Microelectronic System Design Research Group
University Kaiserslautern
www.eit.uni-kl.de/wehn

SoC-Network for Interleaving in Wireless Communications

Norbert Wehn
wehn@eit.uni-kl.de

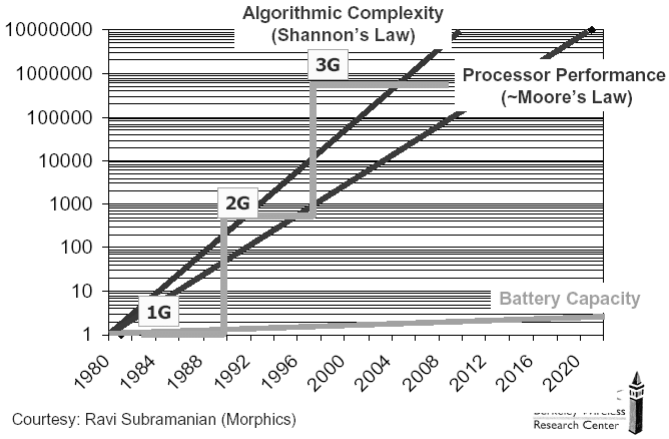
MPSoC'04
5-9 July 2004, Saint-Maximin la Sainte Baume, France

Outline

- **Wireless Implementation Challenges**
- **Interleaving Problem in parallel Architectures**
  - ⇨ **Turbo-Codes**
  - ⇨ **LDPC Codes**
- **Interconnect centric Architectures**
- **Conflict handling**
- **Scalabe Architectures for Channel Decoding**
- **Flexibility Trade-offs**
  - ⇨ **Synthesizable VHDL, AS multiprocessor, FPGA**
- **Conclusion**

Wireless Implementation Challenges I

- **DECT 10 MIPS, GSM 100 MIPS, UMTS x 1000 MIPS**



Courtesy: Ravi Subramanian (Morphics)

Wireless Implementation Challenges II

- **Flexibility**
  - ⇨ **Different QoS, „multi-mode" support**
  - ⇨ **Varying throughput requirements**
- **Low Power/Low Energy**
- **Design Space**
  - ⇨ **algorithms, architecture**

....

Architectural requirements

- **Flexible and scalable**
- **Energy-efficient**
- **Parallel Architectures**
  - ⇨ **Exploit parallelism of algorithm to maximize locality**
  - ⇨ **Efficient communication network**

Wireless Algorithms

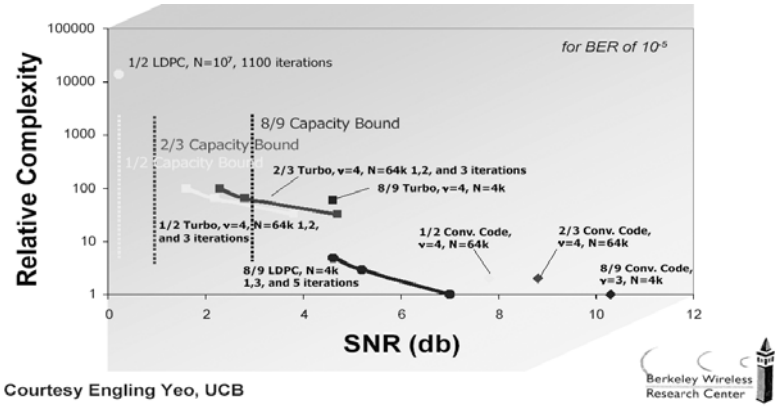
Wireless baseband algorithms

- **Inner modem**
  - ⇨ **signal processing based on matrix computations e.g. multi-user detection, interference cancellation, filtering, correlators**
- **Outer Modem**
  - ⇨ **Channel coding, Interleaving, Data stream segmentation**
  - ⇨ **Parallel architectures not obvious**
- **Channel Coding (forward error correction)**
  - ⇨ **Key for reliable transmission**
  - ⇨ **Performance bound w.r.t. signal-to-noise given by Shannon limit**
  - ⇨ **Key building block in outer modem**
  - ⇨ **Many techniques known**

Channel Coding Techniques I

- **Most efficient Codes: Turbo-Codes, LDPC-Codes**
  - ⇨ **Iterative decoding technique**
  - ⇨ **Computational complexity increased by order of magnitudes**
- **Turbo-Codes (1993)**
  - ⇨ **Revolution in channel coding**
  - ⇨ **E.g. UMTS, DVB, WLAN, CCSDS**
- **LDPC Codes (1996)**
  - ⇨ **Renaissance of Gallagers work (1960)**
  - ⇨ **Competitor to turbo-codes e.g. used in DVB-S2**
- **Very active research area in the communication community**

Channel Coding Techniques II

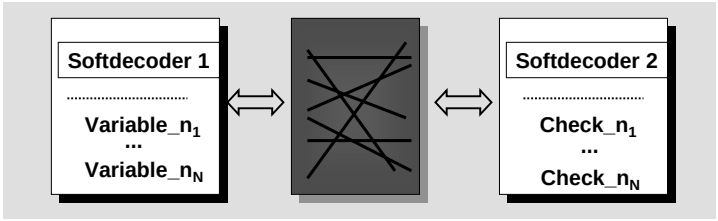


Courtesy Engling Yeo, UCB

- Iterative algorithms on block basis
  - ⇨ Latency
- High throughput architectures
  - ⇨ Data distribution and NOT data calculation is the bottleneck

7

Generic Decoding Structure



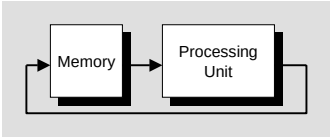
1. Subdecoder 1 processes one complete block
2. After finishing the calculation, data are sent to subdecoder 2
3. Subdecoder 2 starts block processing...
4. Iterate until stopping crtierion fullfilled

- Information exchange takes place „randomly“
  - ⇨ LDPC : Tanner graph (Parity check matrix)
  - ⇨ TC : Interleaver
- Quality of „randomness“ strongly influences communcations performance
  - ⇨ Error floor

8

Architectures

- Serial implementation
  - ⇨ Both subdecoder are serially implemented on a single processing unit
  - ⇨ PU generates 1 data per clock cycle
  - ⇨ Interlaving is a simple adressing problem
  - ⇨ But: very low throughput



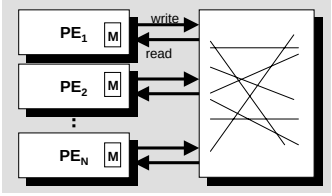
- High Throughput architectures
  - ⇨ Put N serial architectures in parallel
    - – Low architectural efficiency
    - – Large LATENCY
  - ⇨ Better: parallelize the algorithm

Parallel Architectures

- LDPC: inherent algorithmic parallelism
  - ⇨ Each Check-Node and Variable-Node works independently from each other
  - ⇨ Full parallel implementation i.e. instantiating each node
    - – Only feasible for small blocks and no flexibility
- TC: MAP decoding algorithm recursively processes a data block
  - ⇨ Dependency can be broken up by „windowing technique“

↷ Subdecoder parallelism is not a big issue

Partially parallel architecture (N<Blocksize)

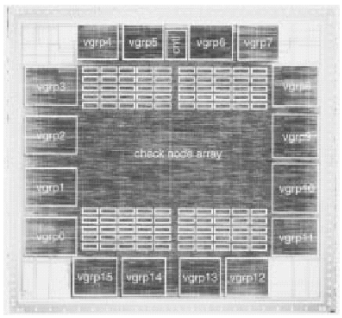


- Computational locality
- But no locality of interconnect

↷ Interconnect centric architecture due to interleaving

Full Parallel LDPC Decoder

- Lucent Chip
  - ⇨ 1 Gbit/s, Rate=1/2, blocksize=1024 bit
- 0.16μm 1.5 V CMOS technology
- Synthesizable VHDL description
- 52.5mm², 1750 Kgates, 64 Mhz

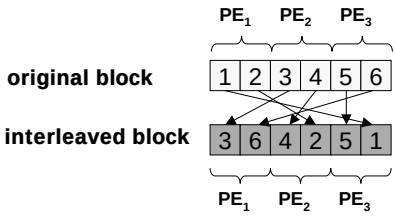


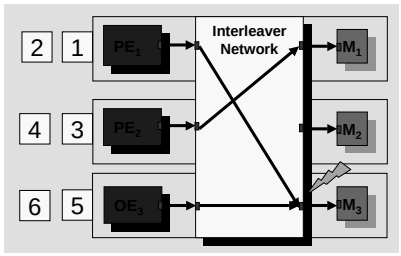
- ⇨ 26624 global nets
- ⇨ Metal 4 & 5 used for top level routing
- ⇨ Special floorplanning and buffer placement tool
- ⇨ Average wire length 3mm
- ⇨ Power consumption dominated by wiring

Full parallel implementation infeasible for large blocksizes

Interleaver Bottleneck

Partially parallel architecture





- Crossbar functionality with *blocking conflict*
- *Throughput/latency* is critical
- *Global routing*
- *Flexibility*
  - ⇨ Blocksizes vary e.g. between 500 and 20000 bits
  - ⇨ About 5000 different interleavers in UMTS

Conflict Handling

1. Conflict free by design
2. Design time conflict resolution
3. Run time conflict resolution

Conflict free by design

- Interleaver is designed according to a fixed architectural template
- But
  - ⇨ Interleaver only optimal for ONE architecture
  - ⇨ Influences communications performance
  - ⇨ No flexibility
  - ⇨ Not compatible with Standard
- Examples: T@MPO Core (IMEC), TurboConcept (Brest)

Design Time Conflict Resolution

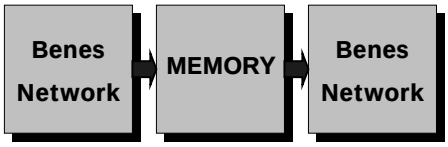
Idea: avoid conflicts at design time for given interleaver

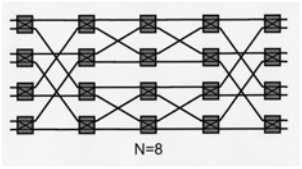
- Minimize access conflicts by suited schedule/assignment
  - ⇨ Stall processing if conflicts are unavoidable
  - ⇨ Throughput degradation
- Use special permutation networks
  - ⇨ E.g. Benes network

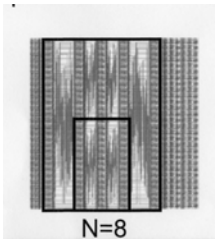
Disadvantages

- Computational complex preprocessing (NP-hard problems)
- Access Pattern must be stored for each interleaver/matrix
  - ⇨ Flexibility costs a huge amount of memory

E.g. Dediacted Permuntation Network







- ⇨ Can process any interleaver
- ⇨ Global wires, pipelining possible
- ⇨ Efficient implementation in full-custom style possible

- ⇨ But: complexity of routing preprocssing, lack of flexibility

Run Time Conflict Resolution

- ❍ Dedicated communication network
- ❍ Maximize locality, scalability

↷ Message passing based networks

- ❍ Topologies
  - ⇨ (chordal) ring, meshes, random...
- ❍ Routing algorithms
  - ⇨ Minimum, deterministic, unicast algorithms
- ❍ Without / with flow control

Advantages

- ❍ Any given interleaver can be supported
- ❍ Negligible latency

But

- ❍ Additional cost for communication network
- ❍ Can be omplex for large degrees of parallelization

Without Flow Control

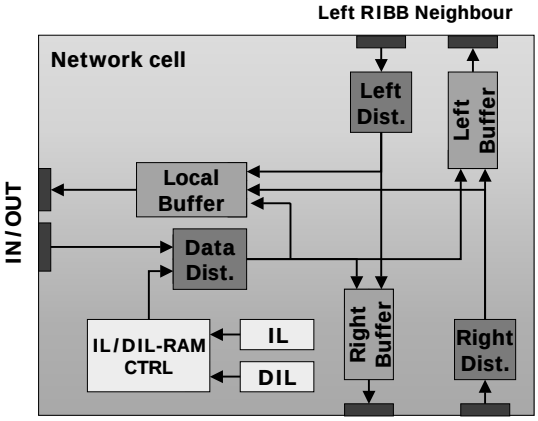
Regular structure (RIBB)

- Network cells connected in a ring
- Nearest neighbour routing, simple physical routing
- But scalability due to ring topology is limited to 8 nodes

Routing decision unit
⇨ determines target buffer

Buffer (FIFO)
⇨ multiple data in, single data out

Throughput
⇨ 1 message / cycle per Link



Flow Control I

- Random Approach (GIBB)
  - ⇨ Random topologies (minimized average distance)
  - ⇨ Balanced shortest path routing
  - ⇨ Limitations on scalability due to topology e.g.
    - Node degree $\Delta = 3$ ⇒ 43 nodes
    - Node degree $\Delta = 4$ ⇒ 310 nodes

<u>No flow control</u>

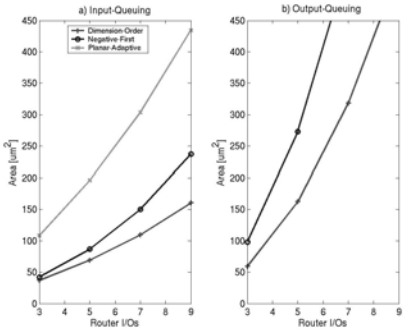
- Buffer sizing is critical
- All possible interleavers must be known at design time

↷ Use of flow control

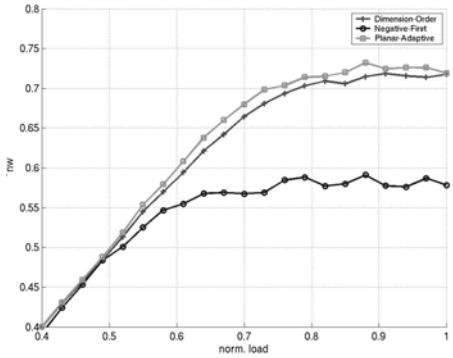
- Buffer sizing is a network parameter
- Introduces some latency
- Deadlock-freedom is essential
  - ⇨ Routing algorithms for regular topologies well known

Mesh Topology

- Limitations on scalability due to 2d-Mesh topology: 16 nodes
- Example: (4,4) 2d-Mesh
  - ⇨ Input Queuing (depth=6)
  - ⇨ Routing algorithms: dimension-order, negative-first, planar adaptive



Router complexity (Input-/Output queuing)



Normalized throughput (input queuing)

Summary Conflict Handling

	Conflict Free	Design Time Conflict Resolution	Run Time Conflict Resolution
Any Standard	No	Yes	Yes
Influence on Code Design	Yes	No	No
Pre-processing	No	Yes	No
Internal Buffering	No	No	Yes
Additional Latency	No	If Stalls allowed	Negligible
Additional Memory	No	Yes	No

Scalable Decoder I

Application of RIBB communication network to scalable decoder architectures

- **VHDL model, Synopsys Tools, 0.18 µm Standard Cell Library**

3GPP compliant Turbo-Decoder

- **Log-MAP, blocksize 5114, 6 iterations, SMAP with 3 SMUs**

Parallel Units	1	4	6	8
Area [mm²]	3.9	9.2	13.0	17.3
Energy / Block [uJ]	48.7	51.7	50.9	55.2
Throughput [Mbit/s]	11.7	39.0	59.6	72.7
Efficiency [norm]	1	1.32	1.47	1.24

21

Scalable Decoder II

(3,6) LDPCode

- **Code rate=1/2, blocksize=10200 bit, 40 iterations**

Check_nodes Variable_nodes	1 2	2 4	5 10
#messages/cycle	6	12	30
RAM [mm²]	12.11	14.66	17.84
RIBB [mm²]	1.97	3.96	10.2
Total area [mm²]	14.31	19.19	29.38
Energy / Block [uJ]	819	686	531
Throughput [Mbit/s]	5.4	10.6	22.5
Efficiency [norm]	1	1.75	3.14

22

Flexibility Trade-offs

- Flexibility („software defined radio")
  - ⇨ Different QoS , „multi-mode" support, varying throughput requirements

What are the costs for flexibility ?

- Investigations of Turbo-Decoder design space under 3GPP conditions
- Communication network
  - ⇨ Run time conflict resolution
  - ⇨ RIBB based communication network
- Different implementation styles (0.18μm technology)
  - – Scalable synthesizable VHDL implementation
  - – Multiprocessor solution
  - – FPGA implementation
- Measure
  - ⇨ Architectural efficiency (throughput/area)
  - ⇨ Design effort

AS Multiprocessor Solution

- Application-customized Xtensa Core
  - ⇨ Increased ILP by e.g. dedicated butterfly and transfer instructions
    - Outperforms state-of-the-art VLIW DSP for turbo-decoding
    - Extended core ~ 100 Kgates
  - ⇨ Single-cycle data interface for interprocessor communication
  - ⇨ Heterogeneous communication network
    - Extended ring interleaver network with buses

Number of Processors	Throughput [Mbit/s]	Area [mm²]	Efficiency [norm]
1	1.48	6.42	1
8	11.58	20.91	2.58
16	22.64	36.98	2.66
32	43.25	70.26	2.67
40	52.83	87.47	2.62

Validated with Tensilica Xtensa API Interface, Tensilica ISS simulator

Implementation Comparisons

- Scalable VHDL implementation versus AS Multiprocessor
  - ⇨ Design Effort x 1.6
  - ⇨ Architectural Efficiency x 5-8

- FPGA Implementation

Device	Parallelization	Throughput	Utilization	Frequency
Xilinx Virtex II-3000	4 MAP units	22 Mbit/s	83% Slices 70% LUT, 41% RAM 3.1 million GE	88.2 MHz

- Comparison for 22Mbit/s throughput

	AS Multiproc	FPGA	AS HW
Design Effort	1	x 1.6	x 1.6 (no place & route)
Architectural Efficiency	16 processors 37 mm^2	1 FPGA	2 MAP units 5 mm^2

25

Conclusion

- Advanced channel decoder architectures for high throughput are interconnect centric architectures
- Different possibilities to resolve interleaving conflicts in partial parallel architectures
- Run time conflict resolution is the most flexible
- Efficient communication networks are mandatory

Thank you for listening!

For further information please visit

http://www.eit.uni-kl.de/wehn

You can download papers describing the techniques presented in this talk

26