

Evaluation of Switch-to-Switch Header Flit Protection Schemes in Networks-on-Chip

Martin Gag, Philipp Gorski, Tim Wegner, Dirk Timmermann





- 1. Einleitung**
2. Konzept zur Fehlererkennung im Network-on-Chip
3. Flächenminimierung der Router
4. Ergebnisse
5. Zusammenfassung



Hardware Trends:

- ~ alle 2 Jahre Verdopplung der Transistorendichte
- geringere Betriebsspannungen
- geringere Ladung in Speicherelementen
- Erhöhung der Taktfrequenzen

Fehlertrends:

- Signifikante Steigerung der Soft Error Rate
- Höherer Anteil an multi-bit upsets
- Mehr Verletzungen von timing-safety Margen
- Mehr intermittierende Fehler durch Prozessvariationen und Prozessrückstände

Einleitung – Soft Errors im NoC

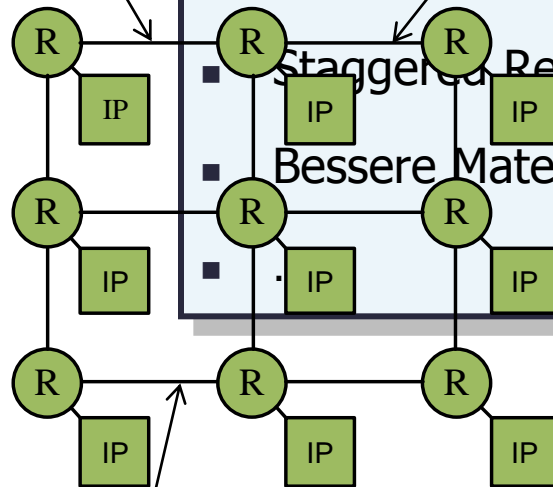


Crosstalk Noise

Radiation

Gegenmaßnahmen auf Physical Layer:

- Spacing, Shielding
- Staggered Repeater
- Bessere Materialien (Low K)

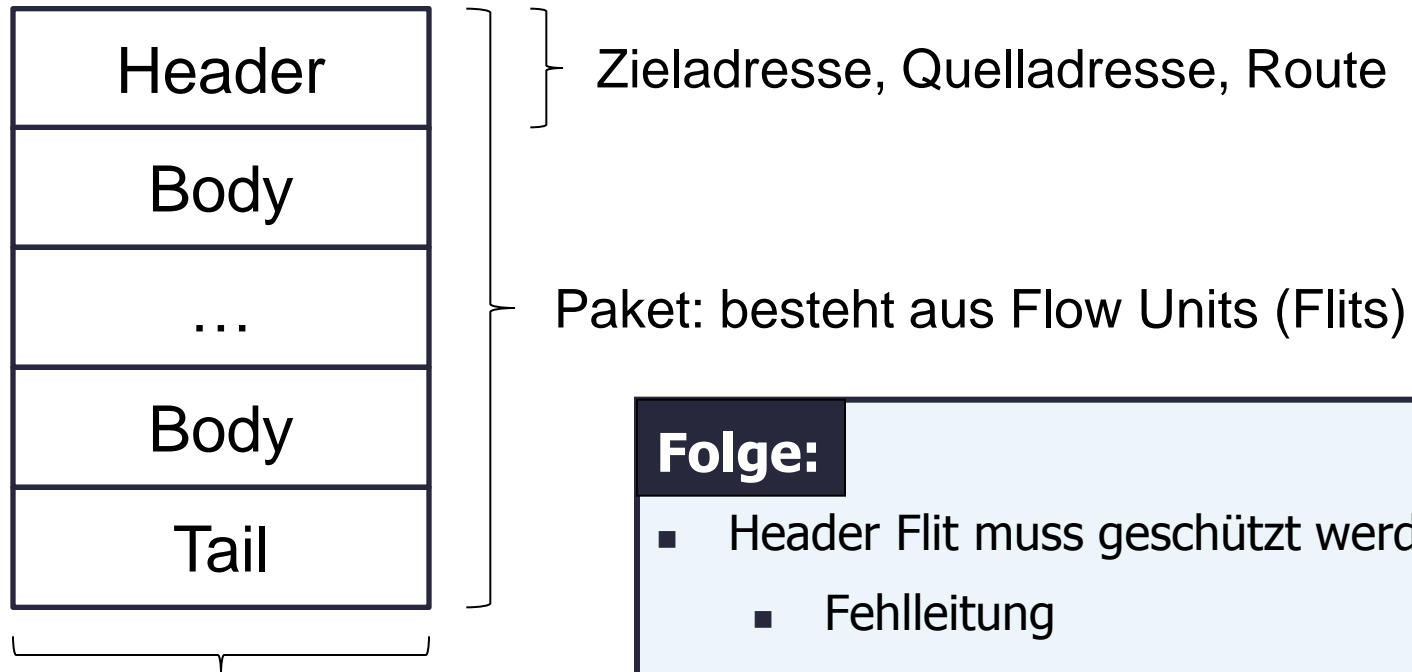


Behandlung auf höheren Ebenen:

- End to End
- Switch to Switch

Therm. N

Konzept zur Fehlererkennung im NoC



Folge:

- Header Flit muss geschützt werden, sonst
 - Fehlleitung
 - Verlust
 - Router-Crash

- Wormhole Switching:
 - Routing Information im Paket-Header -> 1. Flit

Konzept zur Fehlererkennung im NoC – Codierung

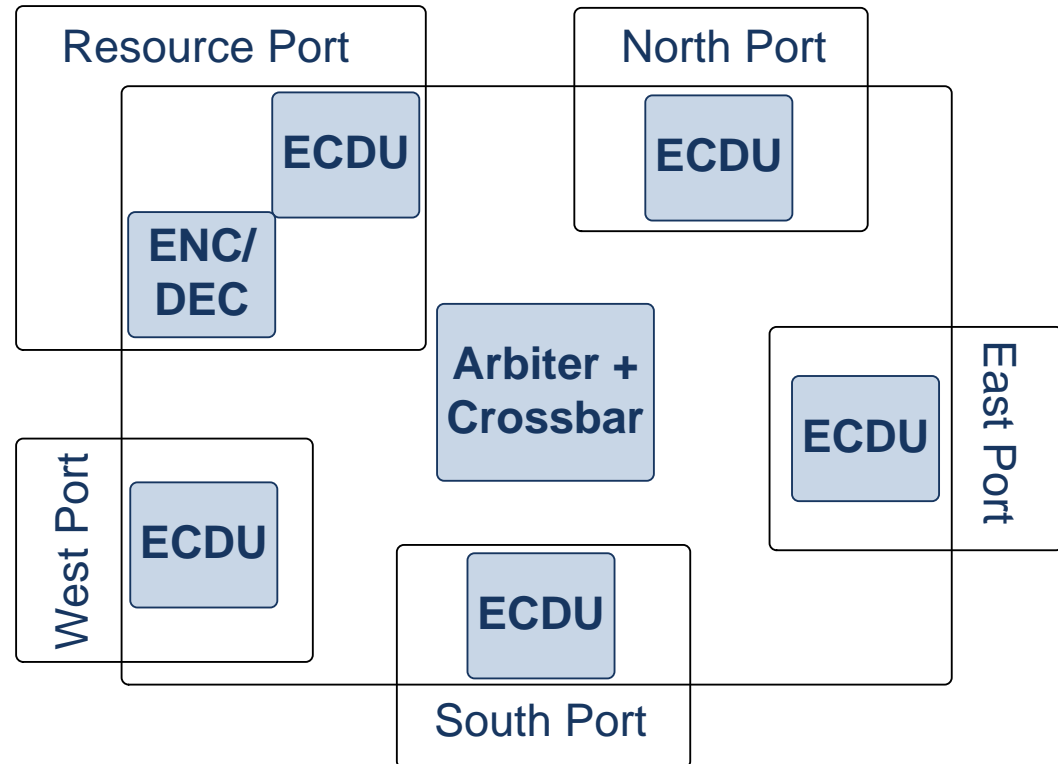


- Hamming Code
 - Auslesen der Nutzdaten direkt möglich
 - Fehlererkennung/-korrektur
 - HC(3,1) } Fehlererkennung (2) ODER –korrektur (1)
 - HC(7,4) } **DED** **SEC**
 - HC(12,8) }
 - EHC(4,1) } Fehlererkennung (2) UND –korrektur (1)
 - EHC(8,4) } **SEC-DED**
 - EHC(13,8) }

Konzept zur Fehlererkennung im NoC

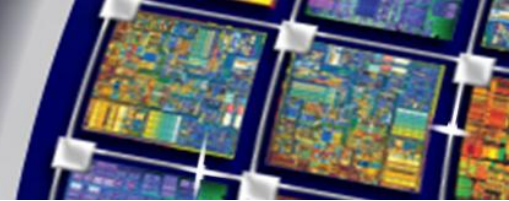
- Ohne Resource Sharing:

- 5 parallele Einheiten

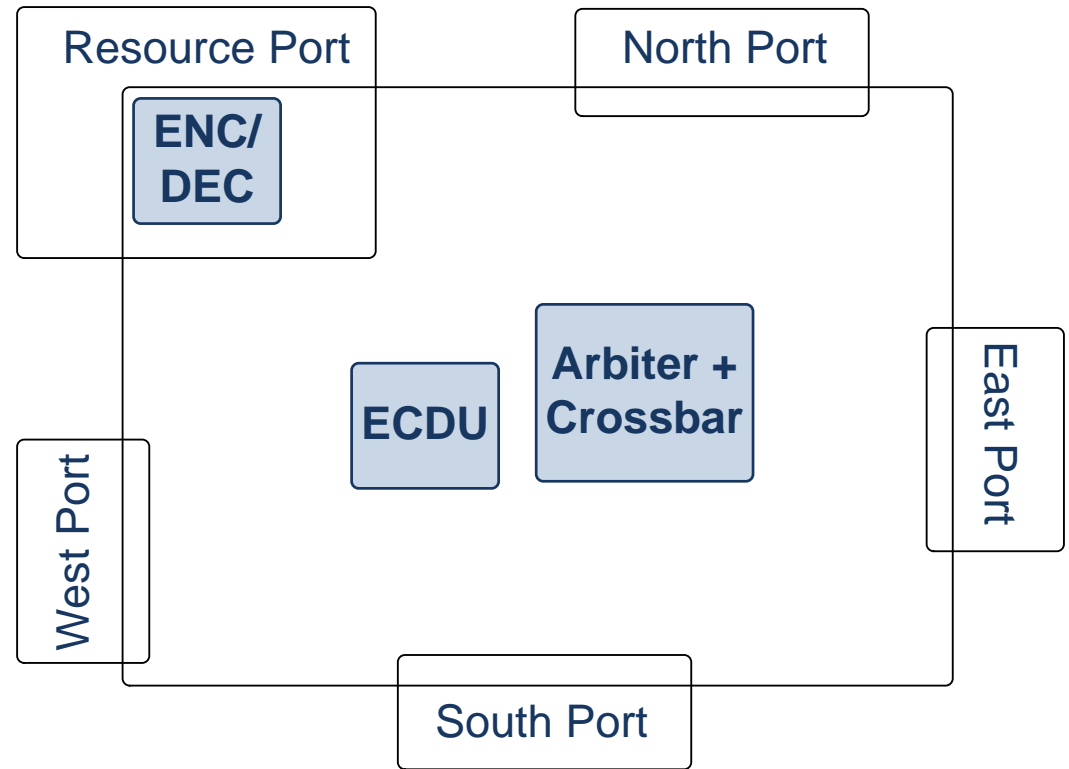


ECDU: Error Correction / Detection Unit

Konzept zur Fehlererkennung im NoC



- Mit Resource Sharing:
 - 2 oder 1 Einheit

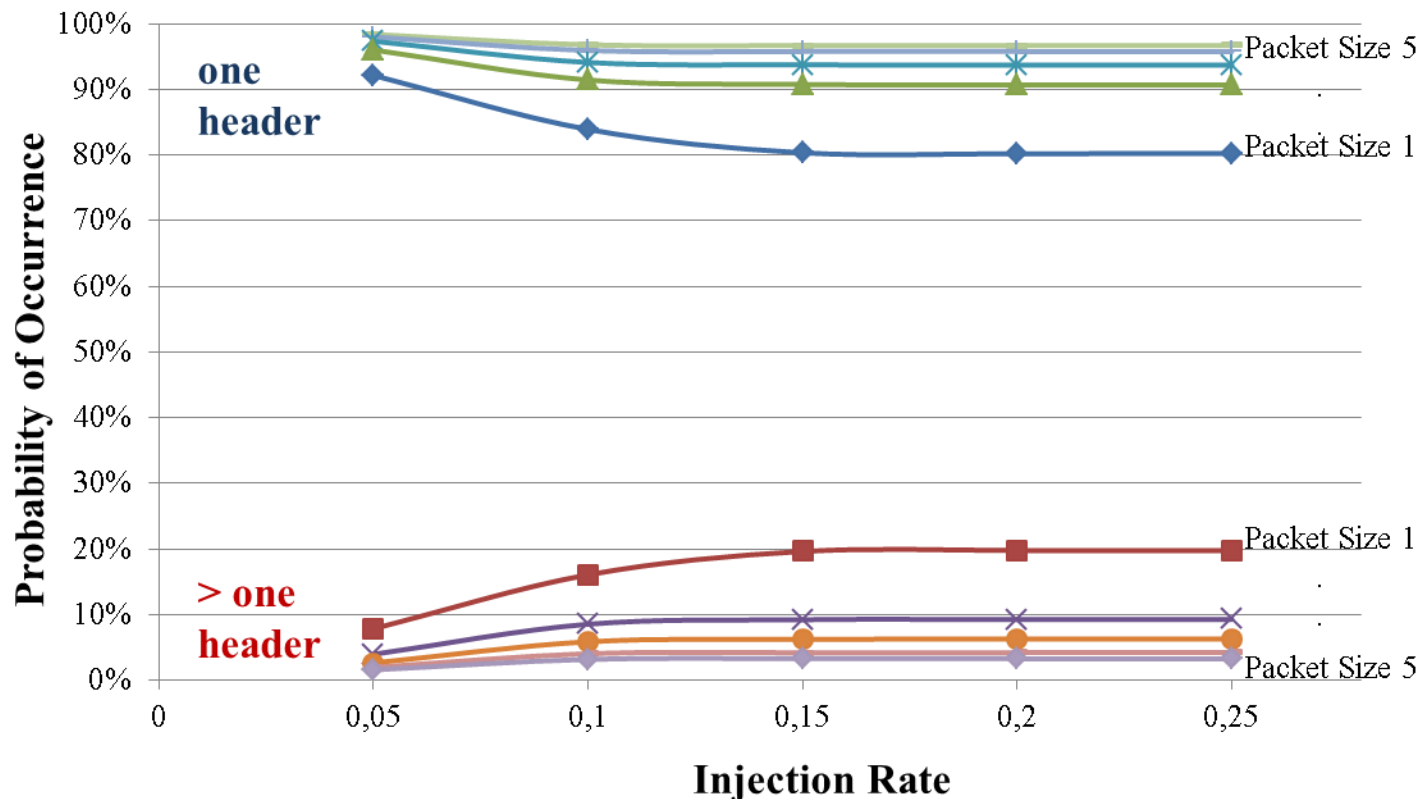


- Geteilte Nutzung der ECDU möglich, wenn nicht viele Header gleichzeitig bearbeitet werden müssen

Konzept zur Fehlererkennung im NoC



- Anzahl der Header Flits im Router in einem Takt:

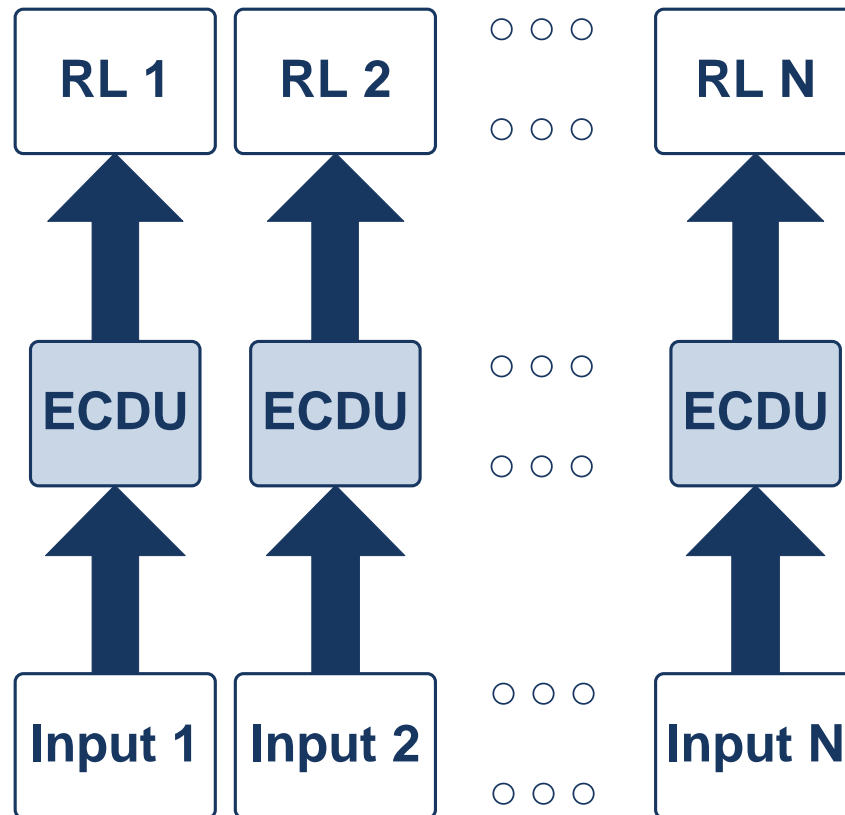


Injection Rate: Anzahl der pro Takt generierten Flits pro IP-Core
Parameter: 10x10 mesh, XY-Wormhole-Routing, random traffic

Flächenminimierung ✦



■ Ohne Resource Sharing



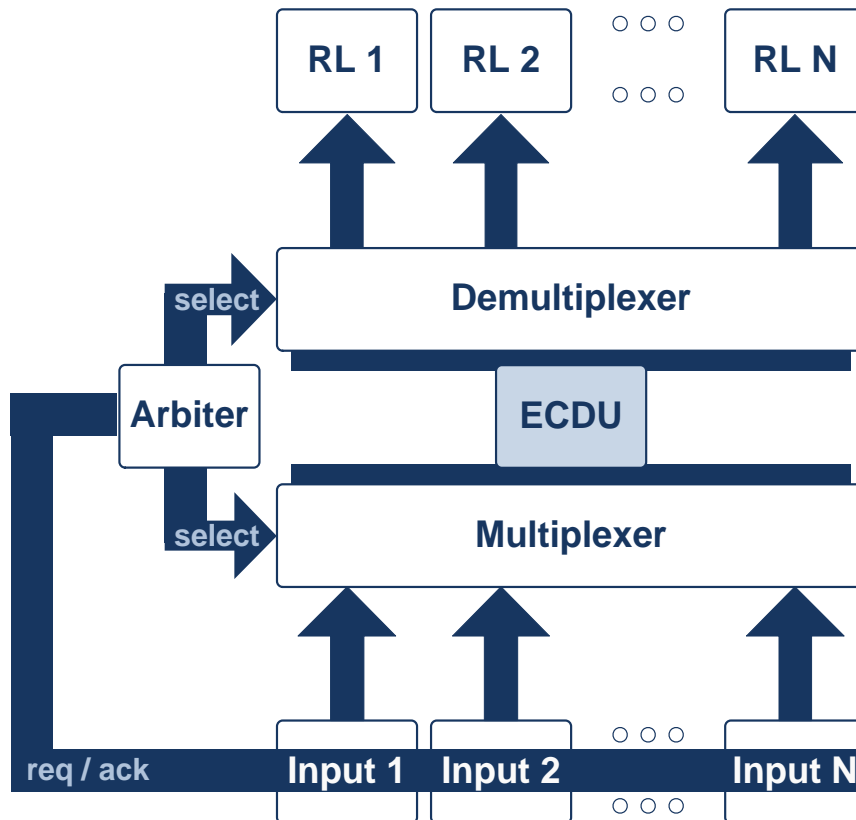
Kosten:

- $A = N * A_{ECDCU}$
- $T = T_{ECDCU}$

Flächenminimierung ✦



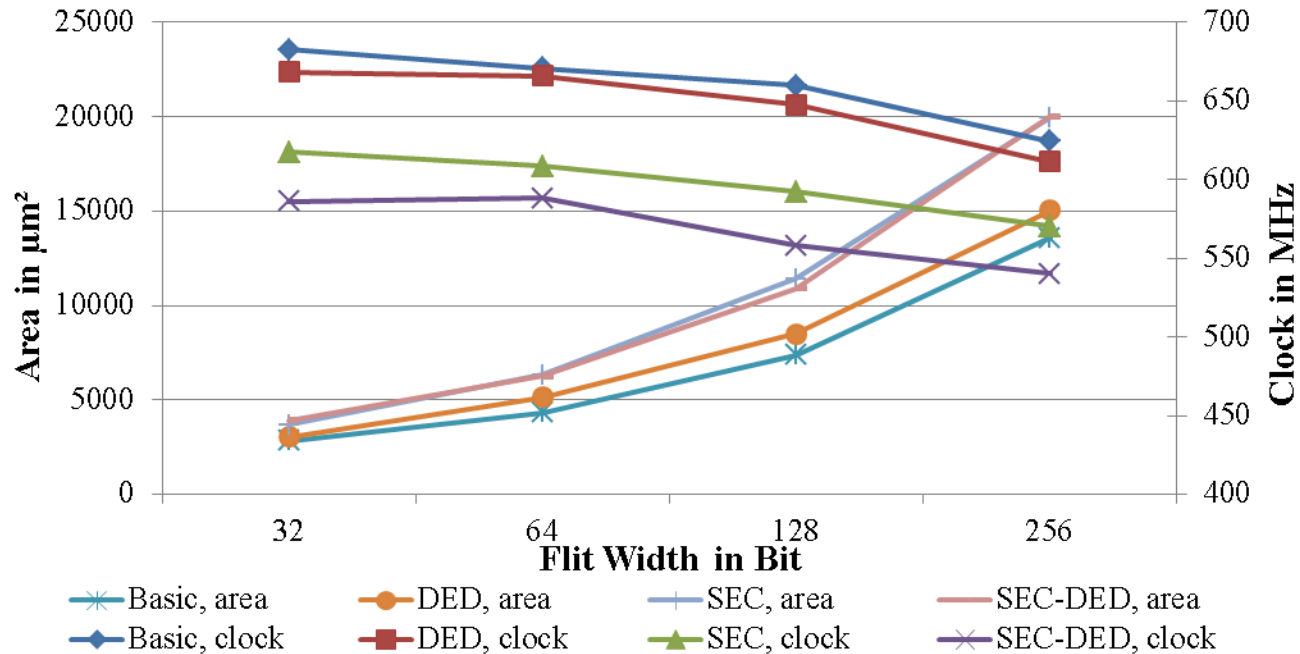
■ Mit Resource Sharing



Kosten:

- $A = A_{\text{MUX}}(N) + A_{\text{ECDU}}$
- $T = T_{\text{ECDU}} + T_{\text{MUX}}$

Ergebnisse – Frequenz, Fläche

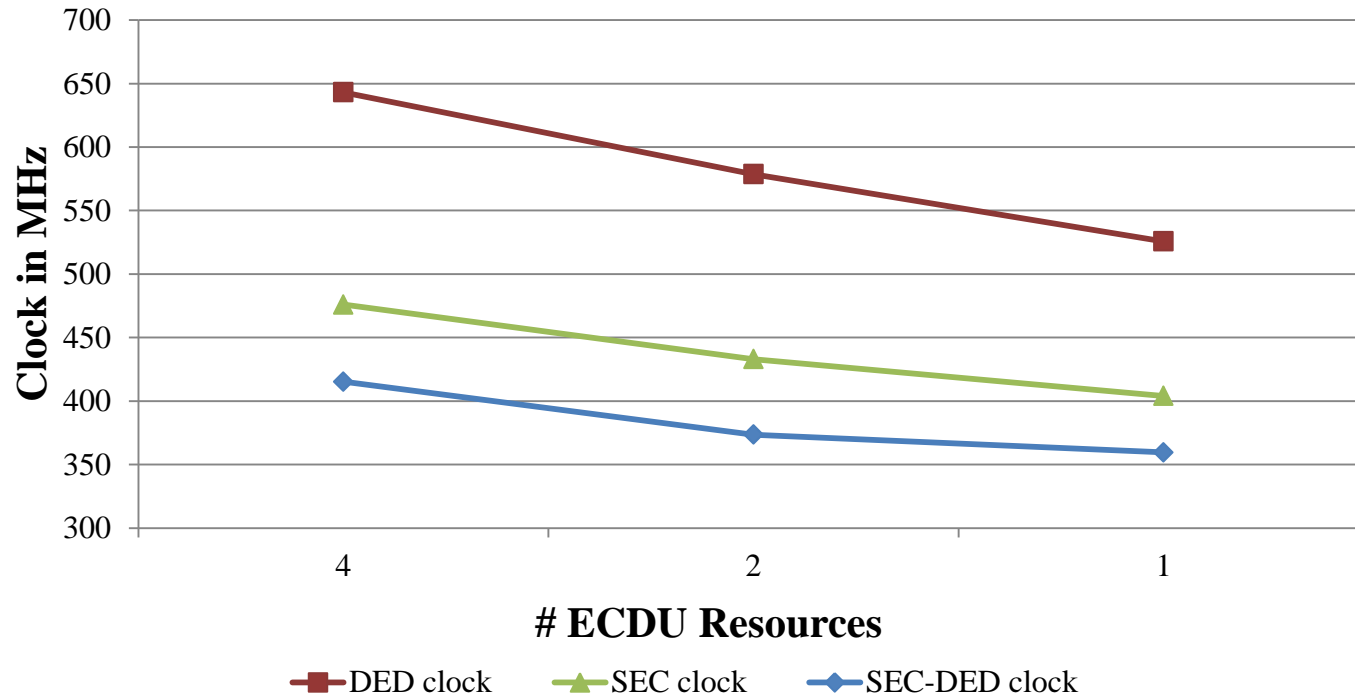


Encoding: keins, HC(3,1), EHC(4,1)

Komplexität:

■ Kein Encoding < DED < SEC < SEC-DED

Ergebnisse – Frequenz, Fläche

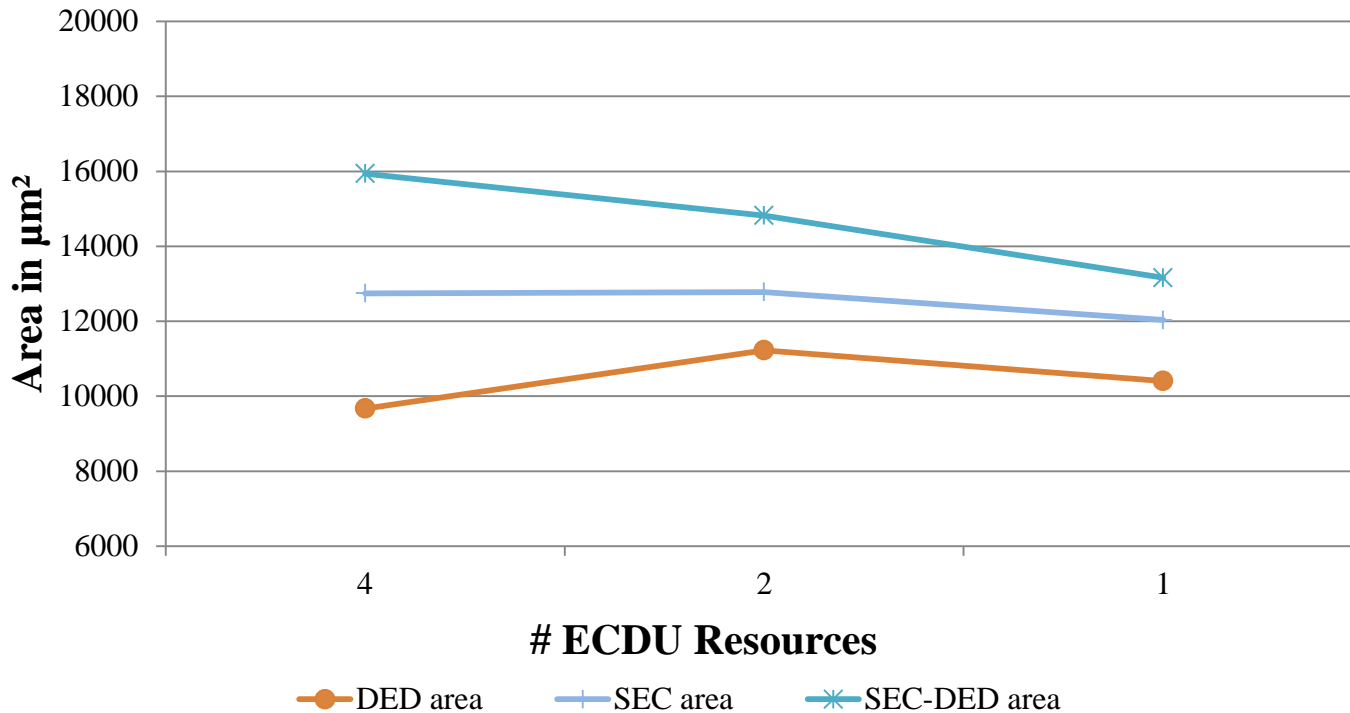


Flitbreite: 128 Bit, Encoding: HC(12,8), EHC(13,8)

Resource Sharing:

- Geringere max. Taktrate

Ergebnisse – Frequenz, Fläche

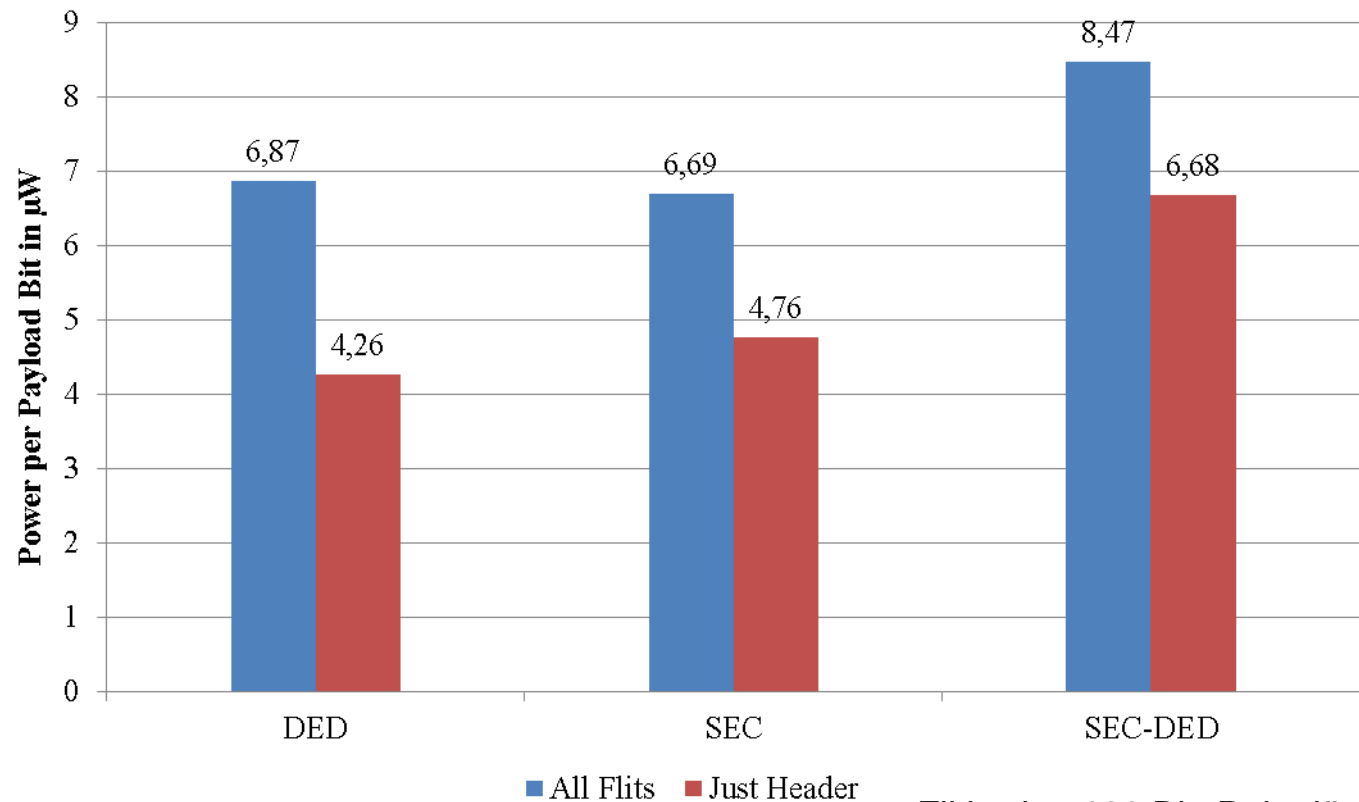


Flitbreite: 128 Bit, Encoding: HC(12,8), EHC(13,8)

Resource Sharing:

- Weniger Fläche, wenn Encoding komplex

Ergebnisse – Power

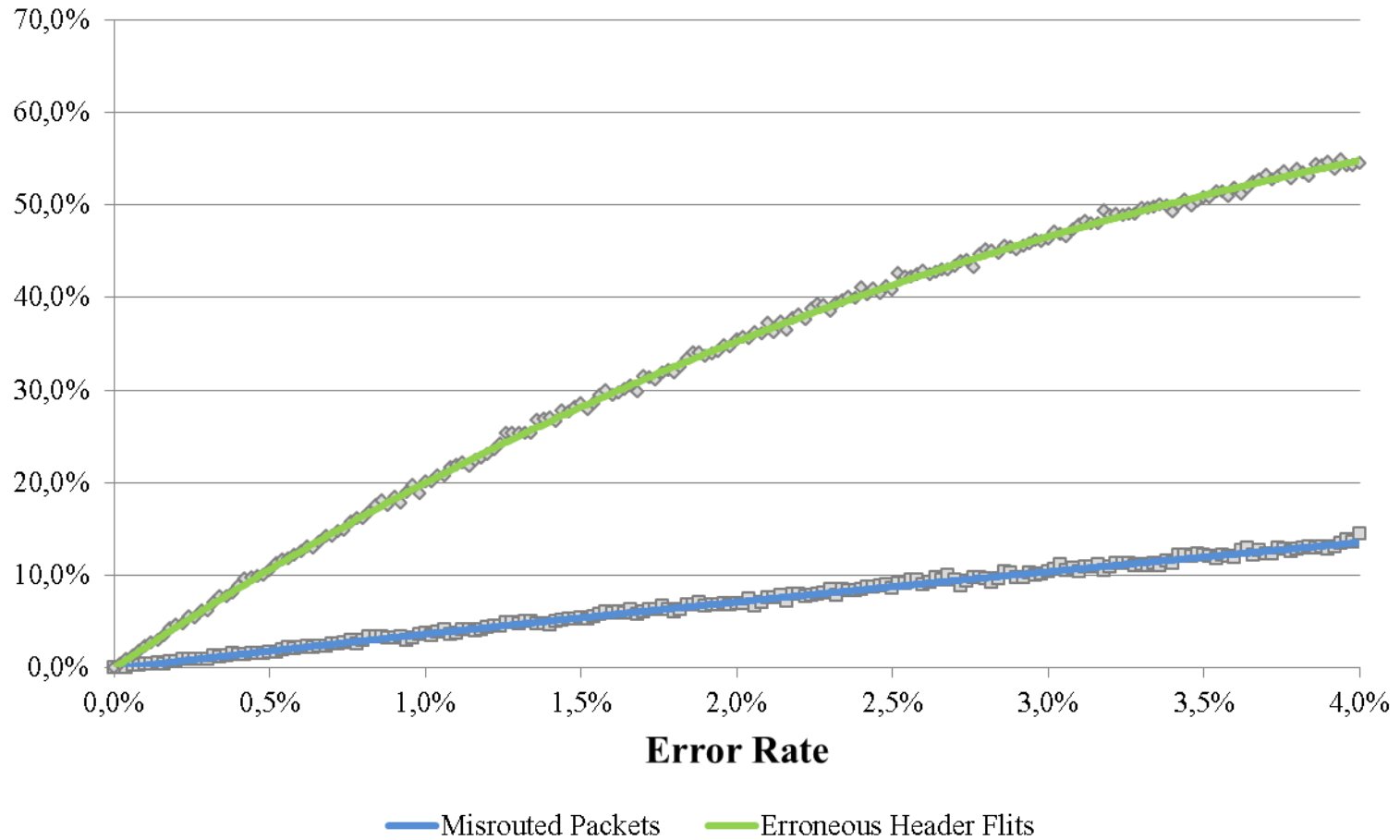
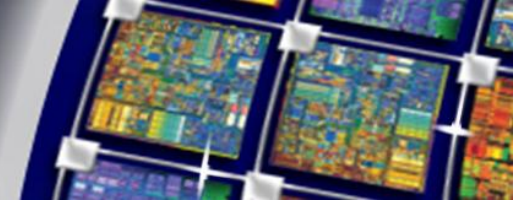


Flitbreite: 128 Bit, Paketlänge: 2

Energieumsatz:

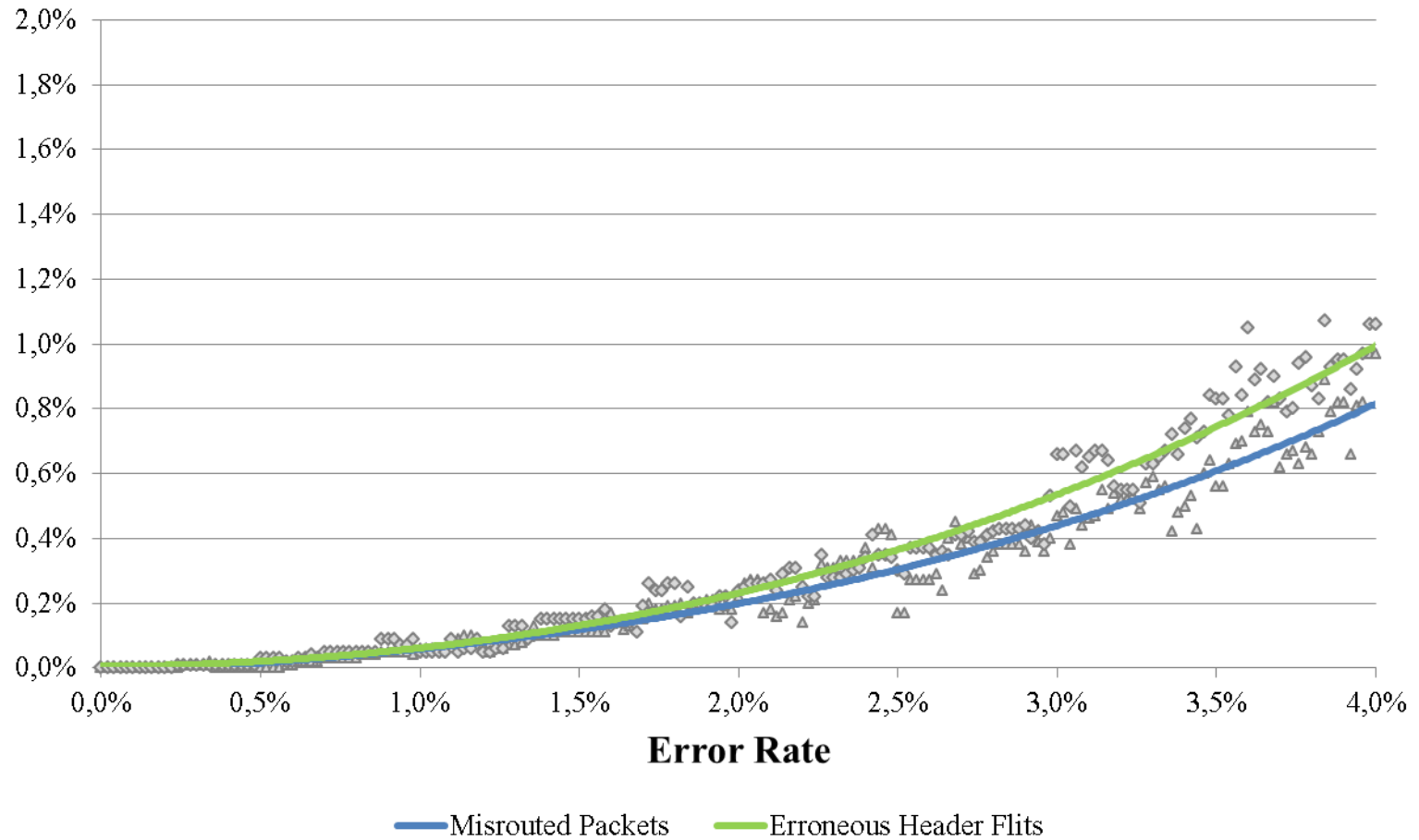
- Sinkt, weil nicht jedes Flit geprüft wird

Ergebnisse – Restfehler



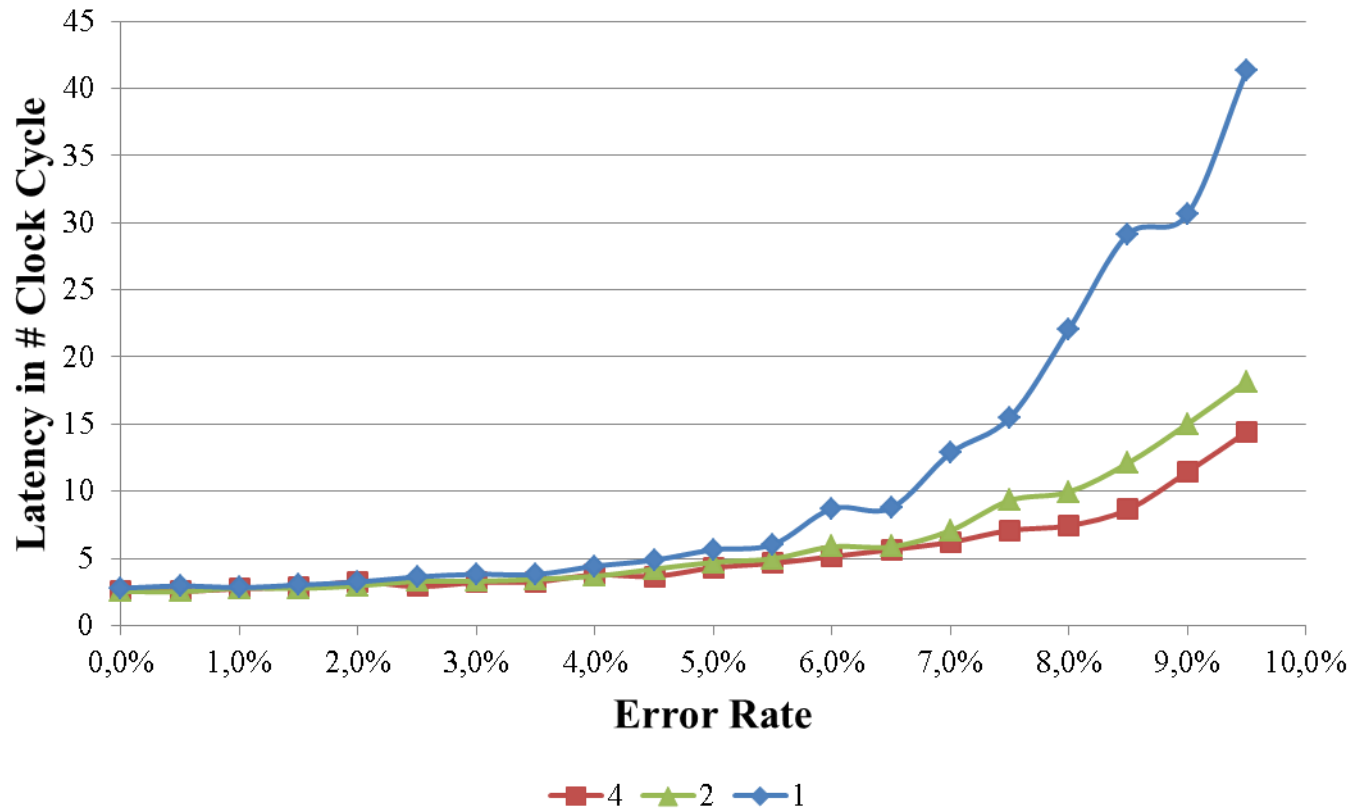
kein Encoding

Ergebnisse – Restfehler



Encoding: EHC(4,1)

Ergebnisse – Latenz



Latenz:

- Steigt nur bei sehr hohen Fehlerraten



Switch to Switch Fehlerbehandlung:

- Prüfung der Header-Flits nötig
- Body-Flits -> End-to-End

Resource Sharing:

- kann Flächenverbrauch senken
- Encoding muss komplex genug sein, hier EHC(13,8)

Ergebnisse:

- Fehlleitungen 25fach reduziert
- Latenz (Takte) wird nicht negativ beeinflusst

Vielen Dank! Fragen?

Kontakt: Martin.Gag@uni-rostock.de



Homepage: www.imd.uni-rostock.de
www.networks-on-chip.com

