



Implementation of a Re-configurable Multi-mode Baseband Terminal Receiver for Enhanced 3G

Klaus Strohmenger
Nokia Research Center



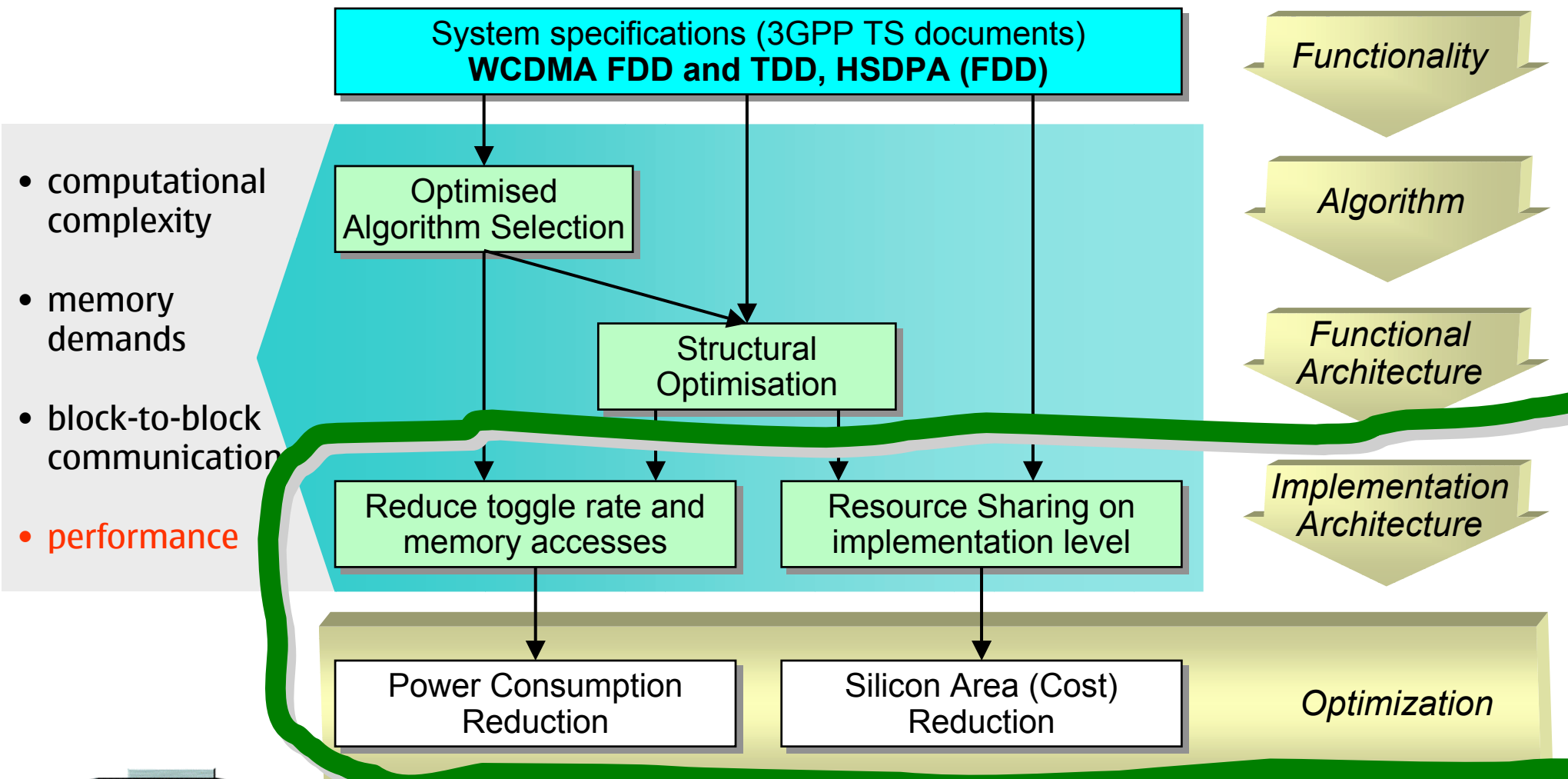
MuMoR Workshop, Lausanne, 26.02.2004 *Klaus Strohmenger* *Page 1*
MUMOR – IST-2001-34561



The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information as its sole risk and liability.

- Overview of multi-mode optimization in baseband area
- Influence of HW/SW Partitioning on implementation complexity
- Multi-mode optimization by HW sharing between Cell Searcher and H-ARQ
- Soft-configurable implementation approaches
 - Available technologies
 - Advantages, challenges and drawbacks
- Example solution for bit-level processing domain
 - Performance and design flow driven solution
 - Power and area optimization driven solution

Multi-mode optimizations





Influence of HW/SW Partitioning on Final Complexity

HW/SW Partitioning Metrics

- Performance and speed issues
- Form factor, manufacturing cost
- Reliability and multi-mode flexibility
- Power consumption

- Availability of components and tools
- Development and design time & cost
- Capability for late spec changes

System Functions

Filter functions

Memory functions

Complex algorithms

Bit-level operations

Implementation

Dedicated HW

SW on GPP or DSP

Soft-configurable technology

Resulting Complexity (Trade-Offs)

• **Computational complexity** (arithmetic and logical operations done in HW, DSP, etc.)

Design analysis and reports

• **Memory requirements** (size, speed, access) demanded by the selected implementation

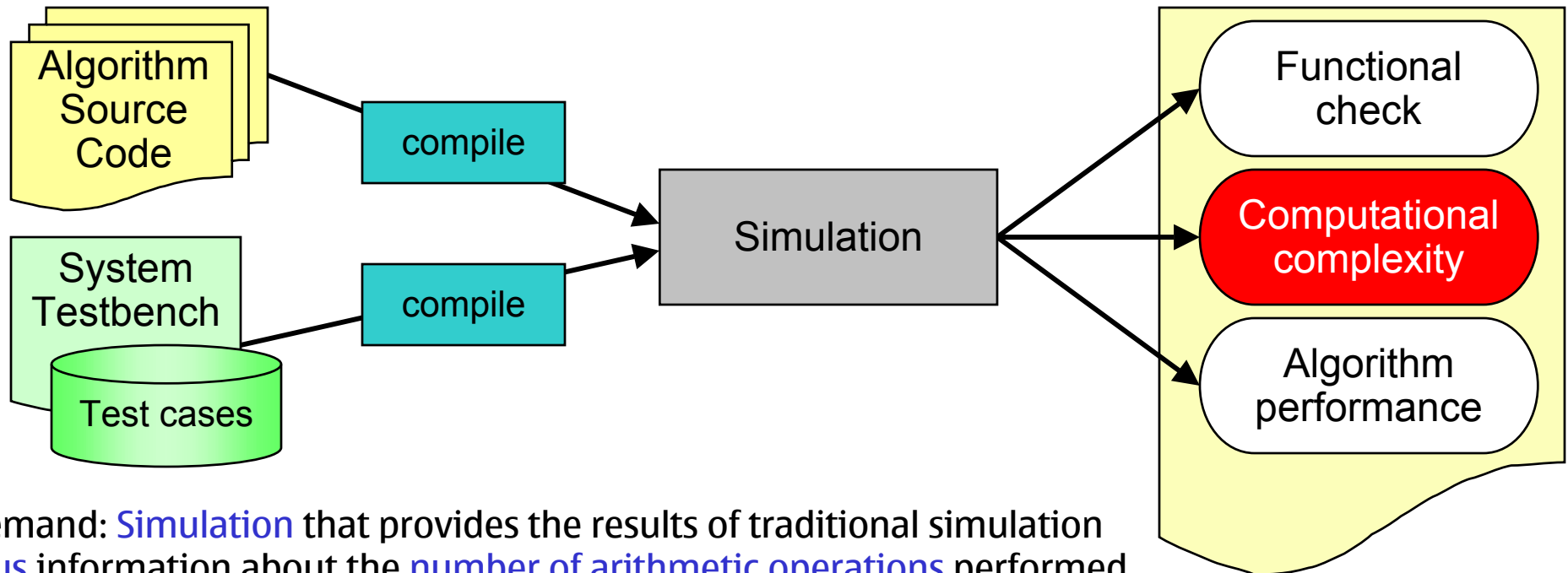
• **Communication bandwidth** of the building blocks and beneficial bus features

GPP: General Purpose Processor





Computational Complexity Analysis for Partitioning



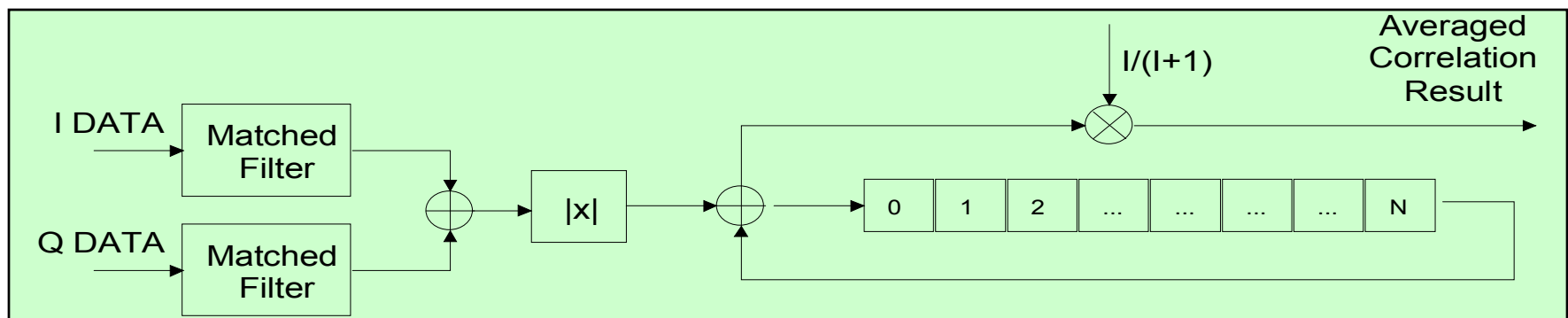
Demand: **Simulation** that provides the results of traditional simulation **plus** information about the **number of arithmetic operations** performed during a simulation for specific test cases to...

- ▶ Compare complexity of one algorithm for **different conditions** (test cases)
- ▶ Compare complexity before and after algorithm **optimisations**
- ▶ Compare different algorithms for **performance and complexity simultaneously**
- ▶ Use complexity numbers as **basis for HW/SW partitioning**



Multi-mode Optimization by HW Sharing (Memory)

- **Scenario:** In **FDD** mode both UMTS and HSDPA channels will be used, but in **TDD** mode only UMTS channels → **FDD centric terminal**
- In **FDD** the primary synchronization code within the SCH is transmitted once per **slot**, in **TDD** this is done once (or twice) per **frame**, I.e. every 15 slots



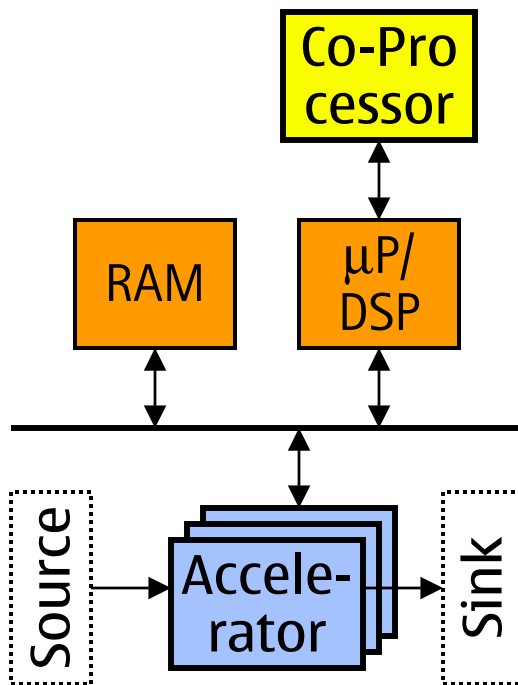
- The **amount of memory** required to store the intermediate correlation results is **much bigger in TDD**. This can be up to several 10k's of soft bit due to oversampling
- In our scenario **this memory can be used in FDD** e.g. as for the incremental redundancy (IR) data of the H-ARQ feature of HSDPA (size up to 29k soft bits)
- Implementation **architecture** has to be adapted, however the **scheduling** is not affected!



HW-Implementation of Re-configurability

- How to cope with **computational hot-spots in a flexible system?**
- 3 categories of flexible architectures can be identified:
 - **Monolithic microprocessor with extensions:**
ASIC-like parts speed up the general purpose processor or DSP for certain (kinds of) computations or functions
 - **FPGA-like structure** with big number of low-complexity processing units:
Symmetrical matrix interconnection network of either identical or varying low-complexity processing units doing parallel processing
 - **Multiprocessor** concepts:
Network of complex processors communicating over a common processor bus and/or via shared memory
- We prefer the **first concept** using a structural flexible/programmable HW to accelerate a DSP as autonomously as possible (here referred to as “**Accelerator**”)

- Definition
 - Accelerator is a block added into the implementation architecture, which speeds up software and adds new functionality to the system.



- Co-Processor
 - Tight coupling with DSP/uP
 - Extension of the instruction set
 - Specific to a DSP/uP
 - Effects DSP/uP pipeline execution

- Accelerator/ASIP
 - Loose coupling with DSP/uP
 - Own set of instructions
 - Independent of DSP/uP
 - Autonomous operation

ASIP: Application Specific Instruction Set Processor



Soft-configurable Implementation (Accelerator, etc.)

	Advantages of soft-configurable implementation	Challenges and Drawbacks of soft-configurable implementation
Compared to pure HW	<ul style="list-style-type: none"> • Increased flexibility compared to normal HW components through direct control and configuration by the attached processor • Due to the increased flexibility the HW component can be used for several (similar) tasks and thus offers potential for better utilization of the silicon area 	<ul style="list-style-type: none"> • In processor-coupled solutions the efficiency can only be exploited, when the data processing is done in data blocks, given a minimum block size • Beside the HW development, which has to be done if a well tailored processing element is not available, SW has to be created and both tested together
Compared to pure SW	<ul style="list-style-type: none"> • The execution speed can be increased by using better tailored HW for algorithms solution than a DSP normally has • Especially operations on bit-level can be implemented much faster and more power efficient 	<ul style="list-style-type: none"> • Utilization of accelerator has to be done explicitly by reserved commands in the SW code • Beside the SW development, HW has to be created (done if a well tailored processing element is not available) and both tested together





DSP Accelerator for Bit-level Processing (Motivation)

- Motivation

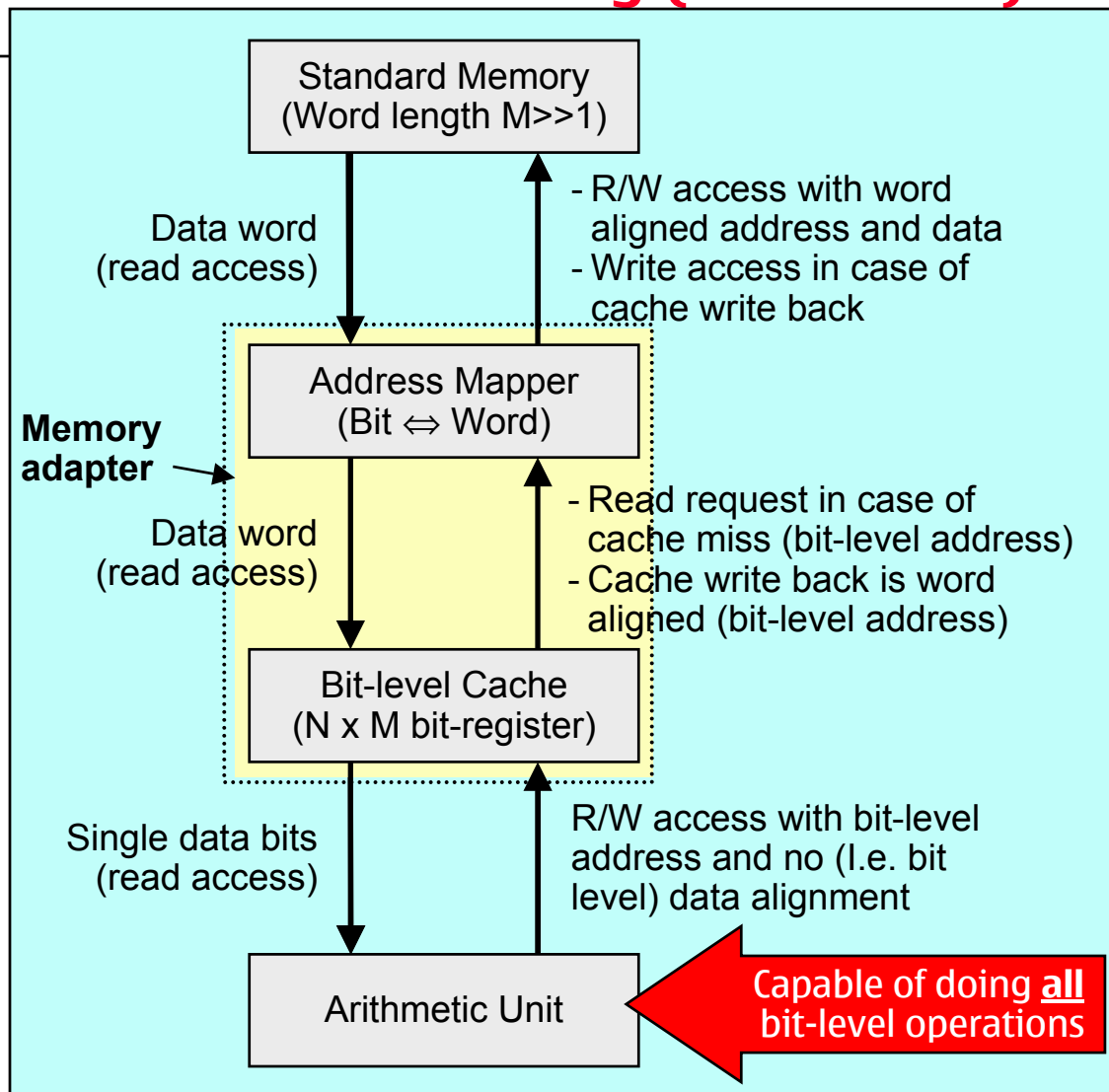
- In PHY Layer Tx and Rx are typically a lot of operations on bit level
- RAMs and DSP operate on words (trouble for pure SW-Radio)

- Target

- Fully exploit the RAM capacity (reduce silicon area)
- Decrease number of memory accesses (reduce power cons.)

- Idea

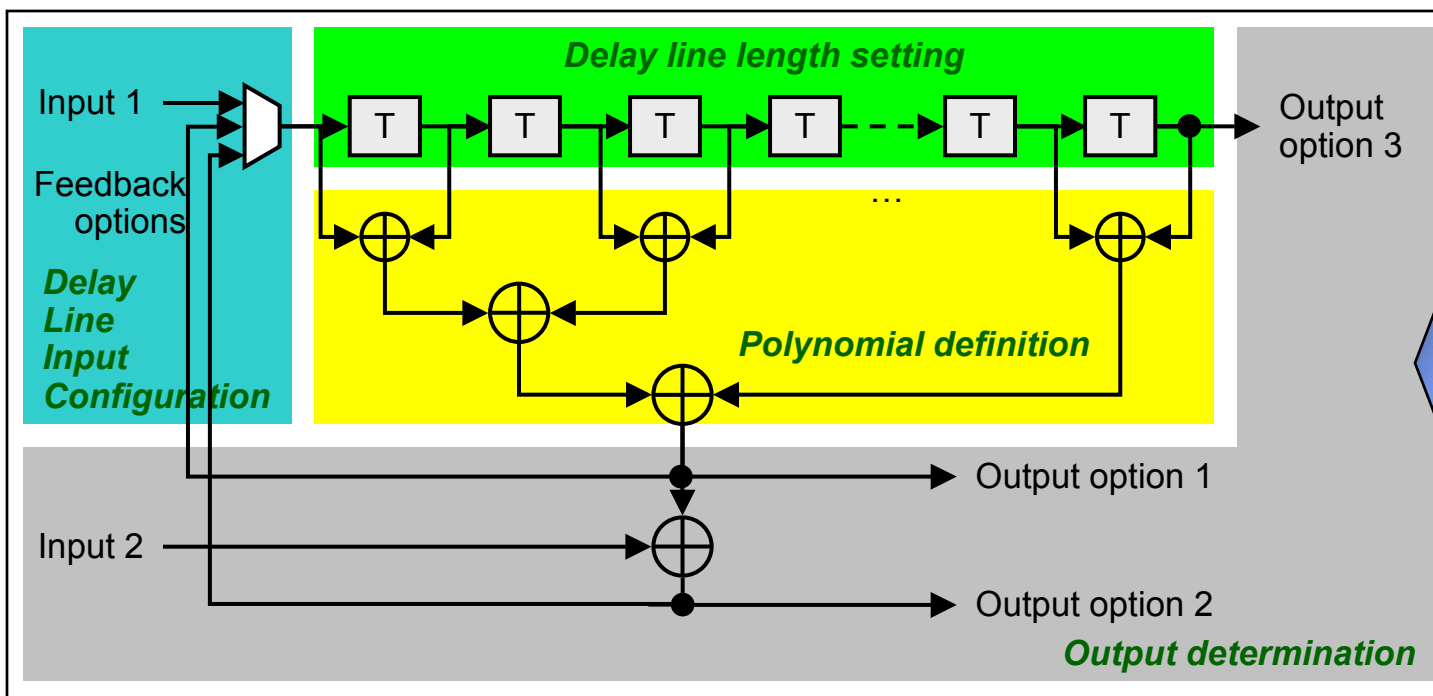
- Bit-to-word memory adapter with cache functionality
- !** But: The area overhead caused by the memory adapter has to be minimised!





DSP Accelerator for Bit-level Processing (Solution)

- Minimize the number of memory adapters to save most additional HW overhead caused by this new component (ideally use only *one* for Tx and Rx)
- This requires to use just *one arithmetic unit*, which is capable of doing *multiple operations* by setting few configuration signals



CRC Encoder	(F, T, H)
CRC Decoder	(F, T, H)
Bit Scrambler	(T)
Bit De-Scrambler	(T)*
Convolutional Enc.	(F, T)
Turbo Encoder	(F, T, H)

**operating on the sign-bit of the soft-bit representation*

F: FDD
T: TDD
H: HSDPA



The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information as its sole risk and liability.



Computations in Specified Reference Channels

Testcase Channels	Time [ms]	Data Block Sizes [bits]					Rx CRC Dec	Data processing [kbps]
		CRC Enc	ConvEnc	TurboEnc	TDDBitSc	TDDBitDS		
HSDPA								2362.5
DL: TS25.101 A.7.1.3 Fixed Reference Channel Definition H-Set 3 (2332 kbps, 16QAM)								2362.5
HS-DSCH	2	-	-	-	-	-	4688	2344
HS-SCCH	2	-	-	-	-	-	37	18.5
FDD								1943.8
DL: TS25.101 A.3.4 DL reference measurement channel (384 kbps)								388.4
DTCH	10	-	-	-	-	-	3856	385.6
DCCH	40	-	-	-	-	-	112	2.8
UL: TS25.101 A.2.4 UL reference measurement channel (384 kbps)								1555.4
DTCH	10	3856	-	11580	-	-	-	1543.6
DCCH	40	112	360	-	-	-	-	11.8
TDD								1159.4
DL: TS25.102 A.2.5 DL reference measurement channel (384 kbps)								1049.2
DTCH	10	-	-	-	-	6557	3856	1041.3
DCCH	40	-	-	-	-	204	112	7.9
UL: TS25.102 A.2.1 UL reference measurement channel (12.2 kbps)								110.2
DTCH	20	260	804	-	724	-	-	89.4
DCCH	40	112	360	-	360	-	-	20.8

HS-DSCH: High-Speed Downlink Shared Channel

HS-SCCH: High-Speed Shared Control Channel

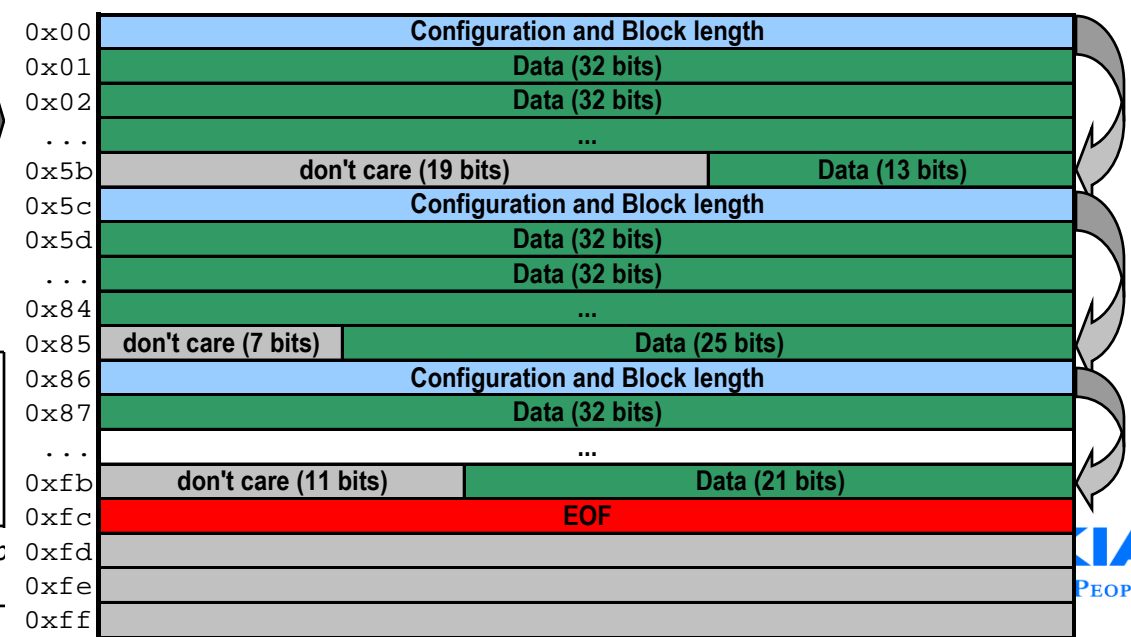
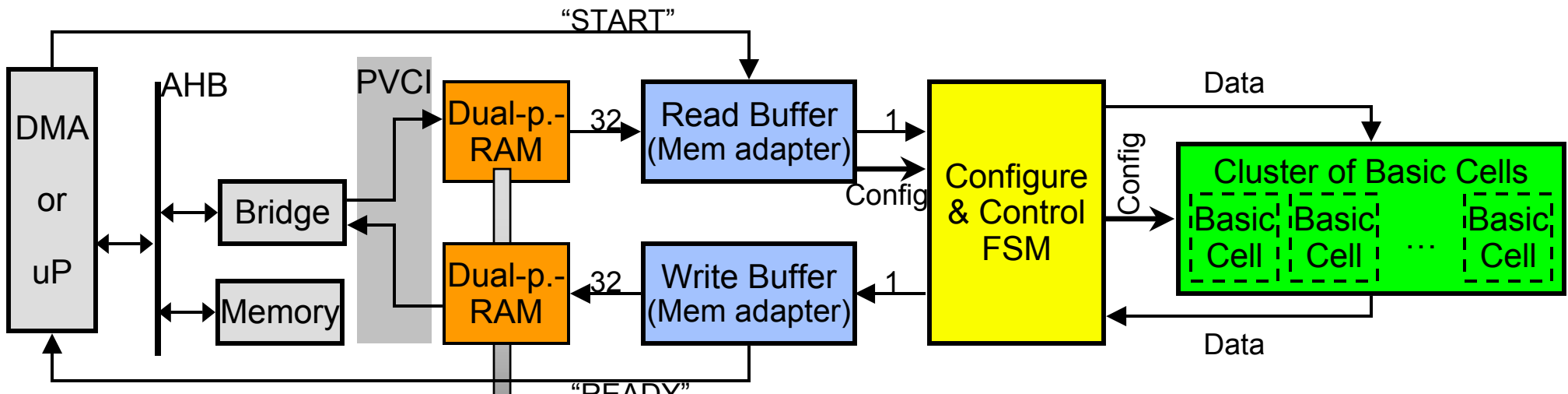
DTCH: Dedicated Transport Channel

DCCH: Dedicated Control Channel





Performance Driven Architecture of the Accelerator



AHB: AMBA High-speed Bus
 AMBA: Advanced Microprocessor Bus Architecture (ARM)
 PVCI: Peripheral Virtual Component Interface (VSIA)



MuMoR Workshop



The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information as its sole risk and liability.



DSP Accelerator for Bit-level Processing (“Basic Cell”)

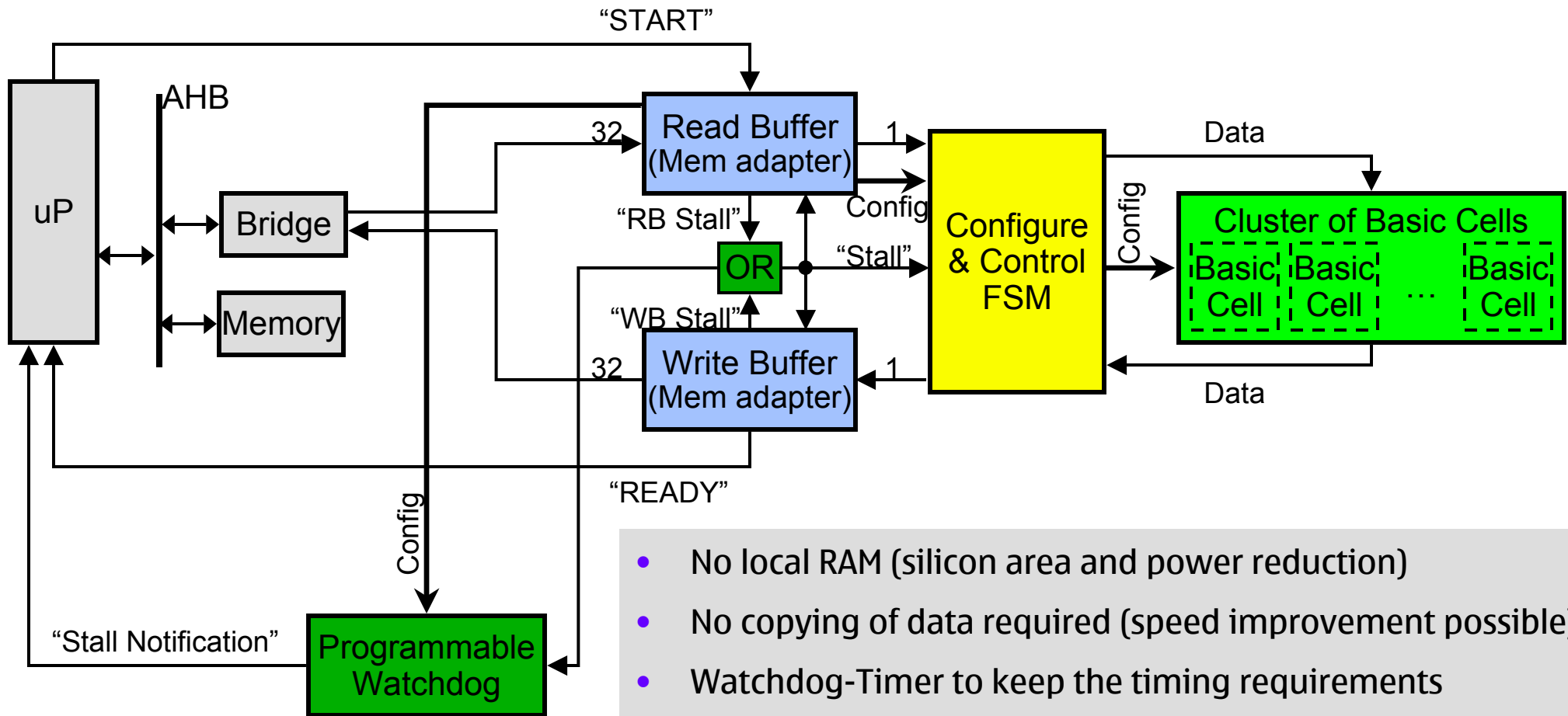
- Implementation using basic cell (4xT) to be e.g. concatenated 6 times (for 24 bit CRC polynomial) ⇒ Scalable for other applications
- When all input data is available locally the accelerator works independent of its environment
 - Required processing time is fully deterministic
 - Important for real-time constraints
 - Minimal interaction with host processor
 - Simple scheduling in HW/SW co-design
- FPGA Device Utilization (Xilinx Virtex-II 6000)
 - IOs required for PPCI (not tied to actual accelerator)
 - Local RAMs: 2 x (2k x 32bit)
 - Actual design is very small

Device Utilization for 2V6000ff1517

Resource	Used	Avail	Util.
IOs	117	1104	10.60%
Global Buffers	1	16	6.25%
Function Generators	845	67584	1.25%
Dffs or Latches	360	70896	0.51%
Block RAMs	32	144	22.22%
Block Multipliers	0	144	0.00%
Block Multiplier Dffs	0	5184	0.00%



Power and Area Optimised Architecture



- No local RAM (silicon area and power reduction)
- No copying of data required (speed improvement possible)
- Watchdog-Timer to keep the timing requirements
- uP has to take more responsibility for timing (→ software)

- Multi-mode optimizations in the investigated systems WCDMA FDD and TDD and HSDPA are possible on implementation level
 - A proper selection of algorithms beforehand is appreciated
 - Computation elements and memory can be shared (support of the architecture)
- Soft-configurable implementations like the presented accelerators offer benefits
 - Because of more flexibility the hardware can be shared between the modes
 - More optimal implementation in terms of power consumption compared to DSP
 - Price: A certain overhead during development has to be challenged (co-design and co-verification)