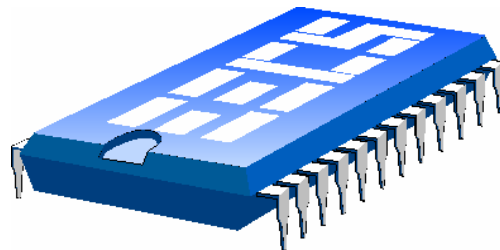


# FLPA-based Power Modeling and Power Aware Code Optimization for a TriMedia DSP

H. Blume, J. von Livonius, T. G. Noll

Lehrstuhl für Allgemeine Elektrotechnik  
und Datenverarbeitungssysteme  
RWTH Aachen



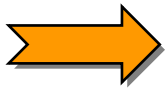
# Outline

- Motivation
- Short Review of FLPA-Modeling
- TriMedia Architecture
- Model Parameters
- Hybrid FLPA/ILPA Model
- Benchmarking
- Summary

# Motivation

## Energy consumption of tasks on DSP architectures depends on:

- Data width of memory accesses
- Compiler options, different configurations of DSP architecture
- Specific architecture elements like EDMA, Co-Processors
- Cache-Misses, Pipeline-Stalls



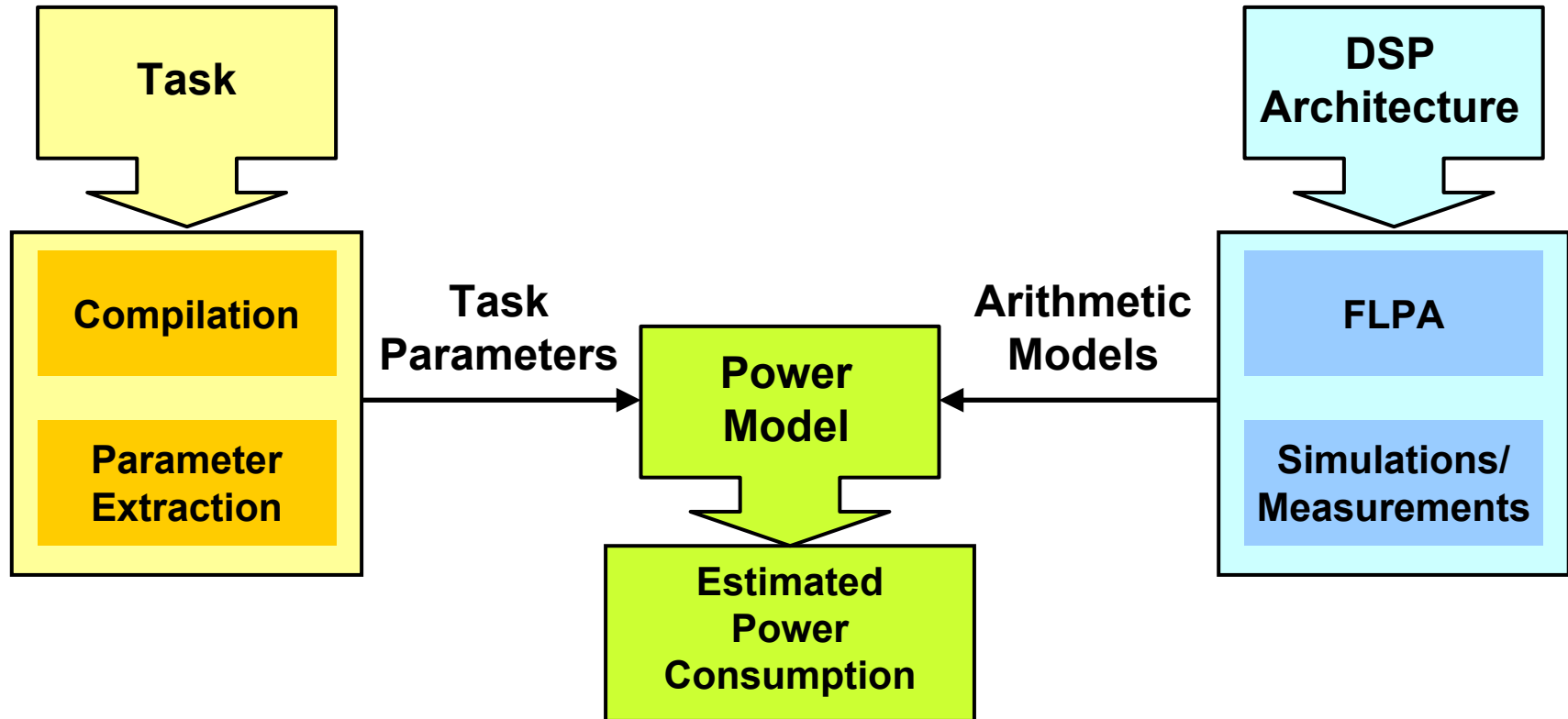
**Methodology for power estimation including all dependencies required**

## Advantages :

- No need for time consuming simulations or measurements
- Easy comparison of implementation alternatives in terms of energy consumption

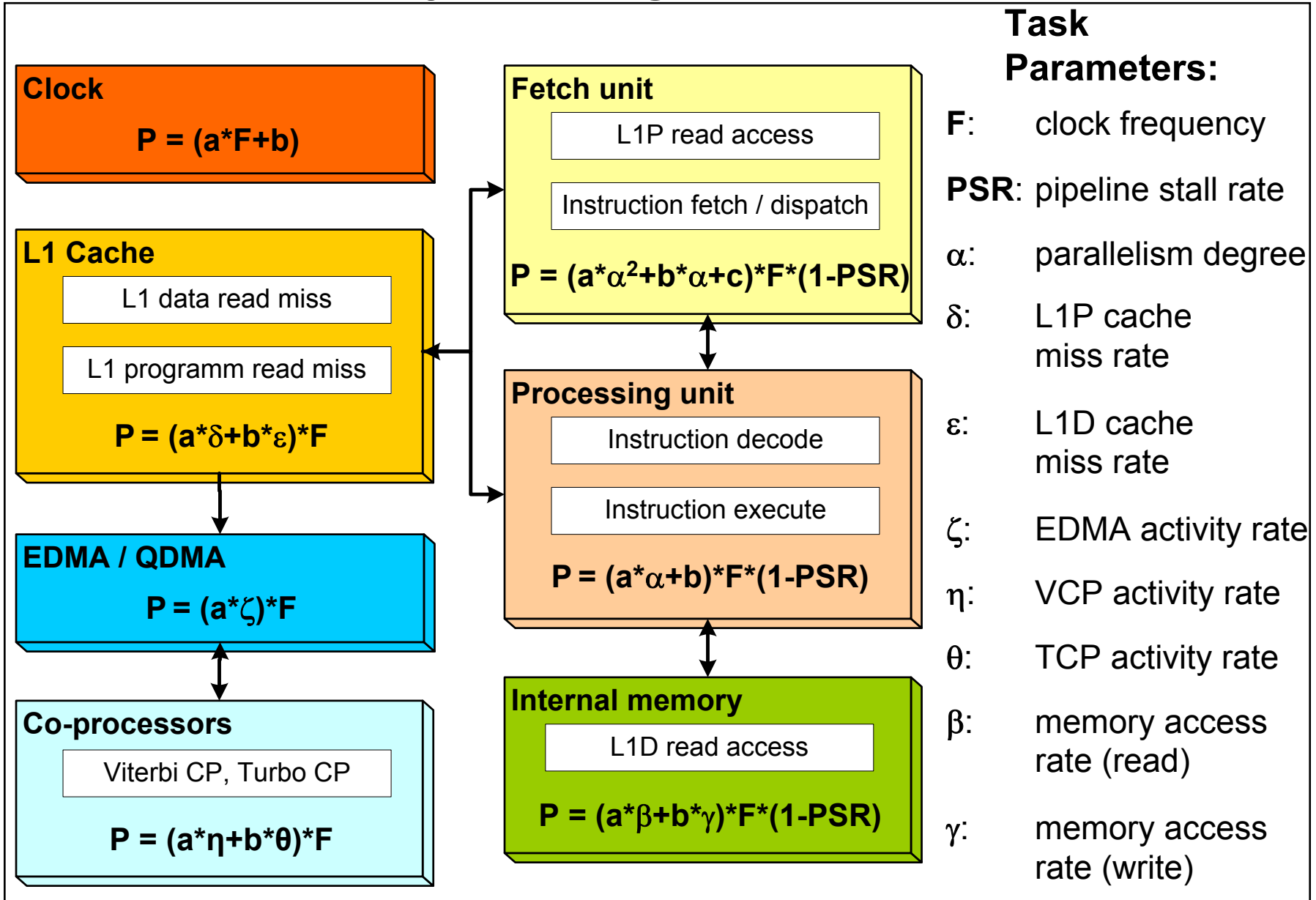
# Functional-Level Power Analysis (FLPA)

- DSP architecture separated into different blocks (Clock, Fetch-Unit, etc.)
- Energy consumption of each block described by arithmetic models
- Determination of algorithmic parameters by analysis of assembly code



[Laurent, Univ. South Brittany, 01]

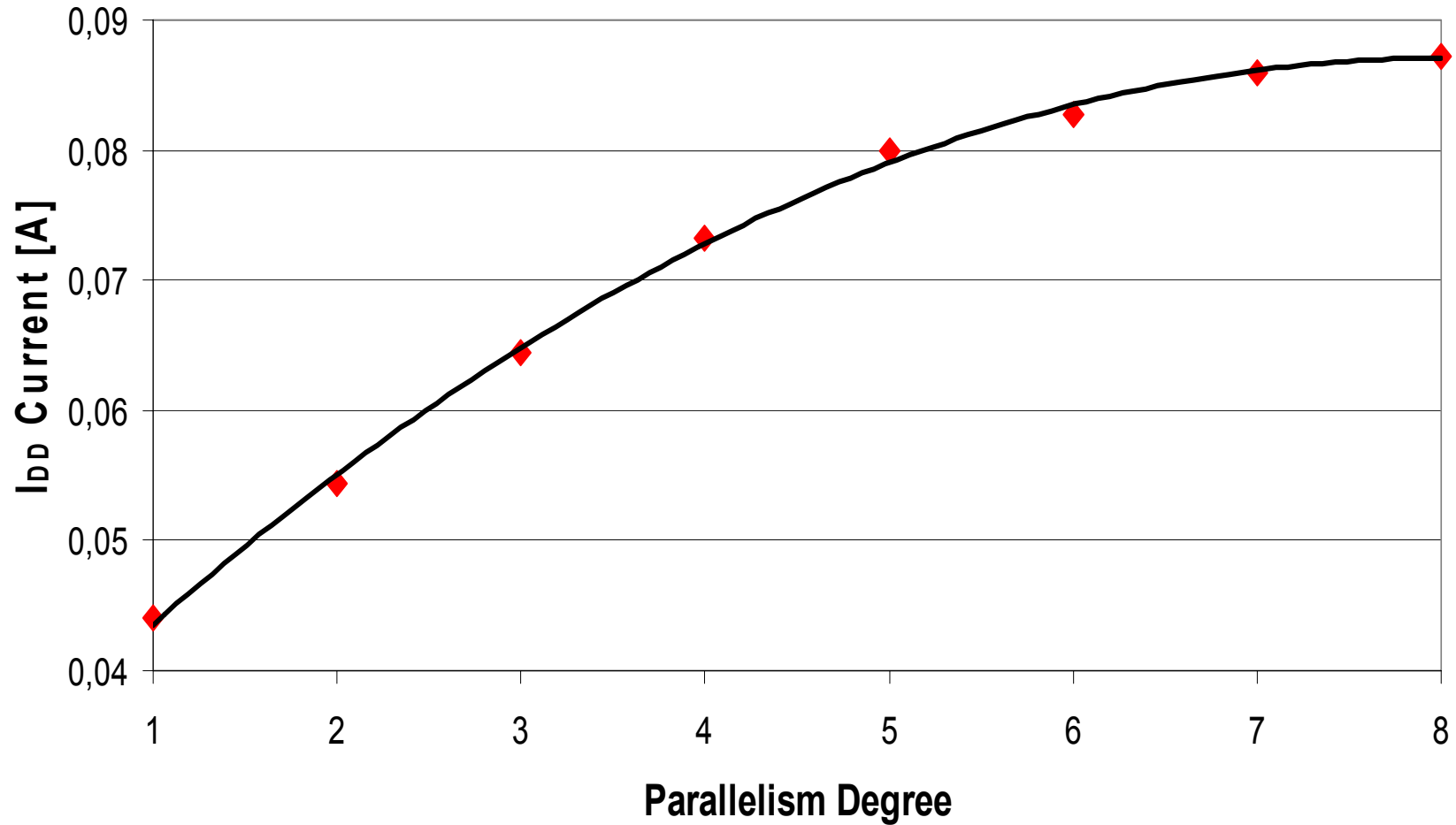
# Review: Exemplary Modeling of the C6416



# Current Consumption Fetch Unit

C6416 DSP

(1000 instructions/loop iteration)



# GUI for Power Estimation

computed automatically by code parser

Power Estimation Tool

## Power Estimation

Algorithmic parameters

Average number of functional units in use:  Parse

Cycles needed for one iteration of the algorithm:

Calculated samples in one iteration of the algorithm:

Internal level 2 memory

Internal memory read accesses

8 bit  16 bit  32 bit  64 bit

Internal memory write accesses

8 bit  16 bit  32 bit  64 bit

EDMA/QDMA

Copied bytes using EDMA/QDMA with CPU in IDLE state:

Copied bytes using EDMA/QDMA with CPU still operating:

Coprocessors

VCP cycles per iteration of the algorithm:  CPU Idle

TCP cycles per iteration of the algorithm:  CPU Idle

Cache/Pipeline

Known pipeline stalls per iteration of the algorithm:

Known L1P cache misses per iteration of the algorithm:

Known L1D cache misses per iteration of the algorithm:

DSP: TI TMS320C6416 @ 500 MHz

Results

Clock Power [mW]:

Fetch Power [mW]:

PU Power [mW]:

Memory Power [mW]:

EDMA Power [mW]:

Cache Power [mW]:

CoPro Power [mW]:

Total Power [mW]:

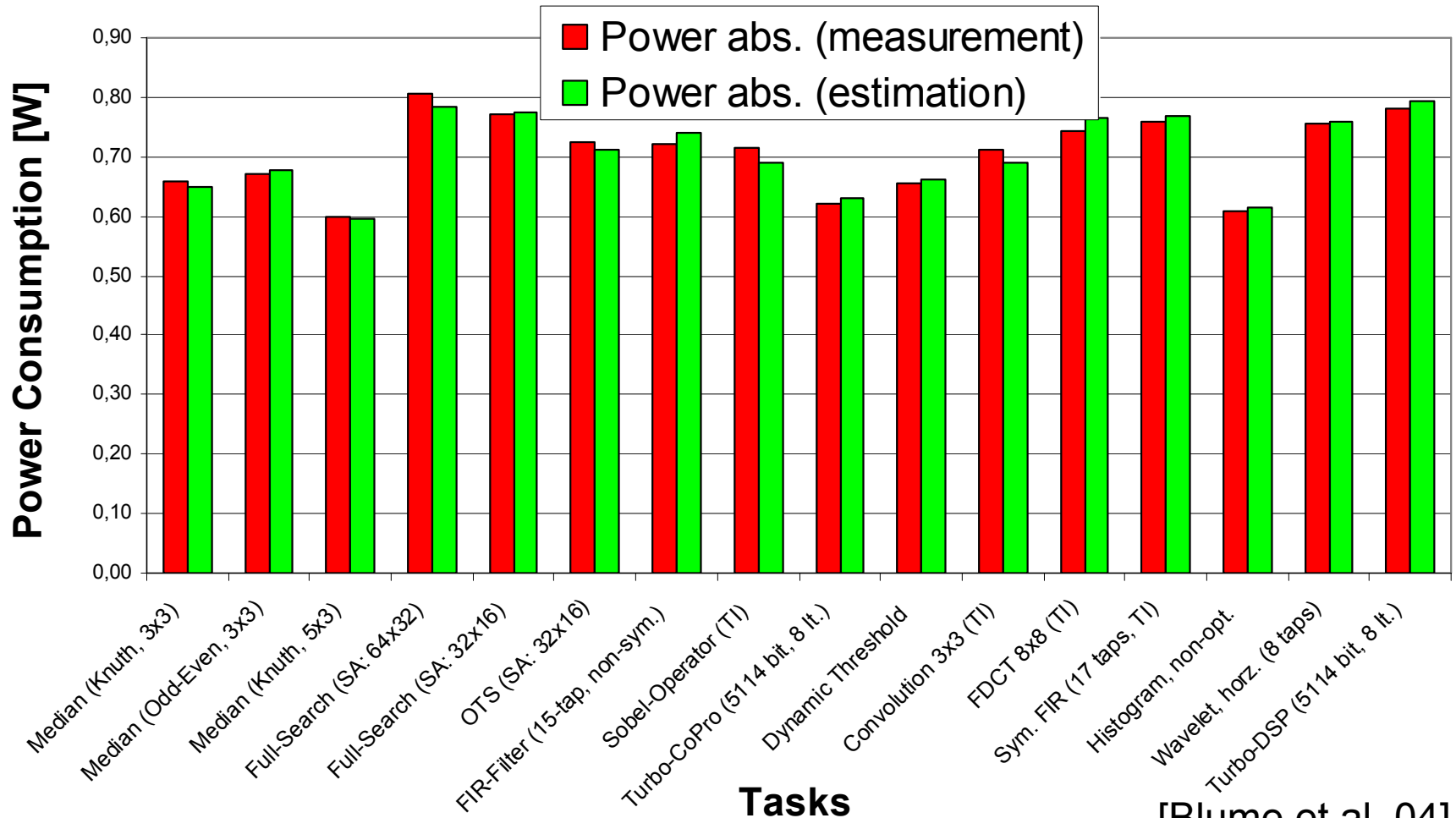
Energy/Sample (f.i.) [nWs]:

Energy/Sample (diff.) [nWs]:

Calculate Quit

# Results for C6416 (I)

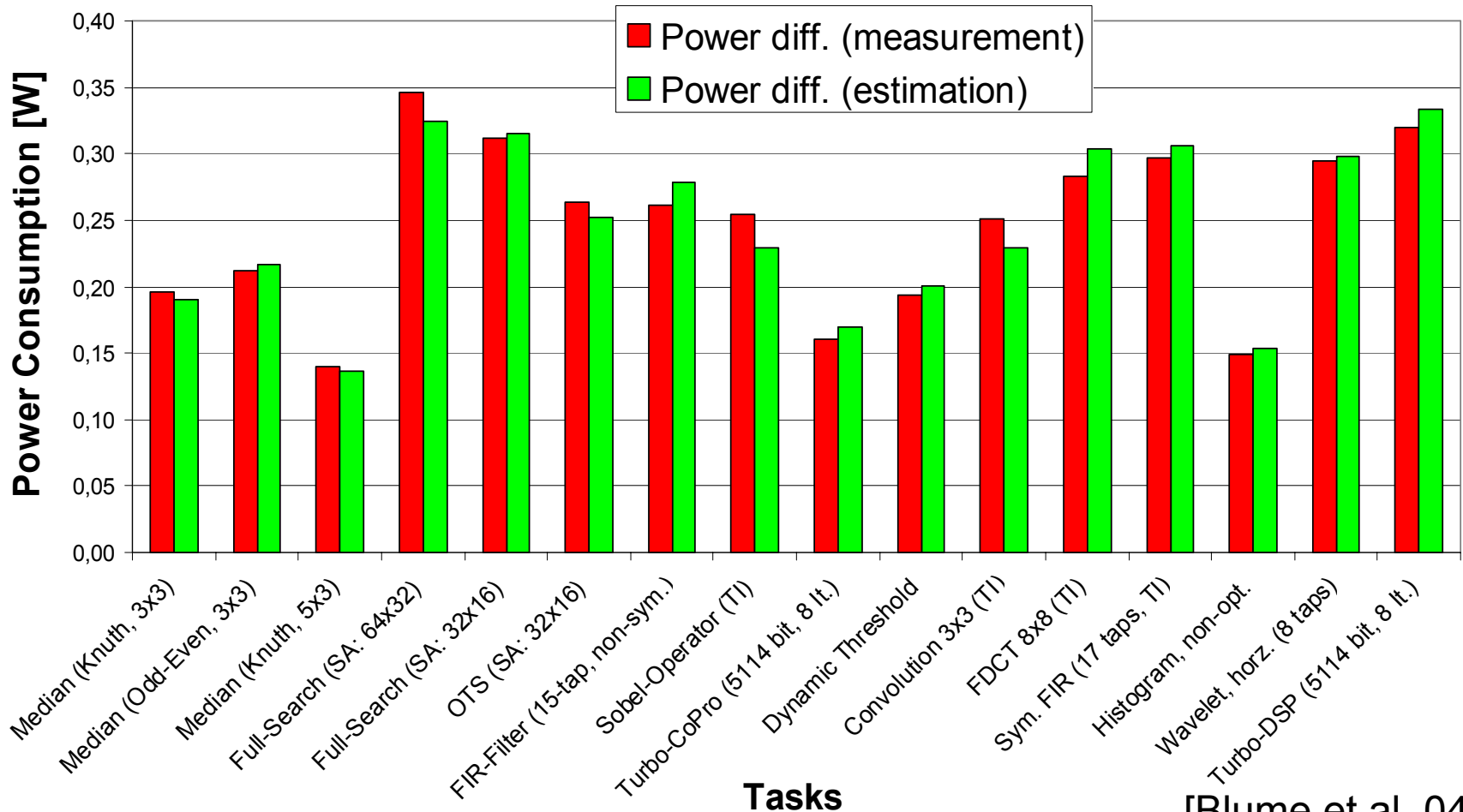
Estimation of **absolute** power consumption of DSP tasks on the C6416 DSP using FLPA (maximum error <3%)



[Blume et al, 04]

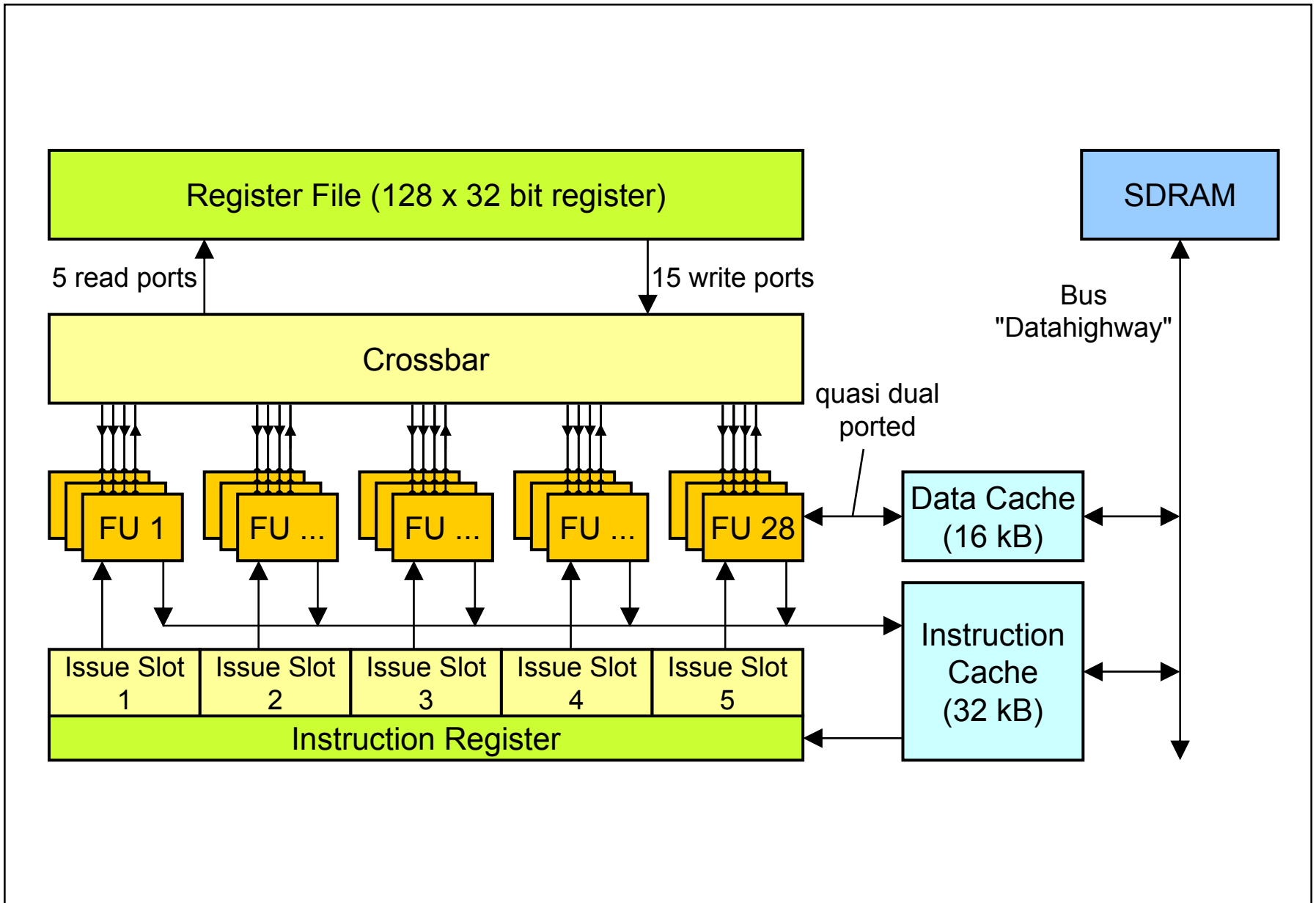
# Results for C6416 (II)

Estimation of **differential** (clock-system not included) power consumption of DSP tasks on the C6416 DSP using FLPA (maximum error <10%)

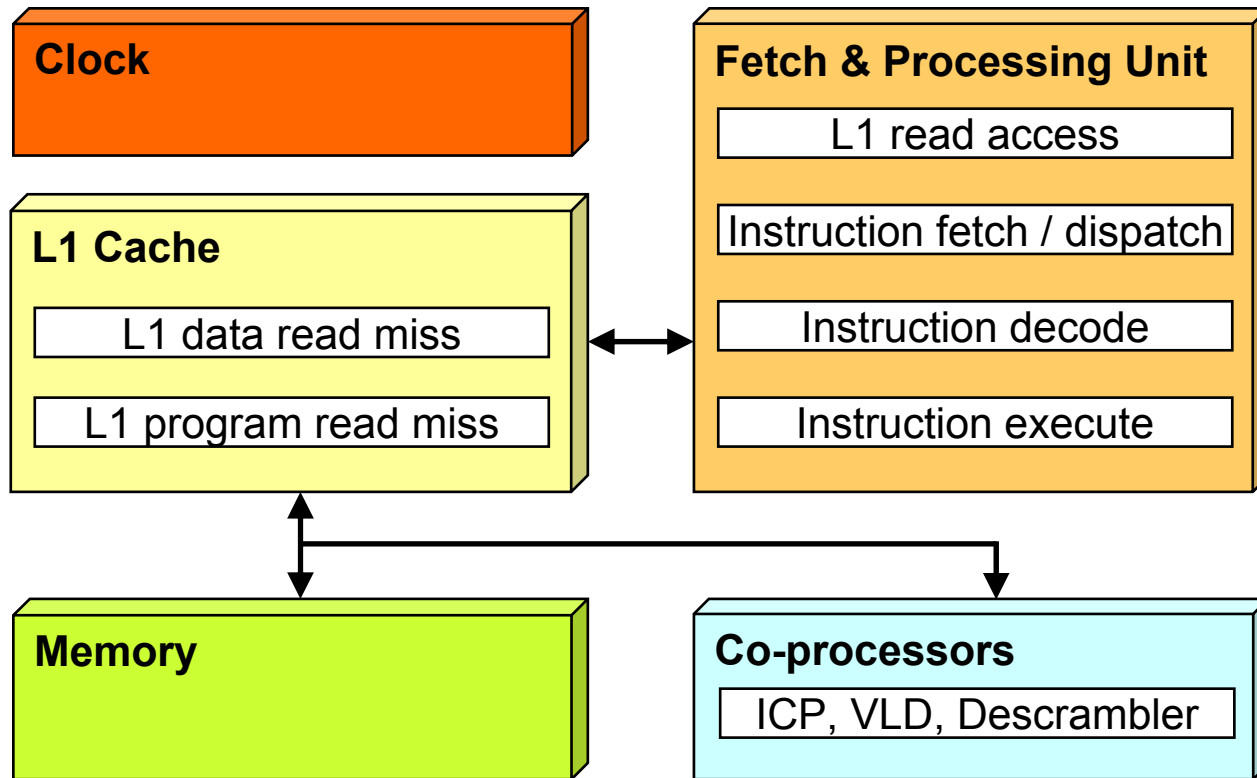


[Blume et al, 04]

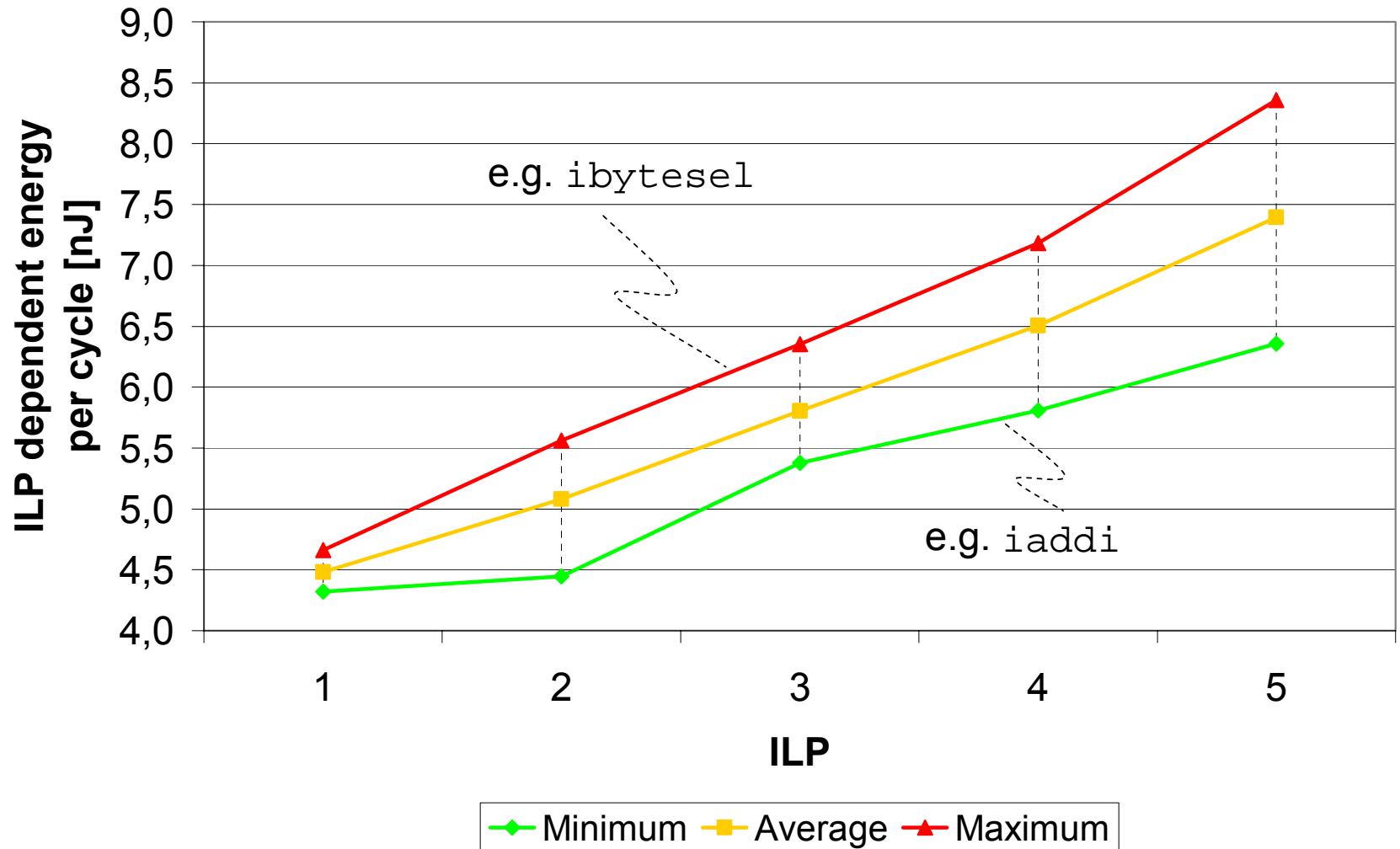
# TriMedia Architecture



# Modeling of the TriMedia TM1300



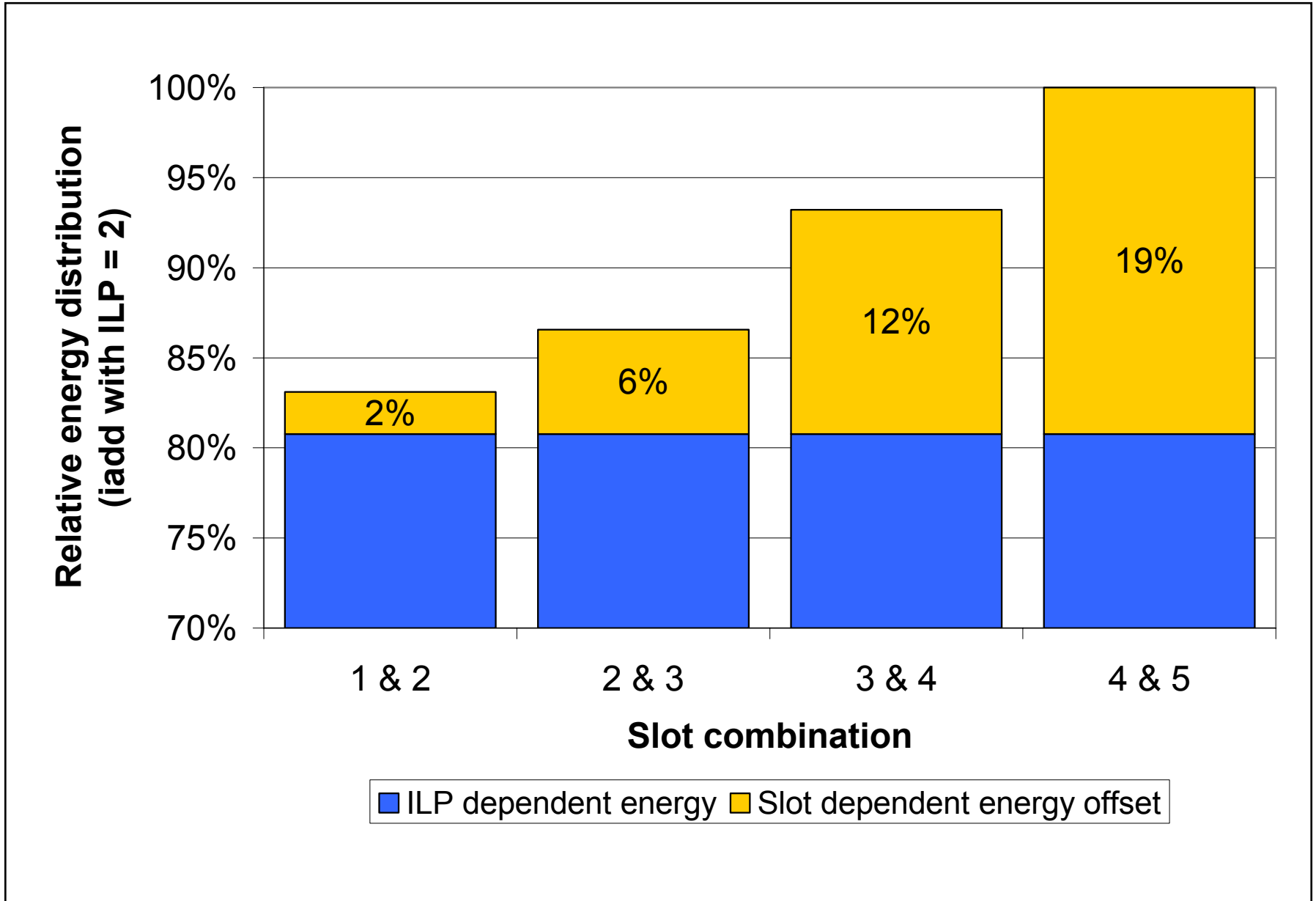
# ILP Dependent Energy



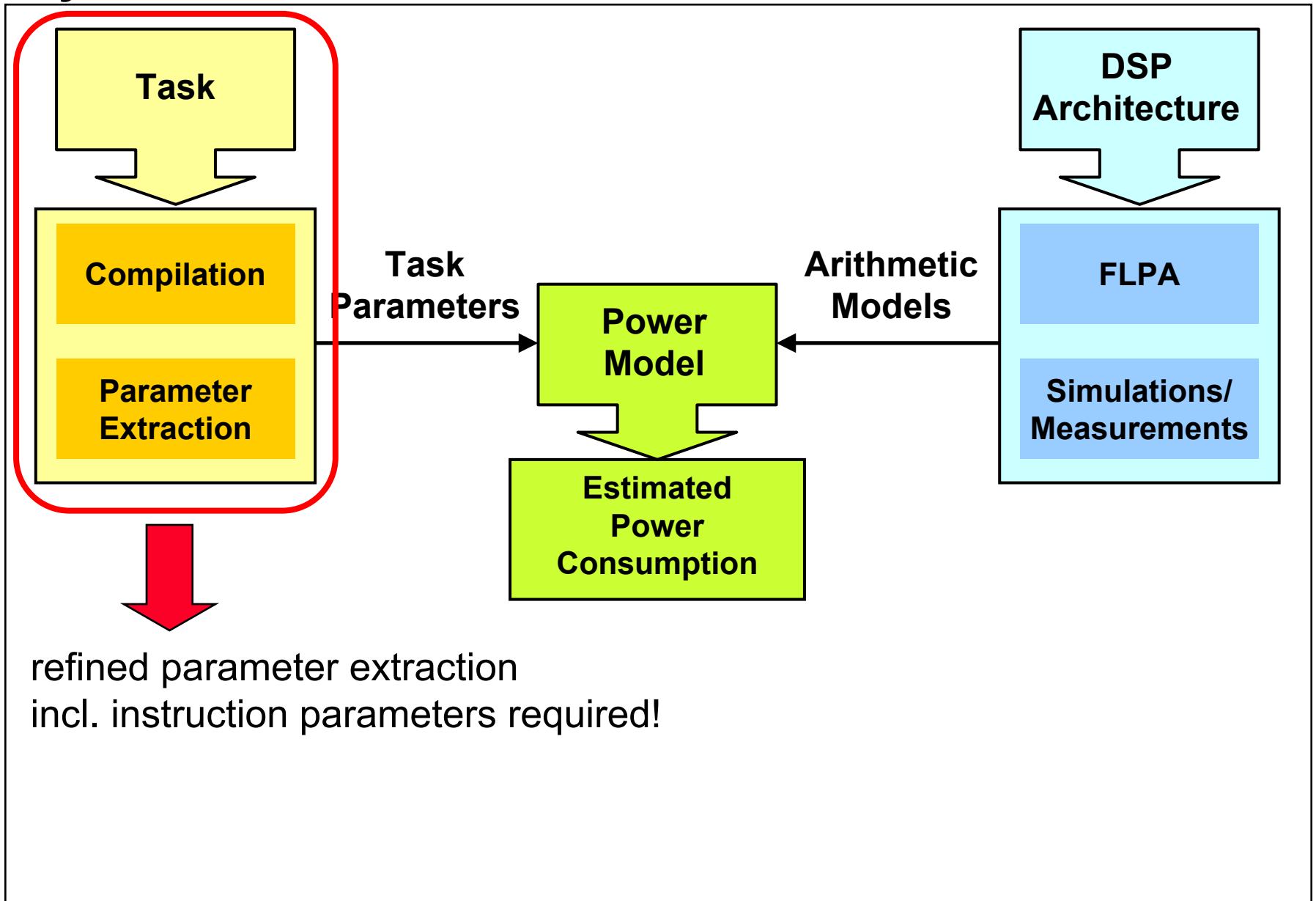
# Slot Dependent Energy Offset



# Influence of the Slot Allocation



# Hybrid FLPA/ILPA Model



# Instruction Dependent Energy Distribution

```
(* cycle 5 *)  
IF r1 iadd r33 r6 -> r13,  
IF r1 ufir8uu r41 r50 -> r54,  
IF r1 iaddi(0xa) r16 -> r56,  
IF r1 nop,  
IF r1 iimm (0xc0) -> r9;
```



Cycle	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5
5	iadd	ufir8uu	iaddi	nop	iimm



**Energy per Cycle depends on:**

- Instruction mix
- Instruction Level Parallelism (ILP)
- Slot allocation

# Instruction Dependent Energy Calculation

```
(* cycle 5 *)  
IF r1 iadd r33 r6 -> r13,  
IF r1 ufir8uu r41 r50 -> r54,  
IF r1 iaddi(0xa) r16 -> r56,  
IF r1 nop,  
IF r1 iimm (0xc0) -> r9;
```

**Code  
Parser**

ilp per cycle

**Profiler**

**Power Model**

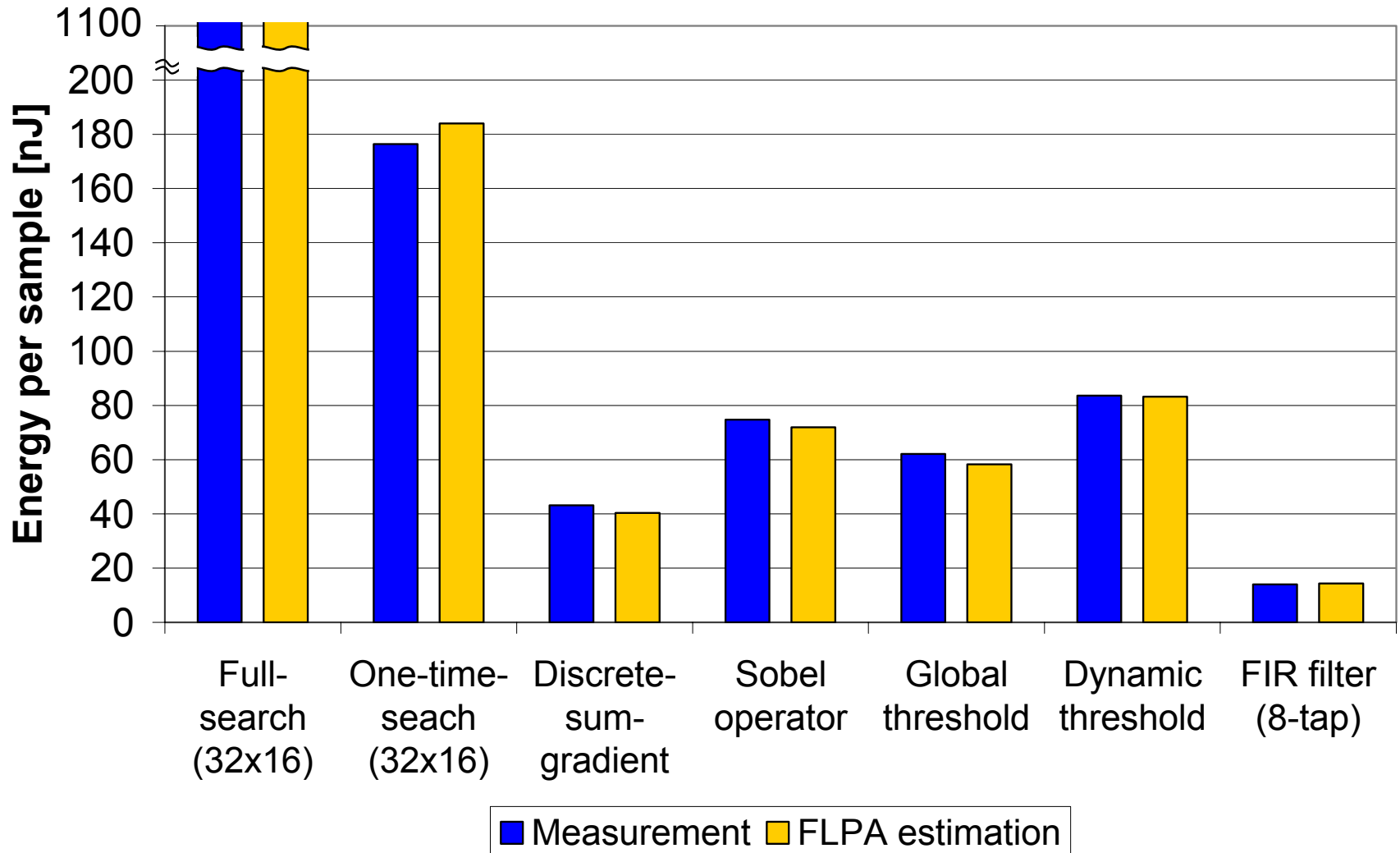
**Database**

decision tree  
and cache  
statistics

ilp and slot specific  
energy offsets

**Estimated  
Power**

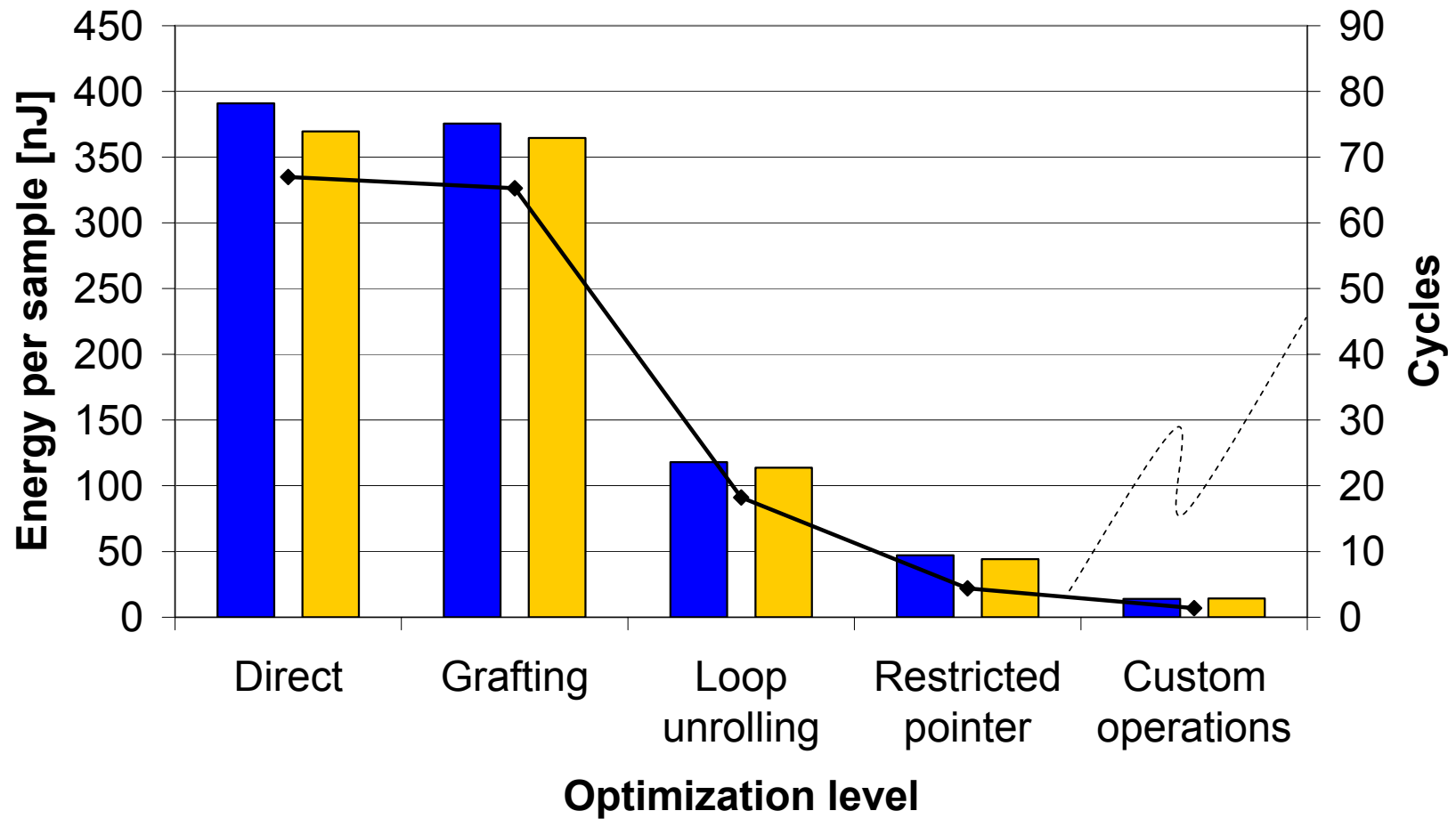
# FLPA Estimation Results for the TriMedia



- Maximum error less than 6 % for the absolute energy conversion
- Energy per output sample differs up to factor 80

# Influence of DSP-specific Code Optimization

- FIR-Filter with 8 taps



→ Maximum error less than 6 % over all optimization levels

# Summary

- Elaboration of a hybrid FLPA/ILPA-model
- Essential parameters:
  - Frequency
  - Cache-Misses
  - # Cycles
  - ILP
  - Distribution of Instructions
- Implementation of a parser for the parameter-extraction
- Estimation results provide a high accuracy which can be applied advantageously even in the code optimization phase