FPL-BENCHMARK ACTIVITIES

Andreas Beyer
Udo Kebschull
Kirchhoff Institute for Physics
University of Heidelberg, 02.09.2010

Motivation

- Observations:
  - Authors selects a different set of design examples to prove their results
    - It is a hard job to implement a full featured design example from scratch
    - There is only poor comparison to other approaches
  - Most well-known benchmark suits are rather outdated
    - e. g. MCNC 93
- First discussions during FPL 09 in Prague

Motivation

- Idea:
  - Turn **fpl.org** into a community web page
  - Use **fpl.org** as a platform to collect and provide FPGA related benchmarks
    - Collect various design examples used in publications and make them available for other researchers who are working at the same research topic
    - Download area for benchmarks
    - Chat room and forum
    - Bring in touch researchers who work in the same research area
  - Community driven benchmark collection

Motivation

- Current status:
  - This is only a preliminary platform implementation to demonstrate how such a benchmark suite could look like
  - We will not be able to implement and maintain all provided designs
    - We need your help!

- However:
  - **fpl.org** could be seen as a basic infrastructure to collect and offer designs which can easily be used for benchmarking your research

Presentation

- Motivation
  - Introduction
  - Benefits
- Implementation
  - Structure
  - Representation
- Future work

Definition of “benchmarking”:

- process of comparing one's performance metrics to
  - industry bests
  - best practices
- dimensions typically measured are
  - quality / time / cost
- Improvements from learning mean doing things
  - better / faster / cheaper

Current situation

- Free (popular) benchmark suites for FPGAs
  - MCNC (Circuits) → 1993
  - RAW (Algorithms) → 1997
  - MiBench (various applications) → 2002
  - IWLS (Circuits) → 2005
- Commercial suites
  - EEMBC
  - AccelFPGA
- mostly 1:1 comparison in papers
  - no possibility for comparison with several projects

Goals

- assembling a set of benchmarks
  - to represent state of the art technologies
  - offer a starting point for further developments
- creating presentation platform
  - structured by fields of operation
  - make implementations comparable
  - follow the evolution process of the codes

Abstract view on development

- This is a community driven benchmark suite
  - We want to support your workflow
  - It’s not our intention to keep you busy
  - overhead of maintainers and contributors has to be kept to a minimum

Abstract view on development

- Define the problem you are working on

Abstract view on development

- Define the problem you are working on
- Describe idea and approach

Abstract view on development

- Define the problem you are working on
- Describe idea and approach

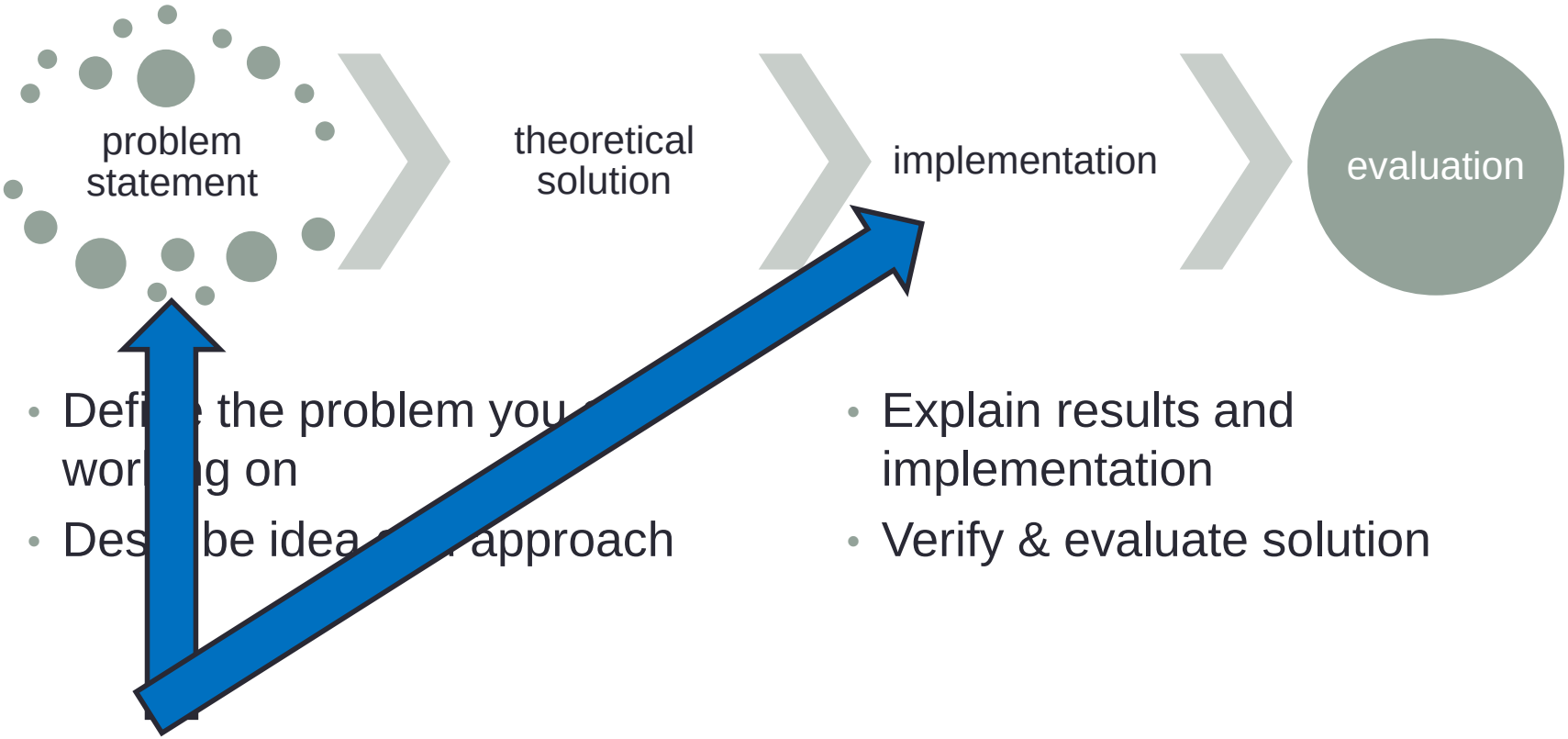
- Explain results and implementation

Abstract view on development

- Define the problem you are working on
- Describe idea and approach

- Explain results and implementation
- Verify solution & evaluate selected metrics

Abstract view on development

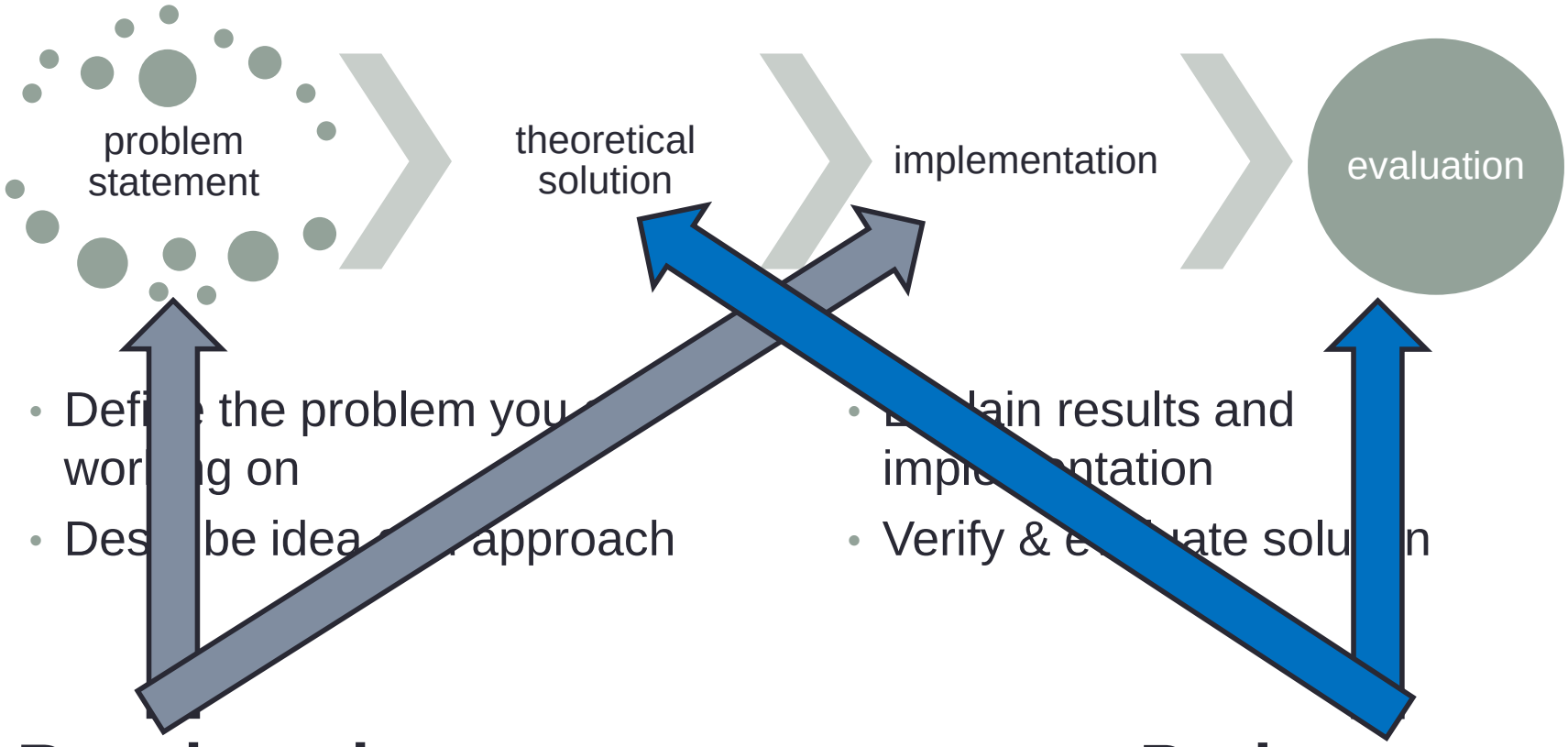


- Define the problem you are working on
- Describe idea and approach

- Explain results and implementation
- Verify & evaluate solution

Benchmark
= Purpose & Code & Tools

Abstract view on development



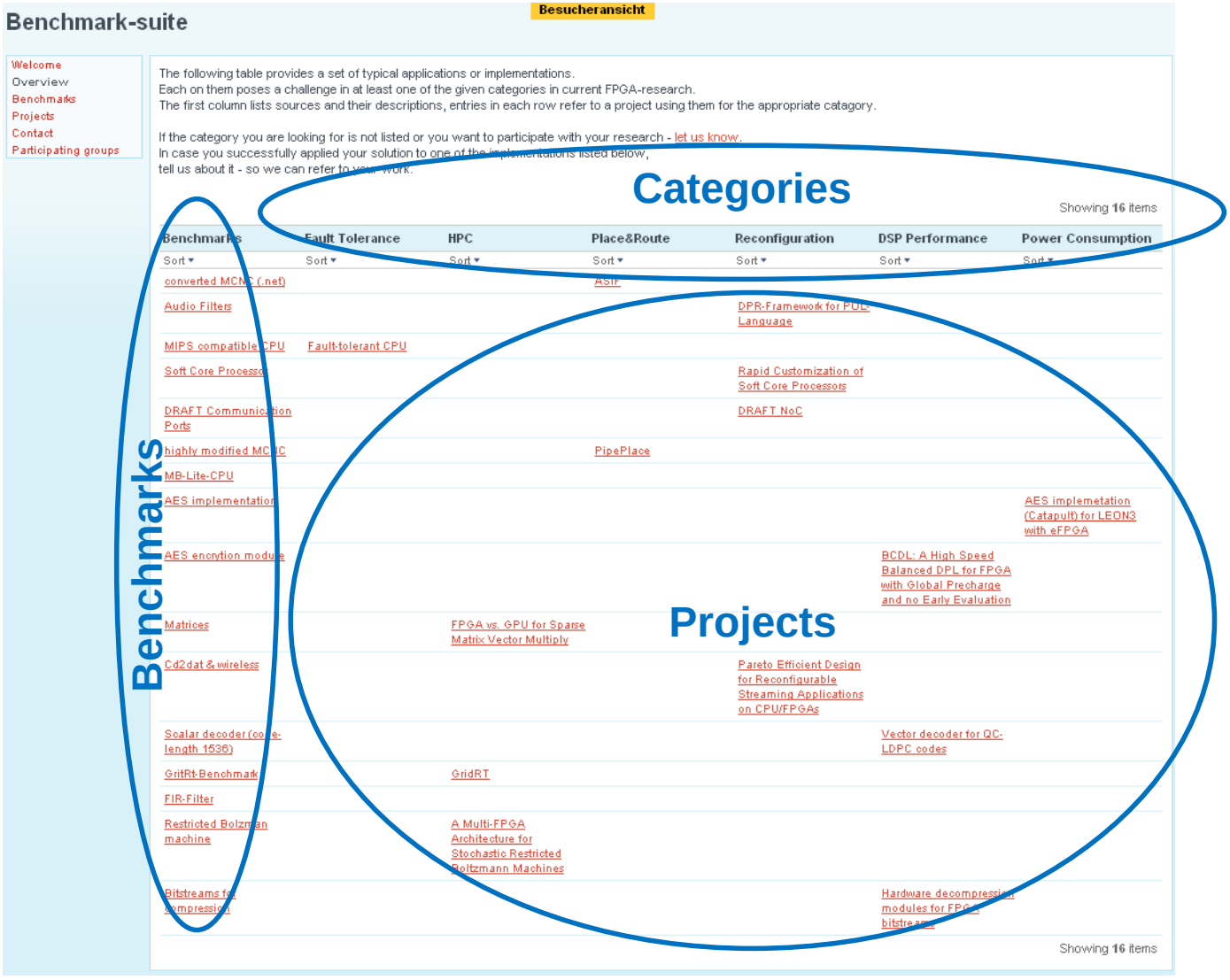
Benchmark
= Purpose &
Code &
Tools

Project
= Approach &
Measurements &
Related publications

Realization

- The suite consists of Projects and Benchmarks
- Each benchmark is used in different projects
  - as input data
  - as basis for further implementations
- Each project is based on one benchmark
  - the posted results are comparable
    - to the original benchmark
    - to other projects aiming for different features
  - projects are arranged
    - in categories according to their field of operation
    - in the same line as the corresponding benchmark

Structure (1) – Overview



Structure (2) – Benchmark

Benchmark-suite

Besucheransicht

Welcome
Overview
Benchmarks
Projects
Contact

Benchmarks >

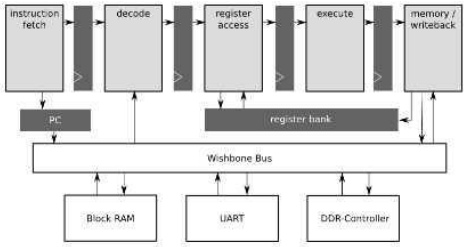
MIPS compatible CPU

posted May 28, 2010 3:23 PM by Andreas Beyer [**updated a minute ago**]

The final version of this CPU is quite similar to the MIPS R2000/R3000 system and is supplied with

- 32 bit address and data width, byte addressed, big or little endian
- a five stage pipeline
- a hardware multiplication and division unit
- interrupt and exception handling as suggested by the MIPS specification
- a Wishbone bus interface

A graphical representation of the CPU system is shown below.



The CPU and any peripherals can be controlled using the standard GCC MIPS cross compiler. The implemented MIPS CPU is a three op rand architecture with 32 general purpose 32 bit registers and several co-processor registers used for interrupt and exception handling. It neither has a cache nor a MMU at this time. The lack of separate instructio and data caches require that both, instruction fetch and memory access share the same bus. The pipelined execution of commands is separated into instruction fetch (IF), instruction decode (DE), register access (RA execute (EX) and memory / writeback (MW). IF-stage fetches the next instruction from the bus and DE-stage decodes it and creates the select signals for the multiplexers in the later stages. RA-stage loads the contents of the registers according to these signals. EX-stage performs the requested logic combination of the previously loaded register contents. MW-stage redirects the result back into one of the registers or per rms a bus access. The Wishbone bus currently grants access to an internal BlockRAM, a UART controller and a DDR-SDRAM controller.

This text describes the non-hardened version of the CPU as it is in the attached archive.
See Overview to find the related "fault-tolerant CPU" project.

Elements	utilization
flip-flops	2254
look-up tables	3502
DSPs	4

Attachments (1)

hw_nFT.tar.gz - on May 28, 2010 4:21 PM by Andreas Beyer (version 1)
27k Download

Anmelden Nutzungsbedingungen Missbrauch melden Seite drucken Zugriff eischränken | Powered by Google Sites

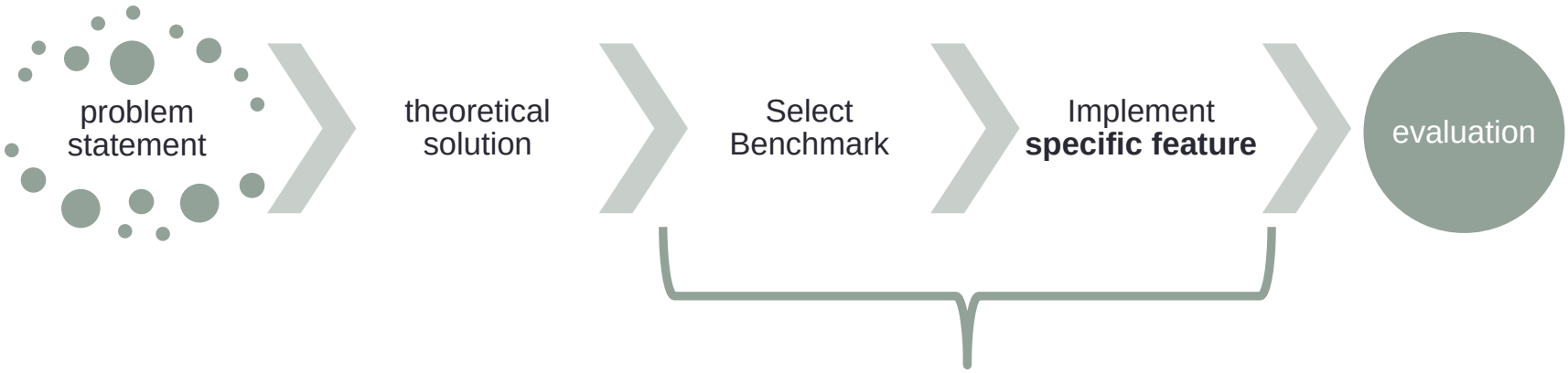
Purpose

Sources

Structure (3) – Project

“cheaper, better, faster”

- Benefits of using benchmarks during development

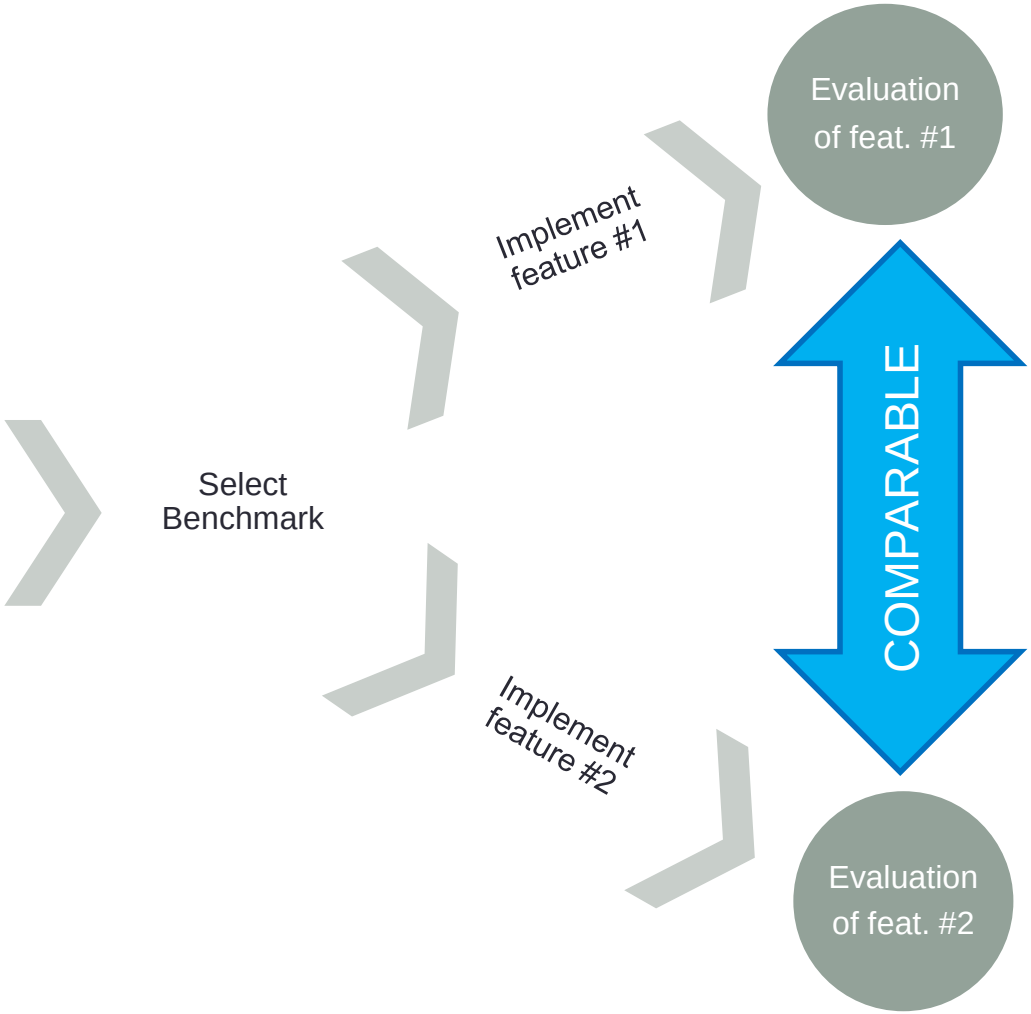


Implementation efforts reduced:

- Start with existing sources from Benchmark
- Implement specific feature or apply mechanism

“comparing one's performance metrics”

- two groups with the same field of operation
- they choose the same benchmark to start from
- implement their feature or apply mechanism
- results are comparable due to using same basis



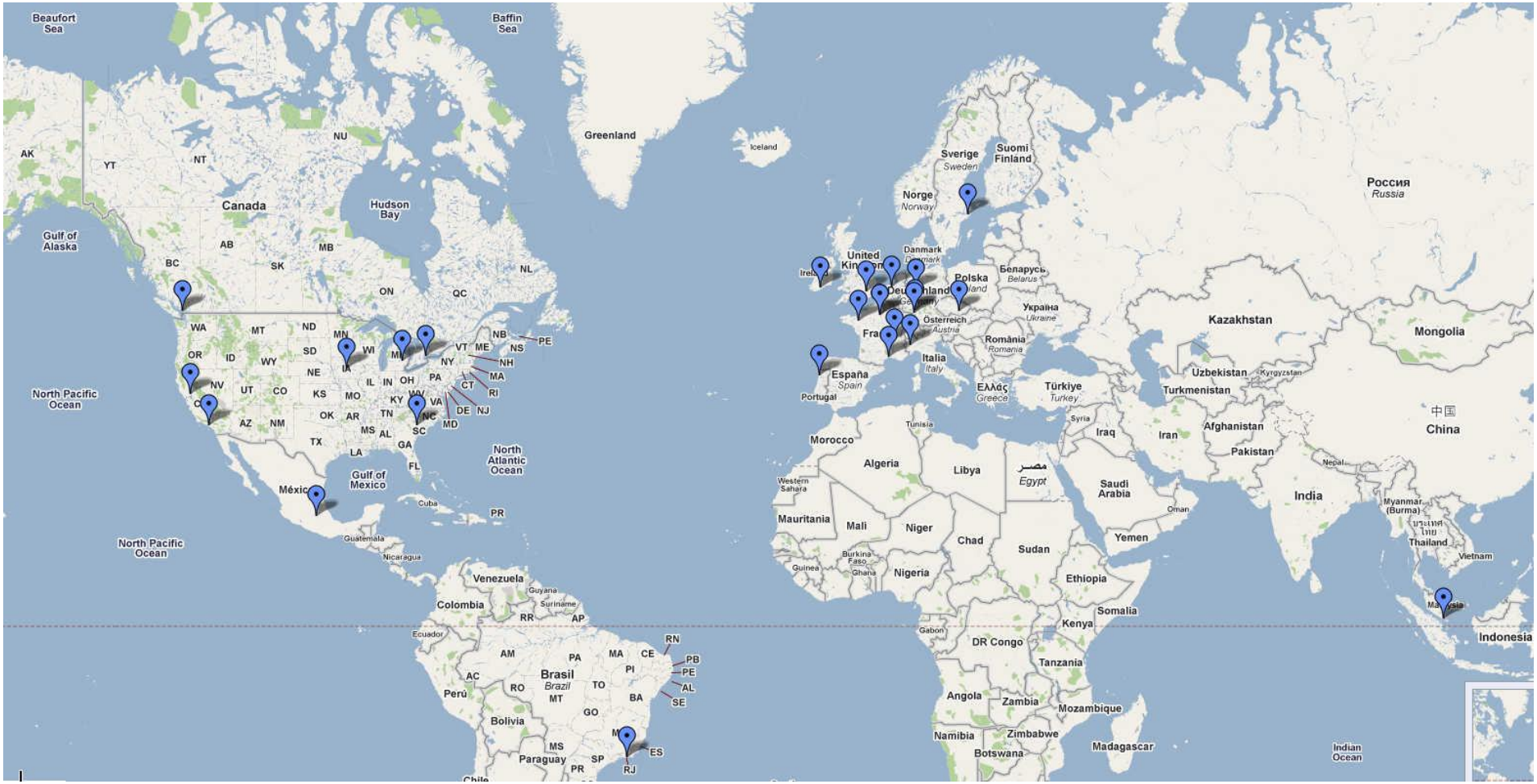
Inspected publications

- All papers of following conferences (~2000)
  - DAC 2008 & 2009
  - Date 2009 & 2010
  - DSD 2010
  - FPL 2008 & 2009
  - FPT 2009
  - ReConFig 2009
  - SOCC 2009
- Criteria
  - Quantifiable results / metrics & measurements
  - Aims for FPGA
  - Preferably evaluated with own benchmarks
  - willing to share sources

Areas of operation

- Relevant for FPL-BS
  - Fault Tolerance
  - HPC
    - FPGA and GPU
  - Synthesis
  - Place&Route
  - Reconfiguration
    - Tools and Application
  - DSP Performance
  - Power Consumption
  - ...
- Unnoticed areas
  - Spiking neural networks
  - Random number generator
  - Motion detection
  - Altered FPGA hardware

Contributors



200 requests → 50 responses → 35 Dialogs → currently 25 candidates → 17 contibutions

How to contribute

- Procedure to publish a new “Benchmark”
  1. contact benchmark@fpl.org
  2. provide your contribution
     - describe purpose or problem statement
     - assemble sources
     - give instructions how to compile
  3. provide the corresponding “Project”
  4. maintain your benchmark
     - Updates
     - Answer questions from other users (forum)

How to contribute

- Procedure to publish your results as “Project”
  1. contact benchmark@fpl.org
  2. provide your contribution
     - describe your approach
     - present metrics & measurements
     - list related publications
     - poste contact information / affiliation
     - tell us the desired category
       - where you fit in
       - to create
  3. maintain your Project
     - Updates
     - Answer questions from other users (forum)

Future work (planned services)

- Automated evaluations
  - Cluster and web interface already set up
  - sources will be “nightly build”
    - for various platforms
    - using different compilers / P&R tools
- RSS-Feed
  - Errata / Updates / News

FPL-Benchmark Suite

Web-link benchmarksuite.beyer-andreas.net

(→ will become www.fpl.org/benchmark soon)

Thank your for your attention