



Implementierungsspezifische Aspekte latenzarmer (Basisband)-Systeme

Dr.-Ing. Markus Petri

19. 1. 2018

Vortrag HU Berlin - Breitbandkommunikation



innovations
for high
performance

microelectronics

Mitglied der

Leibniz
Leibniz-Gemeinschaft

Festplattentransport von Berlin nach Hamburg (300 km):

1 TB 2,5" Festplatte: $0,11 \times 0,08 \times 0,02 = 176 \text{ cm}^3$; 100 g Gewicht (inkl. Verpackung)



Ladevolumen: 100 l = 570 Festplatten

Fahrtzeit: 1,5 h

→ 570 Terabyte / 5400 s = 105 Gigabyte/s



Ladegewicht: 40.000 kg = 400.000 Festplatten

Fahrtzeit 4 h

→ 400 Petabyte / 14.400 s = 28 Terabyte/s

Die Latenz beeinflusst die Gesamtübertragungszeit:

ABER: Auch einzelne Platte benötigt 4 h!

→ Nicht die theoretische PHY-Datenrate, sondern die Gesamtübertragungszeit ist entscheidend!

$$\text{Gesamtübertragungszeit} = \frac{\text{Datenmenge}}{\text{Datenrate}} + \text{Verzögerungszeit}$$

Die Latenz hat einen großen Einfluss auf den Datendurchsatz

(theoretische) Datenrate \neq Datendurchsatz

Datendurchsatz: Anzahl fehlerfreier (Paket-)Übertragungen pro Zeiteinheit

Beinhaltet auch die Bestätigung empfangener Pakete

Beeinflussung durch Verzögerungszeiten (Latenz)

Steigerung nomineller Datenrate + größere Latenz:

→ Geringerer Datendurchsatz möglich

Paketübertragungszeit inkl. unmittelbarer Empfangsbestätigung:

$$T_p = \frac{N_{ps}}{R} + 2T_L$$

Steigerung der Datenrate mit Vergrößerung der Latenz:

$$R_{neu} = n \cdot R \quad T_{Lneu} = m \cdot T_L$$

Wie stark darf sich die Latenz bei einer Datenratensteigerung vergrößern, um die Paketübertragungszeit zu verringern?

$$T_{pneu} < T_p$$

$$\frac{N_{ps}}{R_{neu}} + 2 \cdot T_{Lneu} < \frac{N_{ps}}{R} + 2 \cdot T_L$$

$$\frac{N_{ps}}{n \cdot R} + 2 \cdot m \cdot T_L < \frac{N_{ps}}{R} + 2 \cdot T_L$$

$$m < \frac{N_{ps} \cdot (n - 1)}{2 \cdot n \cdot R \cdot T_L} + 1$$

Bsp: Paketgröße 2 kB, R = 500 MB/s, T_L = 4 µs, n = 2

→ m = 1,25

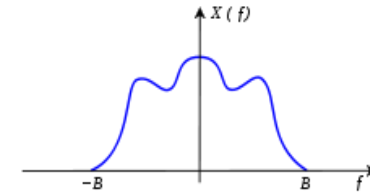
Vergrößert sich bei einer Verdoppelung der Datenrate die Latenz um 50 %, verringert sich die Übertragungsrate um 15 % !

Was ist das Basisband?

Was ist das Basisband?

Basisband:

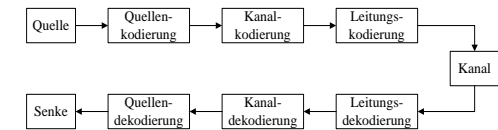
„Natürlicher Frequenzbereich des Nutzsignals“, $f_{\min} = 0 \text{ Hz}$



Basisbandverarbeitung:

Unterste Ebene eines Kommunikationssystems

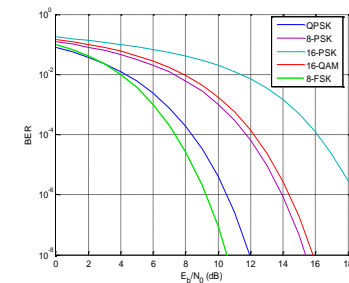
→ Detaillierte Beschreibung durch das horizontale Modell



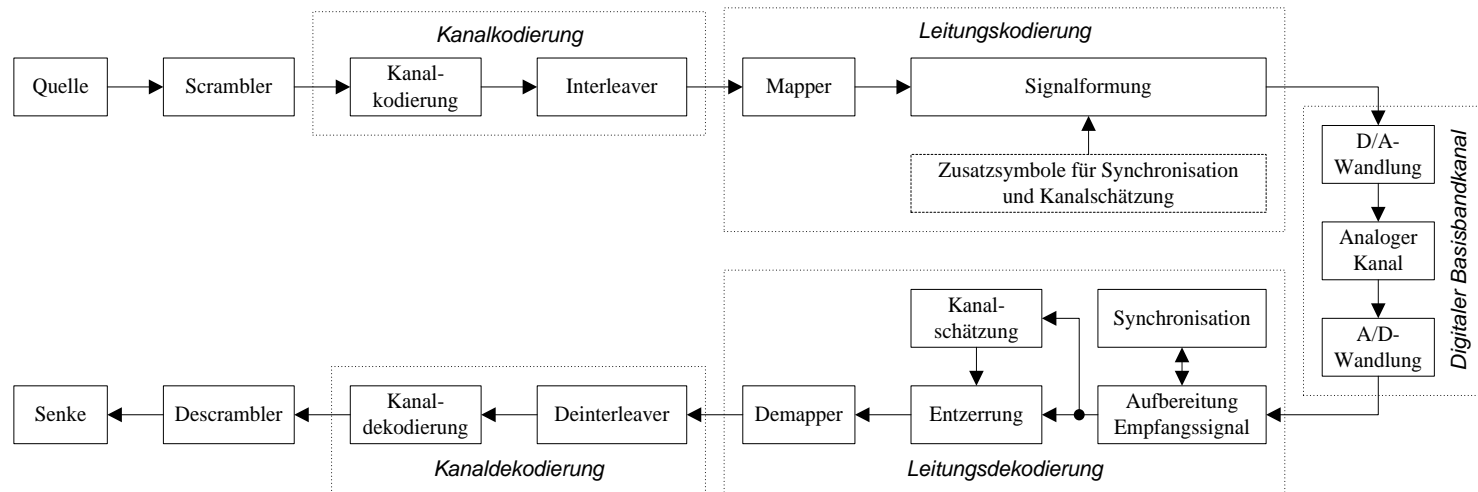
Algorithmen zur Generation und Anpassung des Nutzsignals an den (äquivalenten Basisband-) Kanal

→ Realer Kanal ist analog, Beschreibung durch digitalen Kanal

Abstrakte Modellierung durch Bitfehlerraten



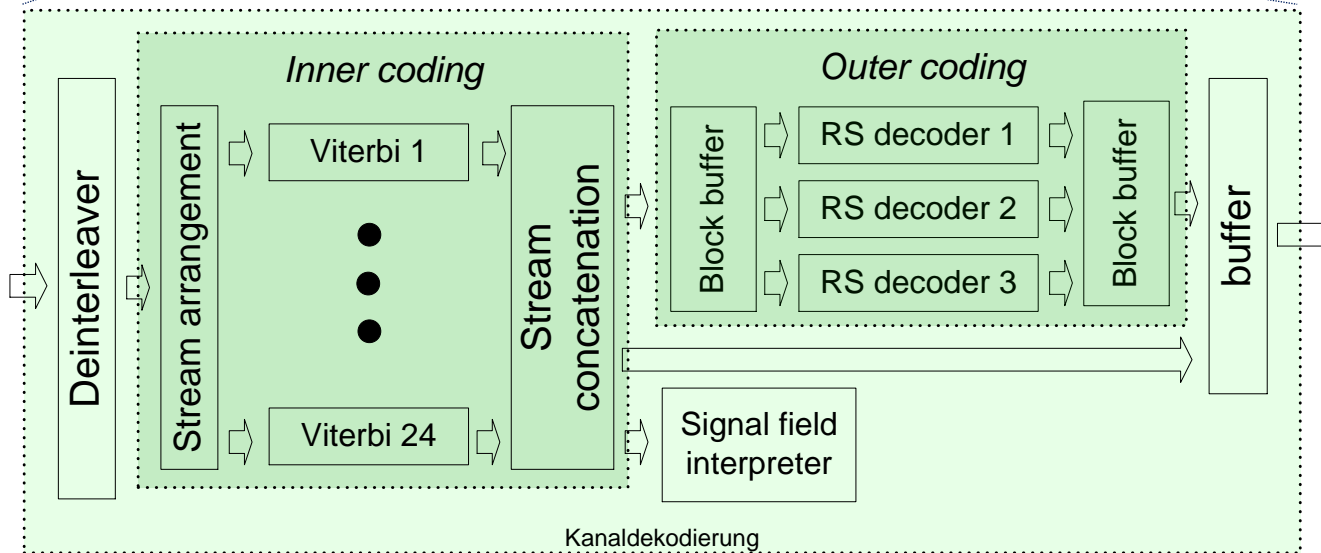
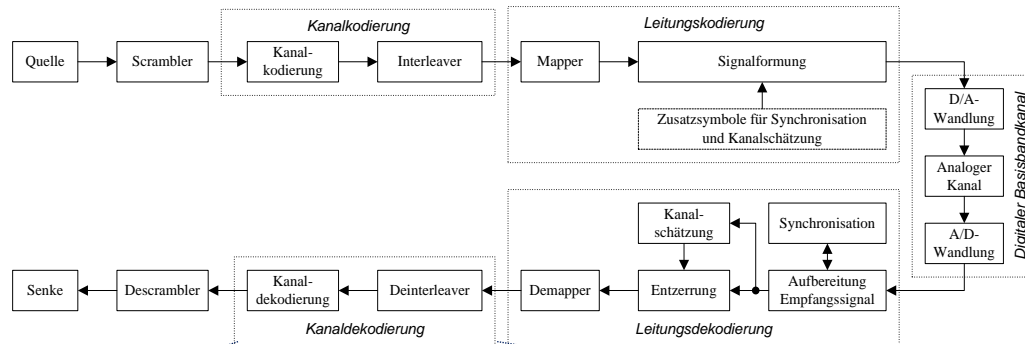
Latenzen im digitalen Basisband



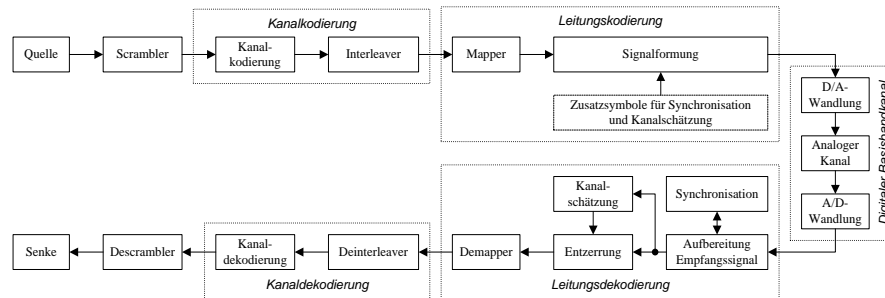
Jeder Block trägt zur Gesamtlatenz des Systems bei

Implementierung hochperformanter Systeme deutlich komplexer

EASY-A VHR-Implementierung der Kanalkodierung



Das Ganze ist mehr als die Summe seiner Teile!



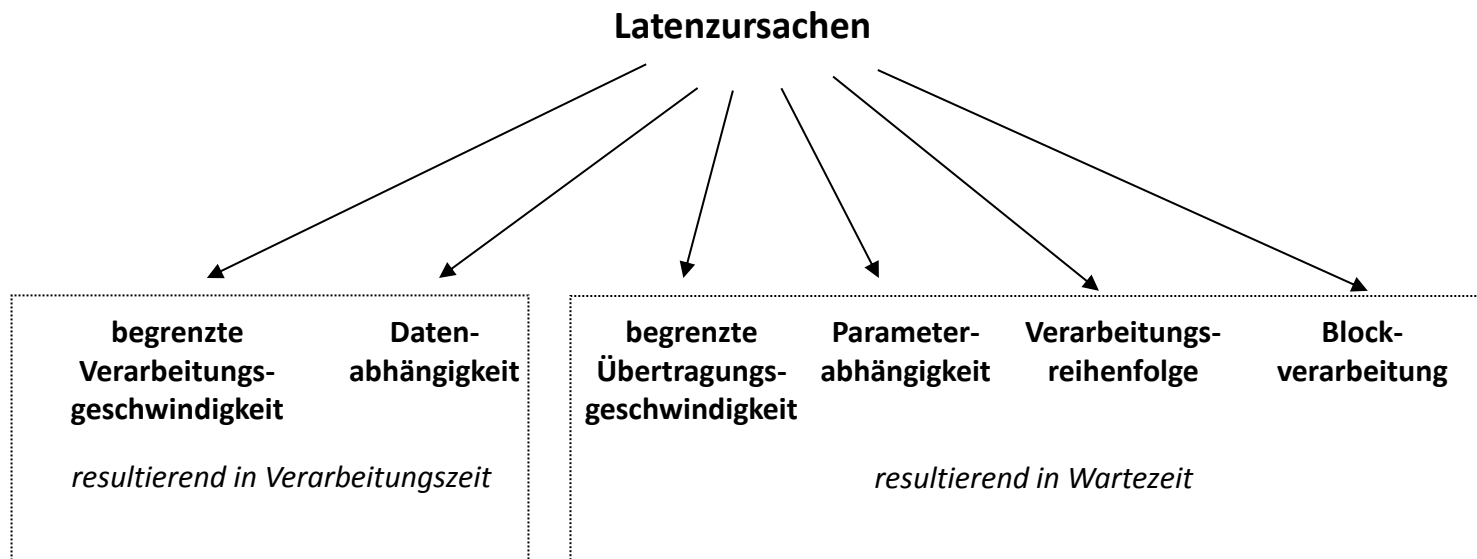
$$\cancel{T_{ges} = \sum_{m=1}^M T_m} \rightarrow T_{ges} = \sum_{m=1}^M T_m(p)$$

Gesamtlatenz:

- Latenzen der Einzelmodule
- Zusammenspiel der Einzelmodule

p: Abhängigkeit von Parametern und gegenseitige Beeinflussung

Ursachen von Latenzen in der Basisbandverarbeitung



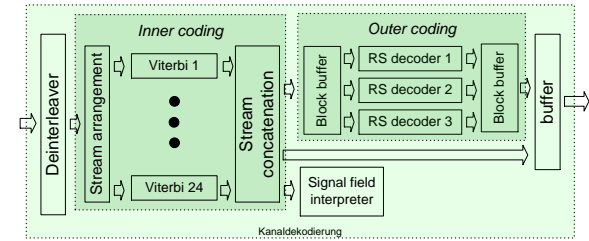
Latenzanalyse paralleler Verarbeitungsstrukturen

Parallelverarbeitung zur Durchsatzsteigerung üblich

→ Parallele Verarbeitungsstrukturen beeinflussen Latenz

→ Unterscheidung in bitparallel und blockparallel

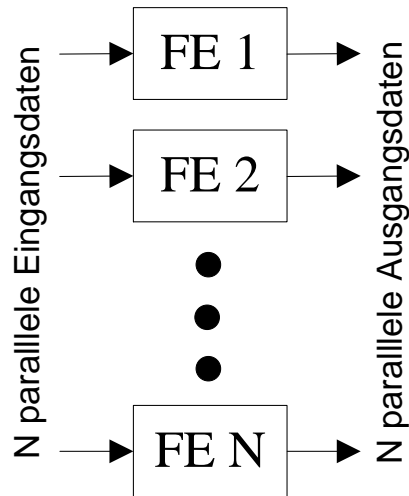
→ Bit: Verallgemeinerung auf atomare Einheit an der Schnittstelle
(z.B. Datenwort aus 8 Bit bei RS-Kodierung)



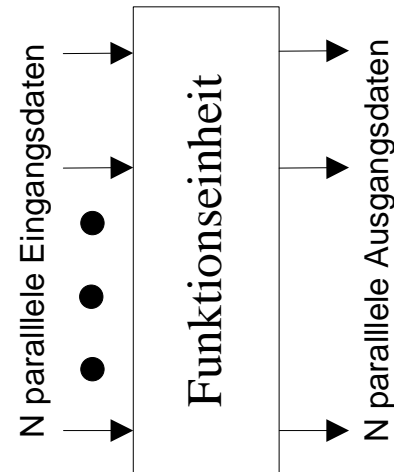
Bitparallele Verarbeitung

Gleichzeitige Anwendung einer Operation auf die parallel eingehenden Bits

2 Varianten: unabhängige und abhängige Verarbeitung

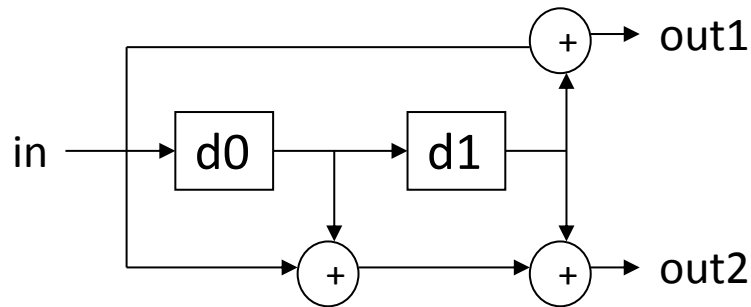


*Bsp: Paralleles Mapping
(Unequal FEC)*



Parallele Faltungskodierung

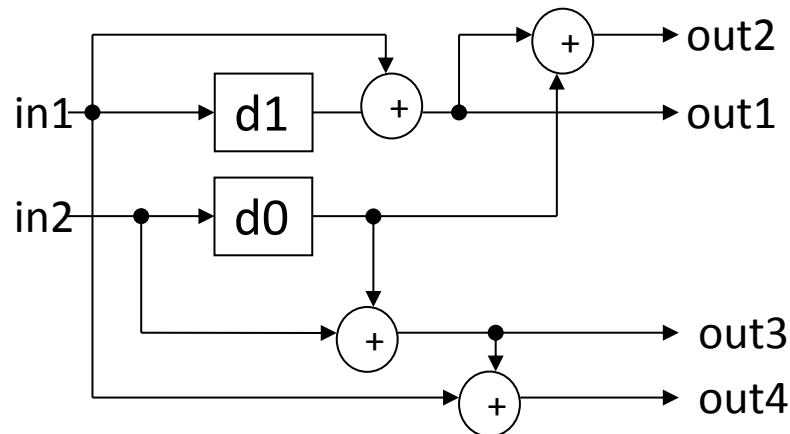
Beispiel: Bitparalleler Faltungskodierer



```

if rising_edge(clk)
  out1 <= d1 xor in;
  out2 <= d1 xor d0 xor in;
  d1  <= d0;
  d0  <= in;
end;

```



```

if rising_edge(clk)
  d0t := d0;
  d1t := d1;
  for n in 1 to 2 loop
    out1(n) <= d1t xor in(n);
    out2(n) <= d1t xor d0t xor in(n);
    d1t := d0t;
    d0t := in(n);
  end loop;
  d0 <= d0t;
  d1 <= d1t;
end;

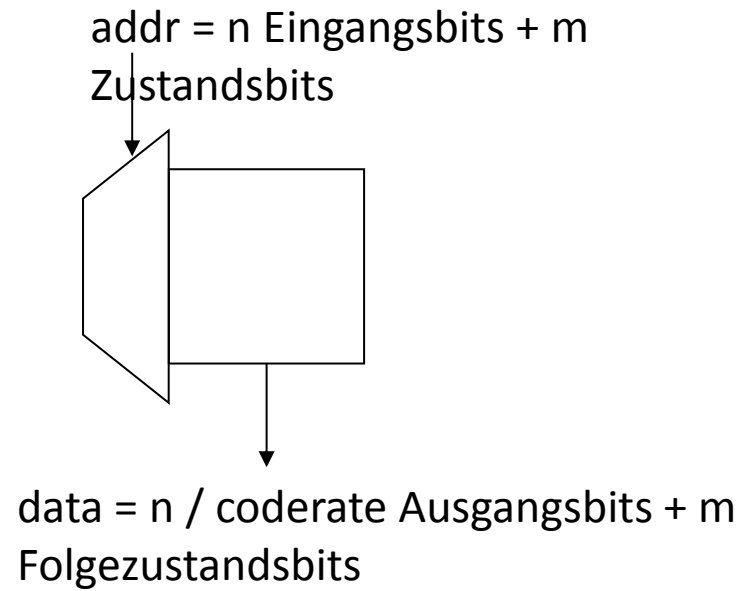
```

Beispiel: Bitparalleler Faltungskodierer in Software

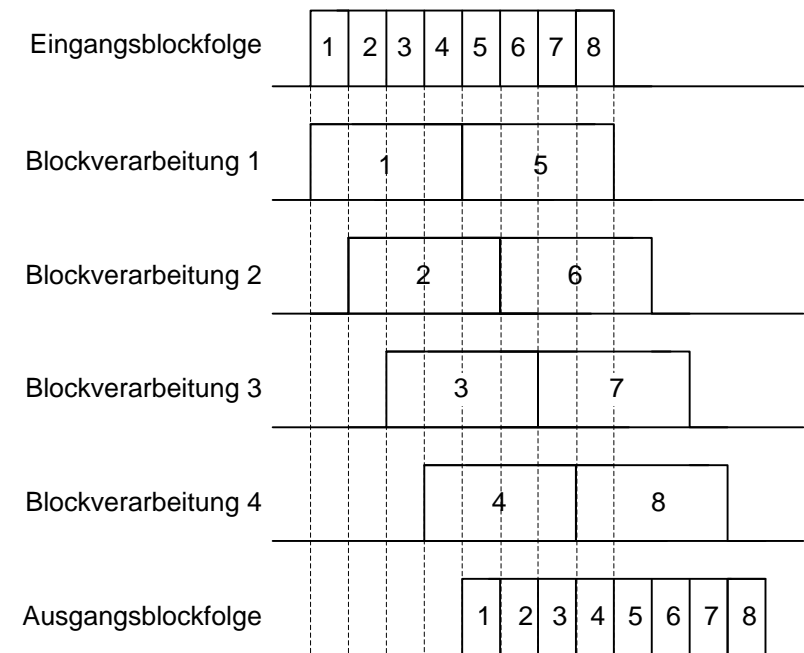
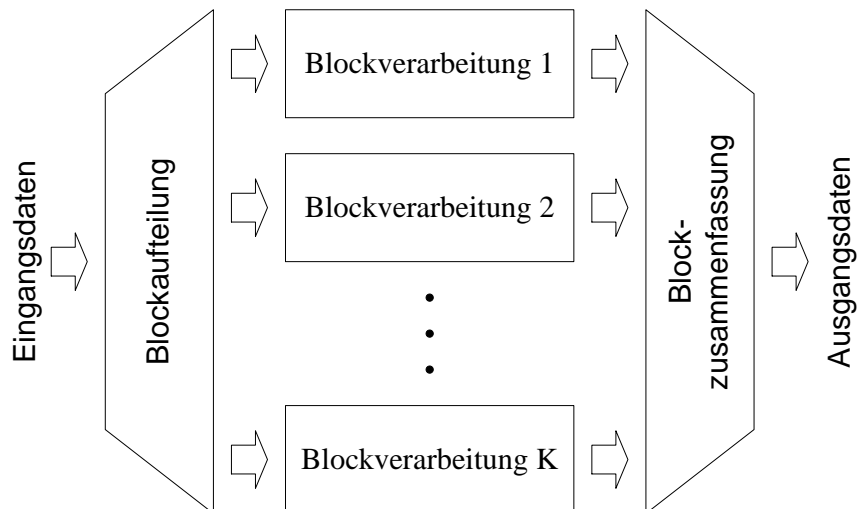
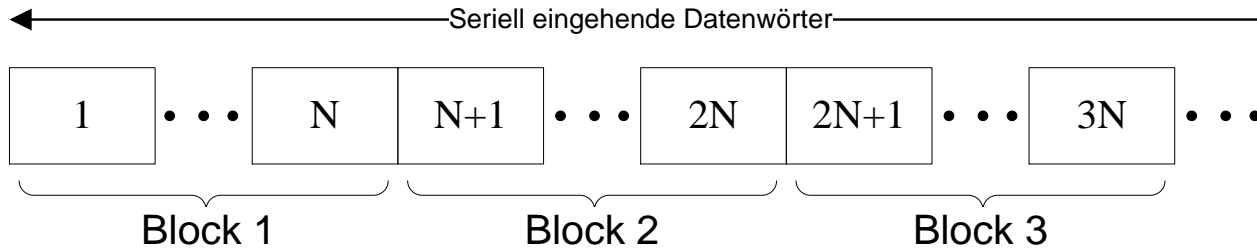
Strukturelle Parallelisierung in Hardware sinnvoll

In Software → LUT besser

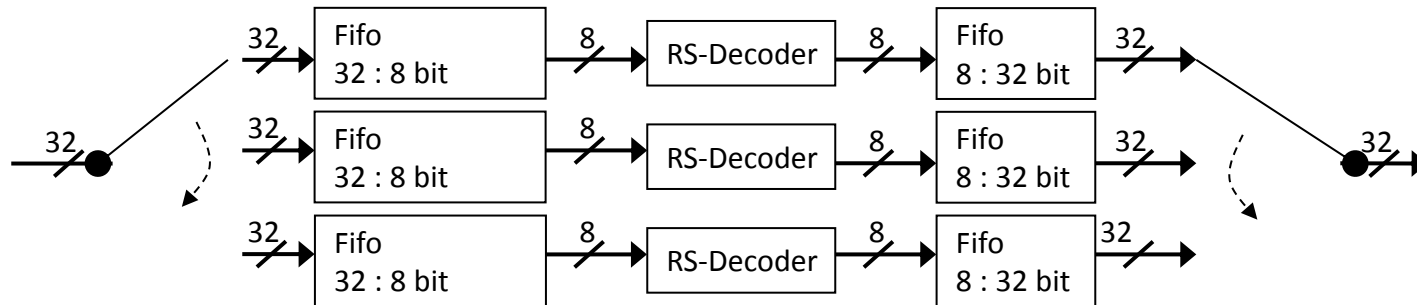
Vereinheitlichung mit Punktierung möglich



Blockparallele Verarbeitung



Beispiel: Blockparalleler Reed-Solomon-Dekodierer

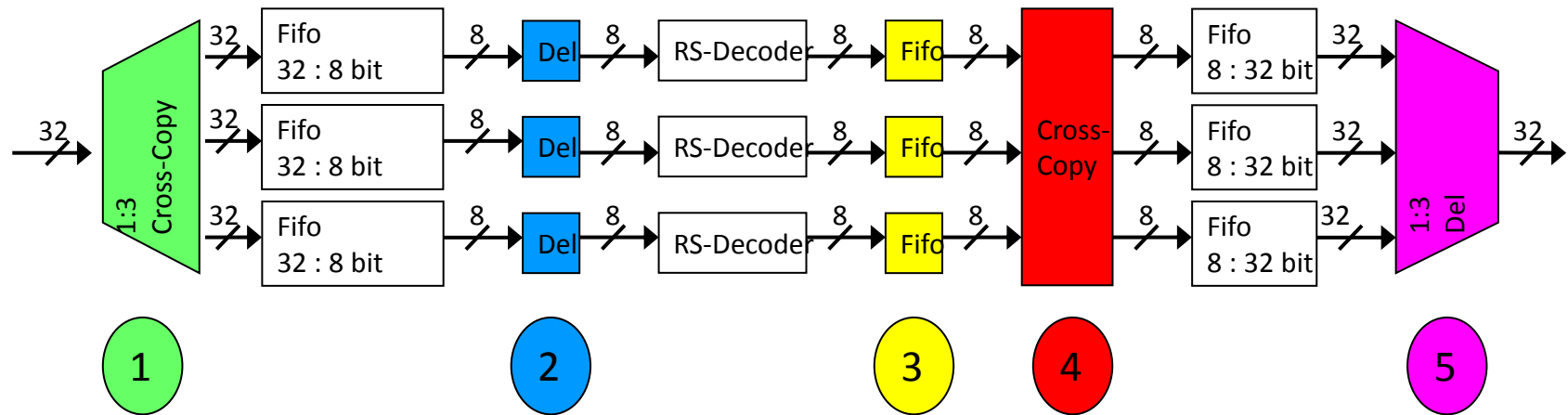


Gesamtdurchsatz: 3x Durchsatz der Einzeldekoder

Diese Lösung ist so nicht richtig!!!

- Block für den ersten Pfad besteht aus 255 Eingangsbytes, es werden jedoch stets Vielfache von 4 eingeschrieben
- Kopieren von einzelnen Bytes und Einfügen von Dummy-Bytes notwendig!

Beispiel: Korrekter blockparalleler Reed-Solomon-Dekodierer



1. Kopieren von Bytes in den nächsten Block + Einfügen von Dummy-Bytes am Ende
2. Löschen überzähliger Dummy-Bytes
3. Speichern des Blocks bis zur Fertigstellung des vorherigen
4. Kopieren von Bytes zum vorherigen Block, Einfügen von Dummy-Bytes am Ende
5. Löschen von überzähligen Dummy-Bytes

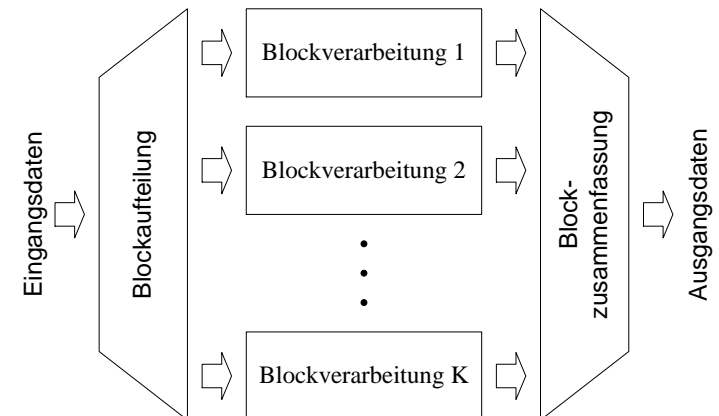
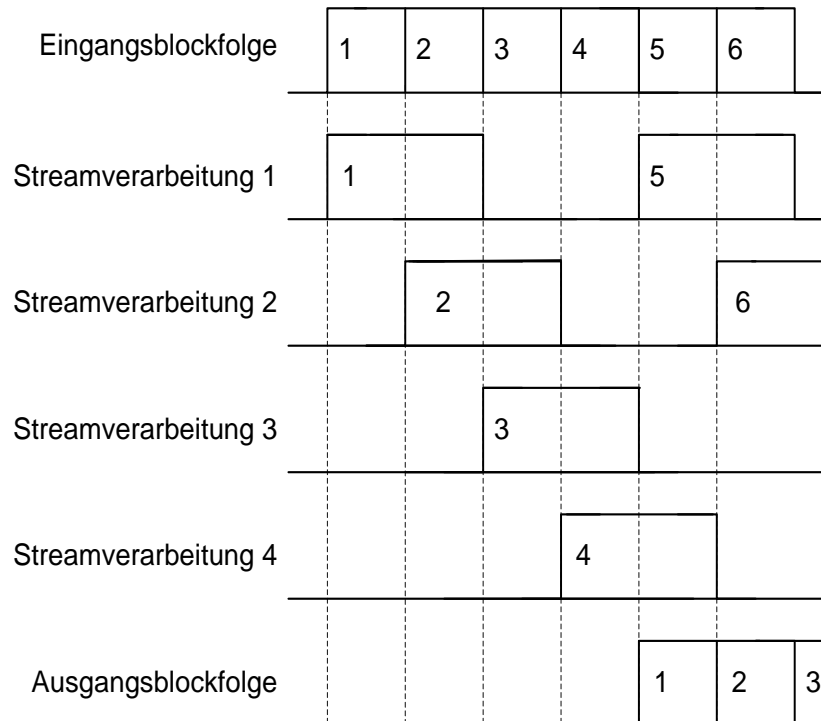
Sonderform: Streambasierte blockparallele Verarbeitung

Aufteilung der nicht in Blöcken organisierten Rohdaten in Teilströme

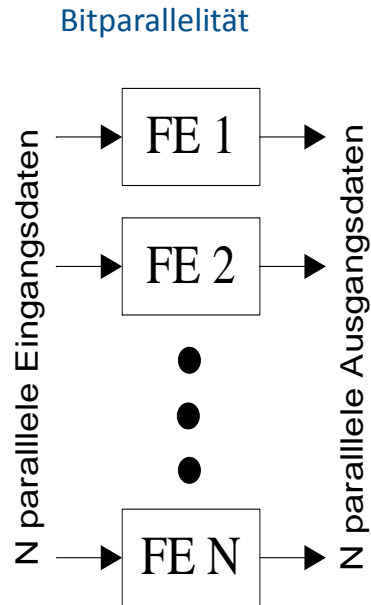
Teilströme werden jeweils einer Block-FE zugeordnet

→ Zwischen den einer FE zugeordneten Blöcken bestehen Abhängigkeiten

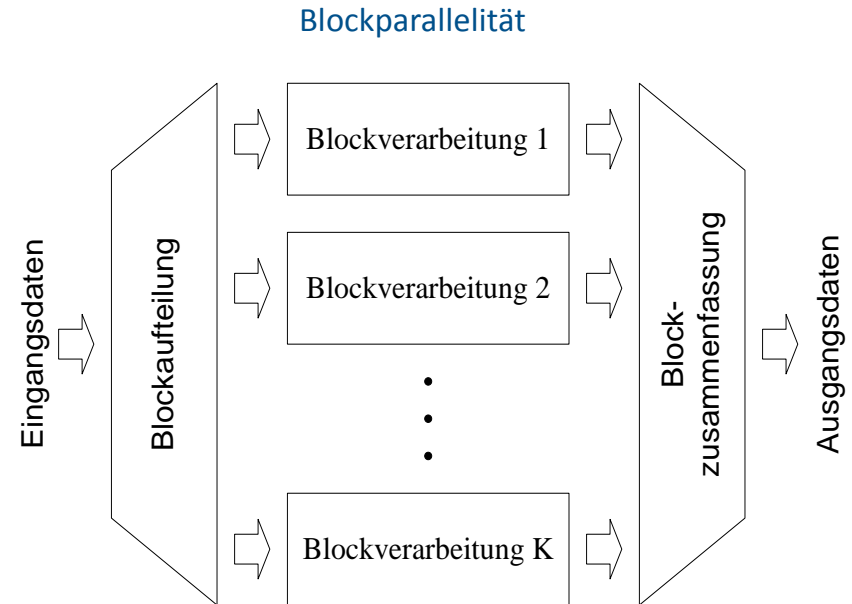
→ Zuordnung kann nicht wahlfrei getauscht werden



Latenzanalyse paralleler Verarbeitungsstrukturen

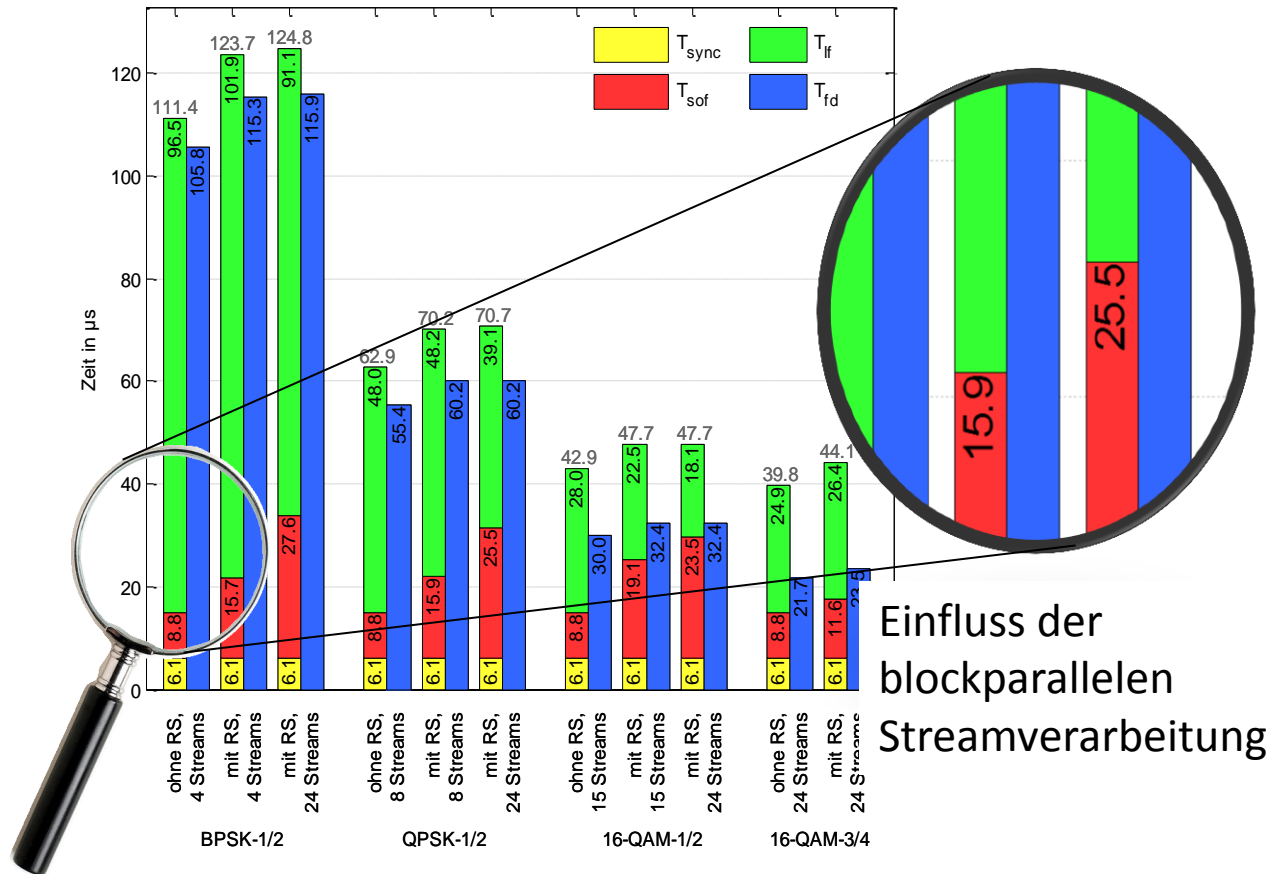


$$T_{Lges} = T_{LFE}$$



$$T_{Lges} = T_{LFE} + \frac{N_{block}}{R_{FE}}$$

EASY-A VHR-E: Analyse der Verarbeitungslatenzen



Aber:



Parallelität ermöglicht auch Latenzverringierung!

Latenzverringierung durch Parallelität

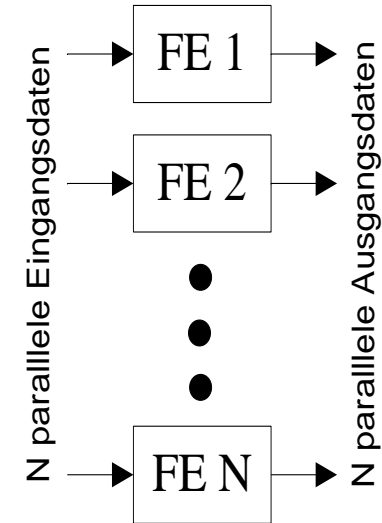
Latenzreduktion durch Bitparallelität möglich

- Verzicht auf dedizierten Interleaver
- Nutzung des impliziten Interleavings

Implizites Interleaving:

- Aufteilung der Bits auf unterschiedliche Funktionseinheiten
- Umwandlung von Bündelfehlern in Einzelfehler

Analyse der Systemperformance notwendig (Framefehlerraten)



Latenzverringierung durch Spekulation

Spekulation: Schlussfolgerung ohne gesicherte Erkenntnis

Gebräuchliche Anwendung in Datenverarbeitung:

- Sprungvorhersage
- spekulative Befehlsausführung
- Prefetching

spekulative Verfahren zur Latenzverringierung:

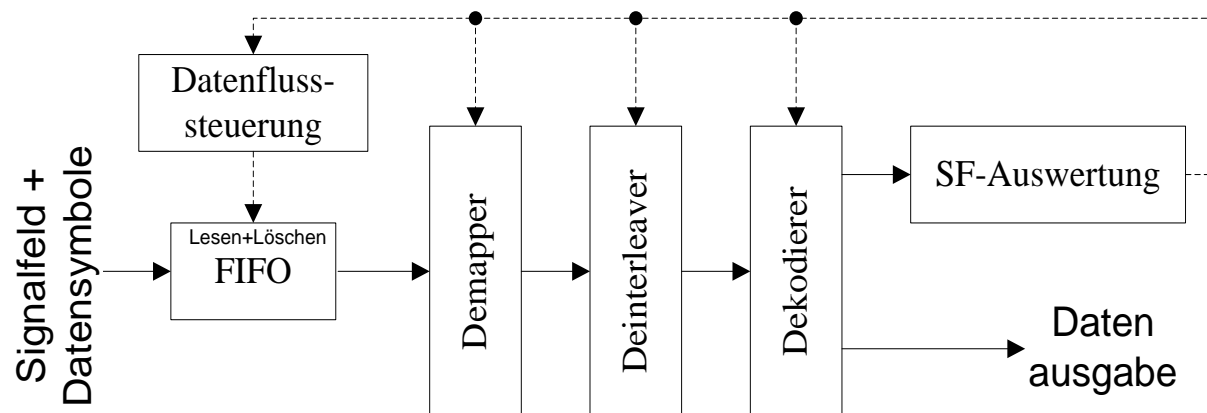
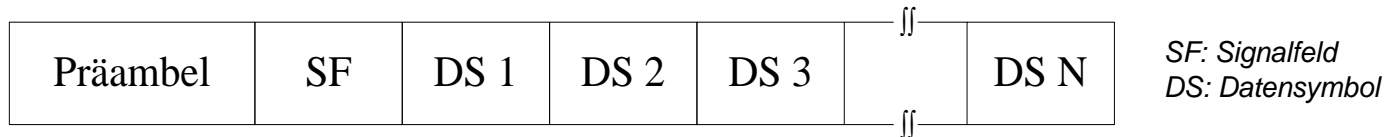
1.) Spekulative Demodulation

→ Spekulation auf die verwendeten Übertragungsparameter
(Modulation, Kodierung, Punktierung)

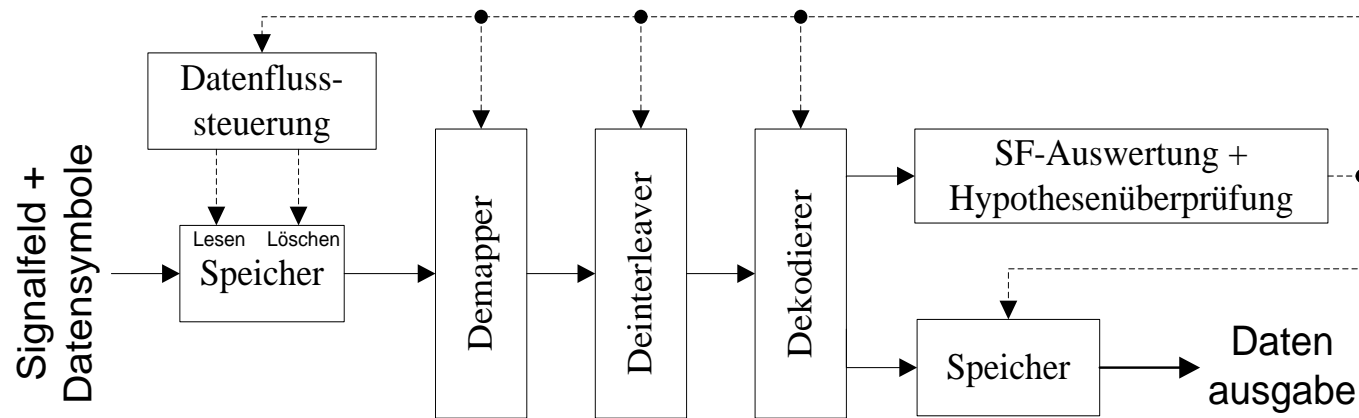
2.) Spekulative Dekodierung

→ Spekulation auf Fehlerfreiheit der übertragenen Daten

Typische Basisbandverarbeitung



Spekulative Demodulation

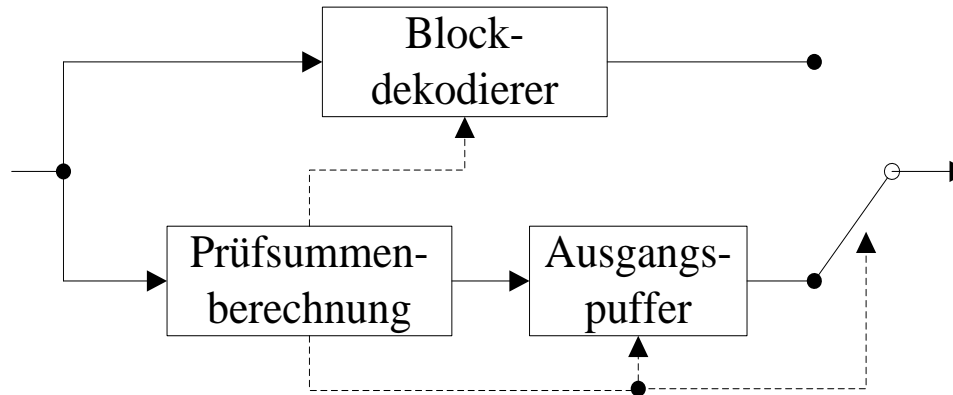


$$T_{Lspec} = \begin{cases} T_{Lnospec} - T_{demap} - T_{deintlv} - T_{decode} & | \text{Spekulation erfolgreich} \\ T_{Lnospec} & | \text{Spekulation fehlgeschlagen} \end{cases}$$

Dekodierung eines systematischen Blockcodes

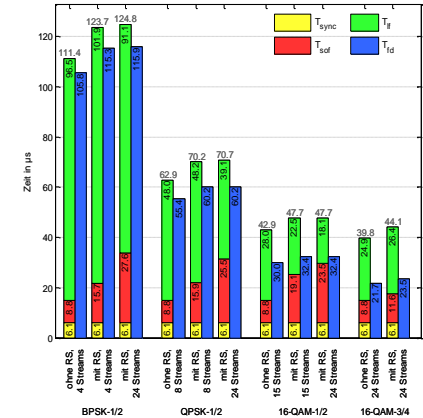
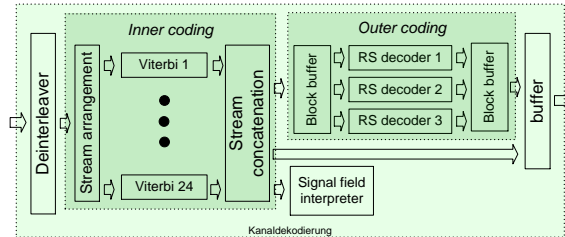
1. Einschreiben des Blocks
2. (iterative) Blockdekodierung
3. Blockausgabe

Spekulative Dekodierung



- Voraussetzung: systematischer Blockcode
- Erfolgsfall: Verringerung der Latenz um Blockdekodierungszeit
- Prüfsummenalgorithmus \leftrightarrow Blockcode

EASY-A VHR-E: Update Latenzanalyse



Empfängerlatenz (BPSK, RS, 4 Streams):

21,5 µs

Implizites Interleaving:

- 1,2 µs (5,6 %)

Spekulative Demodulation (erfolgreich):

- 1,7 µs (8 %)

Spekulative RS-Dekodierung (erfolgreich):

- 2,9 µs (13,4 %)

Gesamt:

16,3 µs

- 5,2 µs (24,2 %)

Zwischenfazit

Steigende Datenraten durch komplexe Algorithmen

→ zunehmender Einfluss von Latenzen auf Datendurchsatz

Systemkonzeption nicht unabhängig von Implementierung + Beachtung Hardwarekenngößen möglich

Spekulative Verfahren: Verringerung von Latenzen aufgrund von Parameterabhängigkeiten + Blockverarbeitung

→ Variable Latenzen: Anpassung MAC-Verarbeitung erforderlich

Deutliche Latenzverringierung durch gezeigte Verfahren in realem 60-GHz-Kommunikationssystem



Interesse an Praktikum / Masterarbeit?

Dr. Markus Petri

IHP – Leibniz-Institut für innovative Mikroelektronik

Im Technologiepark 25

15236 Frankfurt (Oder)

Tel.: +49 (0) 335 5625 445

Fax: +49 (0) 335 5625 671

E-Mail: petri@ihp-microelectronics.com

www.ihp-microelectronics.com



innovations
for high
performance

microelectronics

Mitglied der

Leibniz
Leibniz-Gemeinschaft