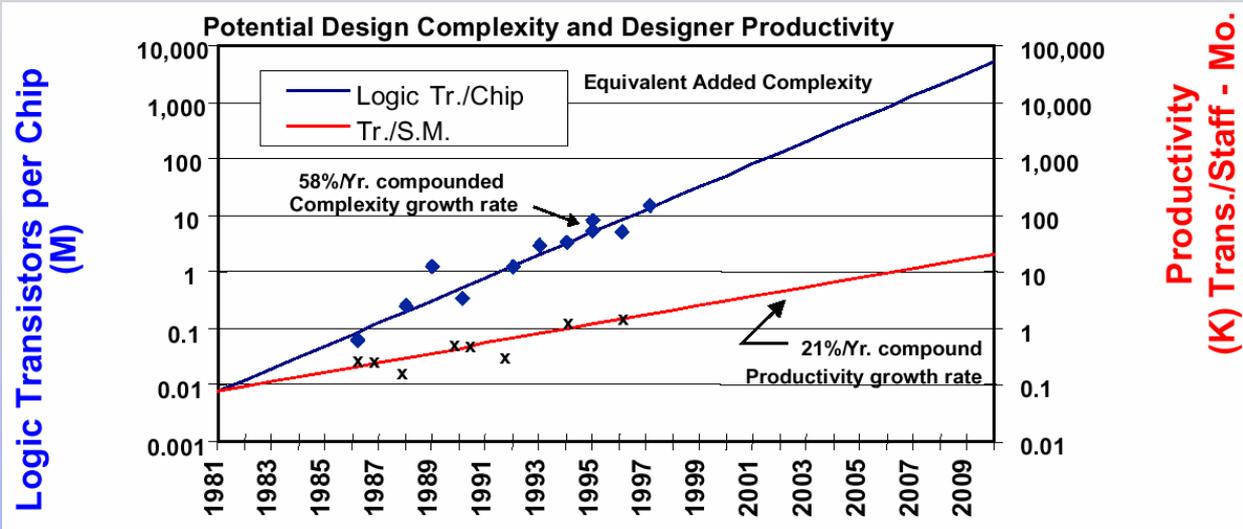


Entwicklung des Mindestabstands zwischen Leitungen

Bedeutung für den Schaltungsentwurf

- kann Entwurfsproduktivität mit Geschwindigkeit des Schaltungsentwurfs steigen?
 - 100% / 3 Jahre Steigerung der Entwurfsproduktivität mit klassischen Verfahren nicht mehr erreichbar
 - als realistisches Ziel gelten 21% /Jahr
 - wesentliche Erweiterung des Entwurfsteams nicht mehr möglich
 - überproportional steigende Kosten für Validierung/Verifikation, heute mehr als 50%
 - Kostendruck
- ⇒ „Design Gap“

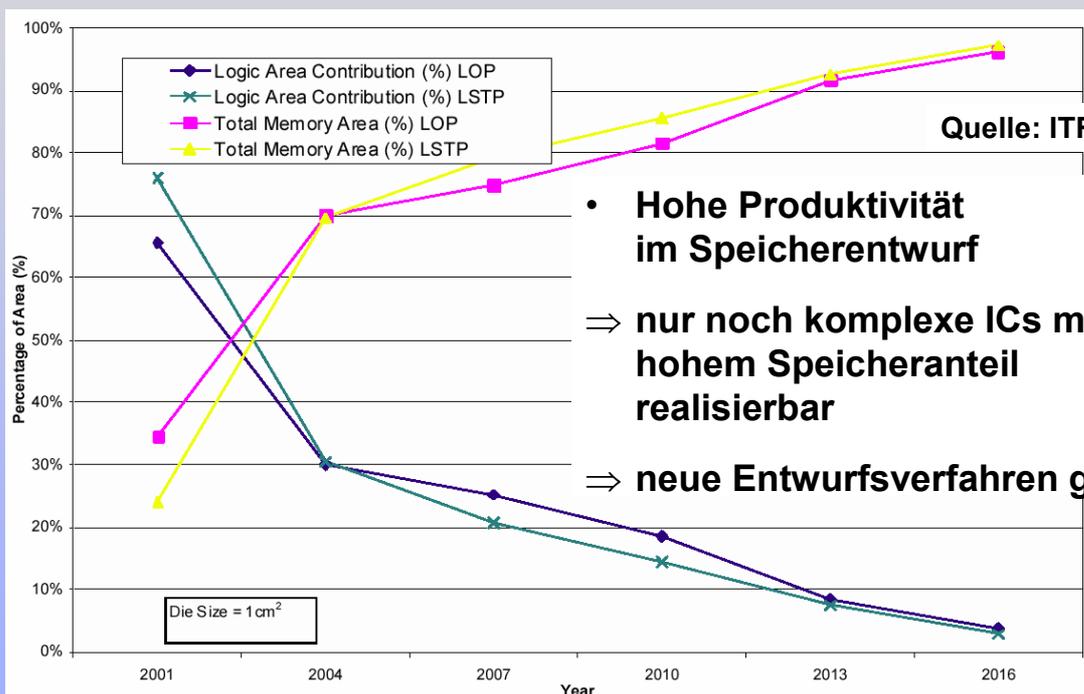
„Design Gap“



- ITRS Roadmap 99 stellt zu geringe Entwurfsproduktivität fest, gibt aber keine Lösungen

Effekt der Design Gap

Denkbare Entwicklung: Nutzung der Chipfläche für Speicher

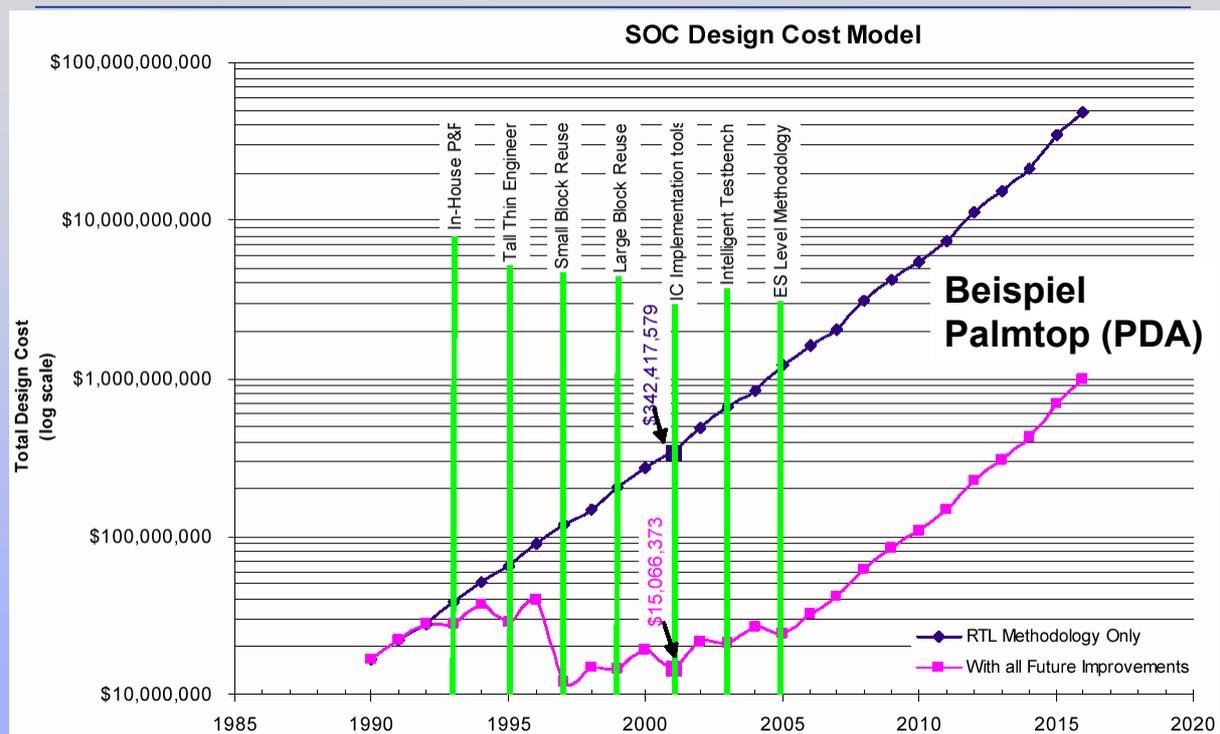


- Hohe Produktivität im Speicherentwurf
- ⇒ nur noch komplexe ICs mit hohem Speicheranteil realisierbar
- ⇒ neue Entwurfsverfahren gesucht

Aber: Höhere Rechenleistung benötigt

- Mobilkommunikation
(Faktor Rechenleistung ca. 10^3 /Generation)
- Bildcodierung: MPEG 2 - MPEG 4 - MPEG 7 -

Produktivitätssteigerung stets stufenweise



Quelle: ITRS 2001

Verfahren zur Produktivitätssteigerung

- wichtigster Schritt zur Produktivitätssteigerung
*Wiederverwendung immer größerer Schaltungsblöcke:
„Design Reuse“*
- Nutzung der verfügbaren Chip-Fläche durch Integration ganzer digitaler Systeme auf einem Chip
⇒ **System-on-Chip (SoC)**
- hochspezialisierte Komponenten erreichen hohe Effizienz hinsichtlich
 - Verarbeitungsleistung
 - Verlustleistung (Batterielebensdauer, Kühlung)
- heute: spezialisierte Parallelrechner
⇒ **Multiprozessor-Systems-on-Chip (MpSoC)**

Integration von Komponenten

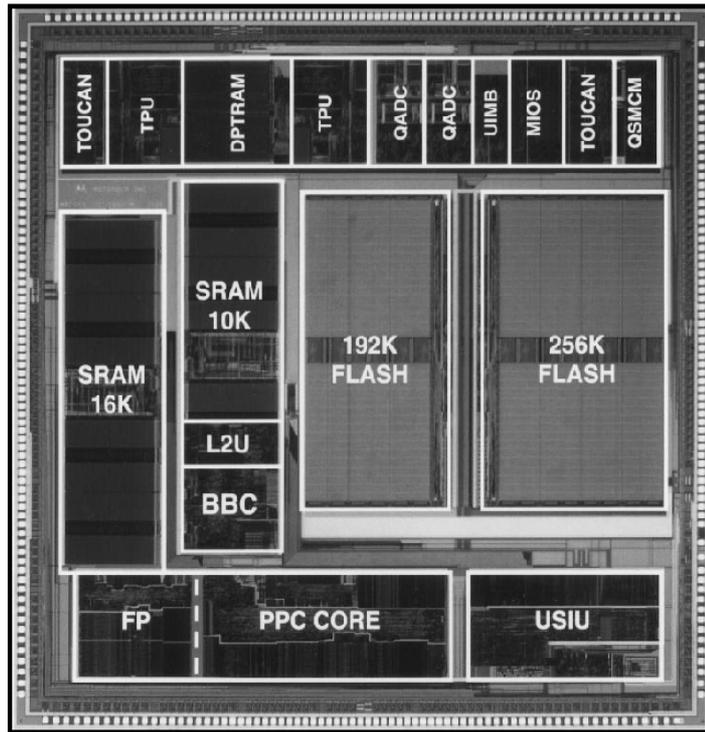
- **1.Schritt:**
Bildung von Standardbibliotheken innerhalb von Firmen
- **Beispiel:**
 - Motorola MPC 555 (1998)
 - Einsatz: Antriebssteuerung im Kfz
 - Wiederverwendung von festen physikalischen Komponenten
„Hardmacros“

MPC 555 Chiplayout

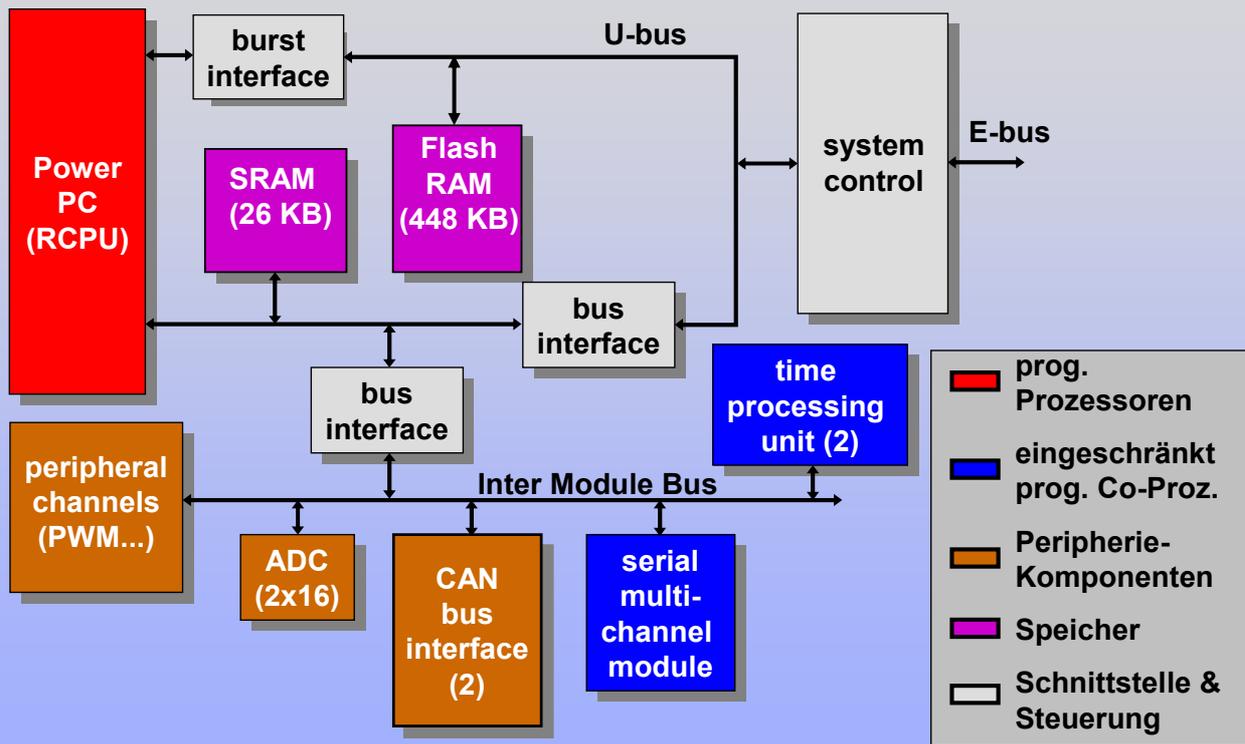
Motorola MPC 555:

PowerPC-basierte
HW Plattform für
Motorsteuerung

Quelle:
Motorola
und Mikroprozessor Report,
April 20, 98



Automobilplattform - Beispiel MPC 555

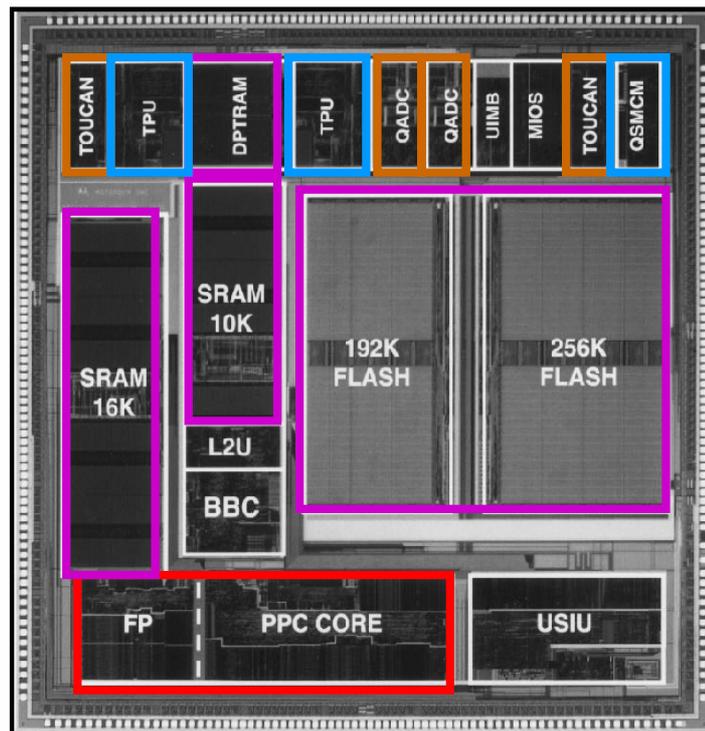


MPC 555 Chiplayout

Motorola MPC 555:

PowerPC-basierte
HW Plattform für
Motorsteuerung

Quelle:
Motorola
und Mikroprozessor Report,
April 20, 98

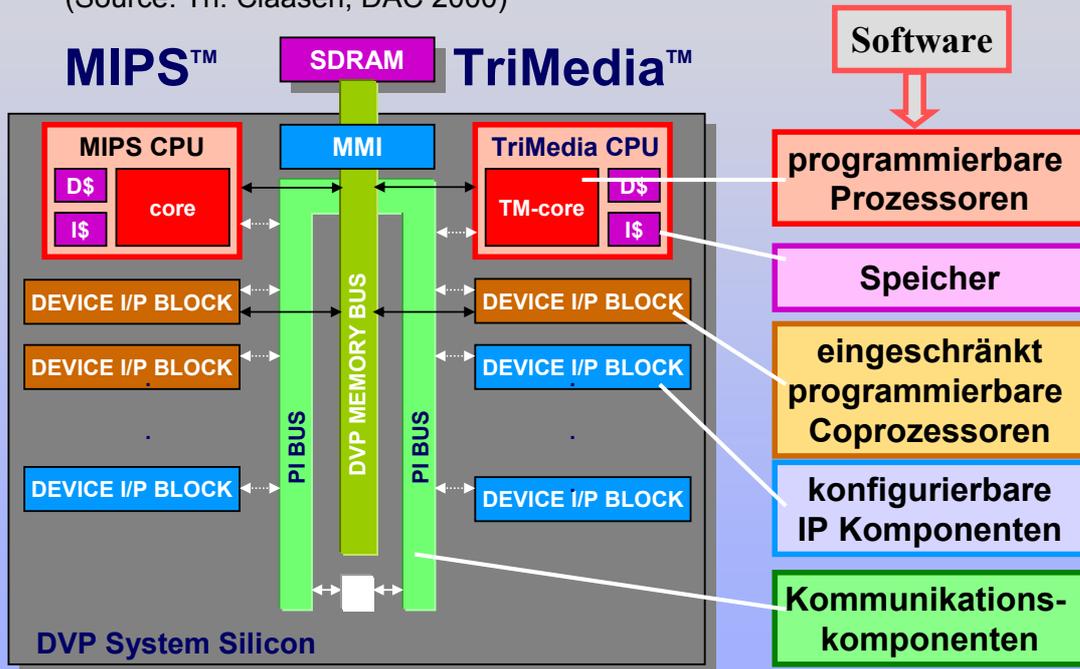


IP Industrie und Plattformen

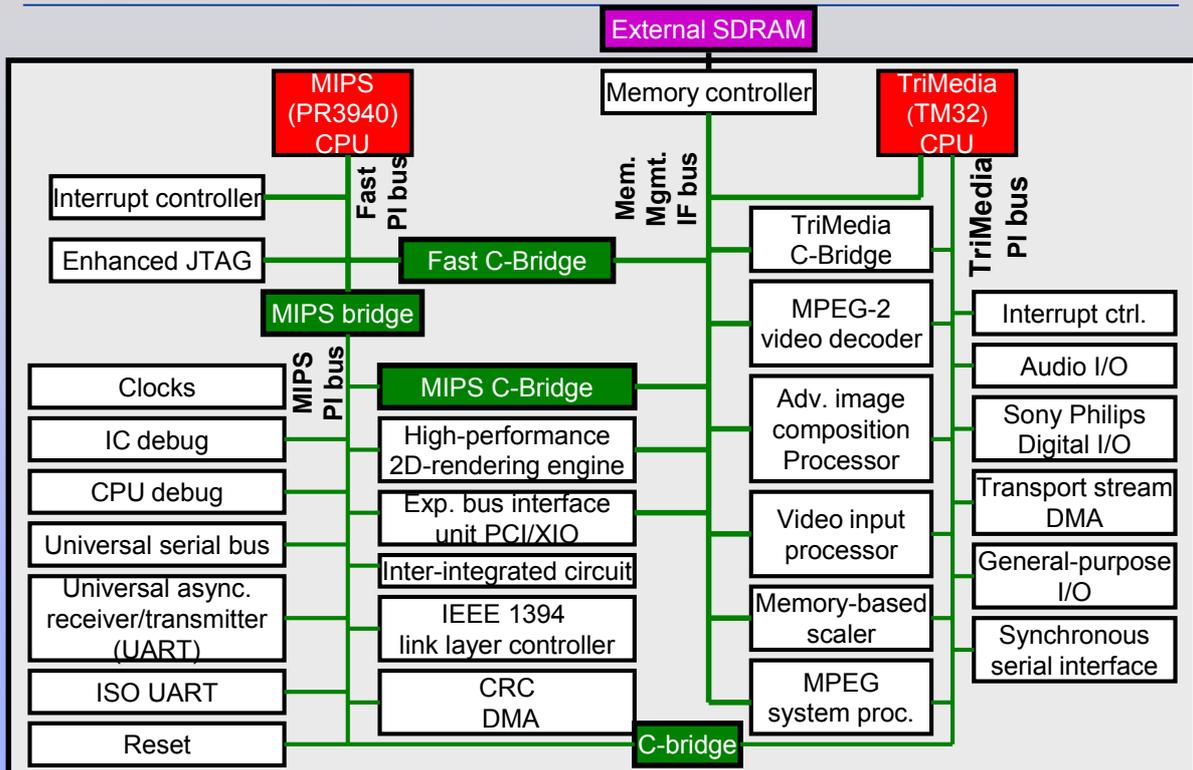
- **2. Schritt:**
Herausbildung einer Komponenten-Zulieferindustrie
 - **Lieferung von Blaupausen „Intellectual Property“ (IP)**
Beispiele: MIPS (Multimedia), ARM (Mobilkommunikation), Texas Instruments (DSP), ...
 - **Prozessoren mit anpassbarer Struktur (ASIPs)**
Beispiele: Tensilica, ImprovSys, ...
 - **konfigurierbare Busse („communication IP“)**
- **Bildung von standardisierten Schnittstellen**
 - Beispiel: internationale Standardisierungsbemühungen in der „Virtual Socket Interface Alliance“ (VSIA)
- **Plattformbildung**
 - **Beispiel: Nexperia als interne Plattform der Fa. Philips**

Beispiel für Plattform

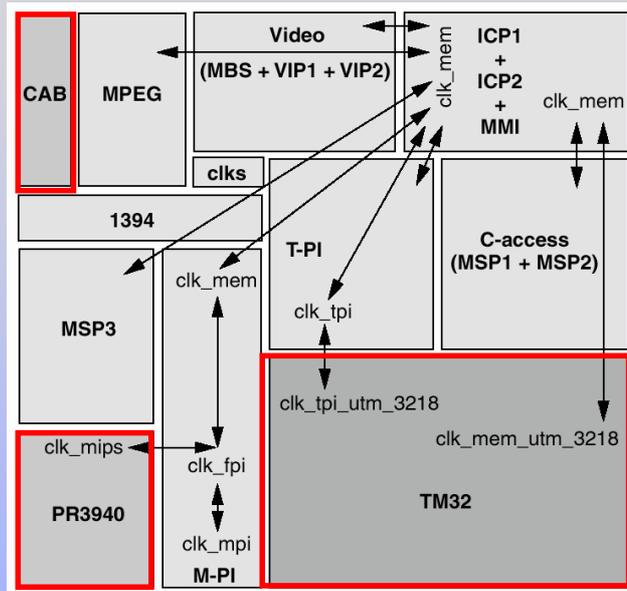
- **Philips Nexperia™ Plattform**
(Source: Th. Claasen, DAC 2000)



Einsatzbeispiel: Viper Setop Box MpSoC

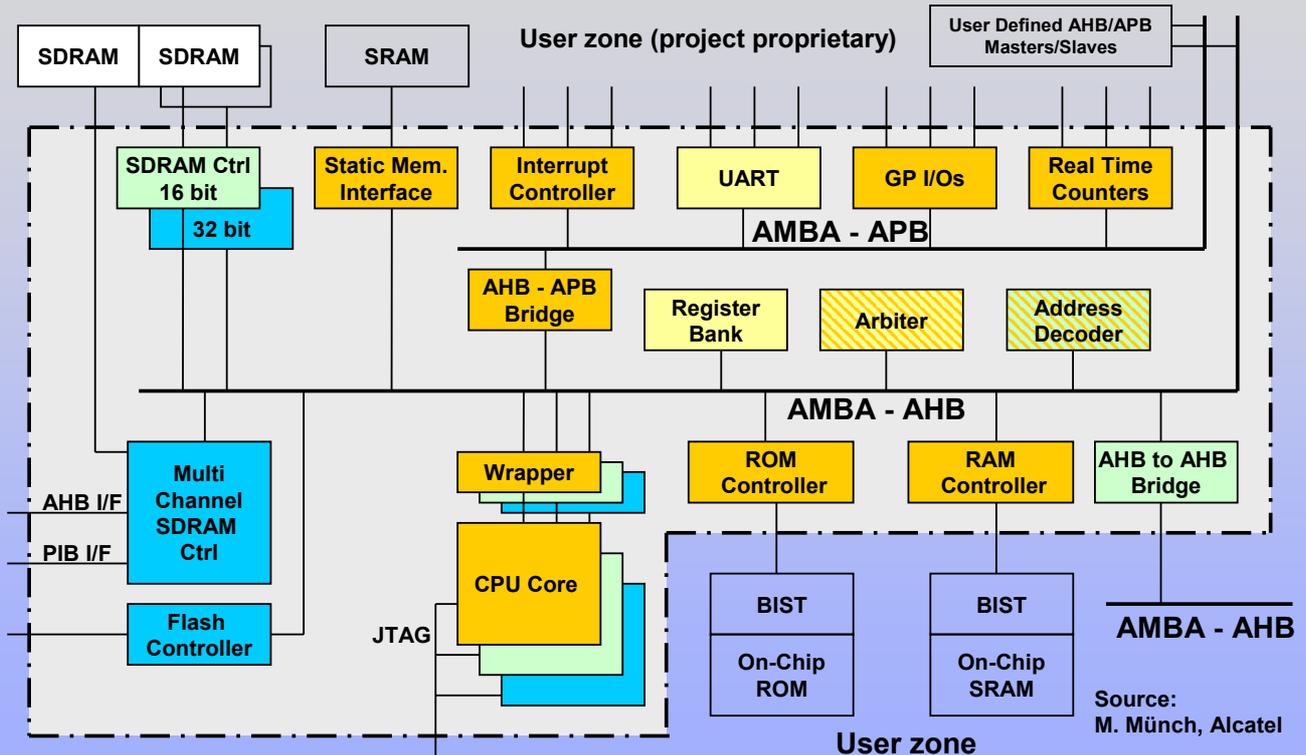


VIPER Chip Layout



- VIPER Hardmacros (zugeliefert)
- VIPER anpassbare Softmacros „Chiplets“

Noch ein Beispiel: Alcatel MpSoC



Source:
M. Münch, Alcatel

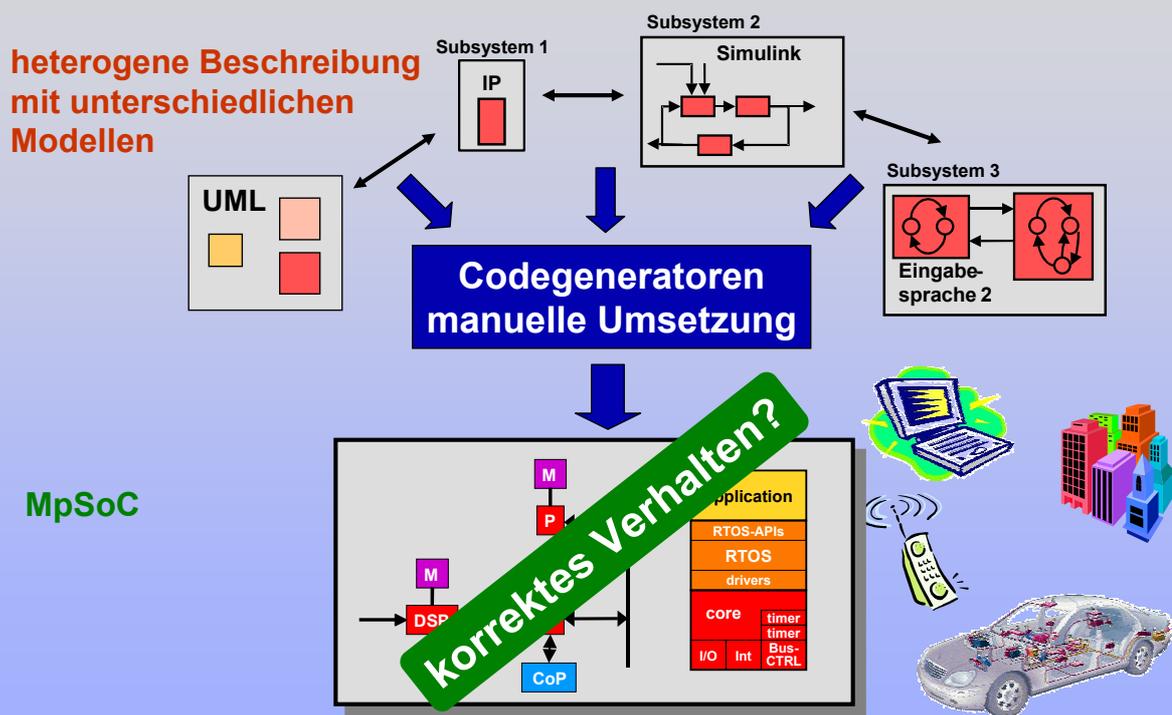
MpSoC sind heterogen strukturiert

- **heterogene Hardware**
 - unterschiedliche Hardwarekomponenten und Kommunikationskanäle
 - heterogene Hardware-Architektur
- **heterogene Software**
 - anwendungsspezifische Softwarebibliotheken (Entwurfsproduktivität!)
 - (verteiltes) Betriebssystem
 - Basissystem zur Kommunikation
- **komplexe, mehrschichtige HW/SW-Architektur**

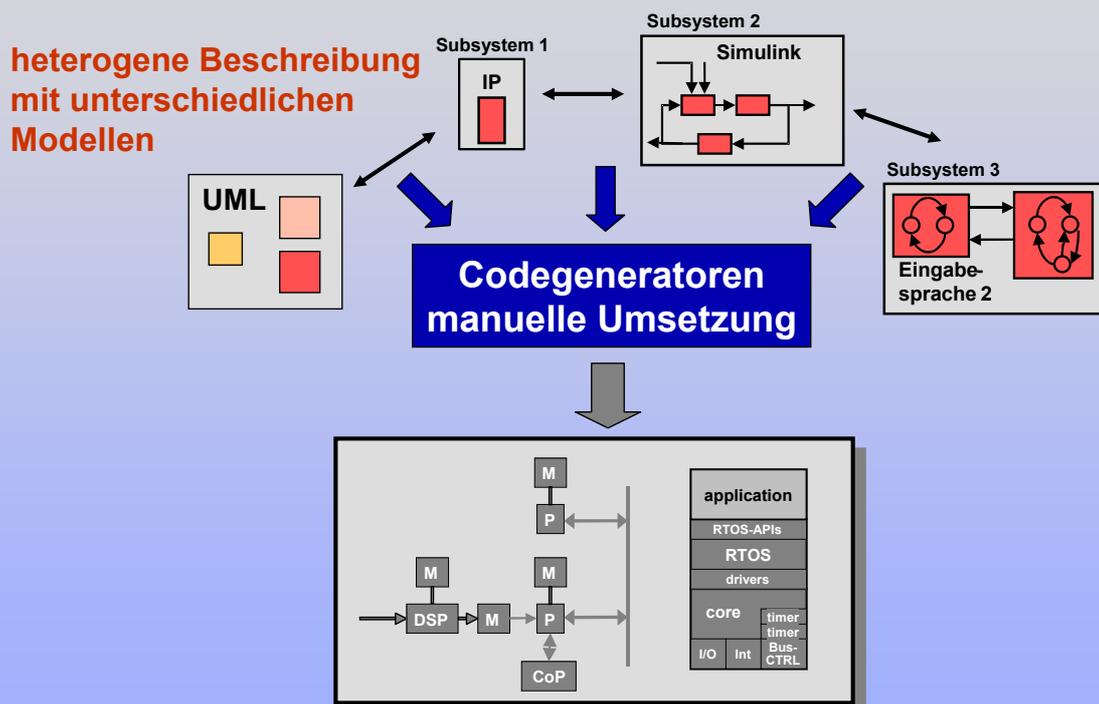
MpSoC-Anwendungen sind heterogen

- **Zusammenfassung mehrerer Funktionstypen auf einem MpSoC**
 - Signalverarbeitung
 - Kommunikationsprotokolle
 - Benutzerschnittstelle
 - Benutzerprogramme (Spiele, Internet-Browser, ...)
 - Wartung- und Update
- **unterschiedliche mathematische Formalismen zur Beschreibung und Optimierung**
 - Signalflussgraphen unterschiedlicher Bedeutung, Automatenetze, C-Programme, Petri-Netze, ...

Entwurfsprozess



Kombination von Sprachen

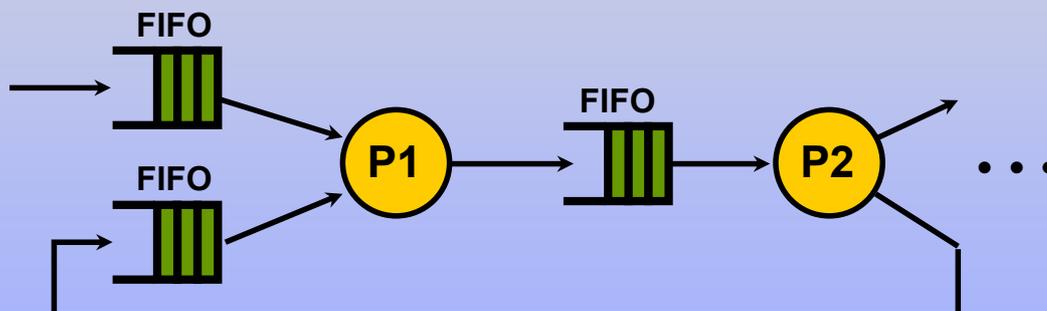


Wichtige Formalismen der Anwendungen

- Formalismen der Anwendung → verwandte Formalismen zur Optimierung und Verifikation
- Beispiele:
 - Signalflussgraph → *Kahn Prozessnetze* für optimierte Ausführungsreihenfolge (Scheduling)
 - reaktive Systeme, z.B. Kommunikationssysteme in Sprachen Statecharts, SDL, Esterel, ... → *Automatennetze*
 - periodisch ausgeführte Flussgraphen (Simulink) → *periodische Prozesse* mit Kommunikation über Register

Kahn Prozessnetze

- Spezielle Klasse von Prozessnetzen (verwandte Klasse Petri-Netze)
- Kommunikation über Warteschlangen mit unbegrenzter Kapazität



Kahn Prozessnetze - 2

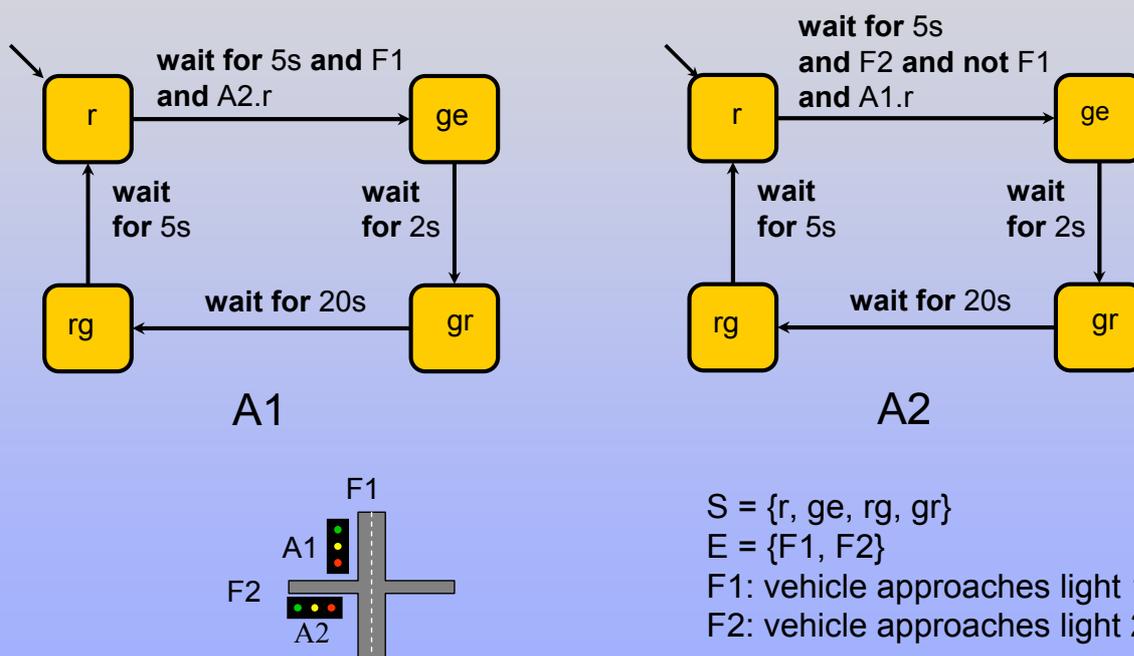
- Ein Kahn Prozess ist **monoton**, wenn

$$F(x_1 x_2 \dots x_n x_{n+1}) \supseteq F(x_1 x_2 \dots x_n)$$

d.h., die Ausgabezeichenkette wächst mit der Eingabezeichenkette
(Beispiel: diskreter Filter, Gegenbeispiel: Sortierverfahren)

- Netze, die nur monotone Kahn Prozesse enthalten, sind funktional **unabhängig von der Ausführungsreihenfolge**
⇒ exzellent für Systemoptimierung nutzbar
- kommerzielle Entwurfswerkzeuge: COSSAP, SPW, DSP Station, ...

Reaktive Systeme - einfaches Beispiel

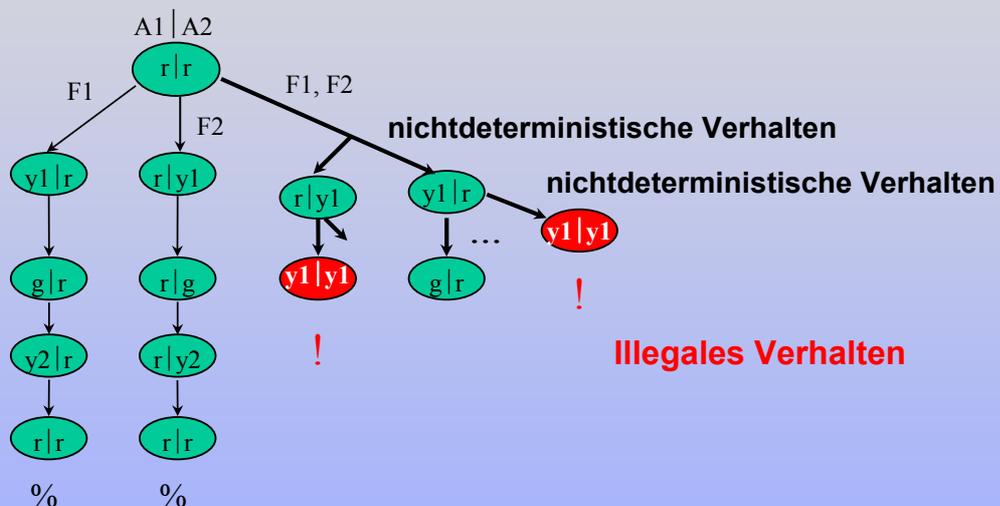


Ampel - Implementierungsbeispiel

- Software : n Befehlszyklen + Signalkommunikation
 - Hardware: 1 bis n Taktzyklen + Signalkommunikation
- ⇒ Automaten-Übergangszeit abhängig von der Implementierung
- ⇒ nichtdeterministisches Verhalten

Kooperation der Automaten

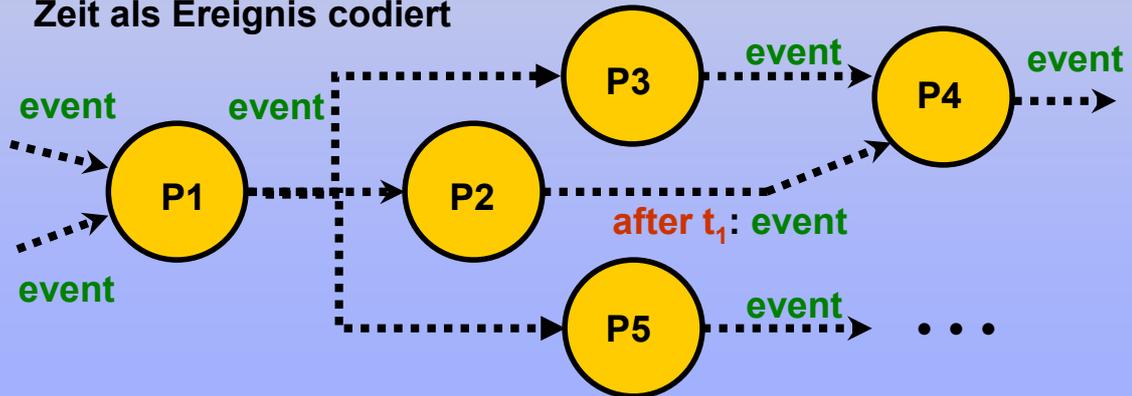
- Erreichbarkeit von Zuständen



- ⇒ Erweiterung des Verhaltens erforderlich
- Semaphore, Monitor, ...

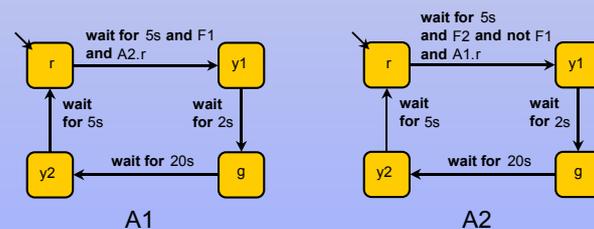
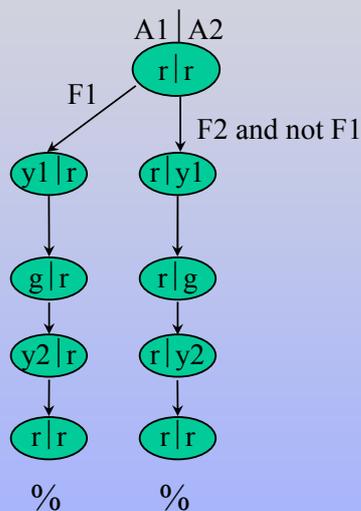
Synchronous Reactive Systems

- Automaten kommunizieren über **events** mit $t_{ev} \rightarrow 0$ (synchron)
- Automatenprozess aktiviert durch jedes Ereignis (**“OR”**)
- Automatenprozess reagiert unverzüglich
- Ausgabedaten unverzüglich verfügbar
- Zeit als Ereignis codiert



Ampel als synchrones reaktives System

- Erreichbarkeitsgraph

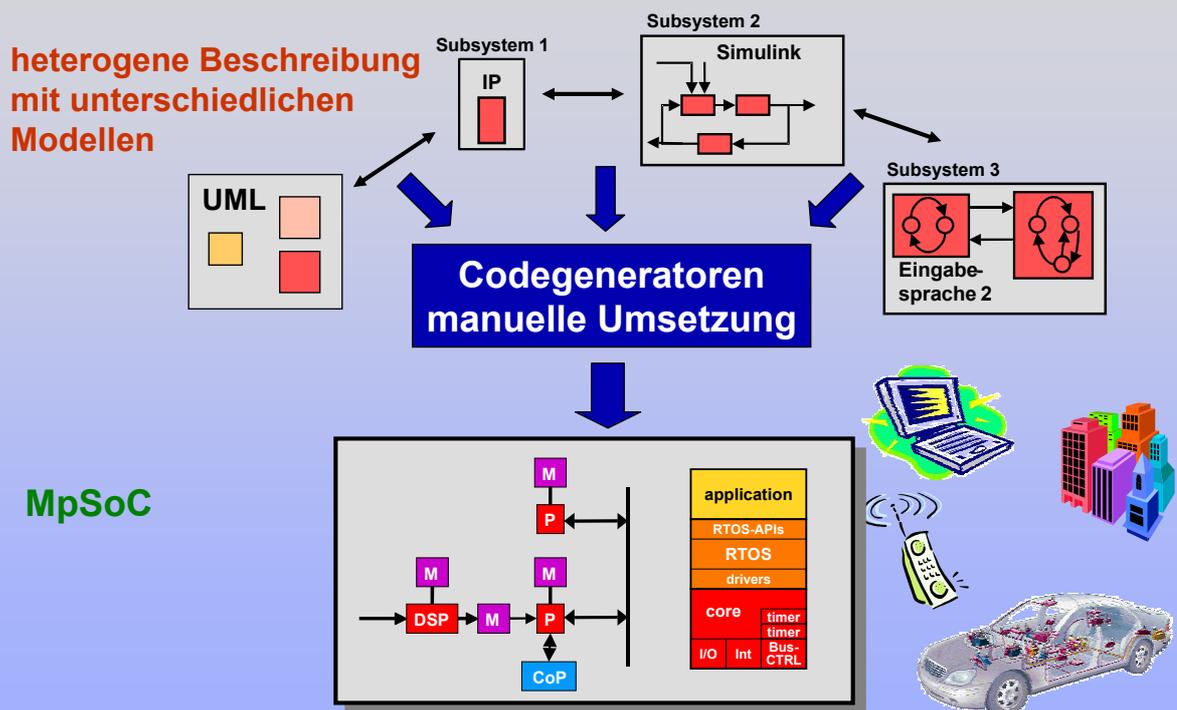


- Implementierung unterliegt komplexen Einschränkungen!

Kombination?

- Kombination verschiedener Modelle besonders kompliziert
- Herausforderung im Entwurf von Kraftfahrzeugen, Mobiltelefonen, Medizinelektronik, ...
- formaler Nachweis der Korrektheit sehr schwierig und praktisch ungelöst
- heute Untersuchung vieler "Fallstudien" durch Simulation
- aktives Forschungsthema

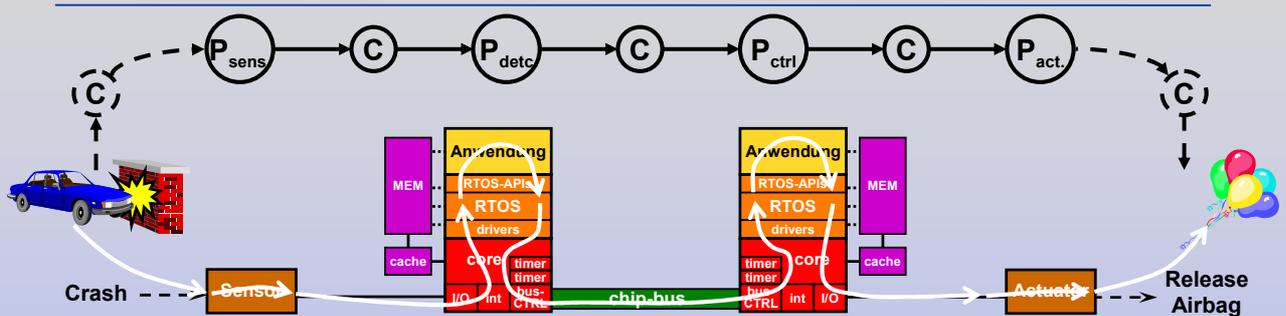
Implementierung auf MpSoC



Korrektes Verhalten

- Kriterien für korrekte Implementierung
 - korrekte Umsetzung der spezifizierten Funktion
 - ausreichende Prozessor- und Kommunikationsleistung
 - Einhaltung von Zeitbedingungen
 - kein Speicherüberlauf
 - kein Blockieren

Anwendungsbeispiel

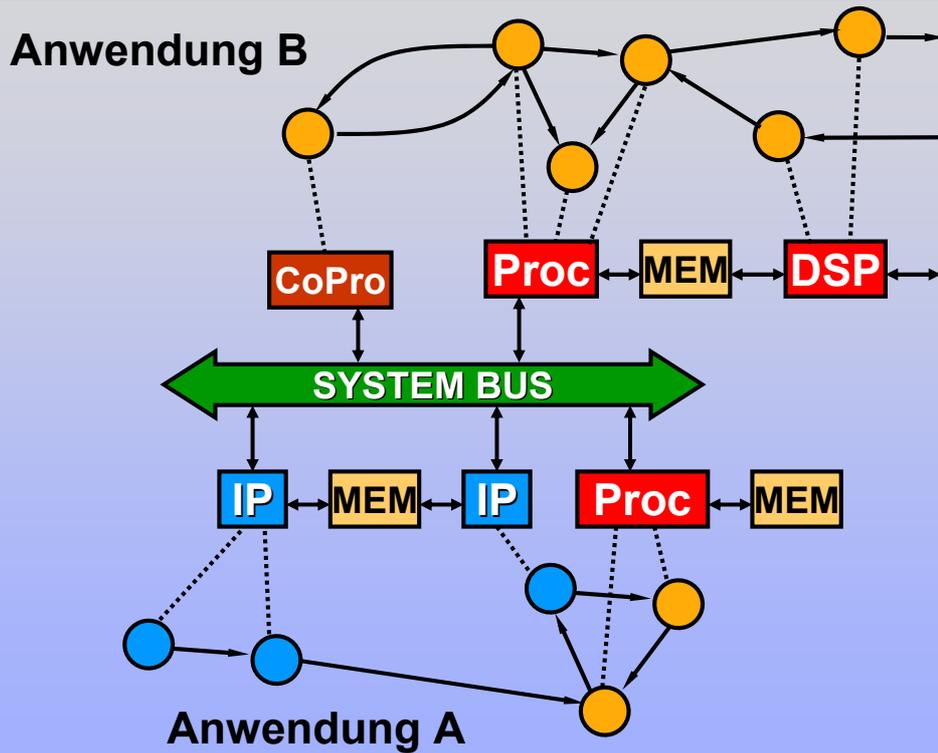


Reaktionszeit?

$$\underbrace{t_{\text{crash}} + t_{\text{sens}} + t_{\text{csens}}}_{\text{physikalische Verzögerung}} + t_{\text{detc}} + t_{\text{fbus}} + t_{\text{ctrl}} + t_{\text{cact}} + \underbrace{t_{\text{act}} + t_{\text{airbag}}}_{\text{physikalische Verzögerung}}$$

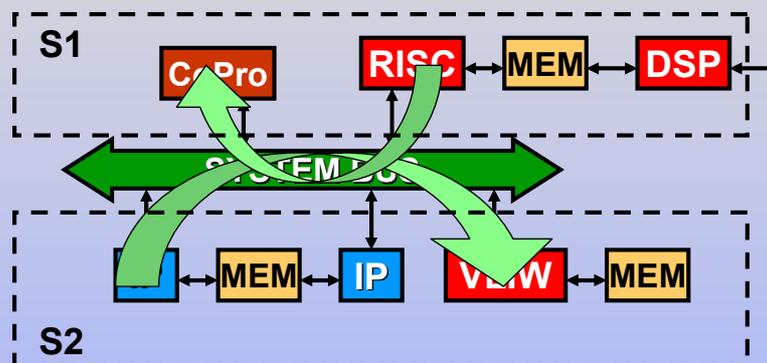
SW-Architektur als Teil der Reaktionskette

Analyse heterogener Systeme



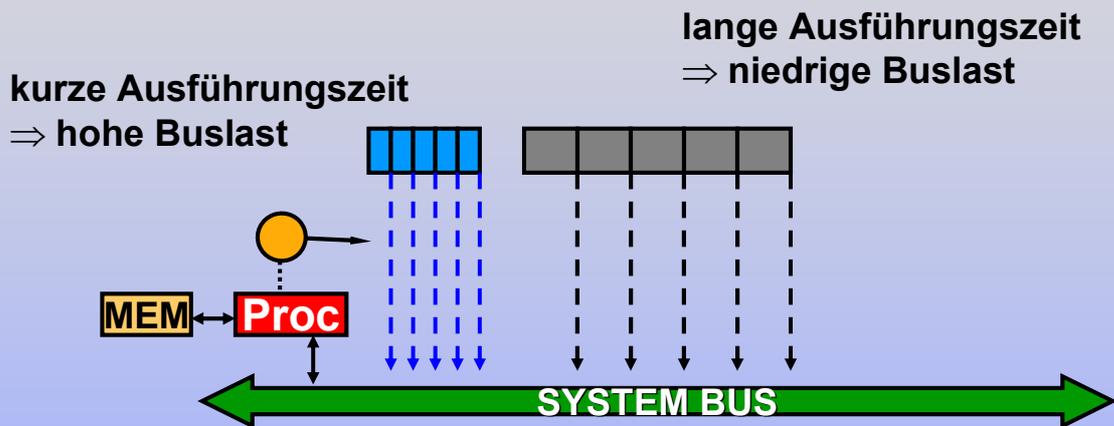
Probleme der Analyse

- komplexe, nicht-funktionale Abhängigkeiten



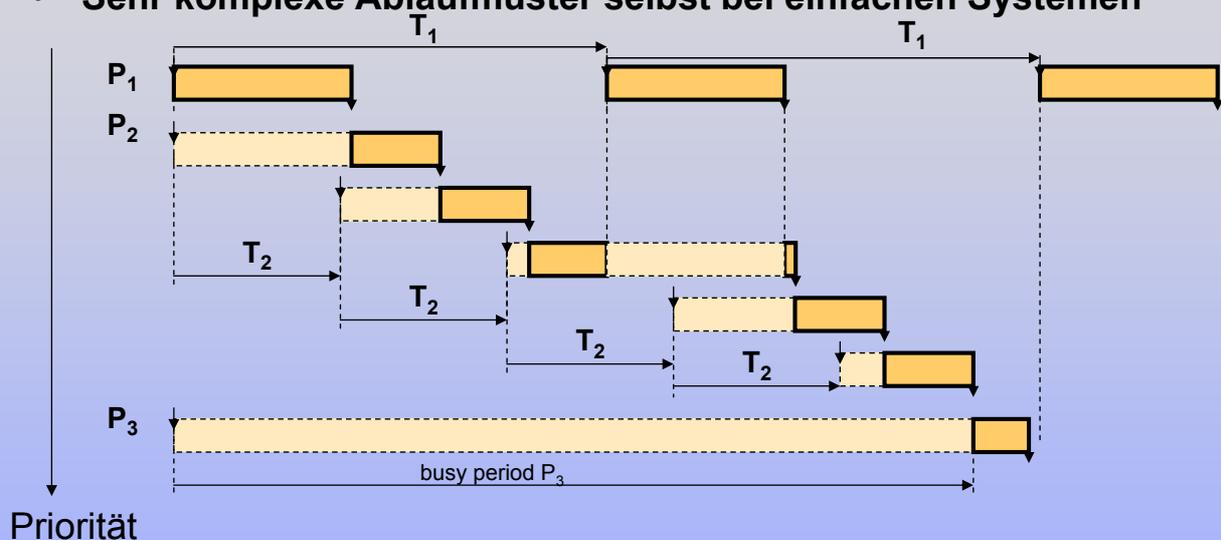
- Beispielprotokoll für Bus: Round-Robin
 - Zeitscheibenverfahren
 - ungenützte Zeitscheiben werden freigegeben
 - ⇒ gegenseitige Beeinflussung

Beispiel für Abhängigkeiten



Beispiel für Ablauf

- Sehr komplexe Ablaufmuster selbst bei einfachen Systemen



- Prozesse mit Unterbrechung und statischen Prioritäten

Analyse von MpSoC - heute

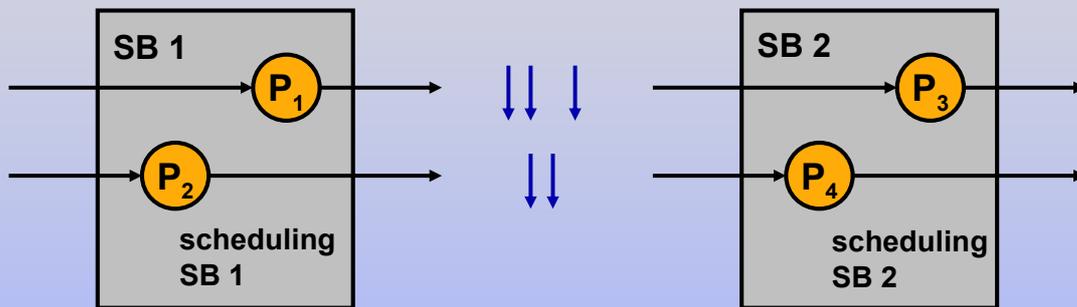
- **typischer Ansatz heute:**
Entwurf von Subsystemen mit nachträglicher Integration m.H. von Prototyping bzw. Simulation
 - **Probleme:**
 - Identifikation von kritischen Fällen
 - Behandlung komplexer Ereignisfolgen
 - Einbindung teilweise spezifizierter Komponenten und „Legacy Software“
 - Behandlung von Varianten
- ⇒ **sehr hohe Kosten (ITRS 99: Systemverifikation erzeugt 40% der Verifikationskosten)**

Formale Analyse

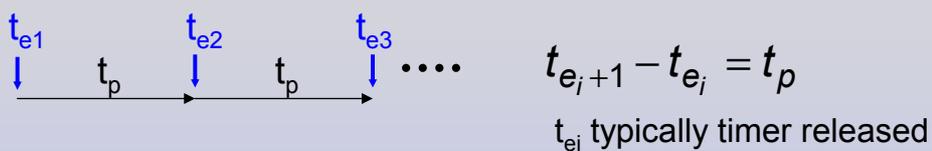
- **Aktives Forschungsthema im Bereich der “Echtzeit-Systeme”**
- **formale Verfahren für einheitliche Subsysteme bekannt**
- **Problem: Integration heterogener Systeme**
- **bekannte Ansätze**
 - **Erweiterung der Analyse auf komplexe Systeme**
⇒ nur für speziell Fälle sinnvoll
 - **hierarchische Analyse**

Kopplung über Ereignisströme

- Subsysteme sind über **Ereignisströme** gekoppelt



Beispiele für Ereignismodelle

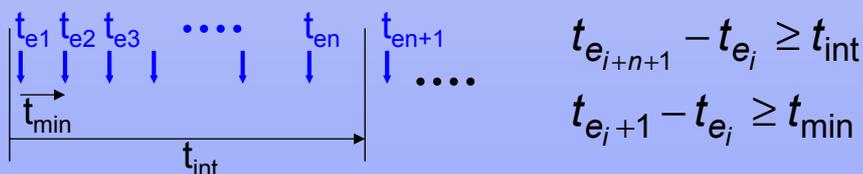


- Periodische Ereignisse mit Jitter**



- Ereignisse mit Minimalabstand**

– **Burst-Ereignisse, sporadische Ereignisse, etc.**



Anpassung von Ereignismodellen

- **Anpassung für kompatible Eingangs- und Ausgangsereignisse**
 - mathematische Transformation zur Adaption der Analyse
- **Anpassung für inkompatible Eingangs- und Ausgangsereignisse**
 - einfügen von Pufferung zur Anpassung der Ereignisströme erforderlich
 - minimale Puffergröße bestimmbar

Was macht die TU in diesem Bereich?

- **Hardware und Software künftiger Chip-Generationen**
 - Theoretische Forschung und industrielle Anwendung
- **Entwurfsmethoden und -prozesse**
 - wie entwerfe ich effizient korrekte Chips
 - Hochleistungsprozessoren für den Elektronischen Film
 - Thomson, ...
 - Netzwerkprozessoren
 - Intel
 - Luft- und Raumfahrt
 - Astrium
- **Internationale Kooperation** mit namhaften Firmen und Universitäten (ETH Zürich, Princeton, Univ. of California, Georgia Institute of Technology, ...)

Beispiele

- **Hochleistungsprozessoren für den Elektronischen Film**
 - Thomson
- **Netzwerkprozessoren**
 - Intel
- **Luft- und Raumfahrt**
 - Astrium, NASA
- **Kraftfahrzeuge**
 - Volkswagen

Beispiel elektronische Filmbearbeitung

- **Hochauflösende Abtastung (digitaler Master)**
- **Bearbeitung der digitalen Bilddaten**

Aufnahme auf Film



Filmabtastung
In hoher Auflösung

Bearbeitung der Bilddaten



Filmaufzeichnung



Projektion

Zusammenfassung

- **Exponentielle Zunahme der Schaltungskomplexität hält an**
- **erforderliche Entwurfsproduktivität nur durch Wiederverwendung und strukturierte Integration erreichbar**
- **Spezialisierung und Wiederverwendung führen zu heterogenen Hardware/Software-Systemen**
- **die korrekte Integration derartiger Systeme ist zum dominierenden Problem geworden**
- **Simulation nicht mehr ausreichend**
- **formale Methoden erlauben eine zuverlässigere Integration**