RWTH AACHEN UNIVERSITY

Re-configurable ASIPs :

Is there any need for these architectures?

28.06.2007, MPSoC 2007, Japan

Heinrich Meyr

ISS

Institute for Integrated Signal Processing Systems

Agenda

- rASIP Application Space
  - Future Wireless Communication Systems
- UMIC Many Core Working Assumptions
- rASIP Design Space
  - Early rASIP Designs
  - Traditional Design Methodology
- Proposed rASIP Design Flow
  - Pre-Fabrication Design Flow
  - Post-Fabrication Design Flow
- Proof of Concept
- Outlook

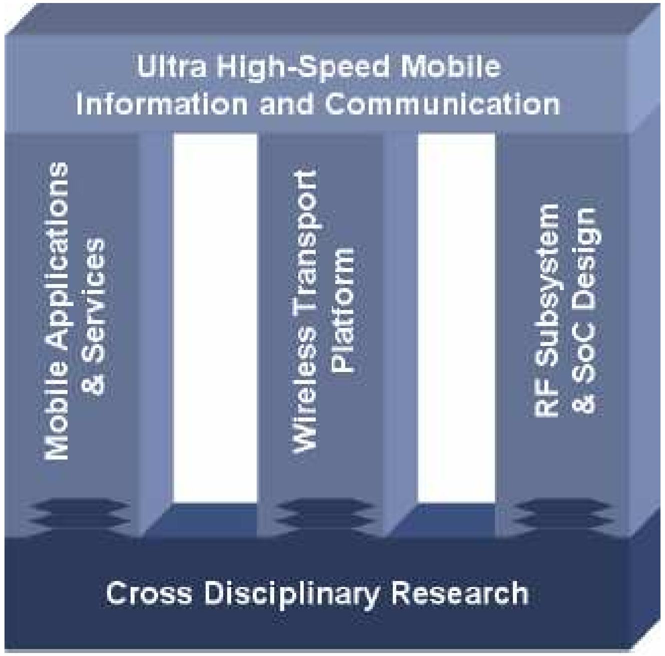
Acknowledgement

- **The research reported in this talk is based on work of area "RF Subystems and SoC Design" of the "UMIC Cluster" at the RWTH Aachen University**
  - Principal Investigators
    - Prof. T.G. Noll
      Institute for Electrical Engineering and Computer Systems
    - Profs G. Ascheid / R. Leupers / H. Meyr
      Chair for Integrated Signal Processing Systems
    - Prof. S. Heinen
      Chair of Integrated Analog Circuits
    - Prof. P. Mähönen
      Ericsson Chair of Wireless Networks
    - Prof. S. Kowalewski
      Chair for Software for Embedded Systems

UMIC Research Cluster at RTWH Aachen

- **Ultra high-speed Mobile Information and Communication**
  - extremely challenging application demands
  - limitations of mobile communications and
  - technology capabilities and constraints

- **The design requires a tremendous leap**
  ⇨ needs collaborative research in
  - Mobile Applications & Services
  - Wireless Transport Platform
  - Radio Frequency Subsystems & System-on-Chip (SoC) Design
  - Cross Disciplinary Methods & Tools
- **Total funding 36mio€ (2006-2011)**
  - www.umic.rwth-aachen.de


Ultra High-Speed Mobile Information and Communication
Mobile Applications & Services
Wireless Transport Platform
RF Subsystem & SoC Design
Cross Disciplinary Research

Agenda

- rASIP Application Space
  - Future Wireless Communication Systems
- UMIC Many Core Working Assumption
- rASIP Design Space
  - Early rASIP Designs
  - Traditional Design Methodology
- Proposed rASIP Design Flow
  - Pre-Fabrication Design Flow
  - Post-Fabrication Design Flow
- Proof of Concept
- Outlook

Future Wireless Systems : Cognitive Radios

- **Must sense or be cognitive of the environment**
  - Other user interference, multipath, noise, etc.
  - Time-variations
- **Must be intelligent to analyze the situation and find the optimal communications protocol, frequency, channel, etc.**
- **Must reconfigure for the channel and protocol required**
- **And…constantly adapt to mobile changing environments**
- **Will have multiple antennas (MIMO)**
- **They will make use of ultra-complex and ultra energy efficient signal processing to optimally use the available bandwidth**

These radios “find the best protocol, frequency, and channel” to communicate over

Agenda

- rASIP Application Space
  - Future Wireless Communication Systems
- UMIC Many Core Working Assumptions
- rASIP Design Space
  - Early rASIP Designs
  - Traditional Design Methodology
- Proposed rASIP Design Flow
  - Pre-Fabrication Design Flow
  - Post-Fabrication Design Flow
- Proof of Concept
- Outlook

The „Key Algorithm“ Proposition of the UMIC Platform

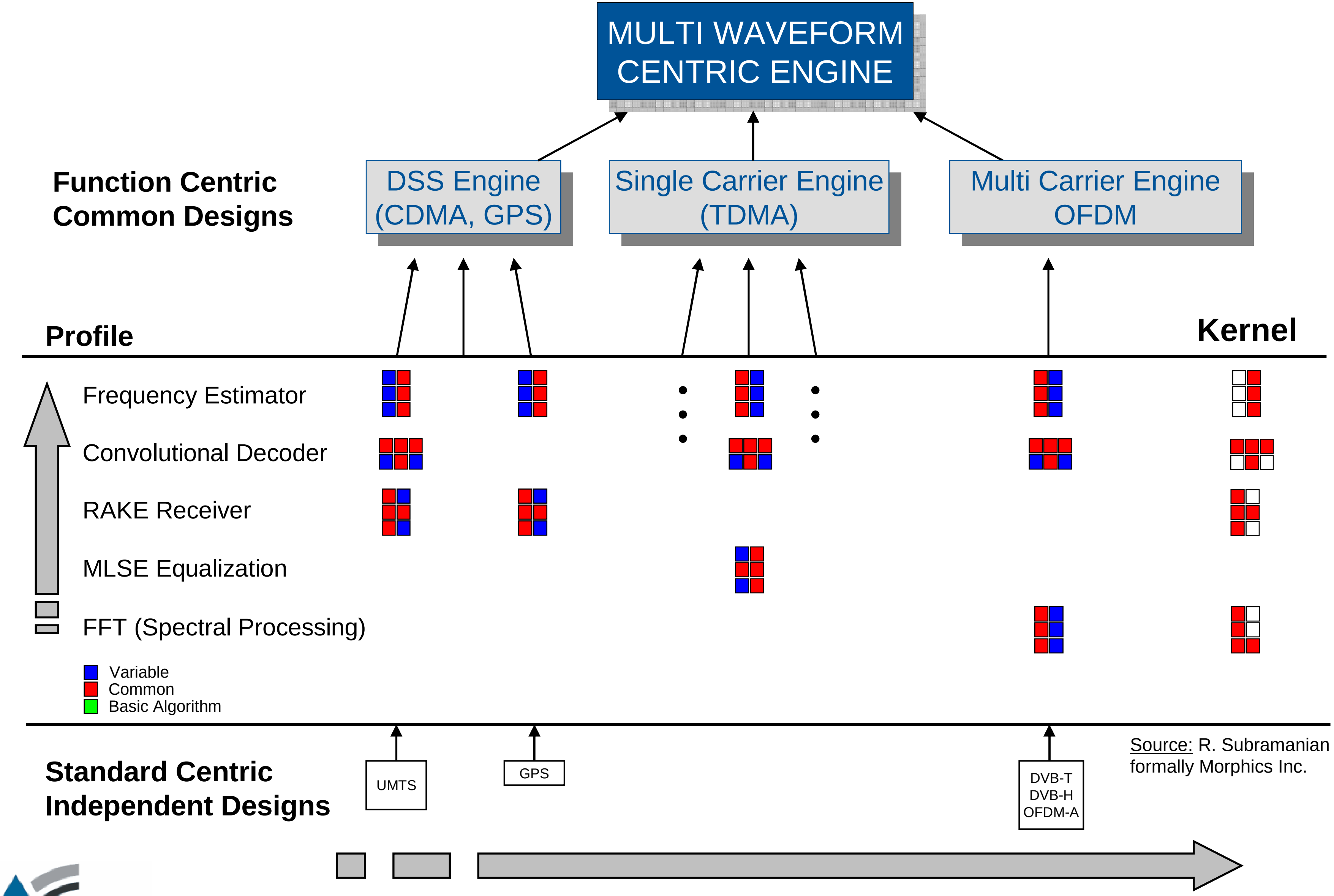
Each signal processing is based on a small number of fundamental algorithms („Nuclei“) that represent a significant amount of the computation.

⇨ **Focus on an efficient composition („design of an MPSoc) or mapping („programming of the MPSoC“)**

References:

- UC Berkeley, EECS TR, June 15, 2006
- Ienne, Leupers: Customizable Embedded Processors, August 2006

HW/SW Partitioning: Spatial and Temporal Mapping



MULTI WAVEFORM CENTRIC ENGINE

Function Centric Common Designs

DSS Engine (CDMA, GPS)

Single Carrier Engine (TDMA)

Multi Carrier Engine OFDM

Profile

Kernel

Frequency Estimator

Convolutional Decoder

RAKE Receiver

MLSE Equalization

FFT (Spectral Processing)

🟦 Variable
🟥 Common
🟩 Basic Algorithm

Standard Centric Independent Designs

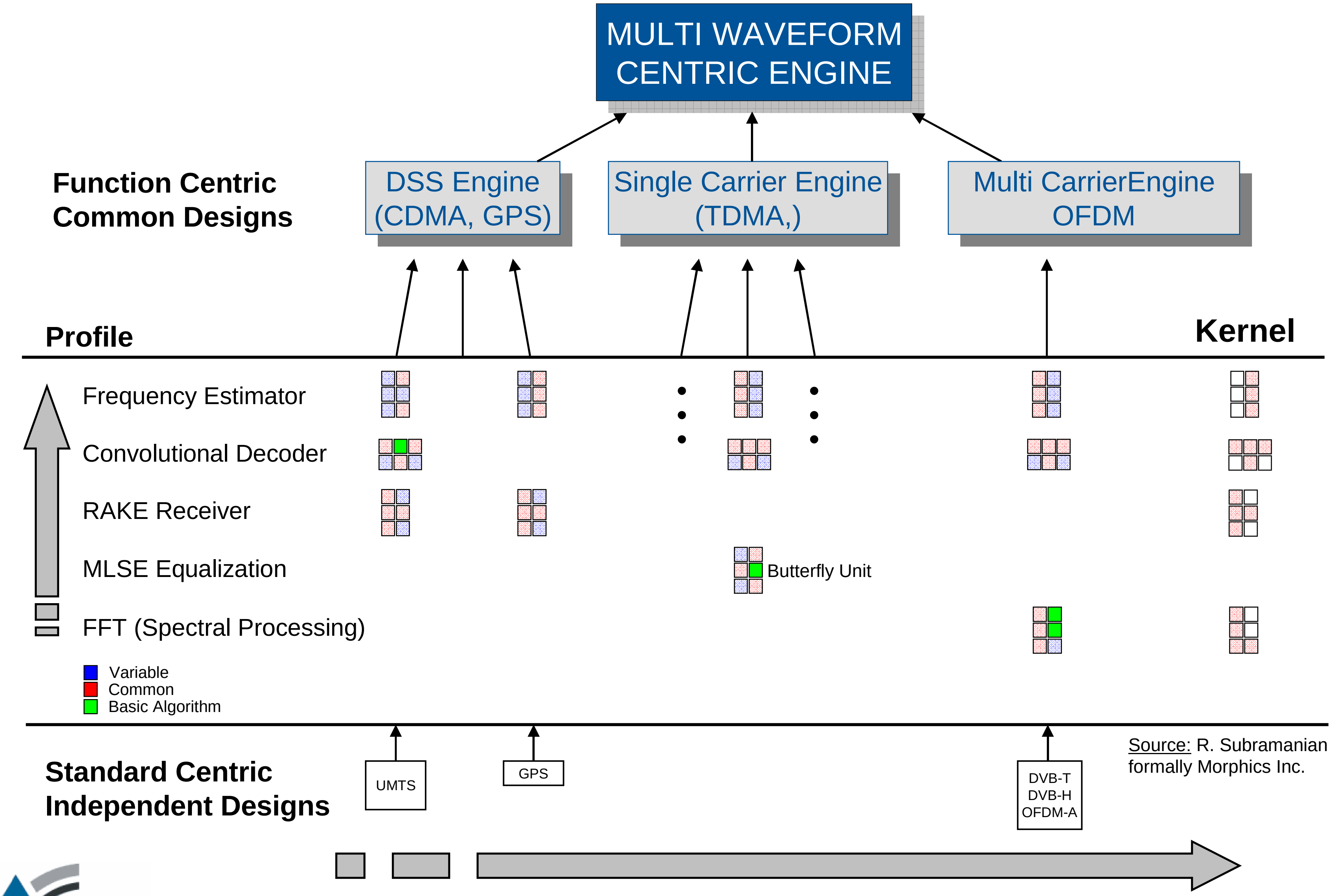
UMTS

GPS

DVB-T
DVB-H
OFDM-A

Source: R. Subramanian
formally Morphics Inc.

HW/SW Partitioning: Spatial and Temporal Mapping



MULTI WAVEFORM CENTRIC ENGINE

Function Centric Common Designs

DSS Engine (CDMA, GPS)

Single Carrier Engine (TDMA,)

Multi CarrierEngine OFDM

Profile

Kernel

Frequency Estimator

Convolutional Decoder

RAKE Receiver

MLSE Equalization

Butterfly Unit

FFT (Spectral Processing)

🟦 Variable
🟥 Common
🟩 Basic Algorithm

Standard Centric Independent Designs

UMTS

GPS

DVB-T
DVB-H
OFDM-A

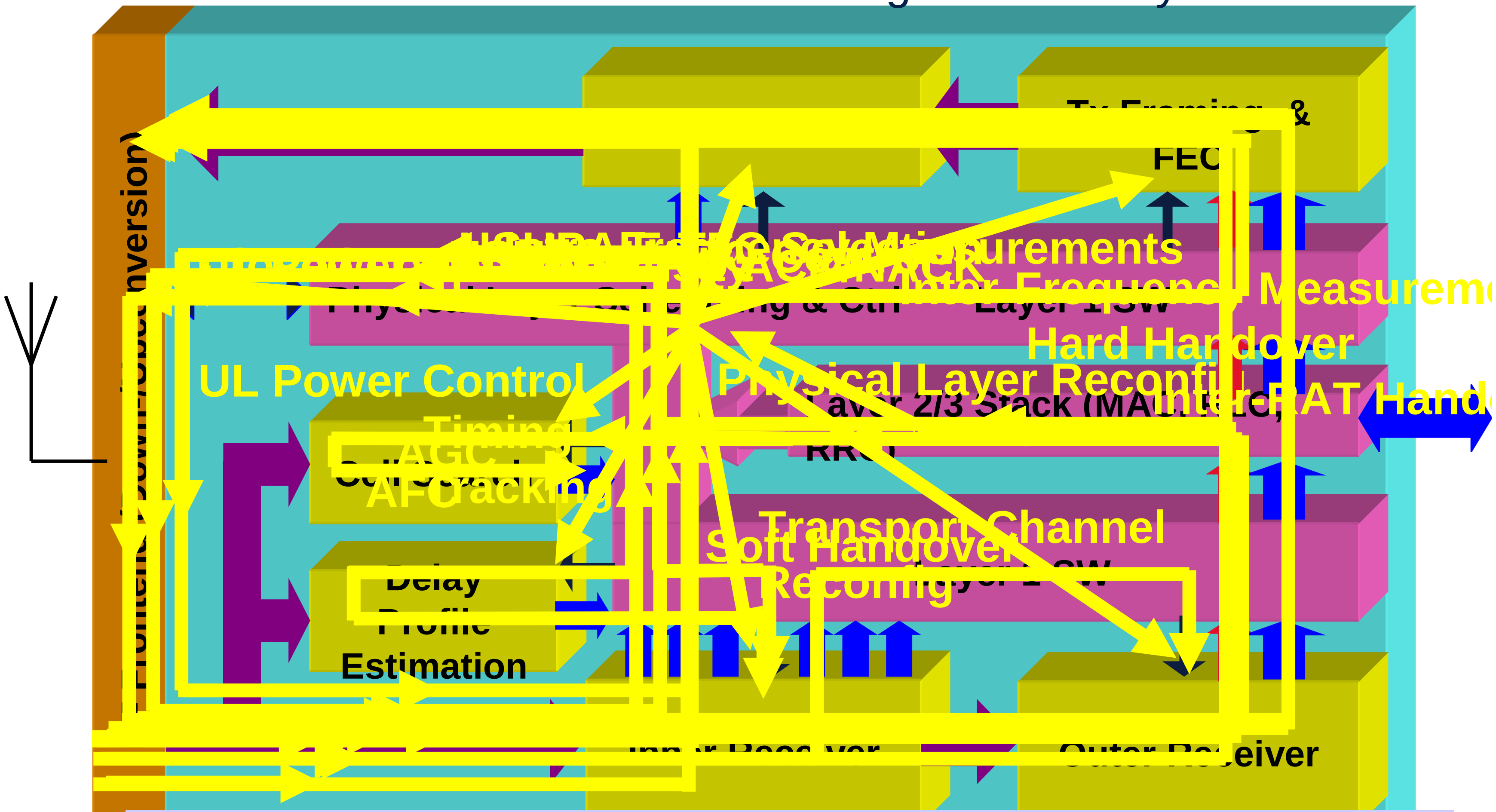
Source: R. Subramanian
formally Morphics Inc.

Source: Dr.H.Dawid, Infineon

Internal use only!

1. System Architecture Concept (HW & SW)

- Remember: demodulation is running continuously

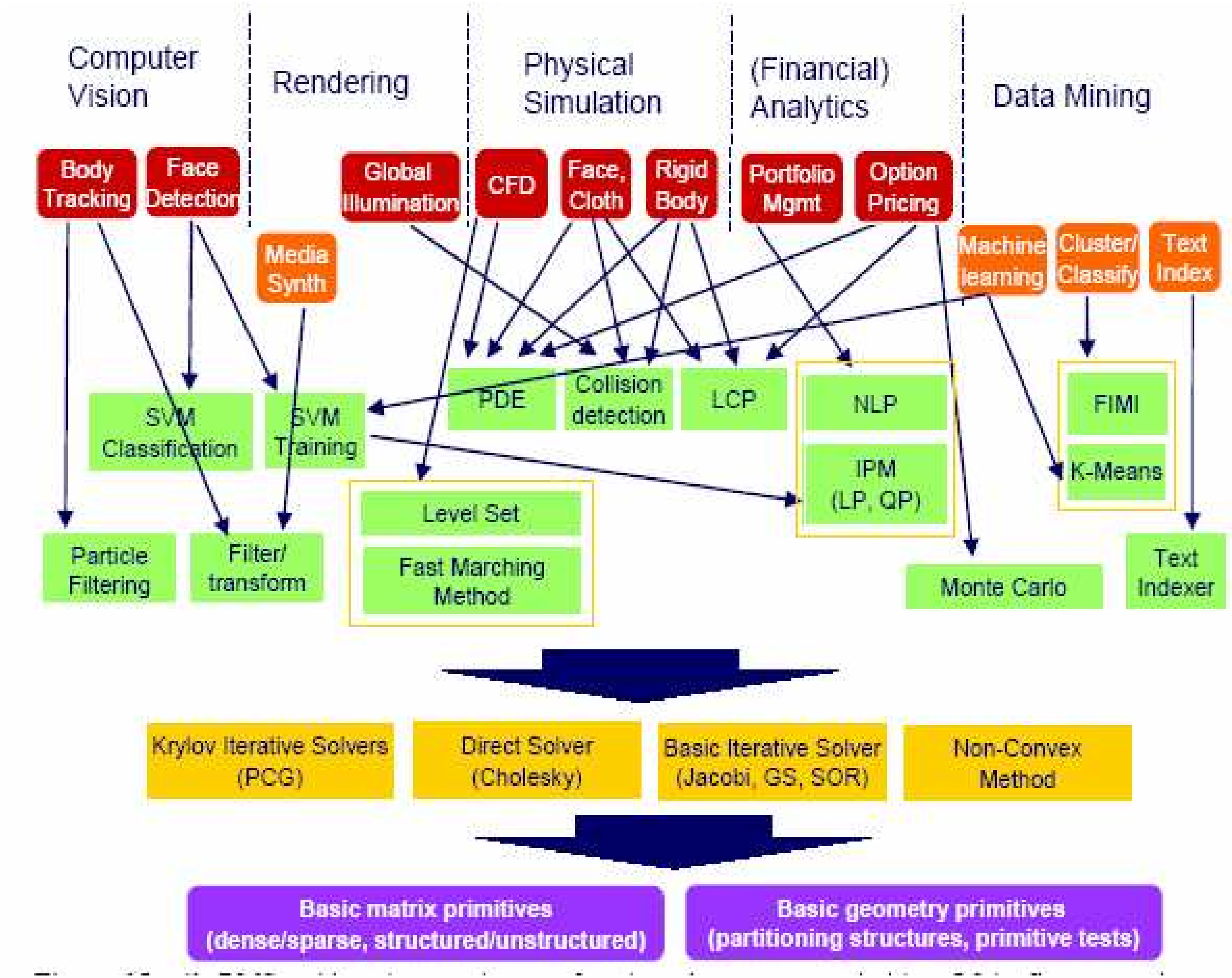


Hidden Latency Requirements (not MIPS) dominate

🟦 Data → L1 Config/Ctrl 🟥 System Information/Higher Layer Ctrl

Architecture

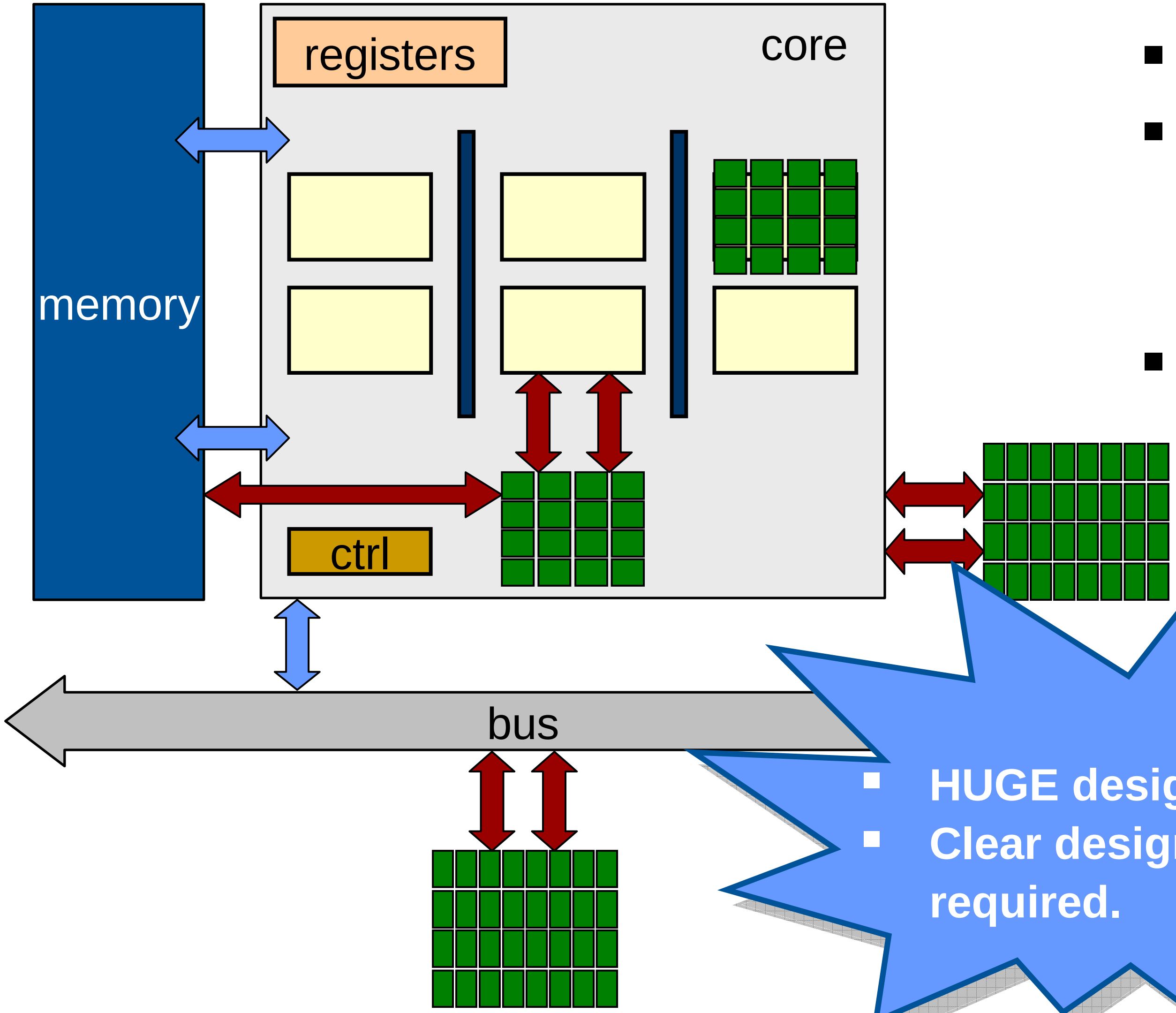
Intel RMS View (Recognition, Mining,Synthesis)



Agenda

- rASIP Application Space
  - Future Wireless Communication Systems
- UMIC Many Core Working Assumptions
- rASIP Design Space
  - Early rASIP Designs
  - Traditional Design Methodology
- Proposed rASIP Design Flow
  - Pre-Fabrication Design Flow
  - Post-Fabrication Design Flow
- Proof of Concept
- Outlook

rASIP Design Space

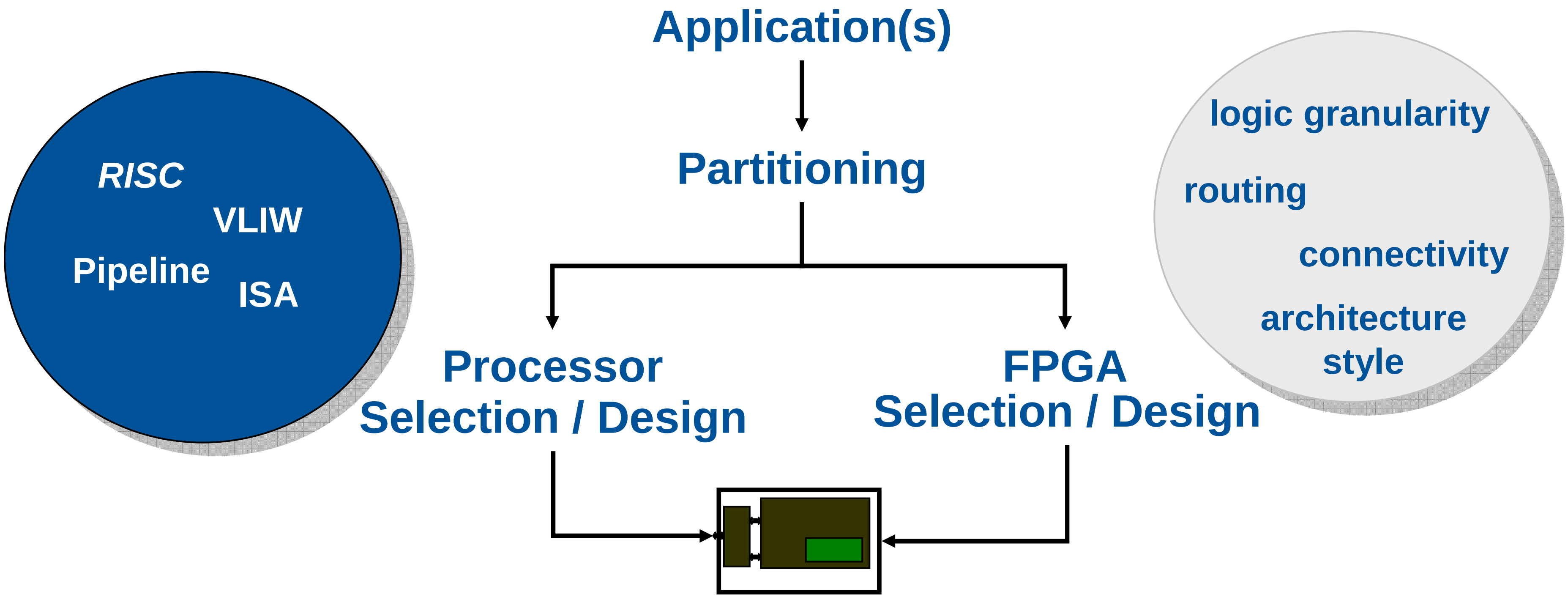


Design decisions:

- base architecture
- reconfigurable part
  - where to put
  - granularity etc.
- coupling and interface

- **HUGE design space!**
- **Clear design methodology required.**

Traditional Design Methodology



Application(s)

Partitioning

Processor
Selection / Design

FPGA
Selection / Design

RISC
VLIW
Pipeline
ISA

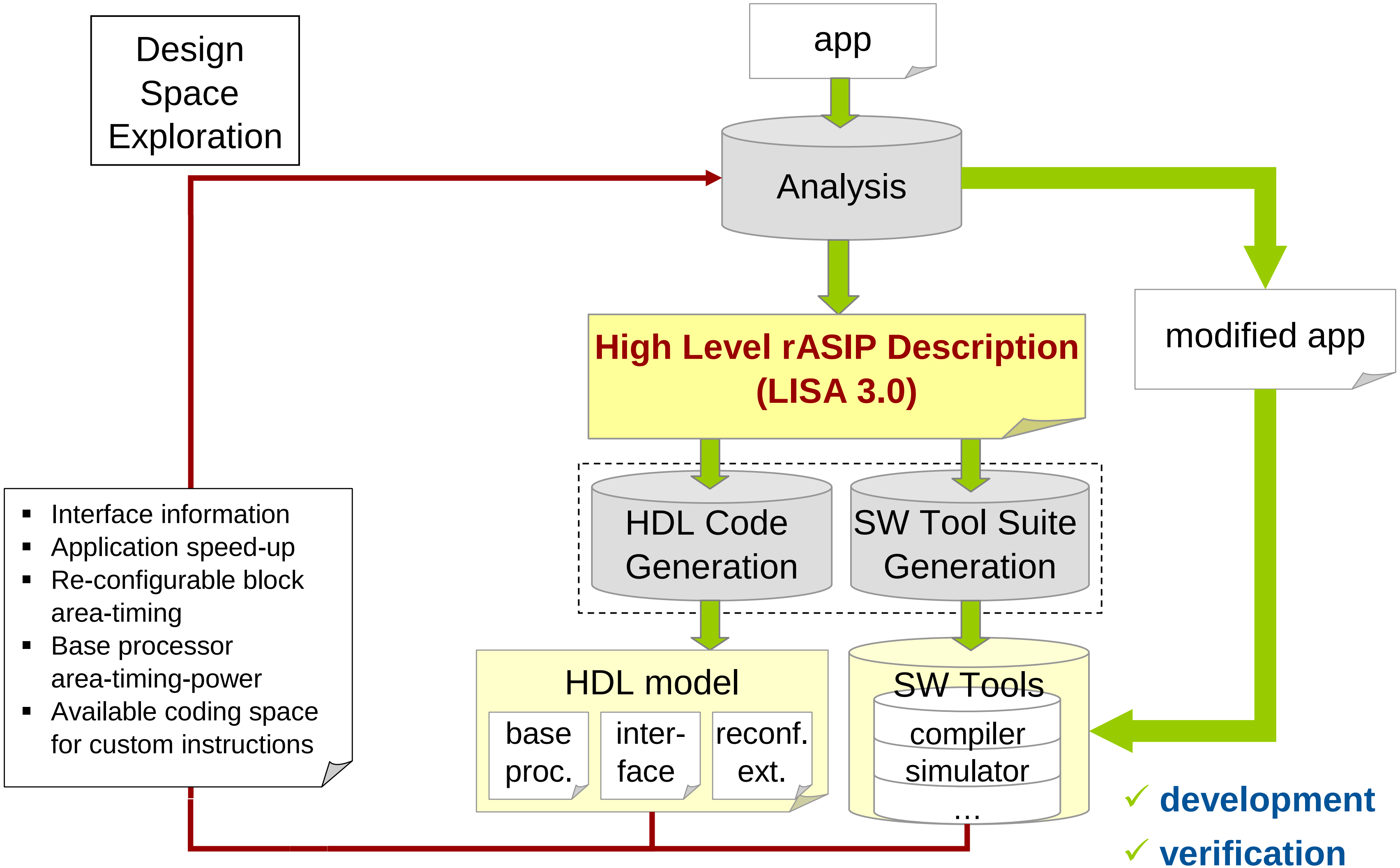
logic granularity
routing
connectivity
architecture
style

- Dedicated/ Partially Re-targetable Tools
- Separate Design Space Exploration
- Partial Design Space Exploration

Agenda

- Reconfigurable systems
- rASIP Application Space
  - Future Wireless Communication Systems
  - UMIC Many Core Working Assumption
- rASIP Design Space
  - Early rASIP Designs
  - Traditional Design Methodology
- Proposed rASIP Design Flow
  - Pre-Fabrication Design Flow
  - Post-Fabrication Design Flow
- Proof of Concept
- Outlook

PRE-Fabrication Phase



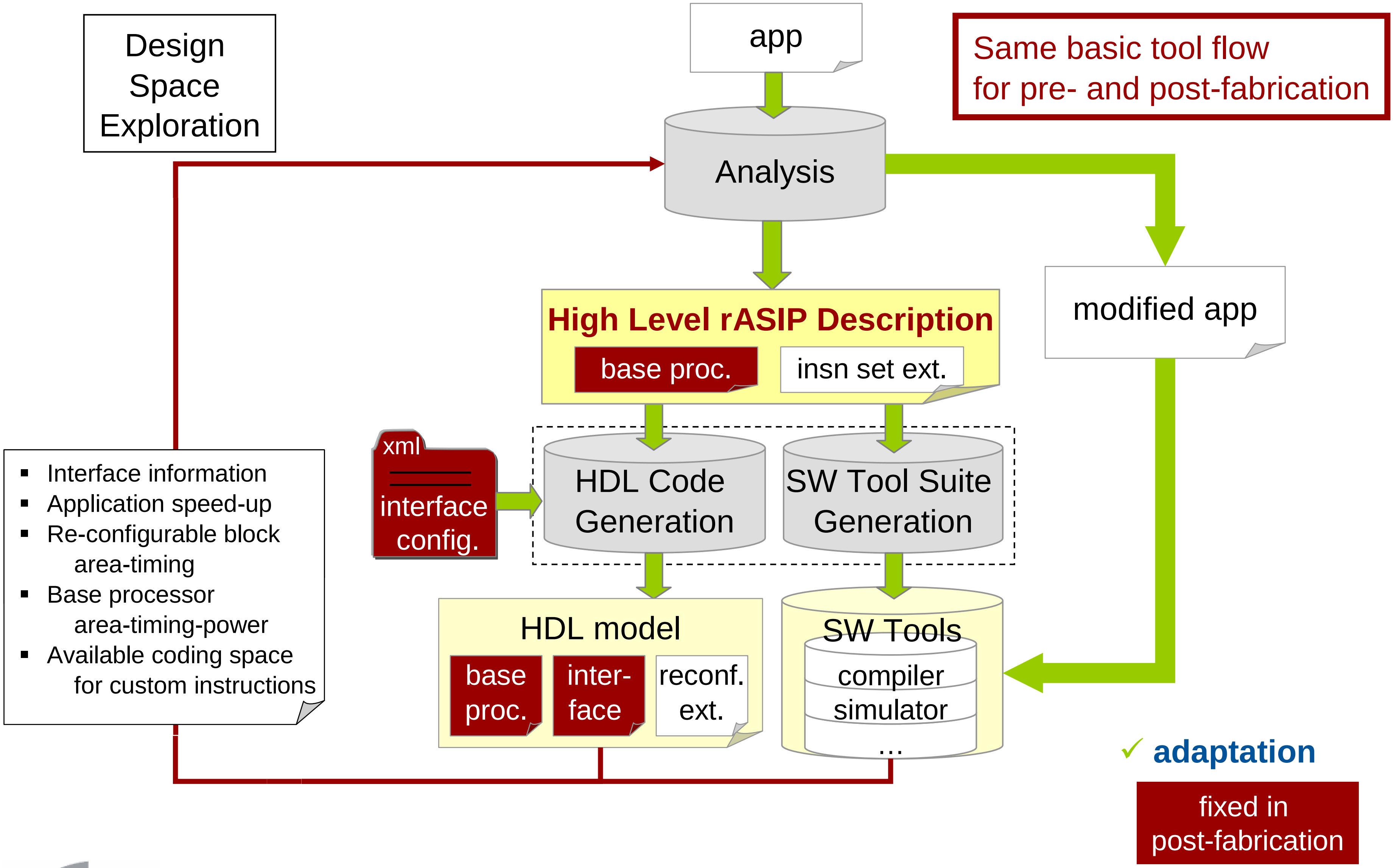
- Interface information
- Application speed-up
- Re-configurable block area-timing
- Base processor area-timing-power
- Available coding space for custom instructions

✓ **development**

✓ **verification**

✓ **exploration**

POST-Fabrication Phase



Design Space Exploration

app

Same basic tool flow
for pre- and post-fabrication

Analysis

modified app

High Level rASIP Description

base proc.

insn set ext.

xml
interface config.

HDL Code Generation

SW Tool Suite Generation

- Interface information
- Application speed-up
- Re-configurable block
  area-timing
- Base processor
  area-timing-power
- Available coding space
  for custom instructions

HDL model

base proc.

inter-face

reconf. ext.

SW Tools

compiler

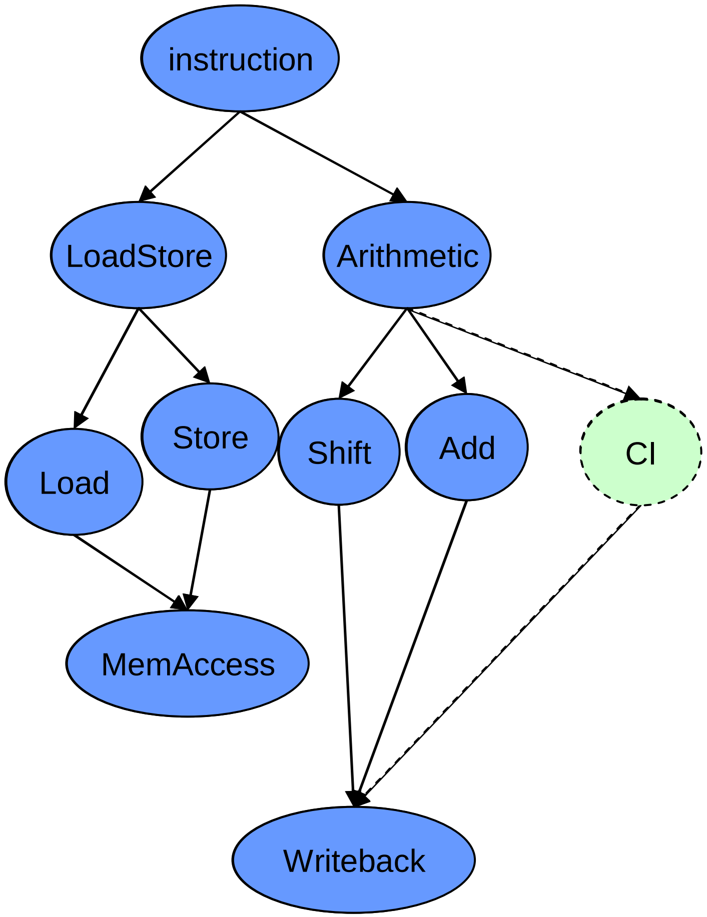
simulator

…

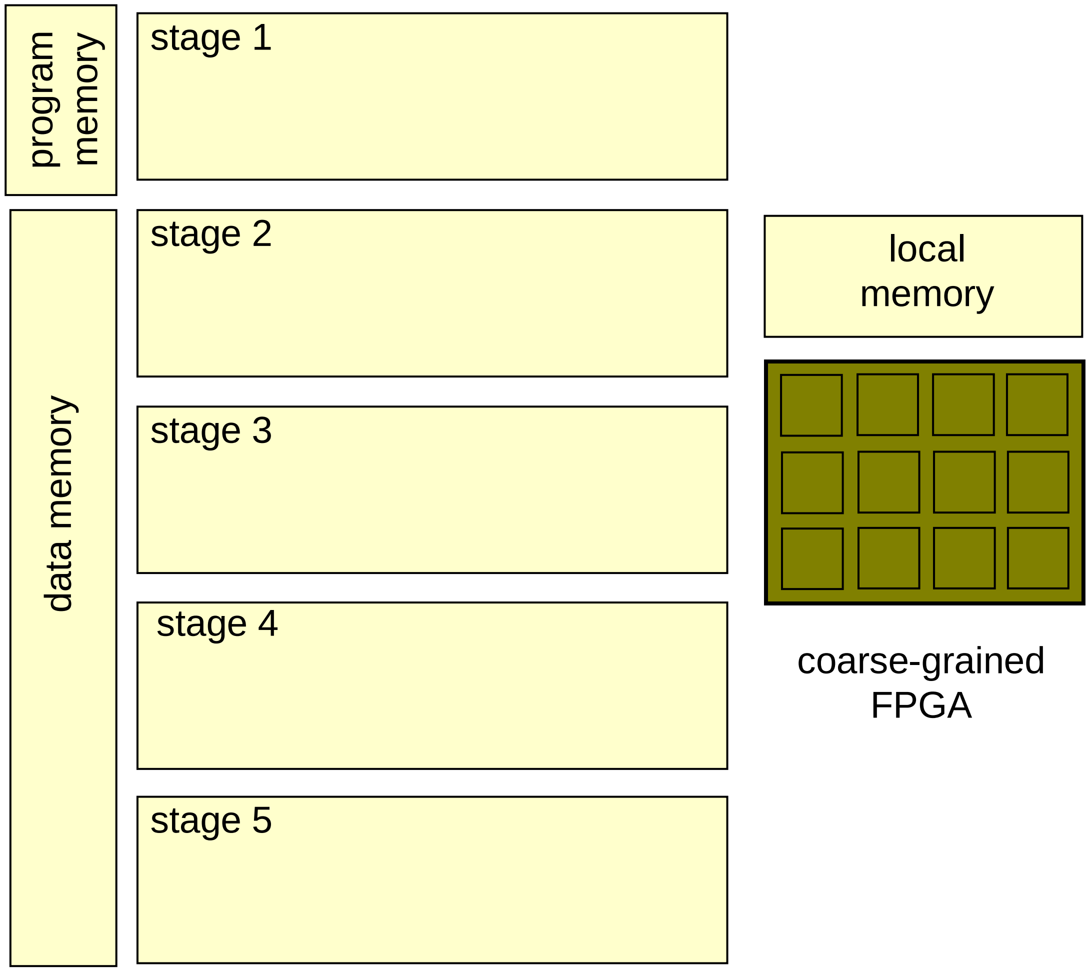
✓ adaptation

fixed in post-fabrication

Stepwise Processor Modelling



step 1 : capturing the ISA



step 2 : structuring the ISA

Agenda

- rASIP Application Space
    - Future Wireless Communication Systems
- UMIC Many Core Working Assumptions
- rASIP Design Space
    - Early rASIP Designs
    - Traditional Design Methodology
- Proposed rASIP Design Flow
    - Pre-Fabrication Design Flow
    - Post-Fabrication Design Flow
- **Proof of Concept**
- Outlook

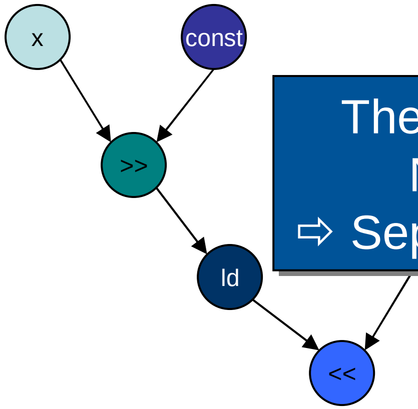
GOST vs Blowfish Hotspot

GOST Hotspot

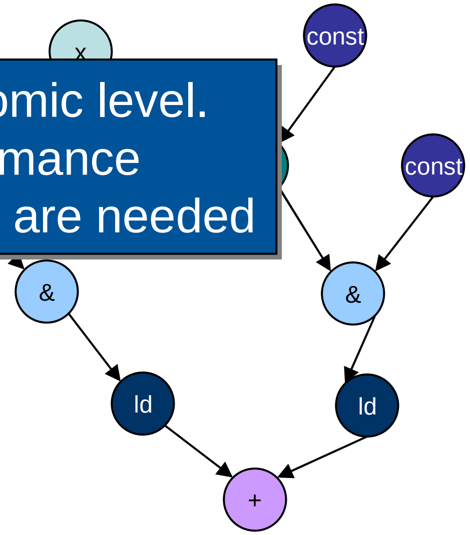
```
word32 f(word32 x)
{
   x = k87[x>>24 & 255] << 24
       | k65[x>>16 & 255] << 16
       | k43[x>> 8 & 255] << 8
       | k21[x & 255];
   // Rotate left 11 bits
   return x<<11 | x>>(32-11);
}
```

Blowfish Hotspot

```
unsigned long F(unsigned long x)
{
            ……
            d = x & 0x00FF;
            x >>= 8;
            c = x & 0x00FF;
            x >>= 8;
            b = x & 0x00FF;
            x >>= 8;

            a = x & 0x00FF;

            y = S[0][a] + S[1][b];
            y = y ^ S[2][c];
            y = y + S[3][d];

            return y;
}
```



The similarity exists only in atomic level.
Not enough to boost performance
⇨ Separate Custom Instructions are needed



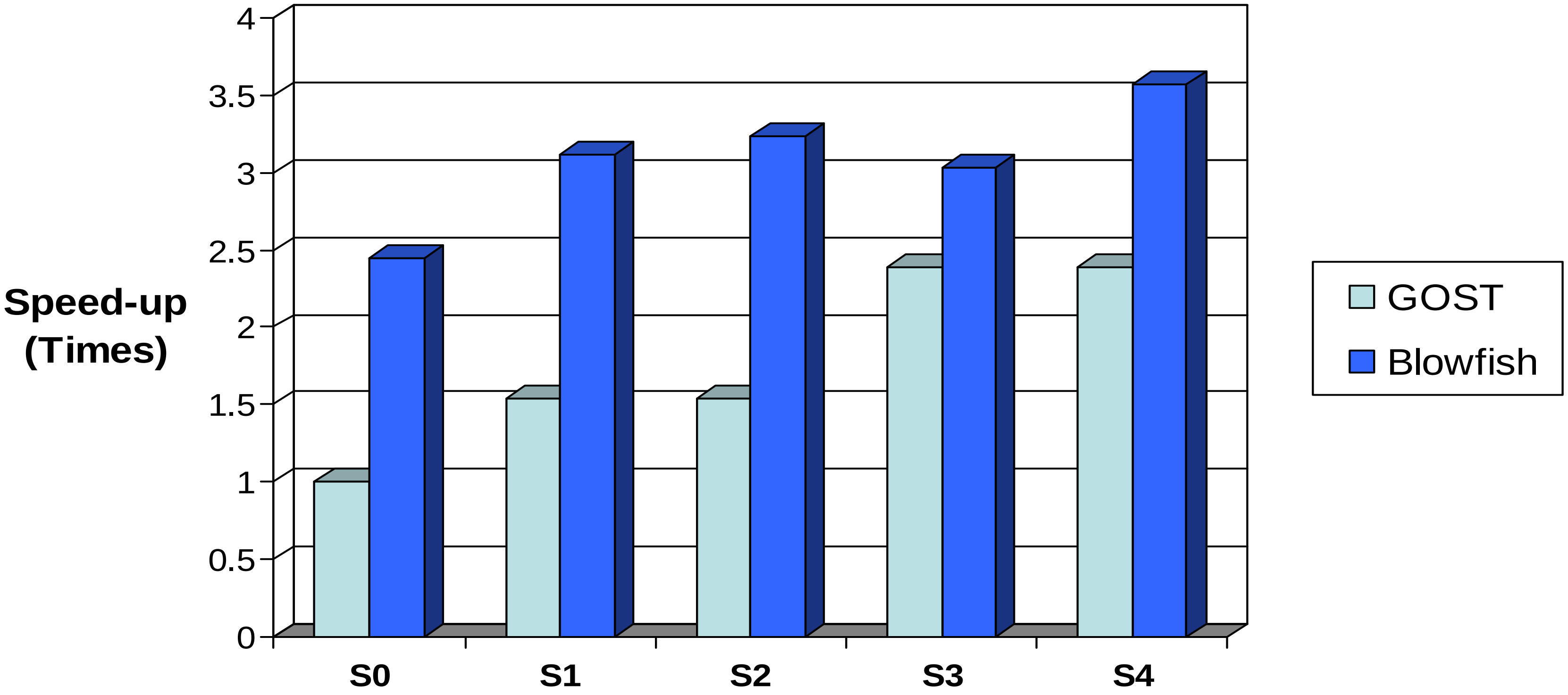
Application Analysis

- **Initial results**
  - Not good for GOST, better for Blowfish
  - Longer operation chains are key to speed-up

- **Identifying common characteristics**
  - Given cryptographic applications access constant S-Boxes from the hot-spots
  - Such S-Boxes can be put inside scratch-pads accessible from custom instructions
  - Conclusion: Scratch-pad access from custom instructions might result in higher speed-ups

Design Space Exploration Results

- GOST: 105 data points
- Blowfish: 120 data points
- Custom Instruction Synthesis with various latencies, area constraints
- From 0 to 4 scratch-pad access per custom instruction
- Total exploration time: 2 man days



Custom instructions Set synthesis tool: R.Leupers,K.Karuri, RWTH Aachen)

Overview: Architecture and Parameters

Cluster

- horizontal LEs per row C_H=[1..64]
- vertical LEs per column C_V=[1..64]
- global inputs north I_N=[0..4]
- global inputs west I_W=[0..4]
- global inputs south I_S=[0..4]
- global inputs east I_E=[0..4]
- output directions D_{CL}=any set from [north,west,south,east]
- output ports (LE-ports)
- granularity M_{CL}=[1..C_H] (only adjacent LEs in a row)

same for all clusters

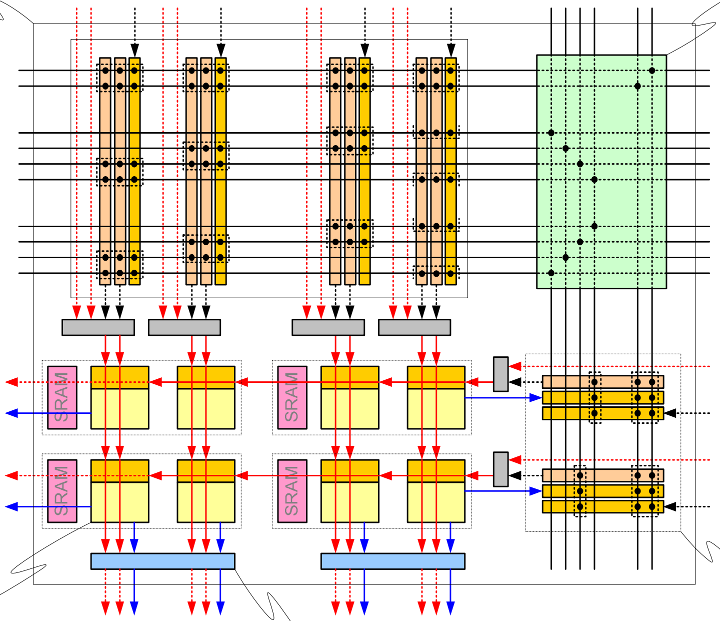
general

- horizontal clusters in eFPGA S_H=[1..256]
- vertical clusters in eFPGA S_V=[1..256]

Routing Switch

- horizontal tracks W_H=[1..256]
- vertical tracks W_V=[1..256]
- switch-positions as index of crossing vert. and hor. lines $P_{RS,i}$=[i_V,i_H] (only F_S=3, no cascaded SPs)
- SP-definition SP_i=set of routes
- per track i: segment length L_i=[1..256]
- granularity M_{RS}=[1..64] (only adjacent switch points)

same for all RS



Logic Element

Dedicated Routing Block (DRB)

- inputs from other LEs and broadcast defined per input of Core Logic [list]

Core Logic (CL) *only for configurator*

- set of functions with according CL-in-/outputs (from DRB and dedicated) [list]

same for all LEs

Register-Stage *only for configurator*

- LEs in a column per register N_{Reg}=[1..C_V]
- register-chain RC=[yes,no]
- LE-outputs with register [list]
- broadcast lines with register [list]

same for all register stages

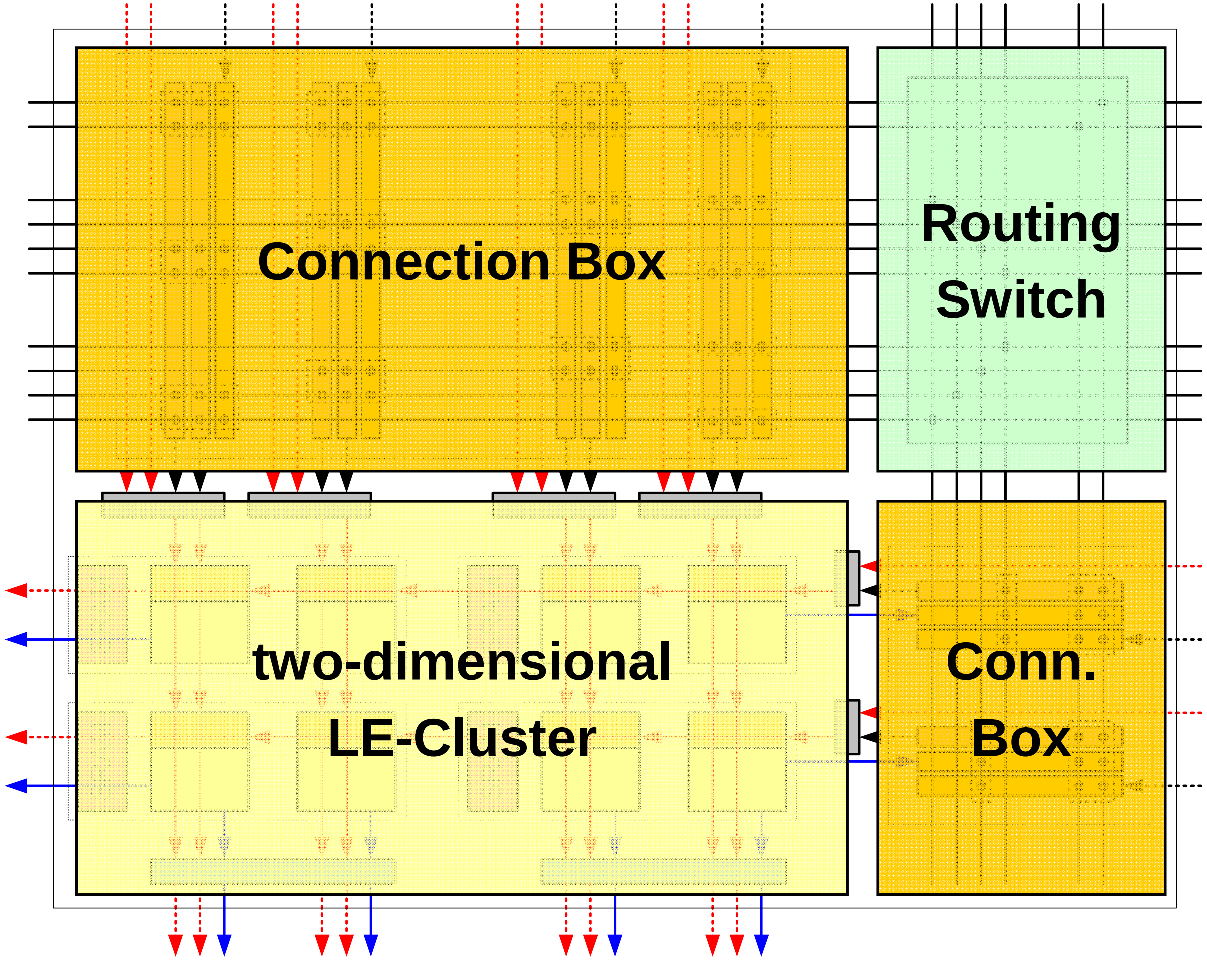
Connection Box

- switch positions per cluster in-/output: set of tracks described by track index i_{track}, LE-index i_{LE} (row or column) and ranges of tracks r_{track} $P_{CP,i}$=f(i_{track},r_{track},i_{LE})
- granularity M_{CB}=[1..64] (only adjacent connection points)

same for all CBs and defined only once per cluster-row (-column)

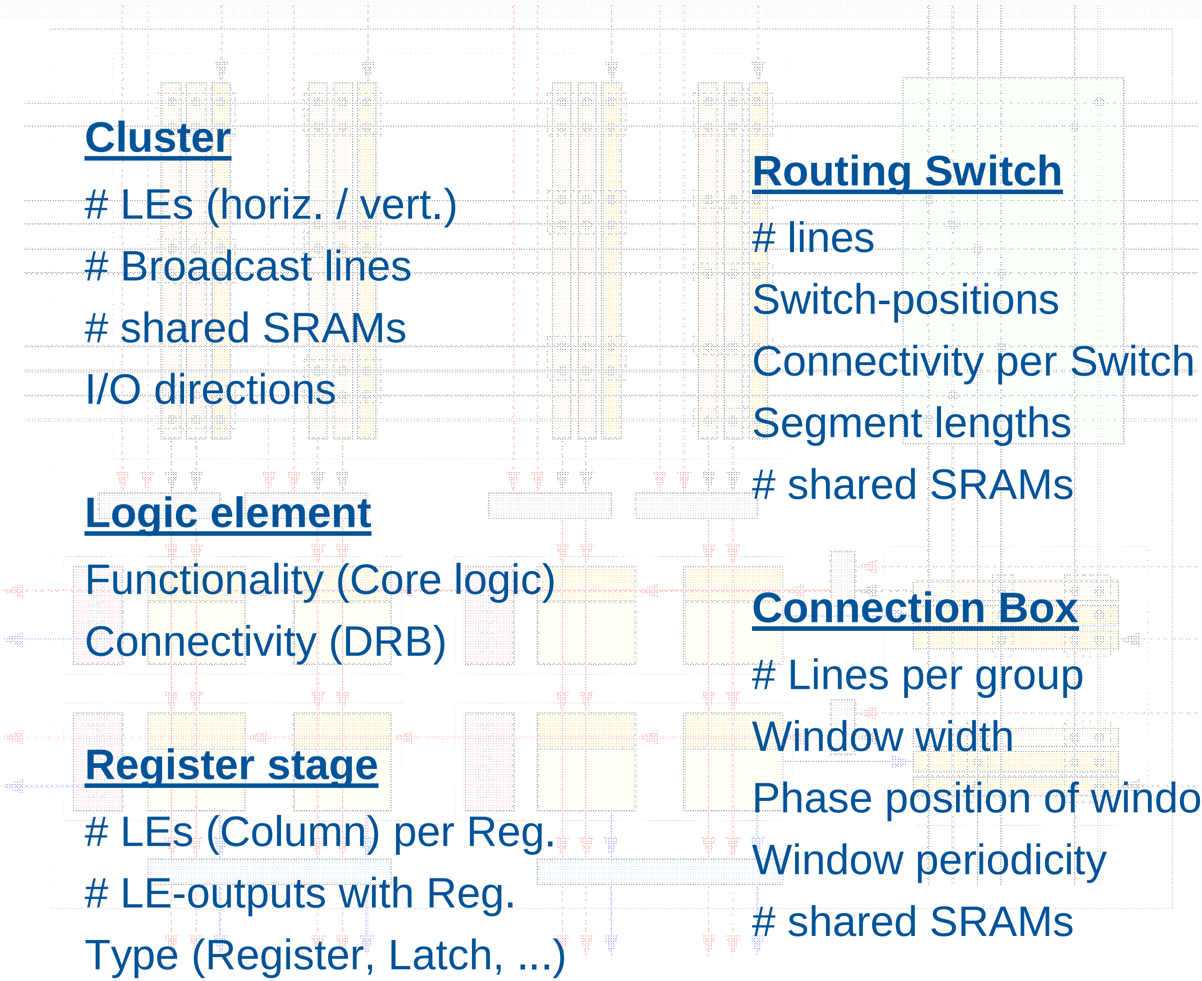
Source: T.G. Noll, RWTH Aachen

Parameterisable FPGA Architecture – Template



Source: T.G. Noll, RWTH Aachen

Parameterisable FPGA Architecture – Template



Cluster

# LEs (horiz. / vert.)

# Broadcast lines

# shared SRAMs

I/O directions

Logic element

Functionality (Core logic)

Connectivity (DRB)

Register stage

# LEs (Column) per Reg.

# LE-outputs with Reg.

Type (Register, Latch, ...)

Routing Switch

# lines

Switch-positions

Connectivity per Switch Point

Segment lengths

# shared SRAMs

Connection Box

# Lines per group

Window width

Phase position of window

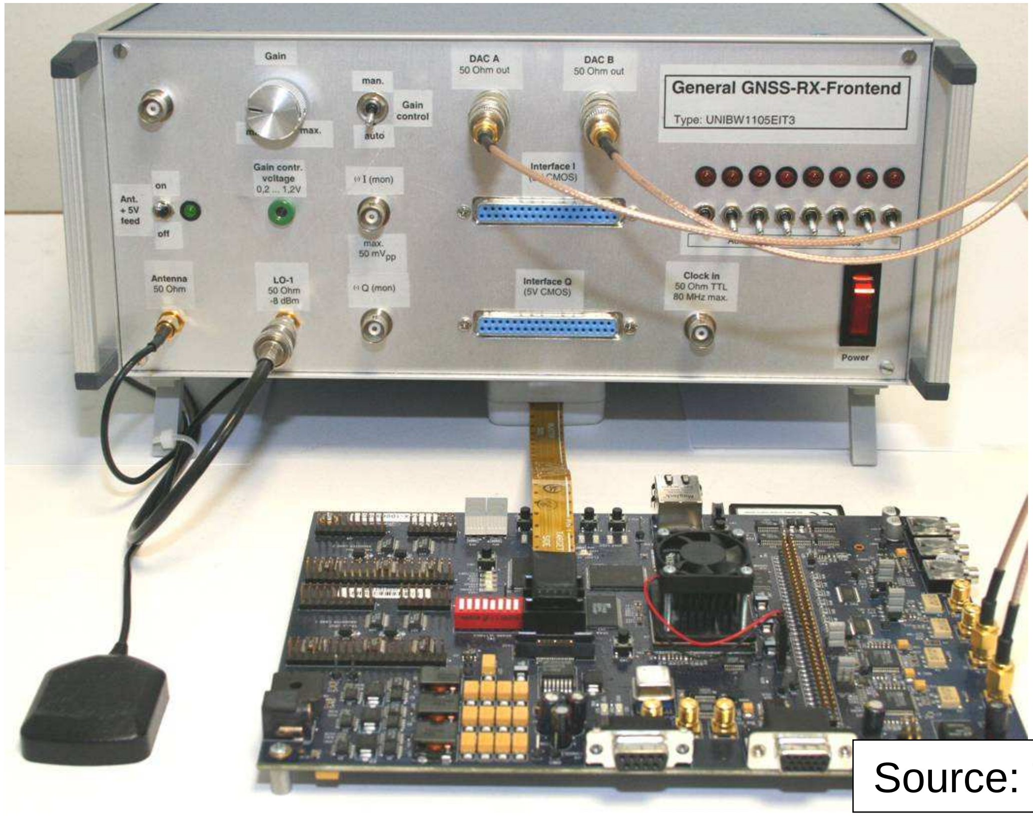
Window periodicity

# shared SRAMs

Source: T.G. Noll, RWTH Aachen

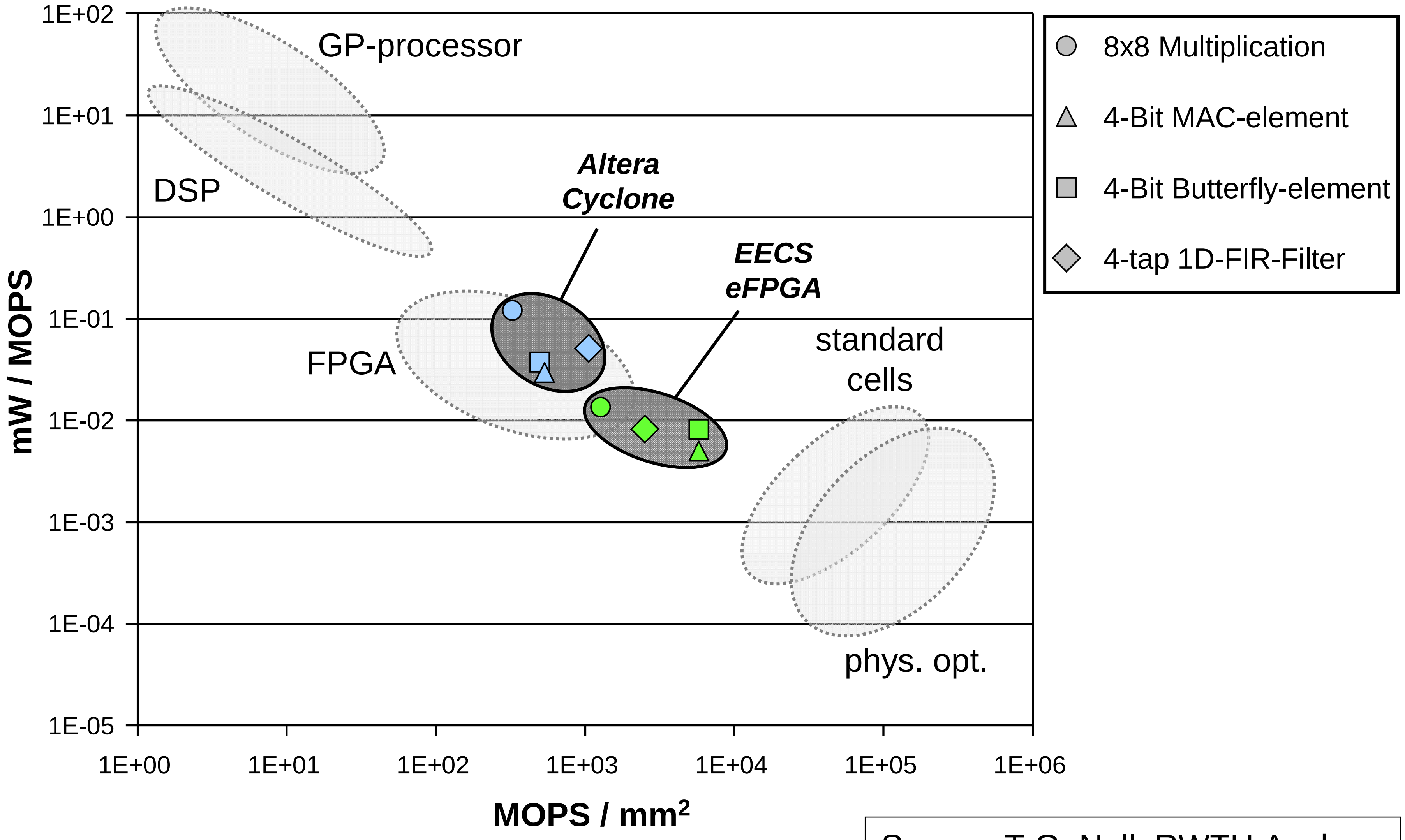
GNSS Prototype Hardware

- **Multioperable GNSS frontend**
  - Supports: NavStar GPS, Glonass, Galileo, Egnos, ...
  - 1575 – 1620 MHz
- **Flexible FPGA based receiver**



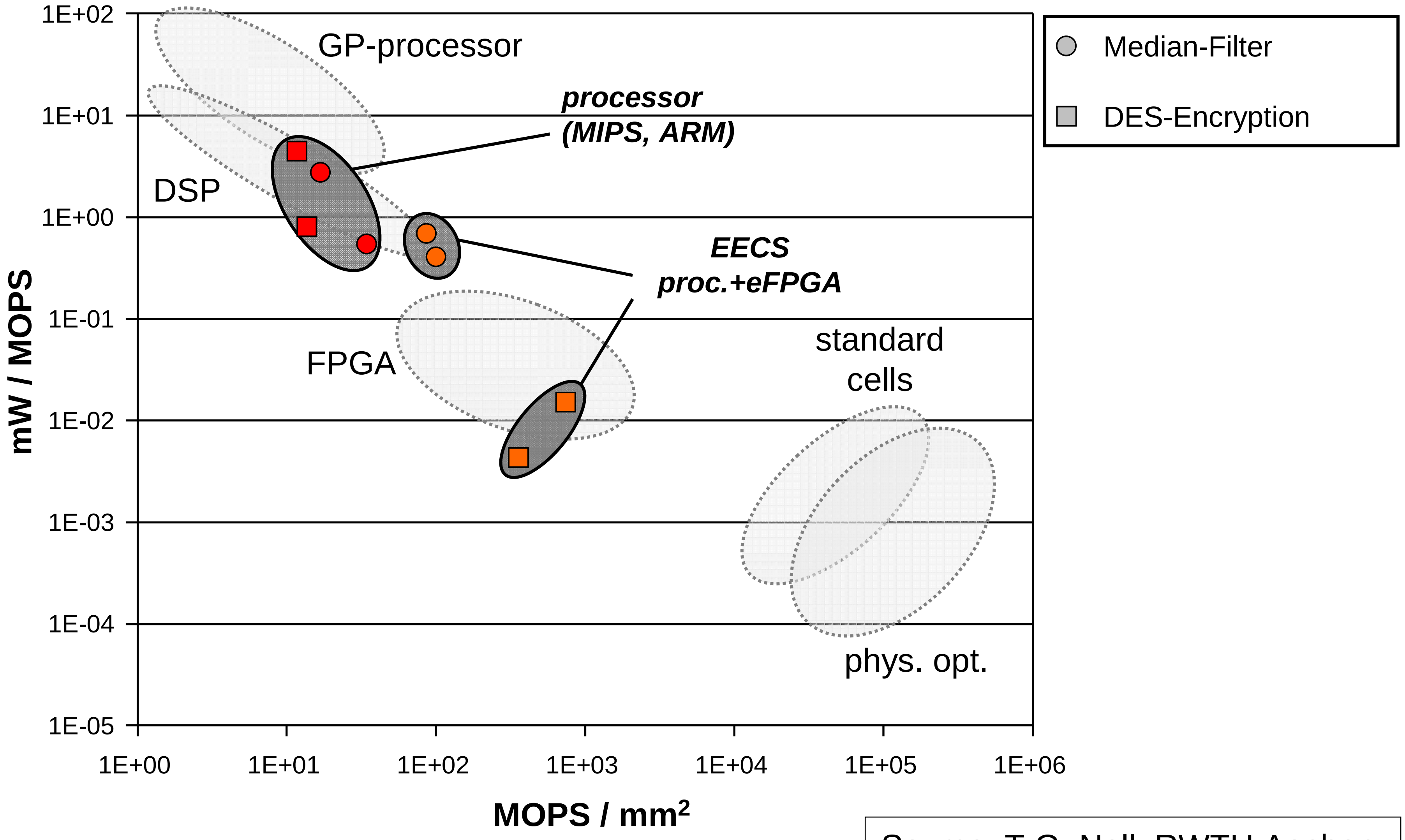
Source: T.G. Noll, RWTH Aachen

Arithmetic oriented eFPGA vs. commercial FPGA



Source: T.G. Noll, RWTH Aachen

Reconfigurable Processor vs. Standard Processor



Source: T.G. Noll, RWTH Aachen

Agenda

- Reconfigurable systems
- rASIP Application Space
  - Future Wireless Communication Systems
  - UMIC Many Core Working Assumption
- rASIP Design Space
  - Early rASIP Designs
  - Traditional Design Methodology
- Proposed rASIP Design Flow
  - Pre-Fabrication Design Flow
  - Post-Fabrication Design Flow
- Proof of Concept
- Outlook

Conclusion

- **rASIPs allow combining the benefits of both ASIP and FPGA**
  - In view of future wireless applications, this is desired
- **Embedded domain-specific eFPGA macros are key for high area/energy efficiency**
- **Pre-fabrication design-space exploration is absolutely essential**
- **Post-fabrication tools for programmability and reconfigurability are absolutely essential**
- **Language-driven (LISA 3.0) rASIP tool suite is proposed**

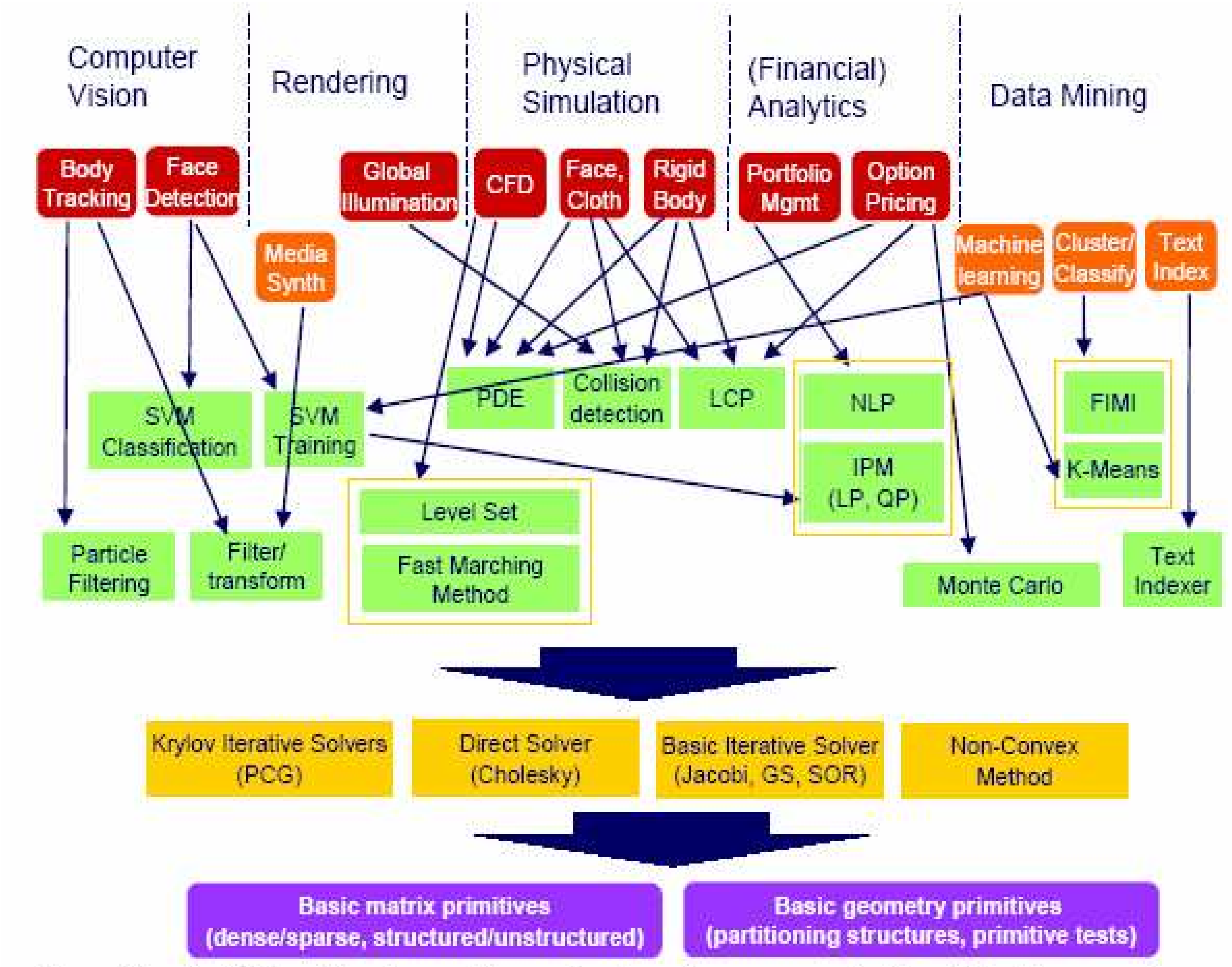
Outlook

- **Processor Design flow with Embedded FPGAs**
  - Current research
    - *Generic Modelling, Exploration and Implementation* of embedded FPGAs
    - Integration of rASIPs into complete SoC design flow (virtual prototyping)

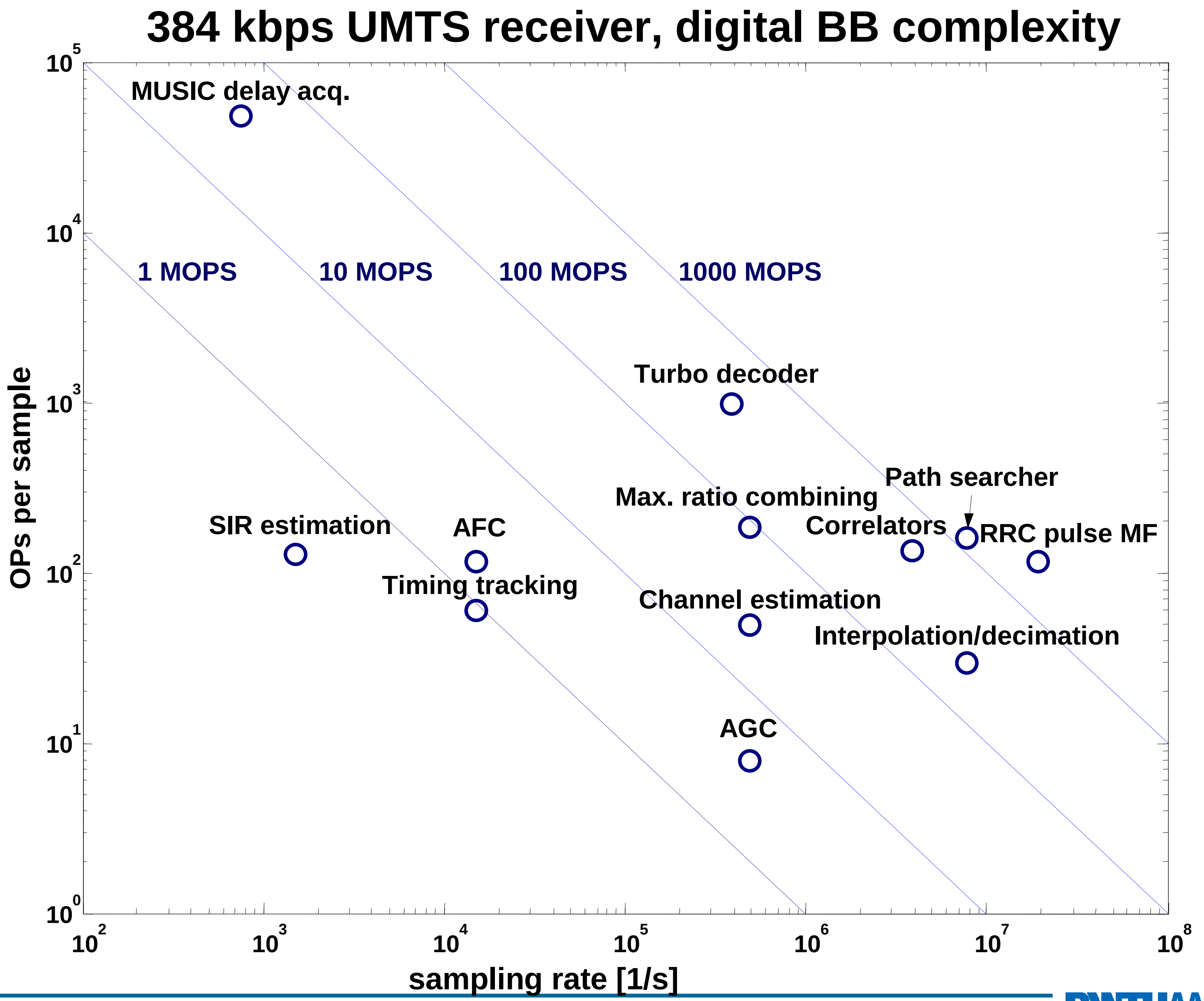
- **Applying the Methodology**
  - Cognitive radio, an upcoming research challenge for system designers, will be handled with the proposed tools and methodology

Thank You

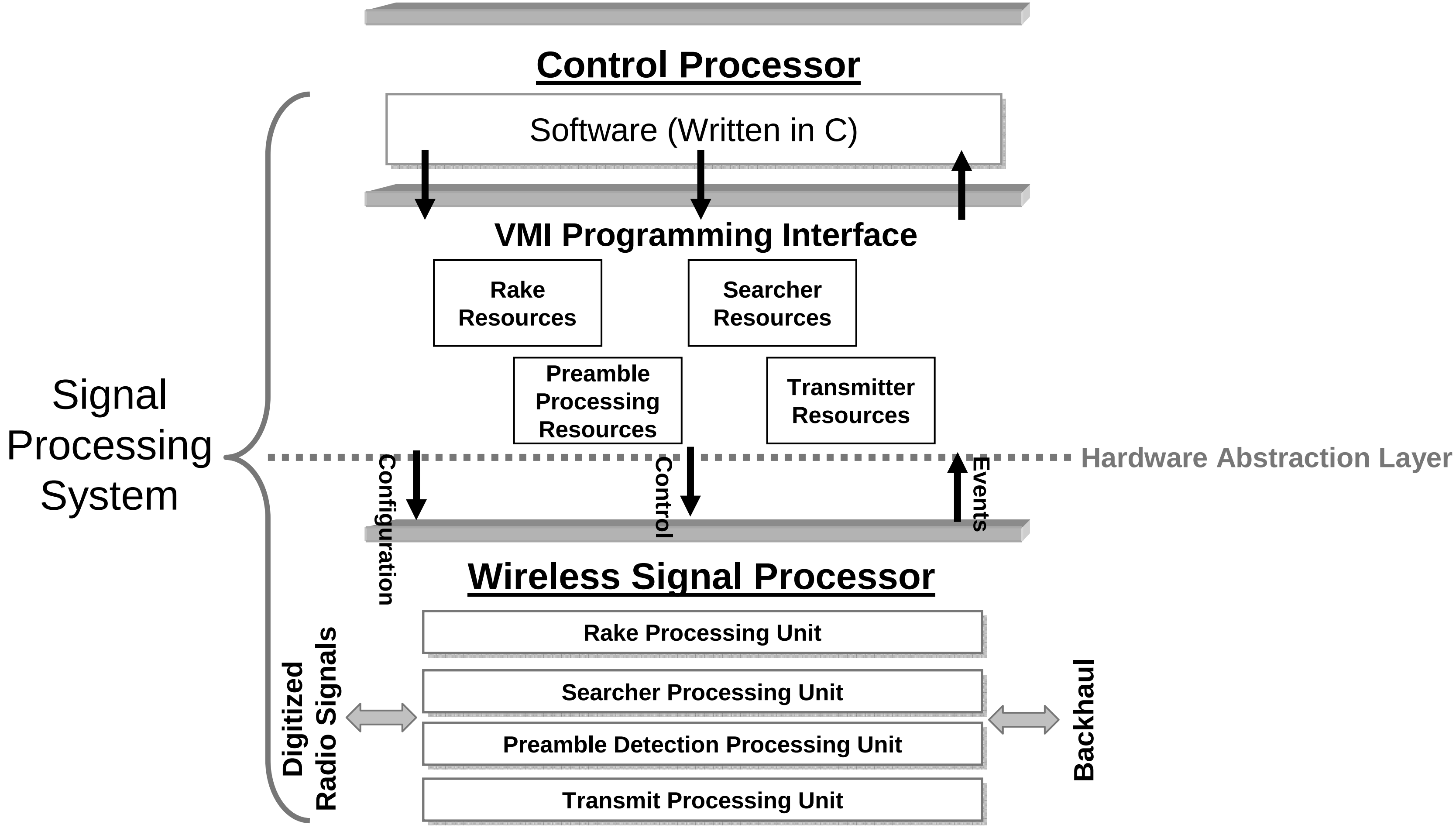
Intel RMS View (Recognition, Mining,Synthesis)



384 kbps UMTS Receiver BB Complexity



SW Programming Model



Signal Processing System

Control Processor

Software (Written in C)

VMI Programming Interface

Rake Resources

Searcher Resources

Preamble Processing Resources

Transmitter Resources

Hardware Abstraction Layer

Configuration

Control

Events

Wireless Signal Processor

Rake Processing Unit

Searcher Processing Unit

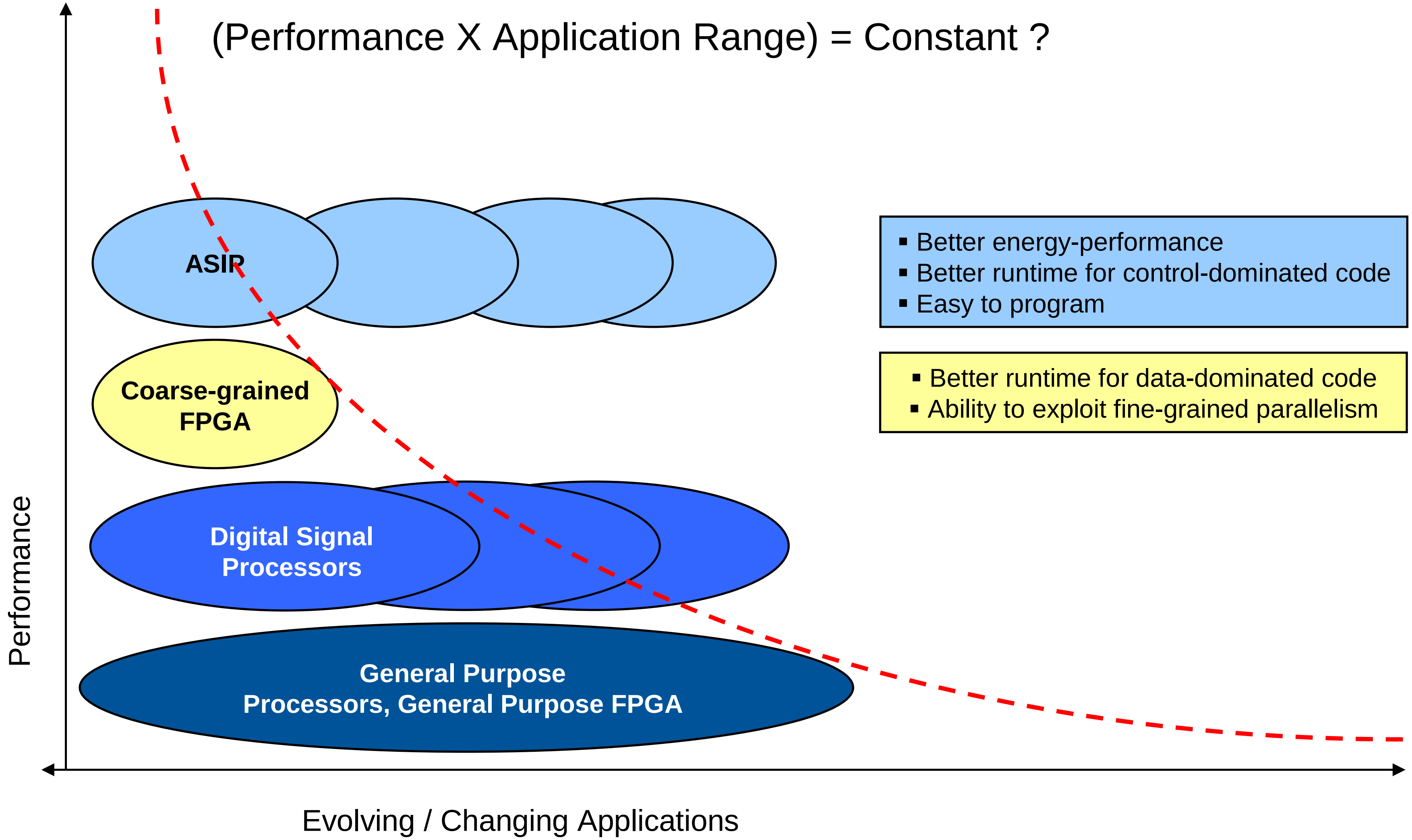
Preamble Detection Processing Unit

Transmit Processing Unit

Digitized Radio Signals

Backhaul

On Flexibility


(Performance X Application Range) = Constant ?
ASIP
Coarse-grained FPGA
Digital Signal Processors
General Purpose Processors, General Purpose FPGA
Performance
Evolving / Changing Applications

- Better energy-performance
- Better runtime for control-dominated code
- Easy to program

- Better runtime for data-dominated code
- Ability to exploit fine-grained parallelism