Zielarchitekturen im Hardware/Software Coentwurf

R. Ernst

Institut für Datenverarbeitungsanlagen
Technische Universität Braunschweig

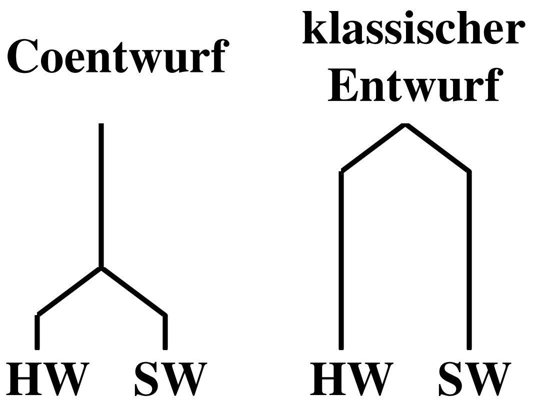
0 Inhalt

1 Einleitung

- **Herausforderungen im Entwurf eingebetteter Systeme**
  - **Wachsende Komplexität der Anwendungen selbst in alltäglichen Massenprodukten**
    - **umfangreiche Systeme mit historisch gewachsener Funktionalität**
    - **Kombination von reaktiven und datenflußorientierten Systemen**
    - **Anforderungen an Flexibilität**
    - **Beispiele: Multimedia, Kfz, Mobilkommunikation**
  - **Wachsende Komplexität der Zielsysteme**
    - **Kombination verschiedener Technologien, Komponenten und Entwurfsstile**
    - **große integrierte Systeme ("Systems-on-a-Chip"), die Komponenten aus verschiedenen Quellen kombinieren (IP-Markt)**
  - **Vielzahl an Randbedingungen und Entwurfszielen**

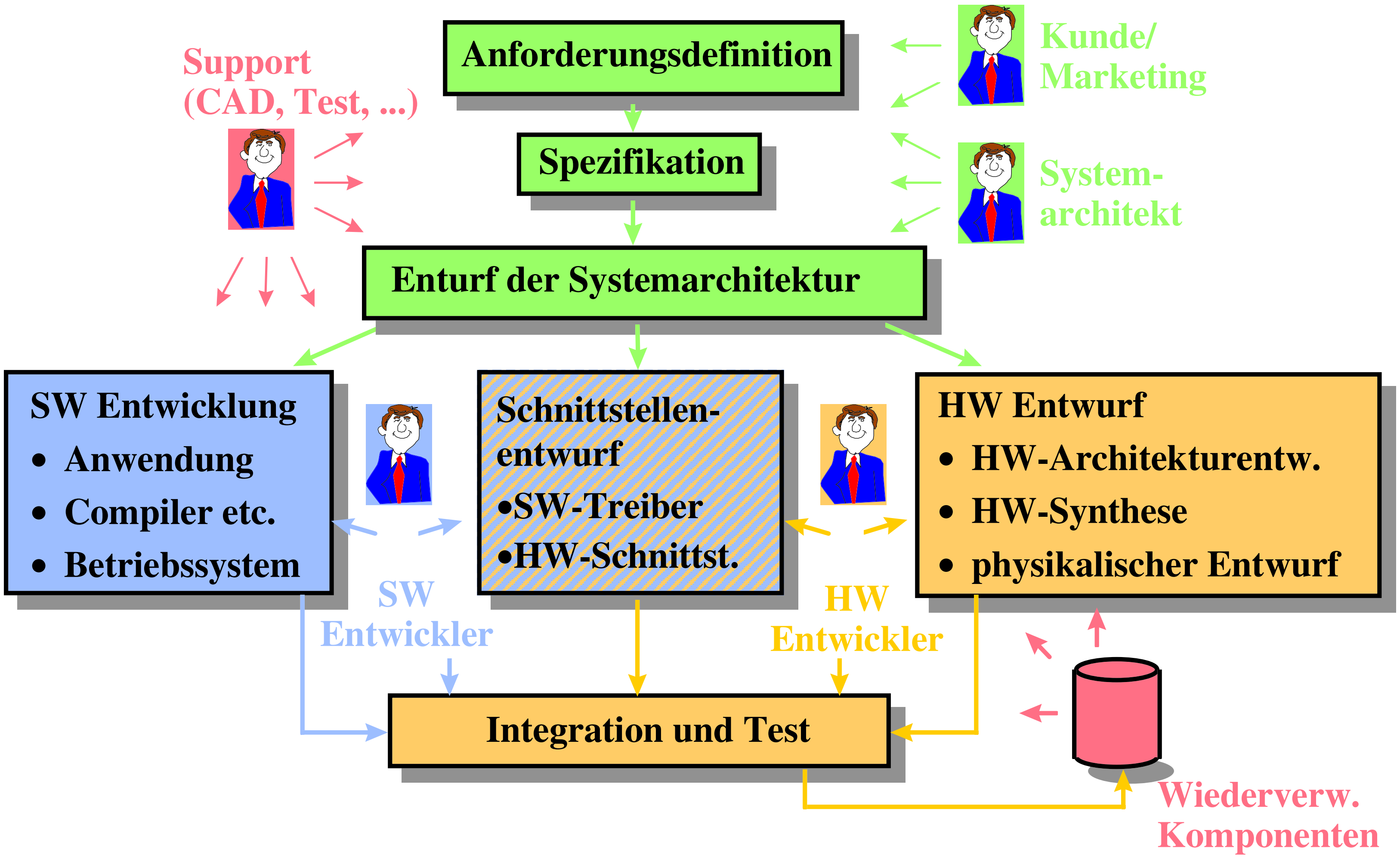
    Beispiele: Kosten, Verlustleistung, Zeitbedingungen, Zuverlässigkeit
  - **Verkürzte und überlappende Entwurfszyklen**

Hardware-/Software Coentwurf



- **Hardware-/Software Coentwurf: Gemeinsamer Entwurf von Hardware und Software**
- **Ziele**
  - **Optimierung des Entwurfs*prozesses***
    - **⇒ Erhöhung der Entwurfsproduktivität**
  - **Optimierung des Entwurfs**
    - **⇒ Erhöhung der Entwurfsqualität**
- **Aufgaben**
  - **Cospezifikation und Comodellierung**
  - **Coverifikation**
  - **Entwurfsintegration und Entwurfsmethodik**
  - **Entwurfsoptimierung und Cosynthese**

2 Entwurfsprozeß für eingebettete Systeme



Anforderungsdefinition

Spezifikation

Enturf der Systemarchitektur

Support (CAD, Test, ...)

Kunde/ Marketing

System-architekt

SW Entwicklung
- Anwendung
- Compiler etc.
- Betriebssystem

Schnittstellenentwurf
- SW-Treiber
- HW-Schnittst.

HW Entwurf
- HW-Architekturentw.
- HW-Synthese
- physikalischer Entwurf

SW Entwickler

HW Entwickler

Integration und Test

Wiederverw. Komponenten

Beobachtungen im Entwurfsprozeß

- **Zunehmend nebenläufiger Entwurf von Hardware und Software mit teilweise noch unvollständiger oder variabler Spezifikation**
  - ⇒ **enge und permanente Kooperation zwischen Hardware- und Softwareentwicklern, Systemarchitekten und Kunden/Marketing gefordert**
- **Enge Zeitfenster für Markteinführung erfordern einen sicheren ("first-time-right") Entwurfsprozeß**
  - ⇒ **frühe Entdeckung systematischer Entwurfsfehler entscheidend**
  - ⇒ **verläßliche Entwurfszeiträume und genaue Vorhersage der Entwurfsdaten ist wichtiger als Minimierung der Entwurfszeit** ***Voraussetzung: genaue Schätzung - heute: Erfahrungswerte***
  - ⇒ **zuverlässige Optimierungsschritte mit geringem Risiko**

Beobachtungen im Entwurfsprozeß *Forts.*

- **Erhöhte Entwurfsproduktivität durch Wiederverwendung von Komponenten und Funktionen**
  - **⇒ Funktions- und Komponentenbibliotheken**
  - **⇒ Problem: Entwurfsmigration zwischen verschiedenen Technologien, Komponenten (Prozessortypen), zwischen Hardware und Software heute *nicht unterstützt***

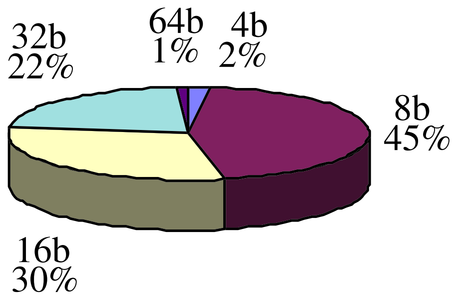
⇒ Reuse von Strukturen
+ konservativer Entwurf basierend auf Erfahrungswerten
+ fehlende Transformationen

⇒ *Beschränkung der Entwurfsoptimierung*

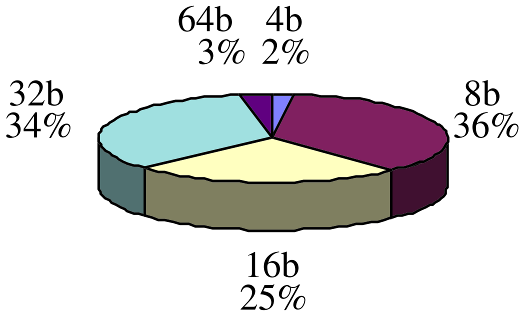
3 Zielarchitektur

Wird denn nicht ohnehin alles im RISC-Prozessor enden?

Abgeschlossene Entwicklungen (Jan 96)



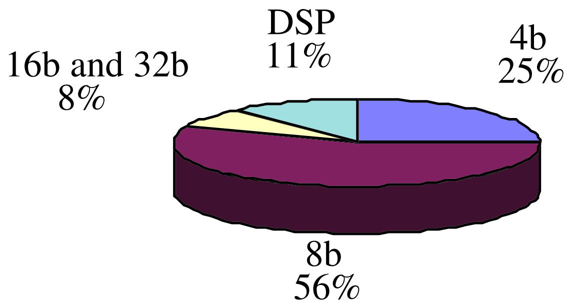
Anzahl begonnener Entwicklungen (Jan 96)



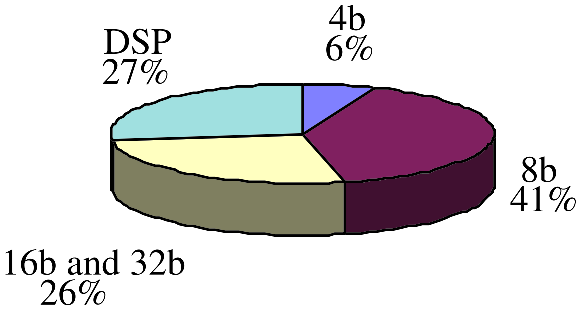
Entwicklungen

(Quelle: Electronic Design: reader survey, 96)

Mikrocontroller 1993 insges. $6.8Mrd.



Mikrocontroller 2000 insges. $25.9Mrd.



Verkaufszahlen

(Quelle: ICE 95)

Anforderungen an die Zielarchitektur

- **Ausnutzung des Spezialisierungspotentials**
    - **A priori definierte Menge von Aufgaben erlaubt Spezialisierung der Architektur**
    - **Spezialisierung wird beschränkt durch**
        - **Eigenschaften des zu spezifizierten Systems**
        - **Anforderungen an Flexibilität (Adaptierbarkeit, Wiederverw.)**
        - **Nichtfunktionale Randbedingungen (Beispiele)**
    - **Vielfalt an gegenwärtigen Architekturen mit unterschiedlichem Spezialisierungsgrad:**

Systematik fehlt

Anforderungen an die Zielarchitektur *Forts.*

- **Bewahrung der geforderten Flexibilität**
  - **Ziele der Flexibilität**
    - **Zulassen später Änderungen**
    - **Planung der Wiederverwendung**
    - **Überdeckung eines Marktsegments (u. a. domänenspezifische Systeme: Beispiele: DSP, Mikrocontroller)**
  - **Analyseverfahren für Flexibilität nur für einfache Fälle und sehr beschränkte Architekturklassen bekannt**
- **Transformation zwischen verschiedenen Architekturen**
  - **Erforschung des Entwurfsraums (Systemarchitekt + Marketing), Entwurfsmigration**
  - **es fehlen Systematik und Verfahren für anwendungsspezifische Architekturen**

Zielarchitekturen im rechnergestützten, optimierenden Coentwurf - Stand der Technik

- **Prozessor-Coprozessor**
  - **Beispiele: VULCAN, TOSCA, COSMOS, PTOLEMY (Kalavade), Arbeiten von Edwards, CHINOOK...**
  - **Beiträge:**
    Optimierung der Kommunikation, Kommunikationssynthese, Lastverteilung (Partitionierungsproblem), Codegenerierung
  - **feste Struktur der Steuerung und der Kommunikation**
- **Anwendungsspezifische Prozessoren (ASIPs)**
  - **RISC-Basis: Beispiel: CASTLE**
  - **DSP-Basis: Beispiele: SGS Thomson, Philips (EPIC), Trimedia (VLIW)**
  - **Beiträge:**
    Optimierung des Befehlssatzes (Vorbild: quantitativer Entwurf von Standard-RISC-Prozessoren), Optimierung der Funktionseinheiten

Zielarchitekturen im optimierenden Coentwurf *Forts.*

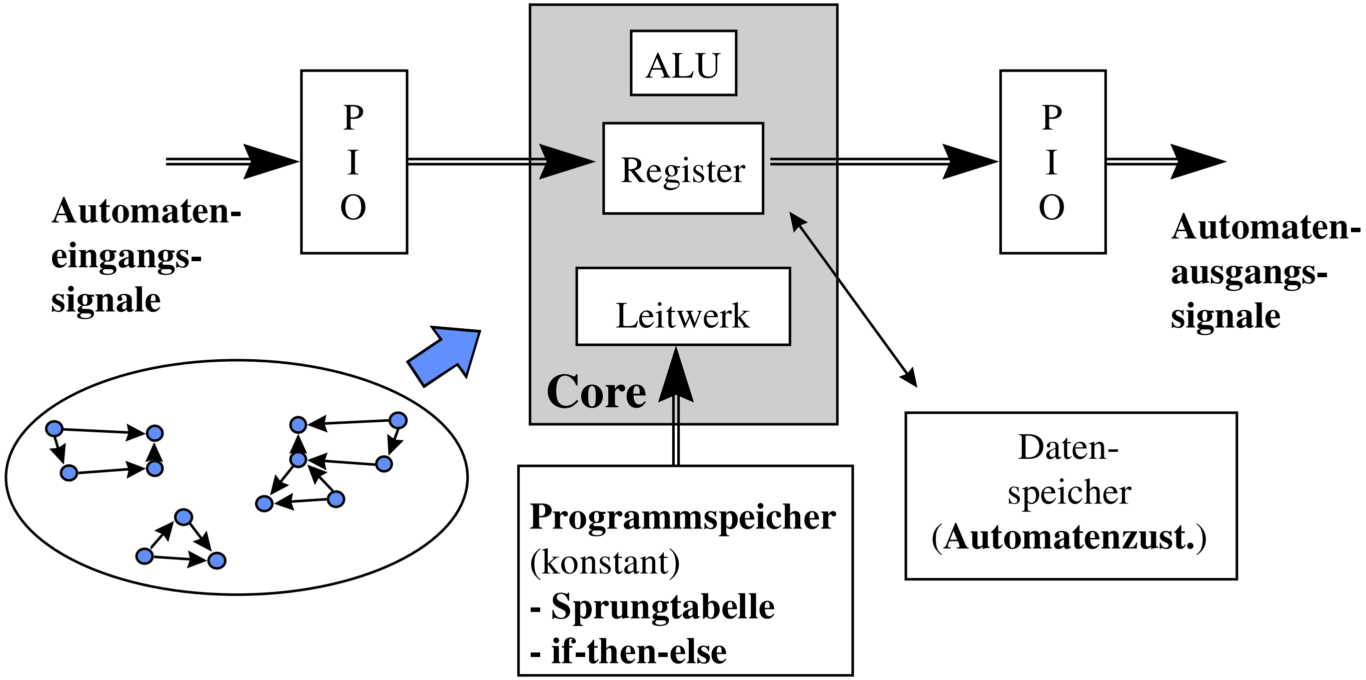
- **Multi-DSP**
  - **Beispiel: Prophid (Philips)**
  - **feste Kommunikationsmechanismen, feste Struktur der Steuerung, festes Speichermodell**
  - **Beiträge:**
    Optimierung der Kommunikation, Speicheroptimierung, Lastverteilung
- **Allgemeinere, benutzerdefinierte Architektur mit vorgegebenen Kommunikationskanälen**
  - **Beispiele: CoWare, Arbeiten von Eles, COSYMA, ...**
  - **Einschränkung der Kommunikationsmodelle, der Steuerung und der Speicherarchitektur**

⇒ Insgesamt:
Große Einschränkungen bei Architekturen und Spezialisierung

4 Systematische Nutzung des Entwurfsraums

- **Analyse des Spezialisierungspotentials einer Anwendung**
- **Transformations- und Optimierungsverfahren für einen weiteren Bereich an Architekturen**
- **Bewertung von gefundenen Lösungen**
- **Schwerpunkt des Vortrags**
  - **Systematik der Spezialisierung unter Berücksichtigung von Anforderungen an Flexibilität**
  - **Analyse und Klassifizierung manueller Entwürfe als Ausgangsbasis**

Beispiel 1: Prozessoren für steuerdominierte Anwendungen: Prinzip



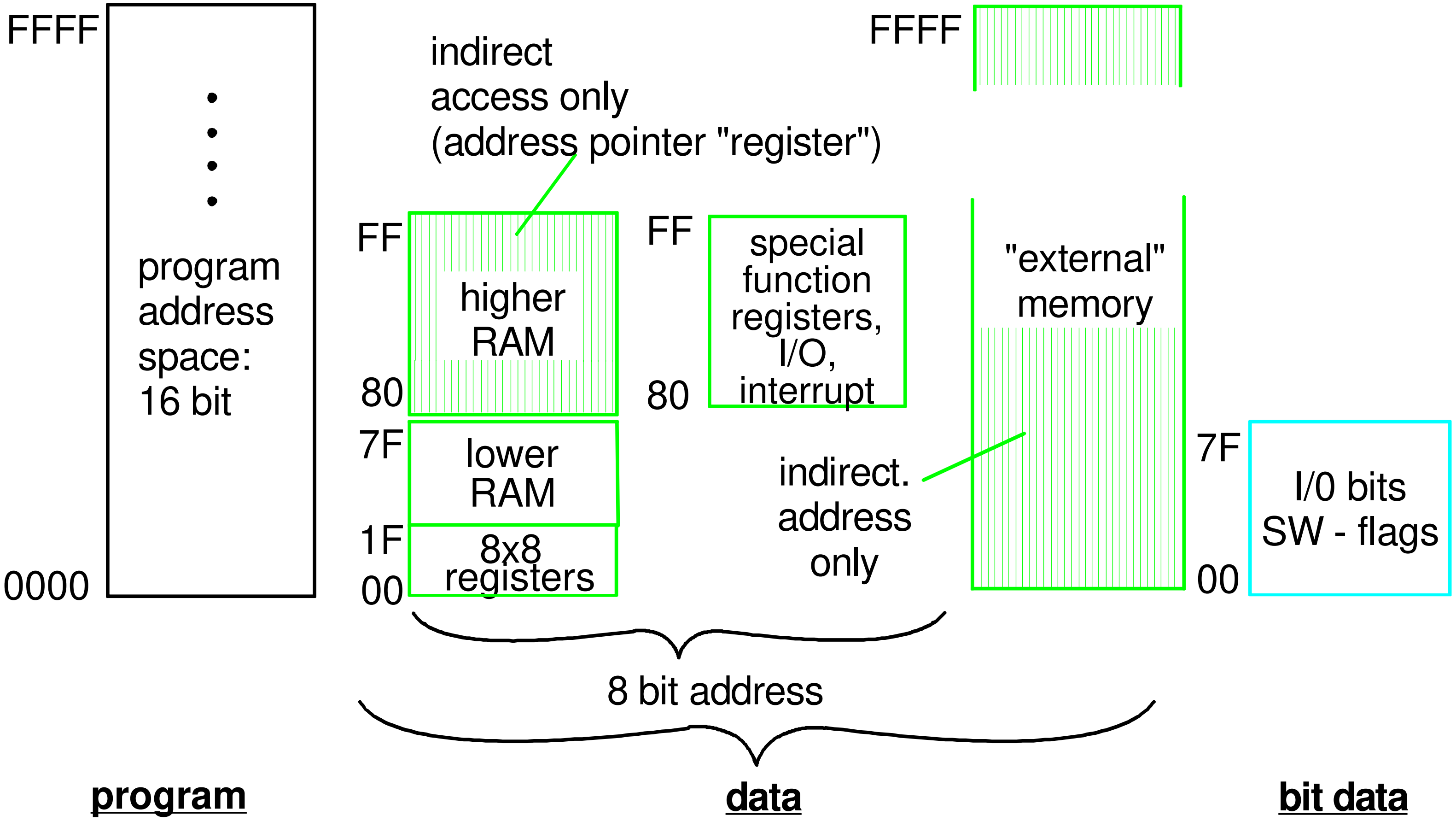
Beispiel 1: Prozessoren für steuerdominierte Anwendungen: *Forts.*

- **Ansätze zur Spezialisierung**
  - **Vorwiegend boolsche Operationen**
  - **wenige Operationen pro Zustandsübergang**
  - **kleiner Datensatz**
  - **häufige, schnelle Prozeßumschaltung**

Beispiel 1: Prozessoren für steuerdominierte Anwendungen: Zielarchitektur

- **8051 - 8 bit Prozessor**
  benutzt in reaktiven Systemen mit geringen Anforderungen an Rechenleistung
  - **Akkumulatormaschine, CISC-Typ, 8 bit Register**
  - **4 Bänke mit 8 Registern, Umschaltung mit Interrupt und Unterprogrammaufruf**
  - **Speicher- und Registeraddressierung (einschl. m-m)**
  - **direkter Zugriff und Verzweigung auf einzelne I/O bits**
  - **Verzweigung relativ zu einem Datenadreßregister**
    ⇒ unterstützt Verzweigungstabellen
  - **komplexer überlagerter Adreßraum mit 8 bis 16 bit Adressierung**
  - **dichte Codierung: 1-3 Bytes/Befehl**

8051 Mikrocontroller: Adreßraum



FFFF

program address space: 16 bit

0000

indirect access only (address pointer "register")

FF

higher RAM

80

7F

lower RAM

1F

8x8 registers

00

FF

special function registers, I/O, interrupt

80

indirect. address only

FFFF

"external" memory

8 bit address

7F

I/0 bits SW - flags

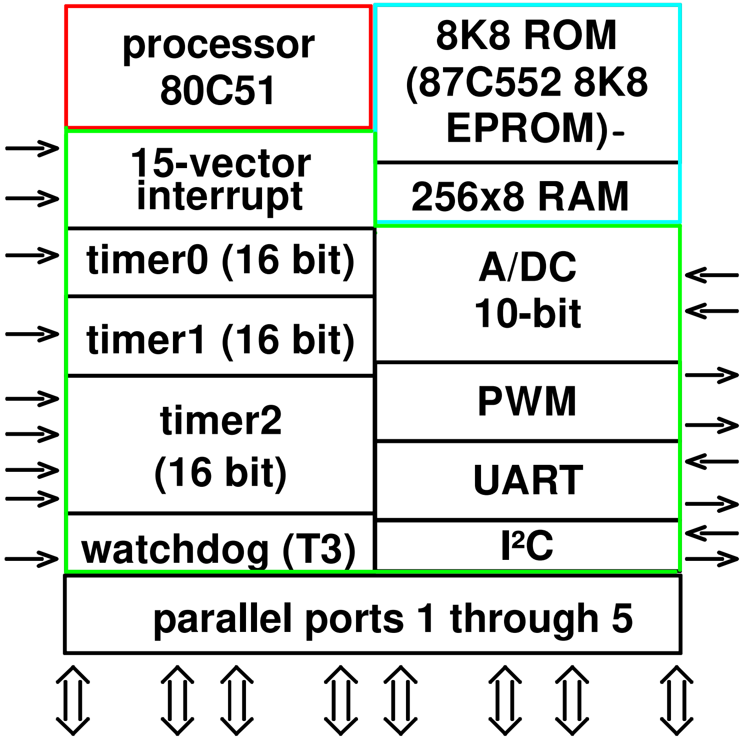
00

program

data

bit data

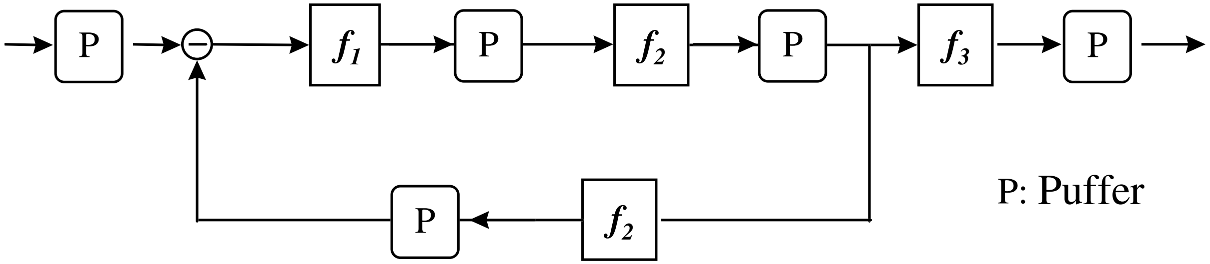
8051 Mikrocontroller: Vollständiges System



Philips 80C552: 8051 basierter Mikrocontroller

Beispiel 2: Prozessoren für datenflußorientierte Anwendungen: Prinzip

- **Überwiegend periodisches Verhalten**
- **Eingabebeschreibung: Flußgraphen**



P: Puffer

- **Anwendungsbeispiele: Signalverarbeitung, Regelungstechnik**
  - **typisch hohe Anforderungen an die Rechenleistung (Durchsatz)**
  - **geringere Anforderungen an Latenzzeit**

    ⇒ großer Entwurfsraum (Puffer, ...)
  - **große Variablenmengen**

    ⇒ Speicherarchitektur ist wesentlich

Beispiel 2: Prozessoren für datenflußorientierte Anwendungen: *Forts.*

- Ansätze zur Spezialisierung
  - in weiten Teilen datenunabhängiger Steuerfluß
  - überwiegend periodische Eingangssignale

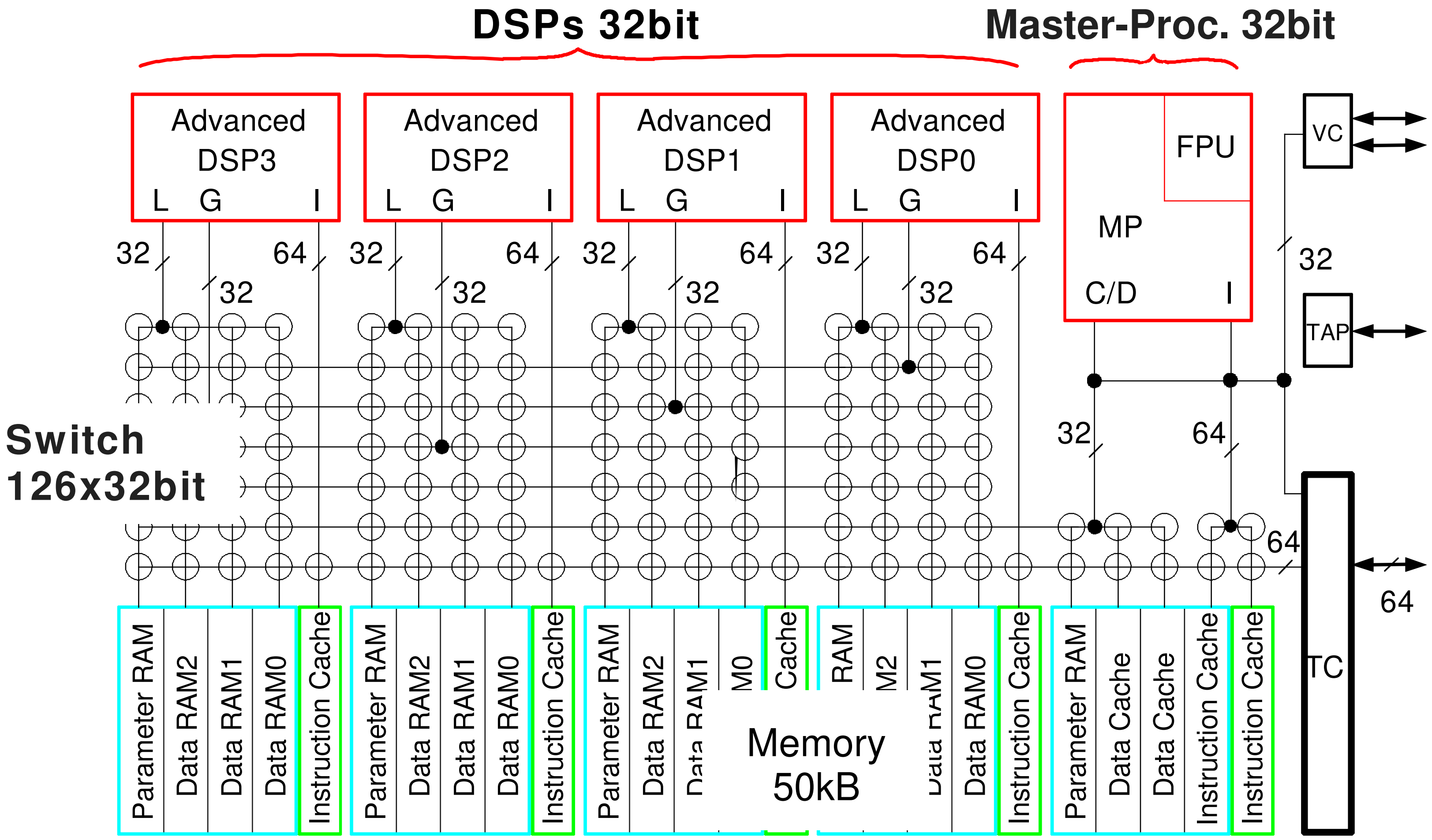
  ⇒ überwiegend statisches Prozeß- und Operationsscheduling für hohe Parallelverarbeitung möglich

  ⇒ Optimierung des Steuerflusses

  ⇒ geringe Anforderungen an Interrupts

  ⇒ reguläre und à priori bekannte Adreßsequenzen

TMS320C80: Multi-DSP für Video-Anwendungen



TMS320C80: Multi-DSP für Video-Anwendungen *Forts.*

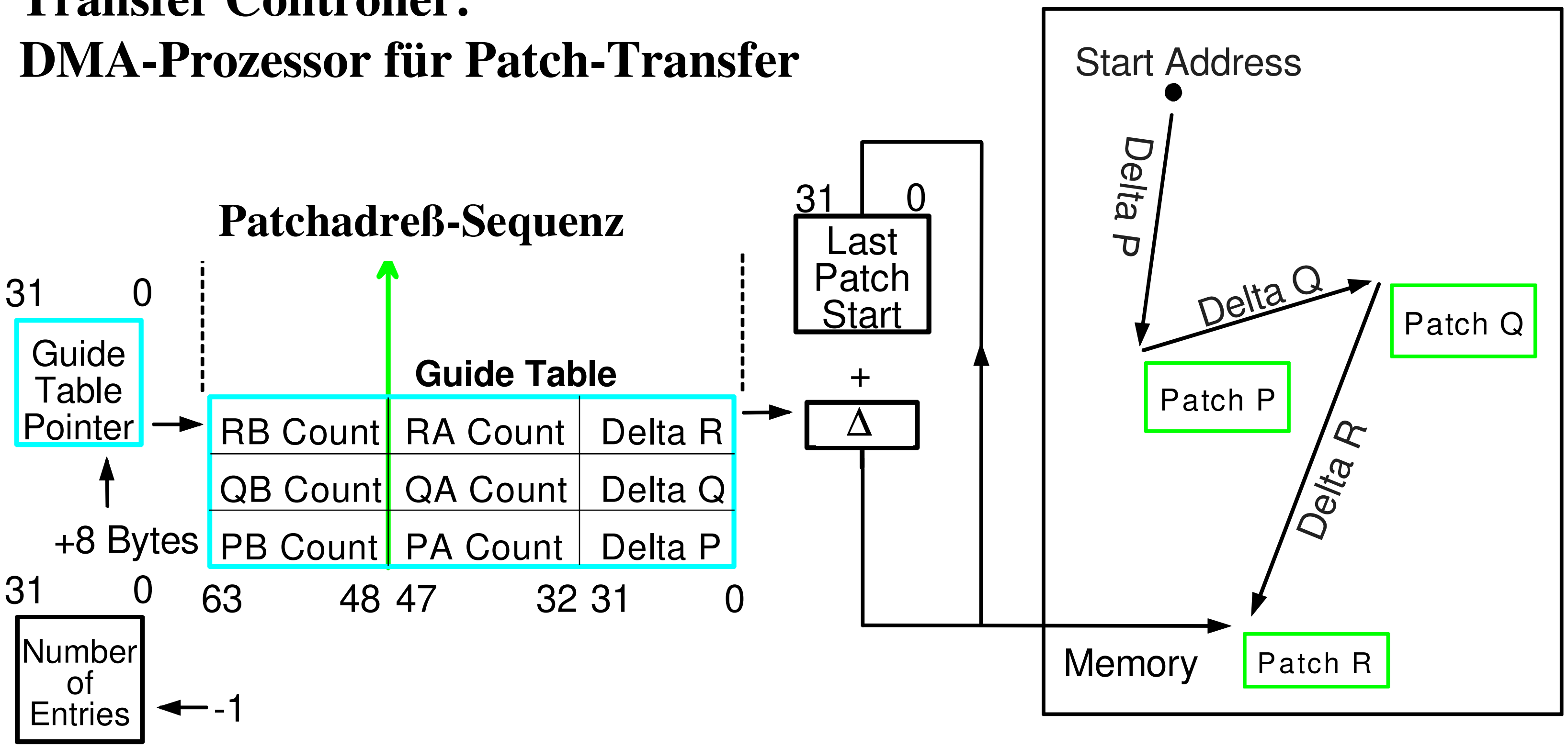
- Steuerprozessor und 4 32bit DSPs
- DSPs:
  - Spezialoperationen für Videoanwendungen (Masken, Pixelexpander)
  - Datenwort in ALU aufspaltbar in 4 Teilworte
  - 2 Adreßrechenwerke für lineare Adreßsequenzen
  - 64bit Befehlswort:
    Operation auf Daten&lokaler Datentransfer&globaler Datentransfer
  - Leitwerk mit Schleifenstack für Schleifentiefe 3, Spezialregister für schnelle Behandlung von 1 Unterprogrammaufruf
  - sehr kleiner Cache (256 Worte)

  ⇒ unterstützt sehr einfache Programme mit hohem Datenvolumen
- Kleine lokale Datenspeicher können nur den Bruchteil eines Videobilds aufnehmen

  ⇒ geschicktes Tiling und leistungsfähiger I/O-Prozessor erforderlich

TMS320C80: Multi-DSP für Video-Anwendungen *Forts.*

Transfer Controller:
DMA-Prozessor für Patch-Transfer



63–48	47–32	31–0
RB Count	RA Count	Delta R
QB Count	QA Count	Delta Q
PB Count	PA Count	Delta P

Schlußfolgerungen aus den Beispielen

- Nutzung von Eigenschaften einer Systemdomäne
  - Anpassung auf größeres Marksegment (Flexibilität)
  - Hoher domänenspez. Spezialisierungsgewinn
  - Spezialisierte Benchmarks (DSP, Mikrocontroller)
  - Problem: sehr unterschiedliche Domänengröße mit verschiedenem Optimierungsgewinn (gemischte Systeme)
- Weit umfangreicheres Repertoire an Spezialisierungstechniken als in ASIPs genutzt
- aber: Codegenerierung für die meisten spezialisierten Architekturen noch manuell

  ⇒ Migration sehr aufwendig

5 Spezialisierungstechniken

- Komponenten
- Systeme
- Kommunikation - Die Infrastruktur

Spezialisierungstechniken für Komponenten

- **Spezialisierter Befehlssatz**
  - **Operator-Chaining**
    - Beispiele: Mult-Akk, Vektoroperationen, Adreßsequenzen
  - **Nutzen von Nebenläufigkeit mit geringen Steuerkosten**
    - Beispiel: nebenläufige Adreßgenerierung in DSPs
    - Kostengewinn in der Befehlscodierung
  - **Auswirkung:**
    Kosten, Programmspeichergröße, Speicherbandbreite, Verlustleistung, Aufwand zur Steuerung

- **Spezialisierte Datenpfade und Funktionseinheiten**
  - **Spezialfunktionen**
  - **Wortlängenadaption**
  - **Auswirkung:**
    Zyklen pro Programm, Verlustleistung
    (aber: evtl. längerer Taktzyklus, kann gewünscht sein: EMV)

Spezialisierungstechniken für Komponenten *Forts.*

- **Spezialisierte Speicherstruktur**
  - **Zahl und Größe der Speicherbänke**
  - **Zahl und Wortlänge der Speicherports**
  - **Zugriffsart:** Multiport, Page Mode, FIFO, Random Access
  - **Auswirkung:**
    Nutzbare Parallelität, Verlustleistung
- **Spezialisierte Verbindungsstruktur**
  - **optimierte Busse und Verbindungen**
  - **Spezialregister**
  - **Auswirkung:**
    Verbindungsaufwand, Steueraufwand, Taktzyklus, Verlustleistung
- **Spezialisierte Steuerung**
  - **Pipelining, spekulative Ausführung, dyn. Befehlsscheduling**
  - **Auslegung des Leitwerks (zentral, dezentral, mikrocodiert, ...)**
  - **Auswirkung:** Nutzbare Parallelität, Kosten, Verlustleistung

Spezialisierungstechniken für Systeme

- **Dekomposition der Steuerung (Steuer-Clusterung)**
  - **Partitionierung der Systemfunktion in lokal gesteuerte Komponenten**
  - **spezialisierte Steuerung der Komponenten**
    - voll programmierbare Komponenten (Prozessoren)
    - Komponenten mit selektierbaren, vordefinierten Steuerequenzen, (FP-Coprozessor, Graphikprozessor, DMA, ...)
    - hart codierte Komponenten

    ⇒ heterogene Multiprozessoren, häufig in eingebetteten Systemen (Mobiltelefon)

    ⇒ Verallgemeinerung der spezialisierten Steuerung
  - **Anwendung globaler Steuerflußtechniken**
    - abhängiger Coprozessor
    - inkrementell gesteuerter Coprozessor (Beispiel: Adreßeinheit im ADSP21060)
    - teilweise abhängige Coprozessoren
(Coprozessoren mit Programmsegment: Transfer Controller, PWM)
    - unabhängige Komponenten

Spezialisierungstechniken für Systeme: *Forts.*

- **Dekomposition der Daten (Daten-Clusterung)**
  - **Verteilung der Daten über unterschiedliche, spezialisierte Speicher**
  - **Partitionierung von Feldvariablen (Tiling)**
  - **nutzt bekannte Datenlokalität zur Speicherspezialisierung (TMS320C80)**
  - **nutzt bekannte Adreßsequenzen**
    - zur Verringerung der Kosten zur Erzeugung von Adreßsequenzen
    - zur Verringerung von Zugriffskonflikten
  - **Planung des globalen Datenflusses**

Kommunikation- Die Infrastruktur

- Kommunikationsmechanismen entscheiden über Lösungsraum
  - Pufferung unterstützt Prozeßpipelining
  - Busse erzeugen Ressourcenkonflikte
  - Spielraum wird heute schon im optimierenden Coentwurf genutzt

Noch einmal: Flexibilität und Spezialisierung

- Die meisten Entwürfe bieten keine reine Hardware- oder Softwarelösung, sondern bieten einen abgestuften Übergang von freier Programmierbarkeit zur völligen Spezialisierung
- Es gibt Flexibilität in Hardware und in Software
  - HW-Flexibilität: Entwurfszeit, Laufzeit
  - SW-Flexibilität: Entwurfszeit, Compilezeit

⇒ Wesentliche Asymmetrie der Implementierung in Hardware und Software, nur Erfahrungwerte zur Nutzung, wird zentrales Problem in “Systems-on-a-Chip”

6 Zusammenfassung

- **Einleitung - Hardware-/Software-Coentwurf**
- **Entwurfsprozeß für eingebettete Systeme**
- **Zielarchitekturen**
- **Systematische Nutzung des Entwurfsraums**
- **Spezialisierungstechniken mit Beispielen**