

Software Defined Radio – Just a Problem of Programming?

Prof. Dr.-Ing. Gerd Ascheid
Venkatesh Ramakrishnan, Martin Witte, Torsten Kempf

- **The *Software Defined Radio (SDR)* Vision**
- **Challenges in Mobile Terminal Design**
- ***Hardware Abstraction Layer* Approaches**
- **The Nucleus approach of UMIC**
- **Outlook**

- **Portability**

- Software is portable onto different platforms
Standard.exe → *Device_1*, ..., *Device_n*

- **Interoperability**

- Different standards

**Contradicting requirements:
Flexible (programmable) &
energy efficient**

- **Flexibility**

- Platform is capable of running different standards
Device ← *Standard_1.exe*, ..., *Standard_n.exe*

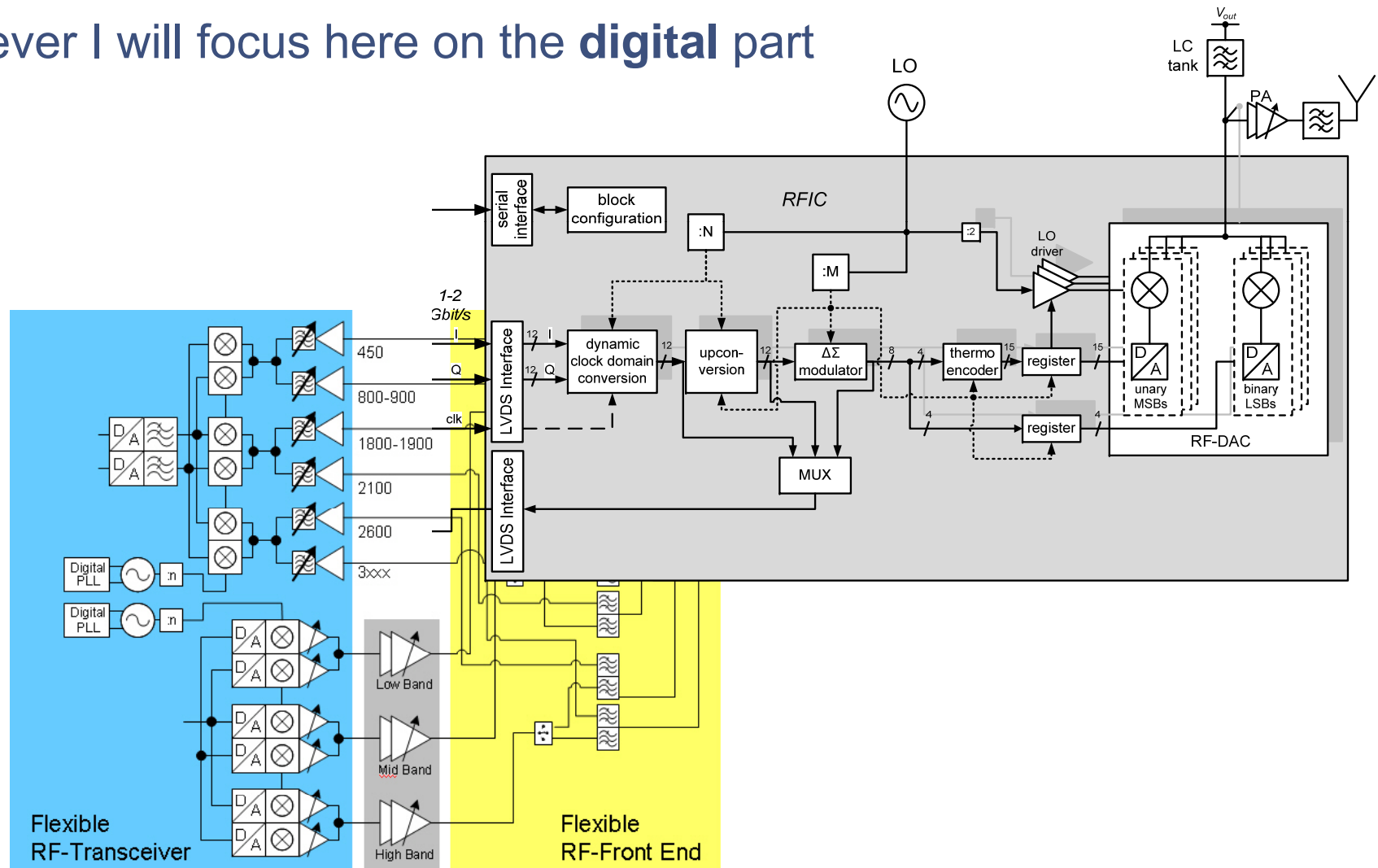
- **Efficiency**

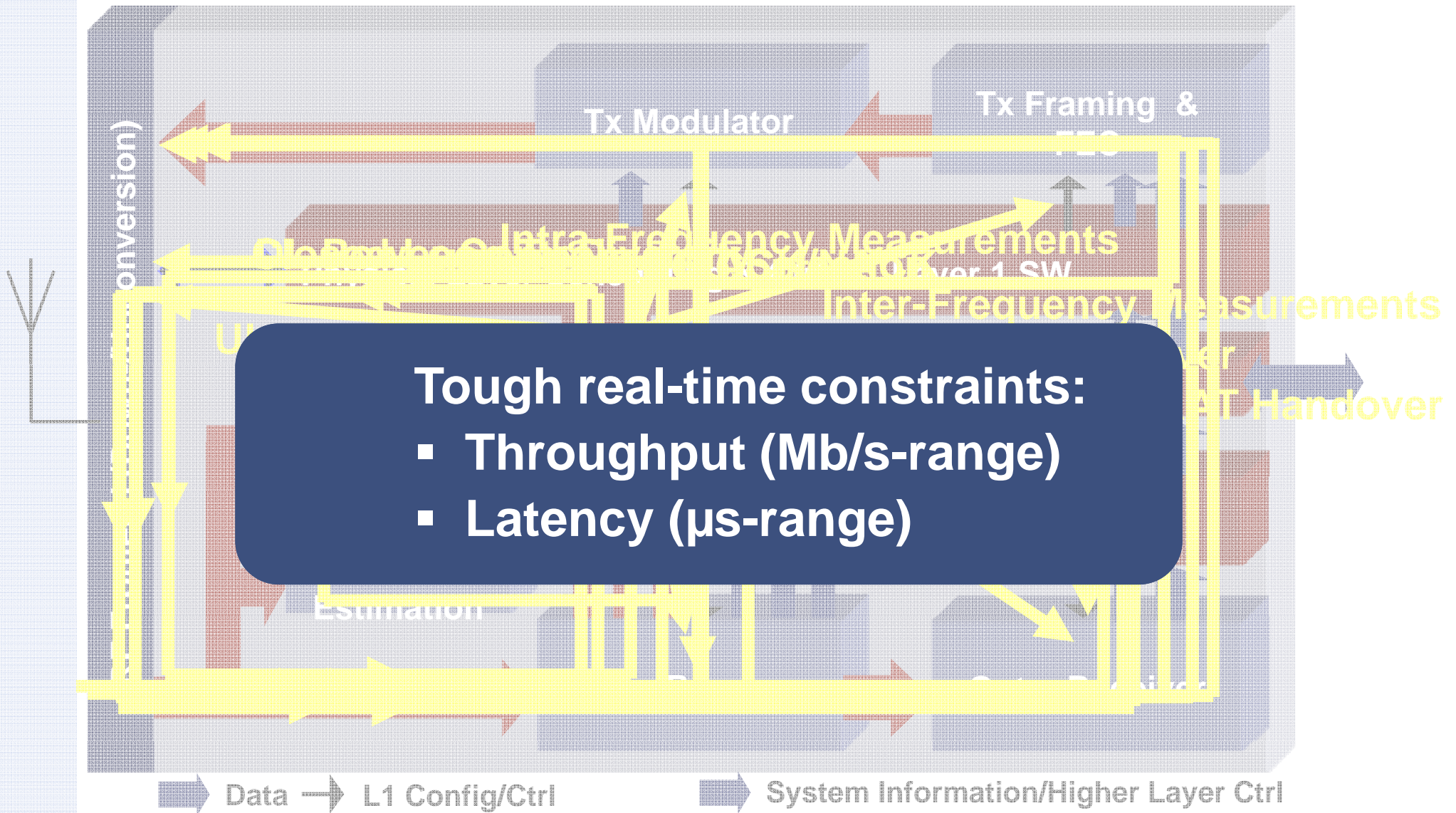
- Power consumption of flexible SDR must be close to power consumption of dedicated device (*battery driven!*)

- **The *Software Defined Radio (SDR)* Vision**
- **Challenges in Mobile Terminal Design**
- ***Hardware Abstraction Layer* Approaches**
- **The Nucleus approach of UMIC**
- **Outlook**

SDR also requires dramatic progress
in **RF front end** design and technology

However I will focus here on the **digital** part



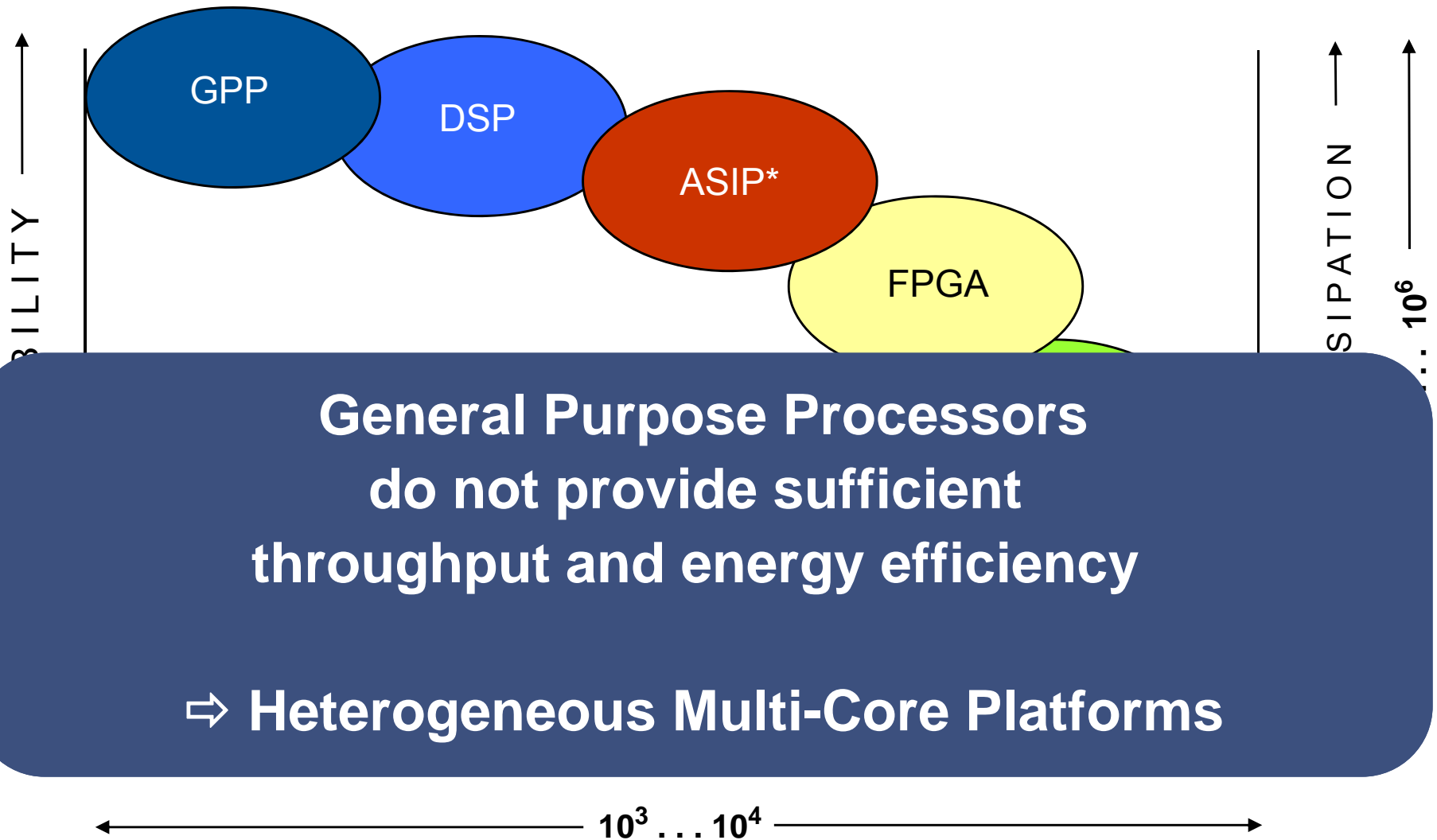


Source: Infineon

**Massive Parallelism required
in the foreseeable future**

	2003	2009	2013
Frequency (MHz)	300	600	1500
GOPS	0,3	14	2458
Operations per Cycle	1	23	1638

Clock frequency increase not sufficient
 ⇒ **Massive parallelism required**
 ⇒ **Multi-core systems**



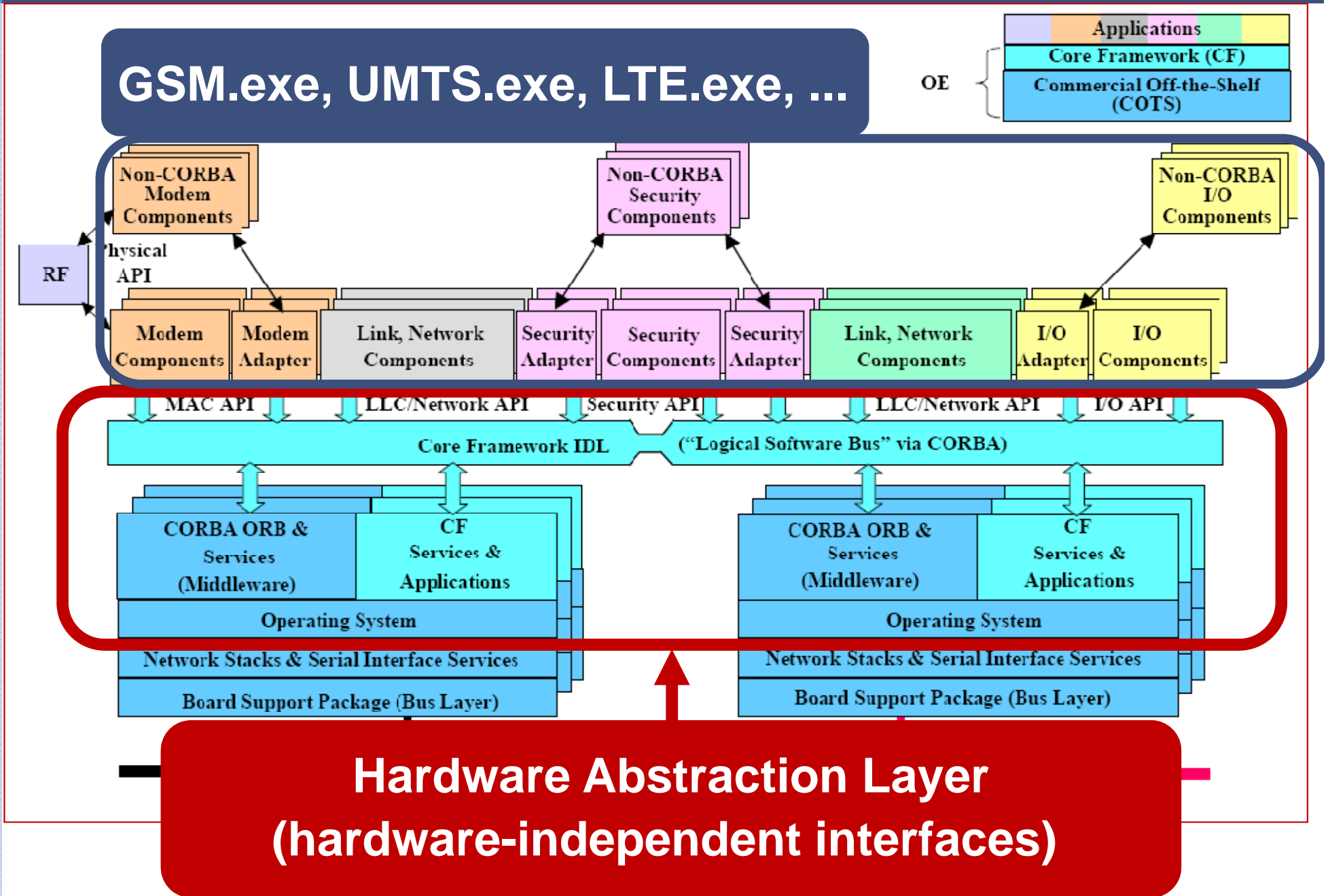
**General Purpose Processors
do not provide sufficient
throughput and energy efficiency**

⇒ Heterogeneous Multi-Core Platforms

Source: T. Noll, RWTH Aachen

* ASIP: Application Specific Instruction-Set Processor

- **The *Software Defined Radio (SDR)* Vision**
- **Challenges in Mobile Terminal Design**
- ***Hardware Abstraction Layer* Approaches**
- **The Nucleus approach of UMIC**
- **Outlook**



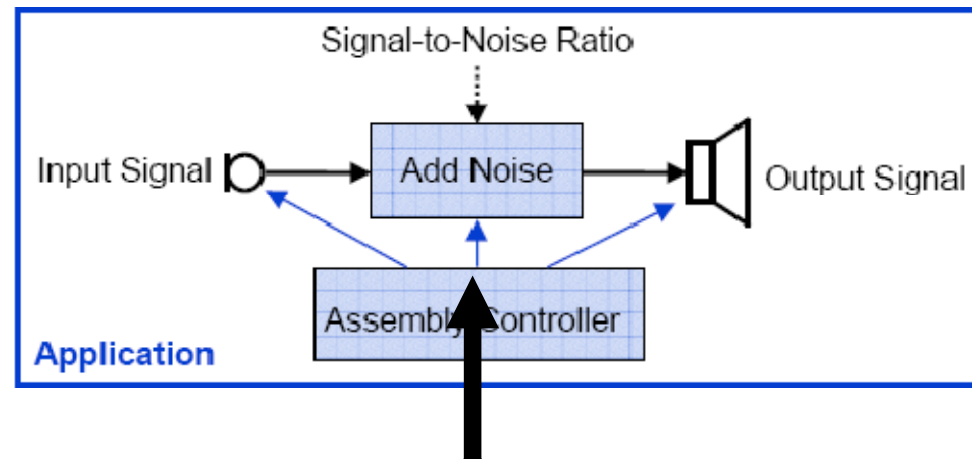
- Independent ORB Express Benchmarks performed by Boeing

Description	Test Span	O/S	Delay Between Invocations
Scenario 1	client/server on same machine	Solaris	none
Scenario 1a	client/server on same	Solaris	70 milliseconds
Scenario 5	machines separated by 10 Mb Ethernet	LynxOS	70 milliseconds
Scenario 6	client and server machines separated by 10 Mb Ethernet	Solaris & LynxOS	70 milliseconds
Scenario 7	client and server machines separated by 10 Mb Ethernet	LynxOS & Solaris	70 milliseconds

Scenario 1a Test	Number of Argument Bytes	Sockets	HARDBest	ORBexpress	TAO
CR Primitives	144	0.230	1.090	0.364	0.977
	24,016	0.642	2.893	0.958	1.603
	144	0.230	1.092	0.383	1.176
	24,016	0.642	5.382	4.957	7.980
	144	0.230	NP	0.9	3.4
	24,016	0.642	NP	24.3	78.9
		Number of Argument Bytes	Sockets	ORBexpress	TAO
CR Primitives	144	0.148	0.268	1.148	
	24,016	1.476	1.886	3.453	
CR NA Records	144	0.148	0.285	1.170	
	24,016	1.476	4.878	21.620	
CR Any	144	0.148	0.595	2.771	
	24,016	1.476	12.950	52.620	

CORBA ORB communication adds around 0.3 ... 10 ms latency !

Source: <http://www.ois.com/>



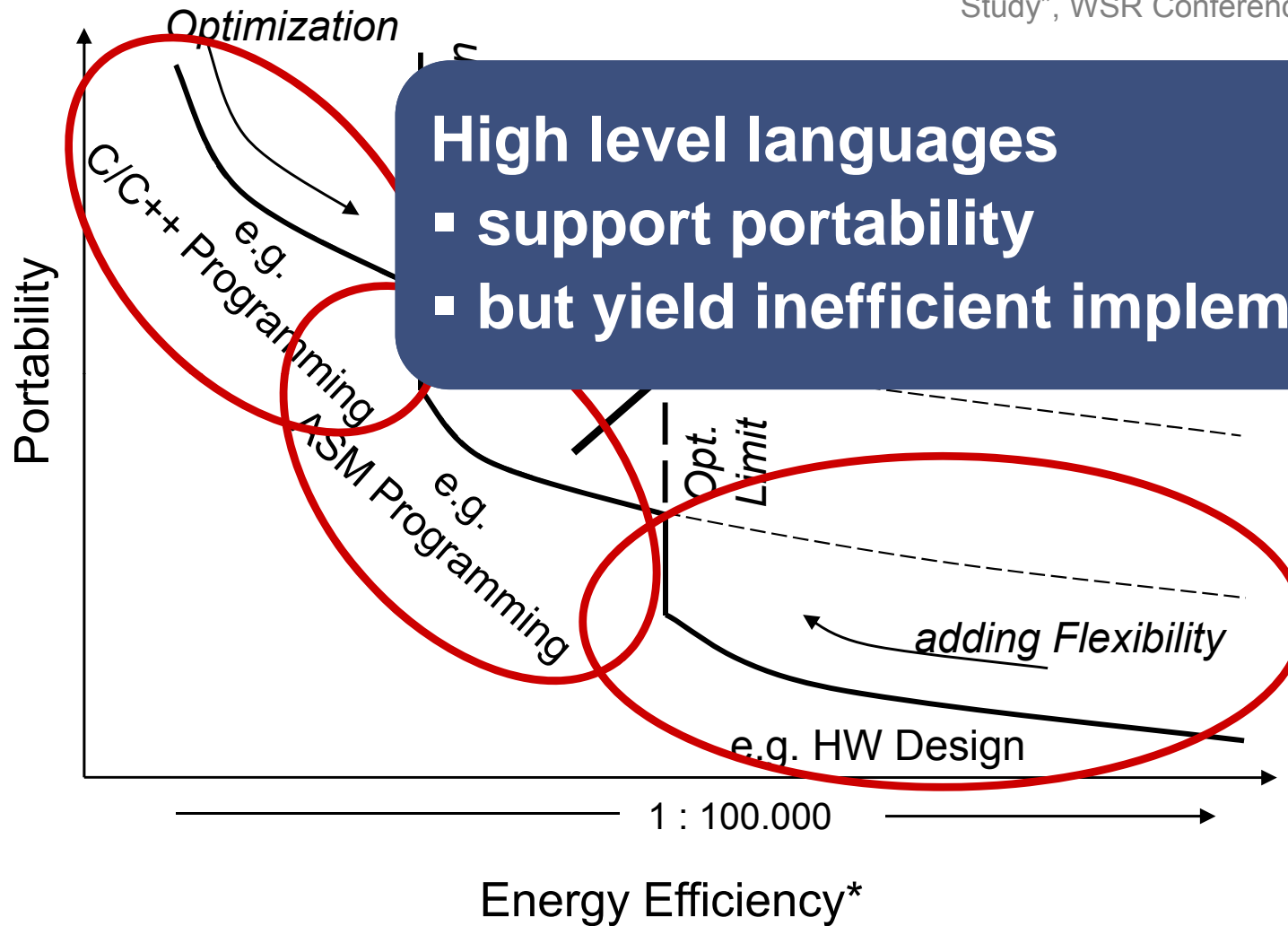
121 lines of functional code lead to
1175 lines of generated code for SCA

Unsuitable for

- Baseband Processing
- Medium Access Control (MAC)

- **The *Software Defined Radio (SDR)* Vision**
- **Challenges in Mobile Terminal Design**
- ***Hardware Abstraction Layer* Approaches**
- **The Nucleus approach of UMIC**
- **Outlook**

Witte et al.: "SDR Baseband Processing Portability: A Case Study", WSR Conference 2008



High level languages

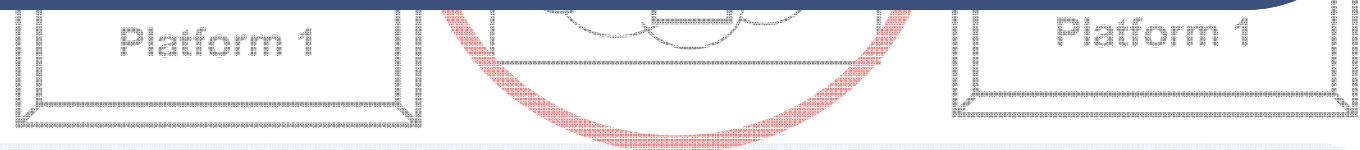
- support portability
- but yield inefficient implementation

* Energy Efficiency = 1 / Energy_Consumption_per_Task

These approaches do not offer sufficient reusability

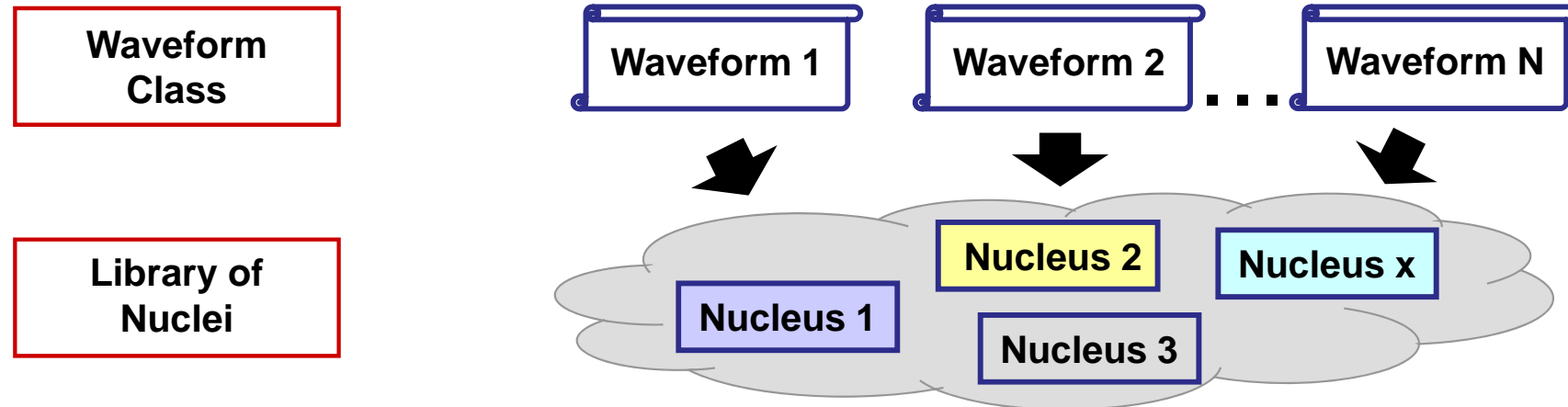
We need

- a flexible waveform description („WDL“)
for functionality and constraints
- a constraint-aware mapping approach (SW \Rightarrow HW)
guaranteeing throughput, latency and energy efficiency



The critical part is the library:

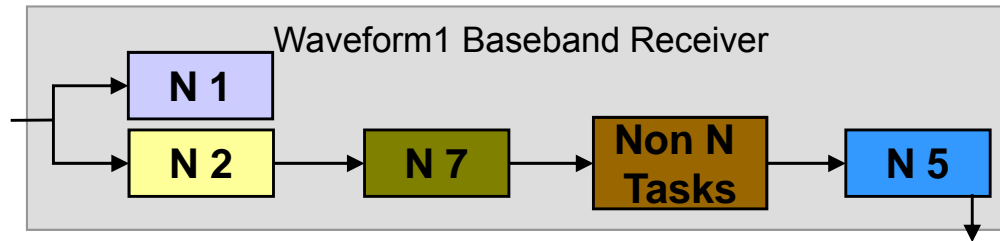
- Proprietary
- Waveform specific components
- Platform specific implementation of components



Nucleus Approach

- Critical, demanding, *algorithmic* kernel
- Captures common kernels across different waveforms
- Neither waveform nor hardware specific

Waveform Description

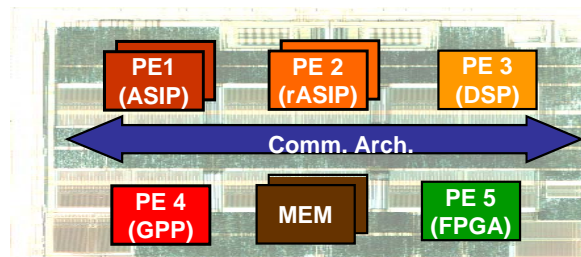


Flexible, efficient tool based mapping

Advantages

- Portable and efficient
- Reusable libraries
- Fast assembly of new transceiver descriptions
- Loadability also of new waveforms

HW Platform



N x: Nucleus X
 NN: Non-Nucleus
 F_{xy}: Nucleus Implementation of Nucleus x on PE y
 NF: Non-Flavor
 PE: Processing Element
 HW: Hardware

Generic Descriptors

- **Algorithmic functionality**
- **Flavours**
= Optimised implementations
 - Algorithmic variants
 - Implementation variants
- **Configuration parameters**
- **Constraints**



Example

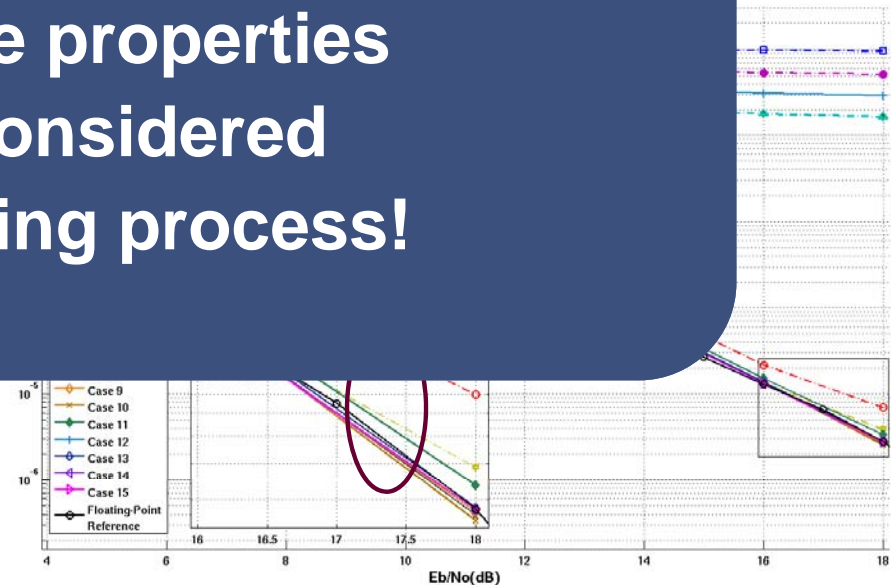
- **FFT**
- **FFT implementations**
 - Radix-2, radix-4, DIT, DIF
 - Assembler, IP-core
- Input/internal data width
- Input/internal scaling
- Rounding scheme
- Processing mode
(e.g. stream/block)
- Input data order
- Input data format

- Flavours and configuration parameters influence performance properties

- Processing properties
 - Latency
 - Throughput
 - Energy efficiency

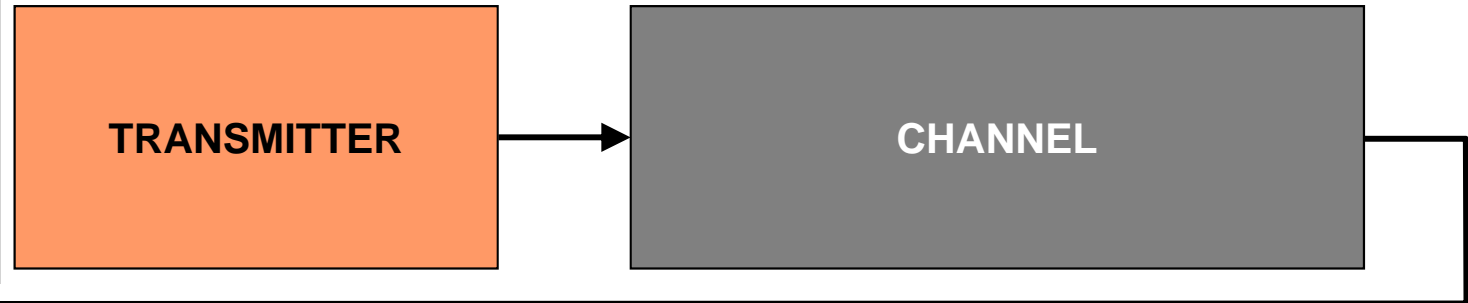
Case	PE	Configuration Parameters				Cycles per frame	
			Input width (bits)	Twiddle width (bits)	Internal scaling		Rounding / Truncation
1	C64x	R-4	16	16	yes	R	6526
2		R-4	16	16	no	T	6262
3	C62x	R-2	16	16	no	T	22388
4	C64x	R-4	32	16	no	R	8327
5		R-4	32	32	no	R	11895
6		R-4	32	32	yes	R	11895
7	Virtex-5	R-2	16	16	yes	R	7364
8		R-2	16	16	yes	R	12453
9		R-4	16	16	yes	R	3450
10	Virtex-5	R-2	16	16	yes	R	3199
					yes	T	-
					no	T	-
					yes	R	3450
				yes	R	-	
				yes	R	-	

Constraints and performance properties must be considered in the mapping process!

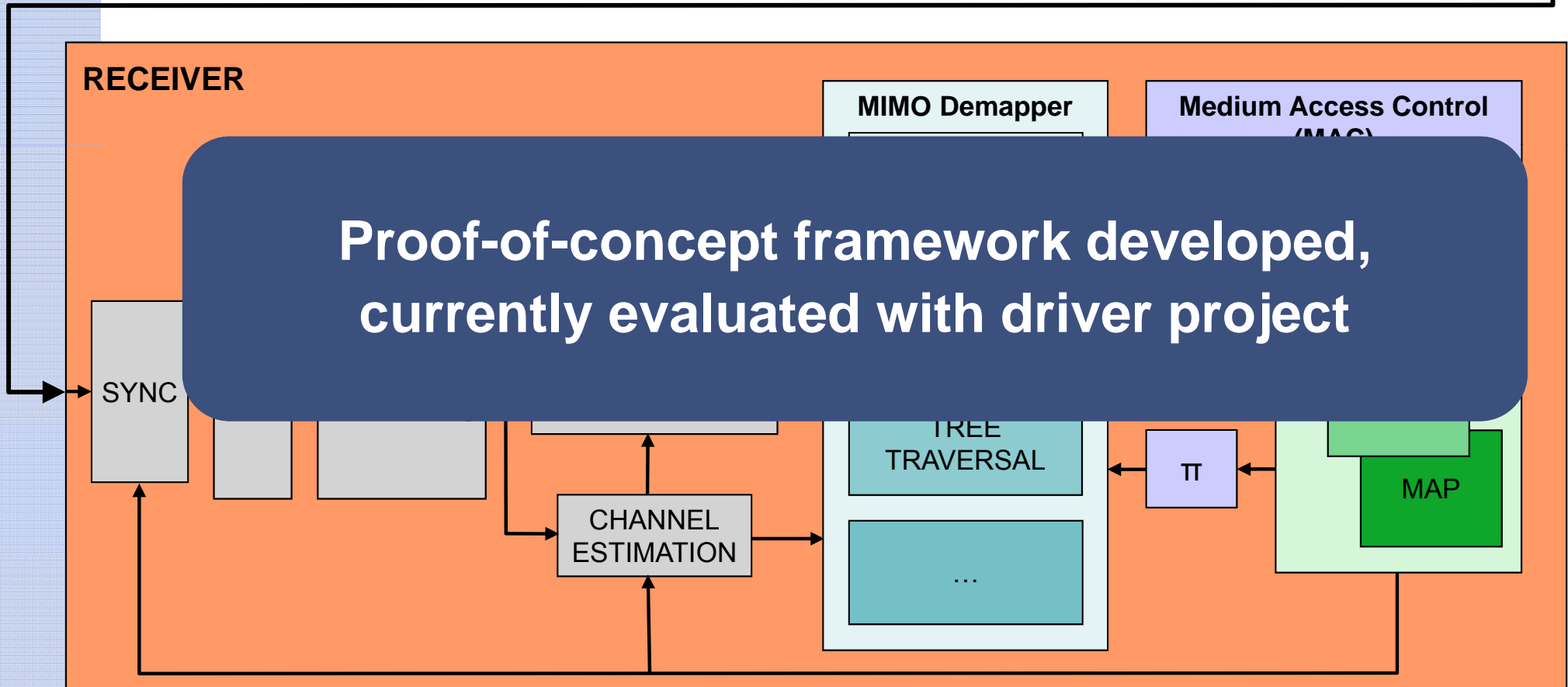


Parameter Flexibility

- Bit rates
- Number of Subcarriers
- Modulation schemes
- Number of antennas
- Coding rate (code)
- Supported modes
- ...



RECEIVER



- **The *Software Defined Radio (SDR)* Vision**
- **Challenges in Mobile Terminal Design**
- ***Hardware Abstraction Layer* Approaches**
- **The Nucleus approach of UMIC**
- **Outlook**

- **Develop complete Nucleus library for driver project**
 - Identification of issues
 - Refinement of approach
 - Proof of concept

- **Prototyping**
 - Design of proof-of-concept ASIC
 - FPGA-based demonstrator (⇒ demo in foyer)
 - Manufacturing & testing of ASIC