Digest of the 1st
FPGA Developers' Forum

Filiberto Bonini (CERN EP-ESE), for the FDF organizers

FPGA Users Group – TWEPP 2024.10.02 – Glasgow, UK

https://indico.cern.ch/event/1381495/contributions/6036030/

Outline (1)

- **New initiative**
  - Kick-off @ CERN, 11-13 June 2024

Outline (2)

- **New initiative**
  - Kick-off @ CERN, 11-13 June 2024
- **FPGA development**
  - Zoom on technical details, rather than the big-picture (physics) goal

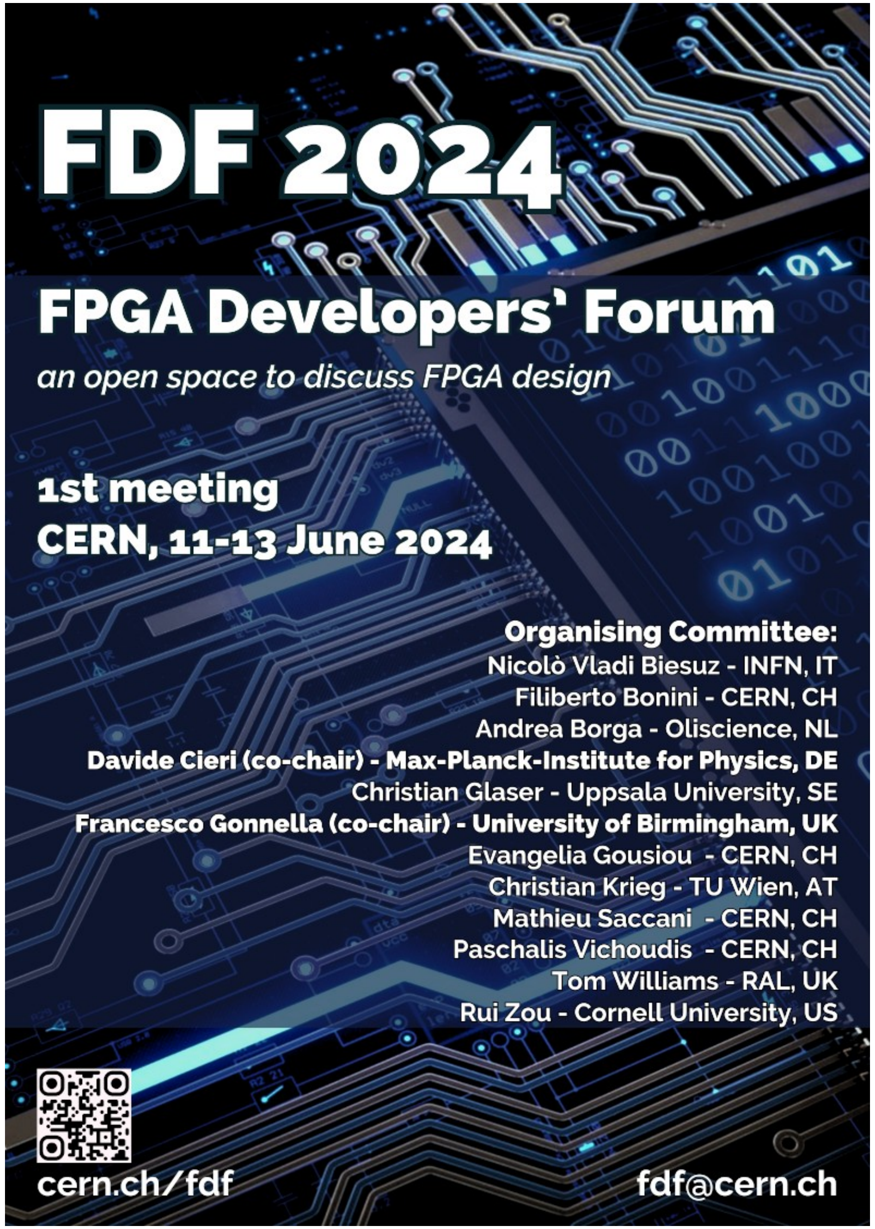
Outline (3)

- **New initiative**
  - Kick-off @ CERN, 11-13 June 2024
- **FPGA development**
  - Zoom on technical details, rather than the big-picture (physics) goal
- **Forum concept**
  - Collaborative community to share lessons & solutions

Outline (4)

- **New initiative**
  - Kick-off @ CERN, 11-13 June 2024
- **FPGA development**
  - Zoom on technical details, rather than the big-picture (physics) goal
- **Forum concept**
  - Collaborative community to share lessons & solutions
- **Positive feedback**
  - From the attendees (130+ in-person, 100+ webcast)

Digest of the presentations



FDF 2024

FPGA Developers' Forum

an open space to discuss FPGA design

1st meeting
CERN, 11-13 June 2024

Organising Committee:
Nicolò Vladi Biesuz - INFN, IT
Filiberto Bonini - CERN, CH
Andrea Borga - Oliscience, NL
Davide Cieri (co-chair) - Max-Planck-Institute for Physics, DE
Christian Glaser - Uppsala University, SE
Francesco Gonnella (co-chair) - University of Birmingham, UK
Evangelia Gousiou - CERN, CH
Christian Krieg - TU Wien, AT
Mathieu Saccani - CERN, CH
Paschalis Vichoudis - CERN, CH
Tom Williams - RAL, UK
Rui Zou - Cornell University, US

cern.ch/fdf

fdf@cern.ch

1. HDL Cores

"It's the Nth time I need this unit…!"

- **Design units, Memories, CDC**
- **Interfaces, Protocols**
- **Functions, Types**
- **Sharing, reusing and improving**
  - vs reinventing the wheel

Examples

- PoC (TU Dresden)
- hdl-modules (Lukas Vik)
- general-cores (CERN BE-CEM-EDL)
- OpenCores.org
- Open HW Repository

HDL Cores – Colibri

“It’s the Nth time I need this unit…!”

- **Design units, Memories, CDC**
- **Interfaces, Protocols**
- **Functions, Types**
- **Sharing, reusing and improving**
  - vs reinventing the wheel

Examples

- PoC (TU Dresden)
- hdl-modules (Lukas Vik)
- general-cores (CERN BE-CEM-EDL)
- OpenCores.org
- Open HW Repository
- Colibri (CERN LHCb DAQ)

Alberto’s talk next

2. HLS & Algorithms

"The applied physicist's corner"

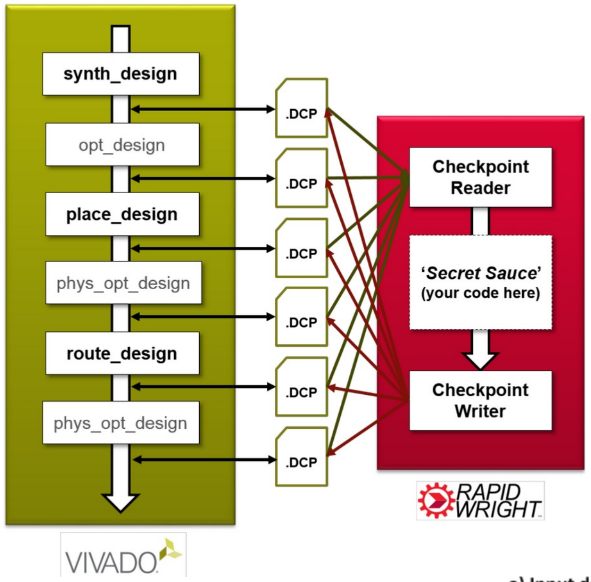
- **HLS being developed by Xilinx for the past 10+ years**
- **More gentle learning curve than HDL for non-EE, physicists**
  - Physics data analysis
  - Machine Learning onto FPGAs

Examples

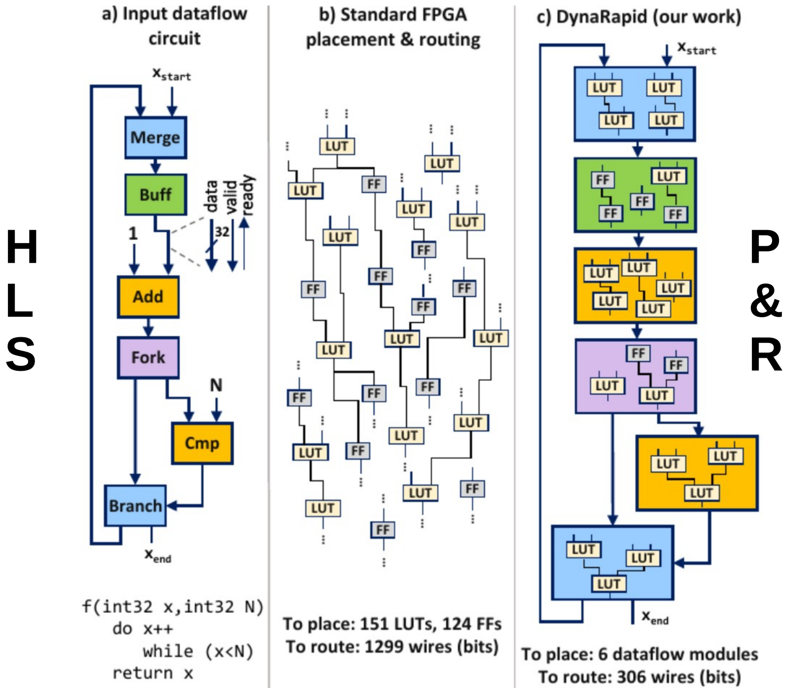
- Conifer : Decision Forests → low-latency FPGA inference → trigger selections
- hls4ml (CERN KT): ML algorithms in FPGA via HLS
- Madgraph framework for physics phenomenology on FPGA
- BondMachine : heterogeneous computing framework

HLS - DynaRapid

- **AMD RapidWright**
  - APIs to manipulate, gateway to Vivado
- <u>**DynaRapid**</u> **: HLS → P&R in < 1 min**
  - Run-time >> QoR
  - Abstraction from RTL up to pre-implemented library of dataflow blocks
  - “Greedy” placer – find legal solution
  - Preliminary academic investigation
- **Challenges**: preserving routability, timing
  - Intra-module beyond boundaries (place)
  - Inter-module: not reachable from outside



A. Guerrieri (EPFL, Hes-so)



```
f(int32 x,int32 N)
  do x++
    while (x<N)
  return x
```

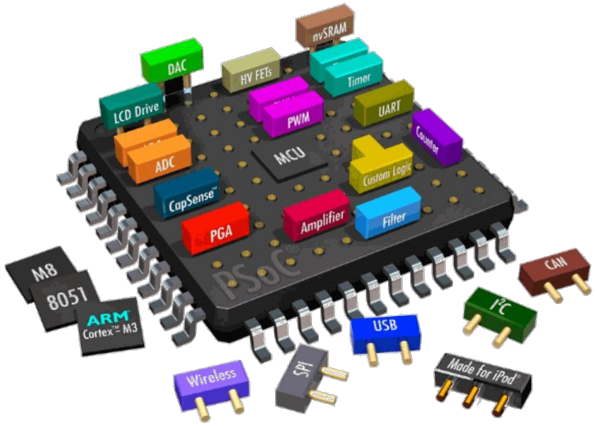
To place: 151 LUTs, 124 FFs
To route: 1299 wires (bits)

To place: 6 dataflow modules
To route: 306 wires (bits)

3. Solutions to common design challenges

“Someone must have run into this…!”

- **Timing closure, Floorplanning**
- **Tuning of IP cores**
- **Exploring other COTS devices**
- **HW/SW co-design**



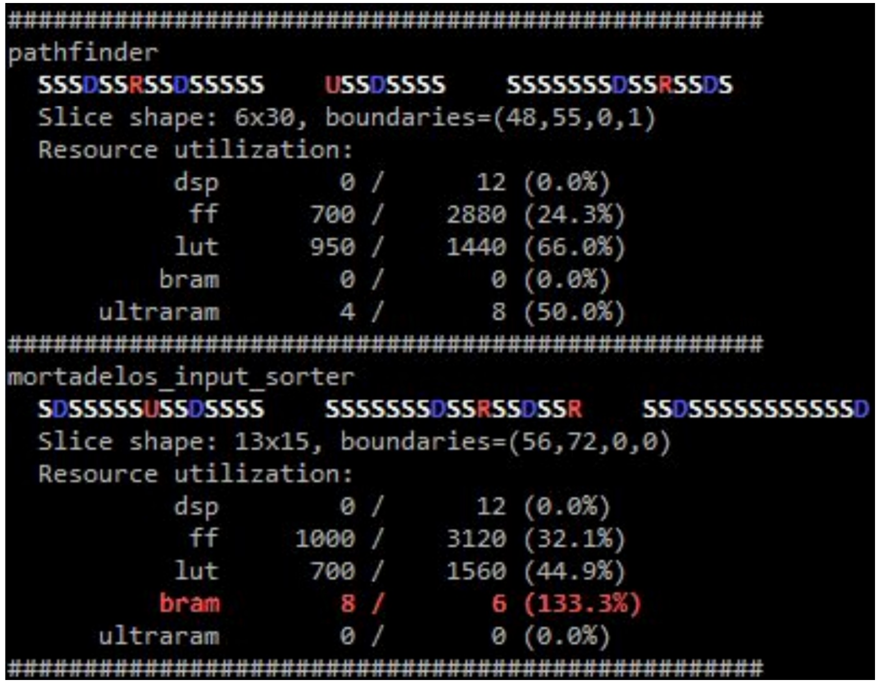
Examples

- Fine-grained placement constraints
- Phase determinism in transceivers
- RFSoC in astroparticle physics experiments
- LoCod : FPGA-accelerated CPU for heterogeneous computing platforms

Solutions – Placement Helper Framework

- Floorplaning can fix timing, but tedious with 100s of pblocks
- Tool for modular, structured designs (e.g. algorithm replicas) that automatically:
  - Reads **resource map** from AMD device fabric **file** (e.g. column types: Slice, bRam, Dsp)
  - **Replicates** placement, **resizes** pblocks boundaries
  - Monitors resource **utilization**
  - Outputs complex **XDC** pblocks constraints file
- User hierarchically maps design to ‘**boxes’** via Python
  - Column-based absolute coordinate system (vs X/Y)
  - Fine-grained, modular allocation (vs clock-regions)
  - Iterative process, maintainable as design evolves
- *Alvaro’s FDF and TWEPP talks*

A. Navarro (CIEMAT)



```
##################################################
pathfinder
  SSSDSSRSSDSSSSS    USSDSSSS    SSSSSSSDSSRSSDS
  Slice shape: 6x30, boundaries=(48,55,0,1)
  Resource utilization:
           dsp        0 /       12 (0.0%)
            ff      700 /     2880 (24.3%)
           lut      950 /     1440 (66.0%)
          bram        0 /        0 (0.0%)
      ultraram        4 /        8 (50.0%)
##################################################
mortadelos_input_sorter
  SDSSSSSUSSDSSSS    SSSSSSSDSSRSSDSSR    SSDSSSSSSSSSSSD
  Slice shape: 13x15, boundaries=(56,72,0,0)
  Resource utilization:
           dsp        0 /       12 (0.0%)
            ff     1000 /     3120 (32.1%)
           lut      700 /     1560 (44.9%)
          bram        8 /        6 (133.3%)
      ultraram        0 /        0 (0.0%)
##################################################
```

4. Development tools

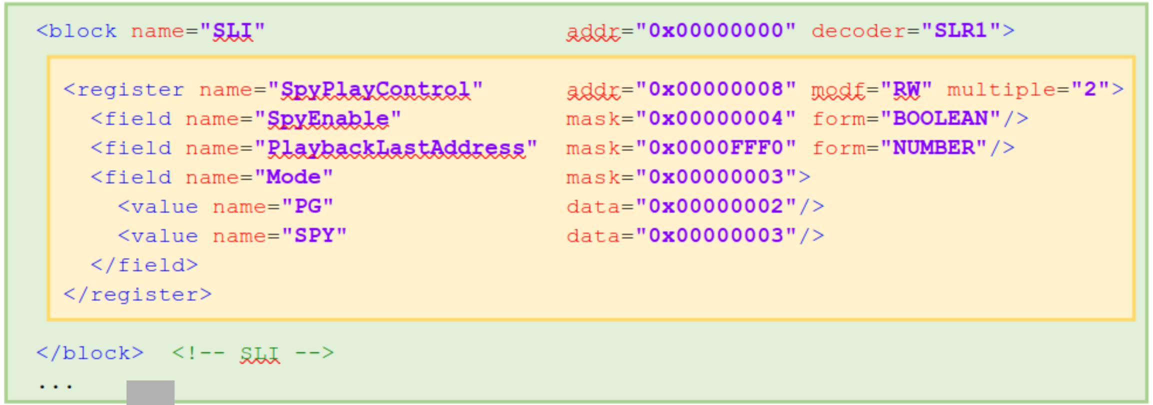
“Software should be doing this…!”

- **Reg-map (control & monitoring)**
  - Automatic code generation (HDL+SW)
  - Keep HW/SW in-sync
- **Project frameworks, revision control**
  - Coordinate teamwork, Reproducibility

Examples

- Hog : project management + CI
- ipbus : reg-map + transport layer + SW
- hdlmake : synth + sim project management
- YML2HDL : YAML → both VHDL and [System]Verilog libraries
- Cheby : YAML→HDL + C header + docs
- hwcompiler : XML→VHDL reg-map + C++ access functions
- Wupper-CodeGen : reg-map w/ automatic address
- airhdl, Linty, Sigasi, vhdl-style-guide

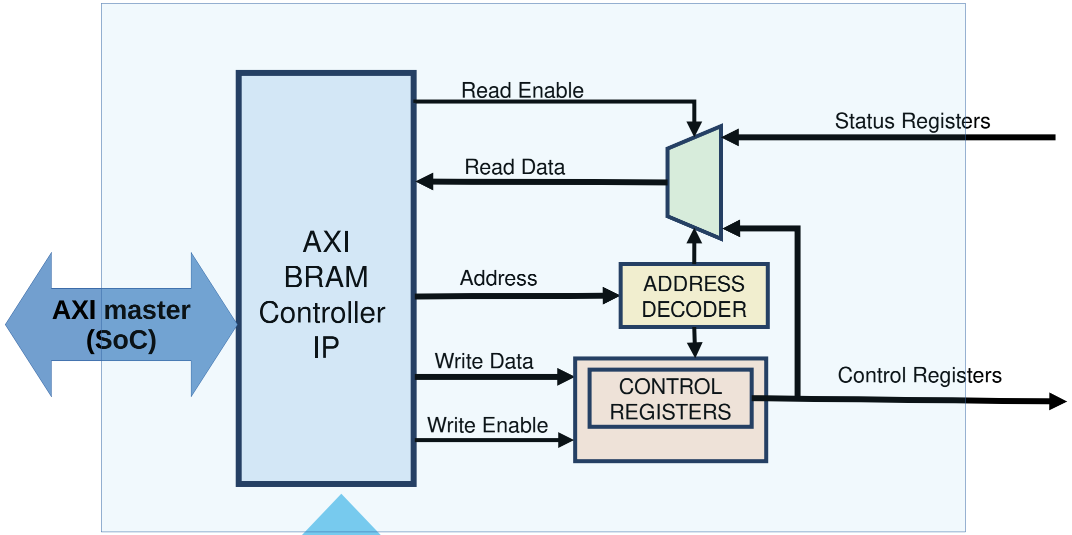
HardwareCompiler automatic code generation

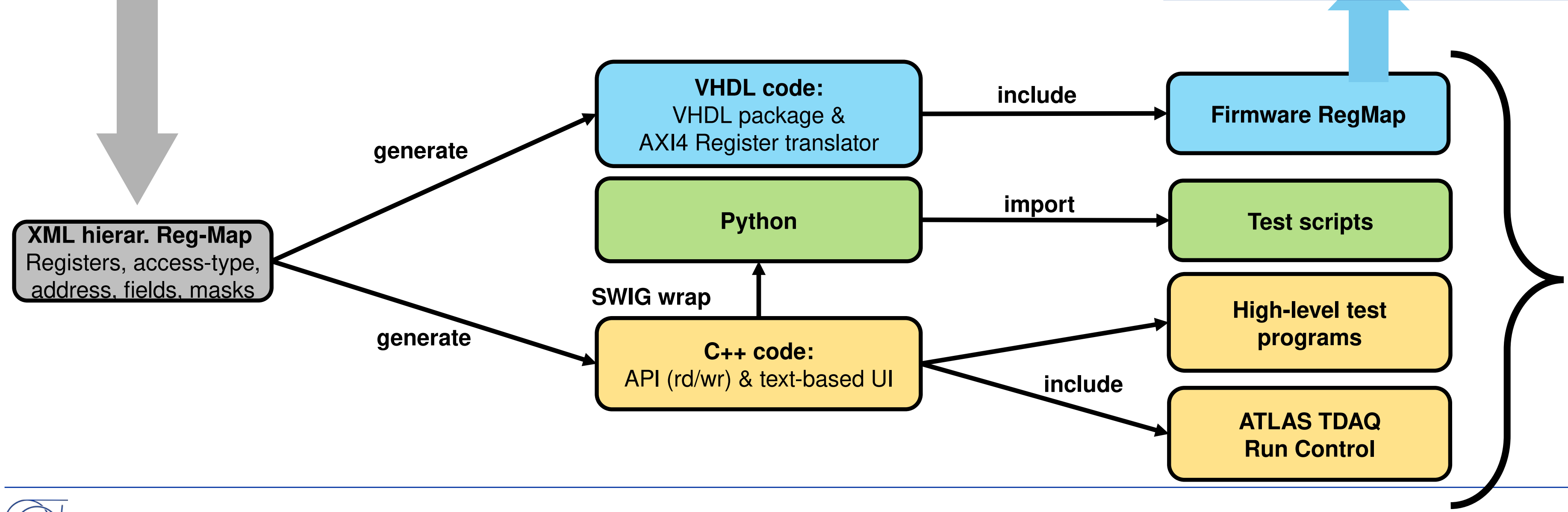


```
<block name="SLI"                        addr="0x00000000" decoder="SLR1">

  <register name="SpyPlayControl"        addr="0x00000008" modf="RW" multiple="2">
    <field name="SpyEnable"              mask="0x00000004" form="BOOLEAN"/>
    <field name="PlaybackLastAddress"    mask="0x0000FFF0" form="NUMBER"/>
    <field name="Mode"                   mask="0x00000003">
      <value name="PG"                   data="0x00000002"/>
      <value name="SPY"                  data="0x00000003"/>
    </field>
  </register>

</block>  <!-- SLI -->
...
```





IN SYNC
Reflecting XML hierarchy

A. Kulinska, R. Spiwoks (ATLAS L1CT)

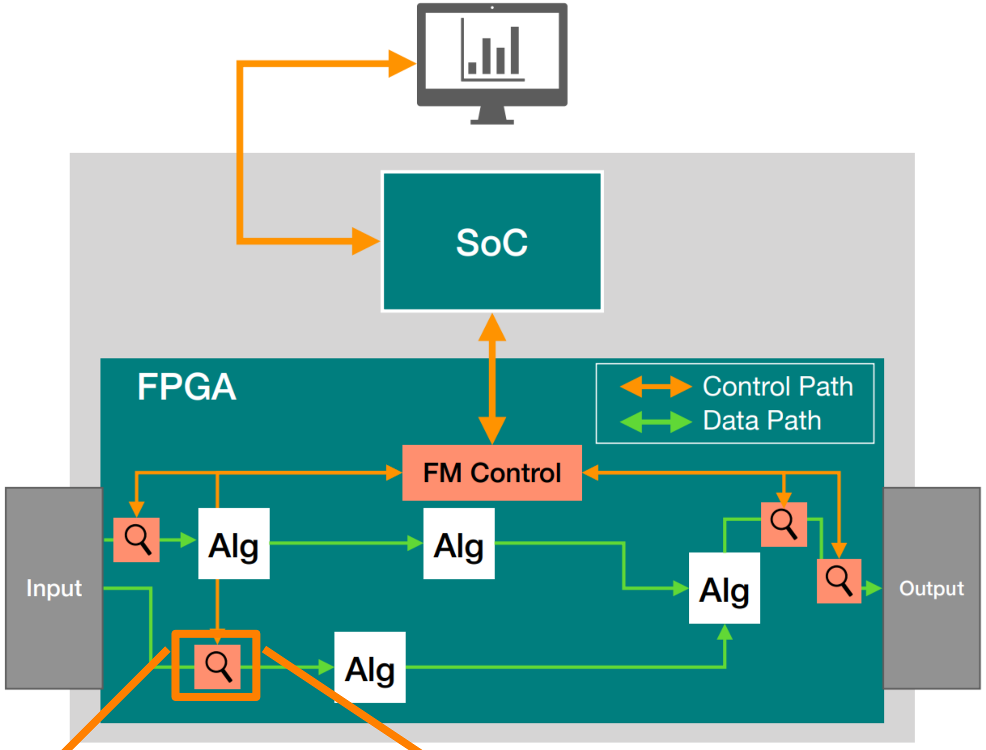
5. Verification & Simulation

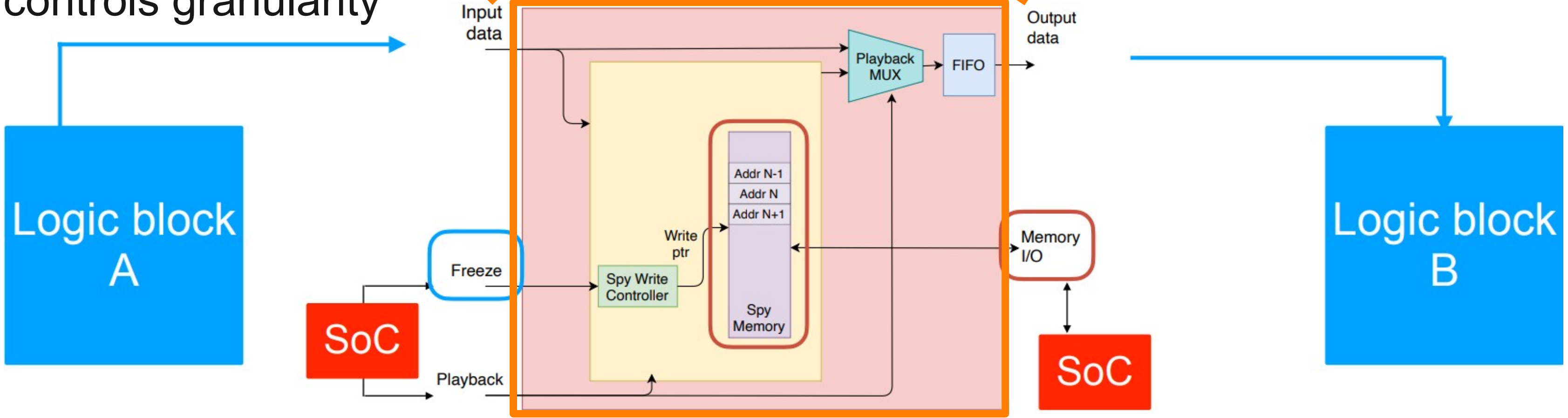
- **Verification frameworks**
  - Coverage, Error-injection, TLM, Scoreboards, Randomization, Utils
- **Formal verification**
- **Open-source simulators**

- UVVM vs OS-VVM
  - Free and open-source
- cocotb (cosimulation library for writing testbenches in Python)
- SymbiYosis (OS), YosysHQ (commercial but special license for OS projects)
- GHDL

Validation – SpyBuffers

- **Online “Plug&Play” debugging module**
  - **Spy** : Freeze & Readout (0-latency I/O)
  - **Inject** : Load & Playback (optional loop-mode and FIFO/CDC)
- **Fast Monitoring (FM)** block controls multiple SBs
- RTL technology and vendor **independent**
- **Parametrizable** memory & SB controls granularity
  - Resource utilization
- Control SW (ipbus), TB (cocotb)
- gitlab.cern.ch/spybuffer





I. Longarini (UCI)

What's next?

Seminars

"Smash HDL Bugs on the Fly"

- David Racodon (CTO, Linty Services)
- 2024.09.24 via Zoom (recorded, Indico)
- Stay tuned for next ones

FDF-2025

May 28-30th @ CERN

- Save the date!

Resources

- Newsletter (cern.ch/fdf-news)
- Discourse Forum (cern.ch/fdf)
- FDF-2024
  - Indico
  - CDS + YouTube
- fpga.developers.forum@cern.ch

Thank You

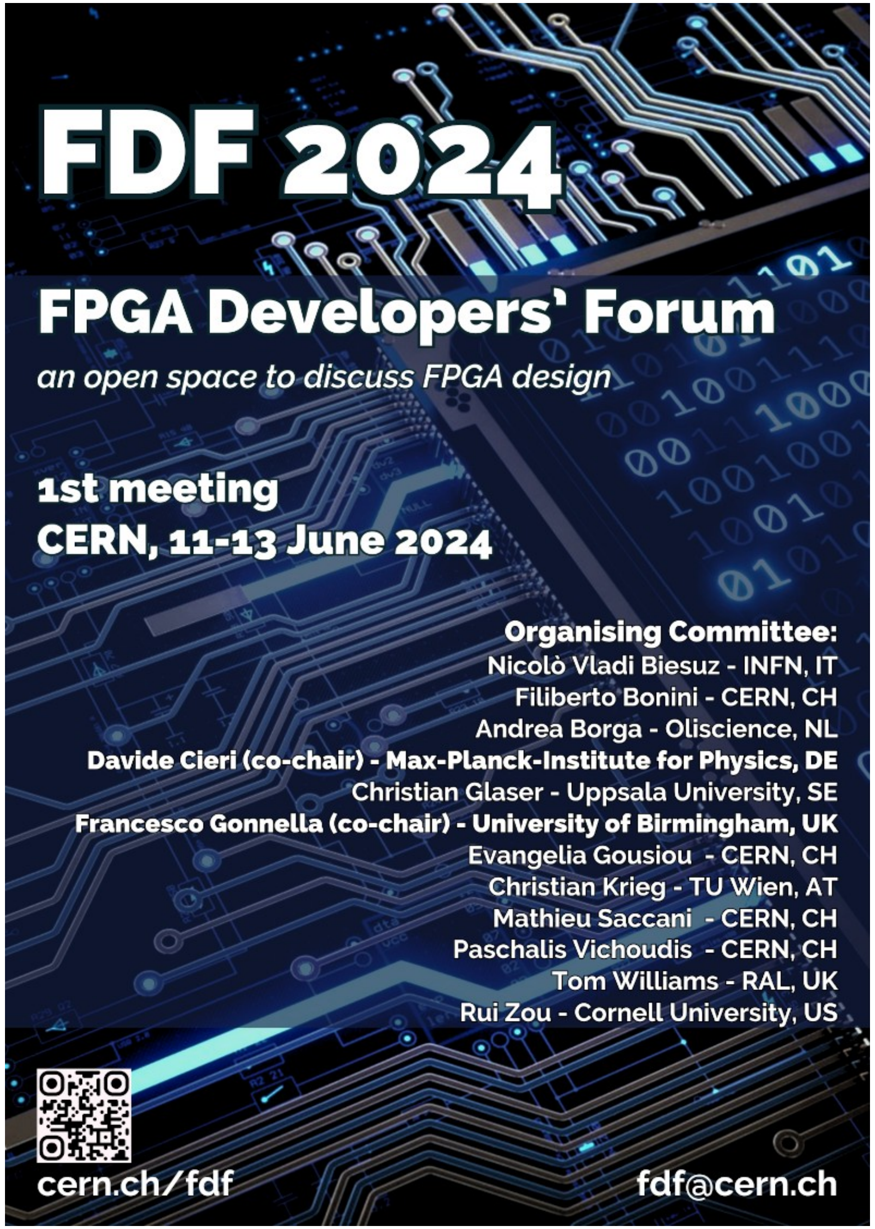
home.cern

Organization

Call for abstracts

- *Did you address an FPGA problem that others may face too?*
- **Tracks**
    1. Shareable HDL Cores
    2. HLS & Algorithms
    3. Solutions to common design challenges
    4. Development tools
    5. Verification & Simulation

- Received: 69 – Selected: 30 – Session time: 18h



FDF 2024

FPGA Developers' Forum

an open space to discuss FPGA design

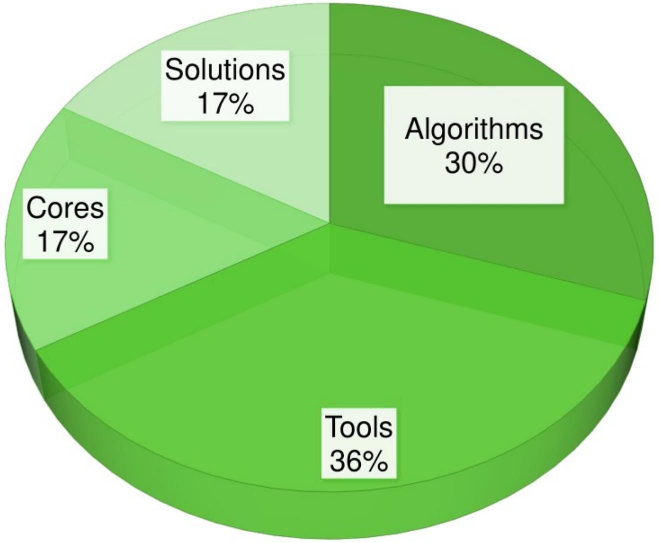
1st meeting
CERN, 11-13 June 2024

Organising Committee:
Nicolò Vladi Biesuz - INFN, IT
Filiberto Bonini - CERN, CH
Andrea Borga - Oliscience, NL
Davide Cieri (co-chair) - Max-Planck-Institute for Physics, DE
Christian Glaser - Uppsala University, SE
Francesco Gonnella (co-chair) - University of Birmingham, UK
Evangelia Gousiou - CERN, CH
Christian Krieg - TU Wien, AT
Mathieu Saccani - CERN, CH
Paschalis Vichoudis - CERN, CH
Tom Williams - RAL, UK
Rui Zou - Cornell University, US

cern.ch/fdf

fdf@cern.ch

- **69 submitted abstracts, 30 accepted**
  - 9 Algorithms
  - 11 Tools for development]
  - 5 HDL-Cores
  - 5 Solutions to everyday design problems



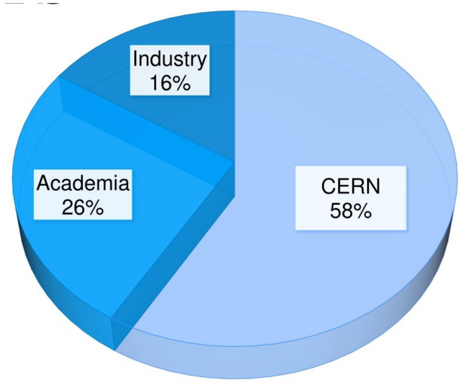
My Take

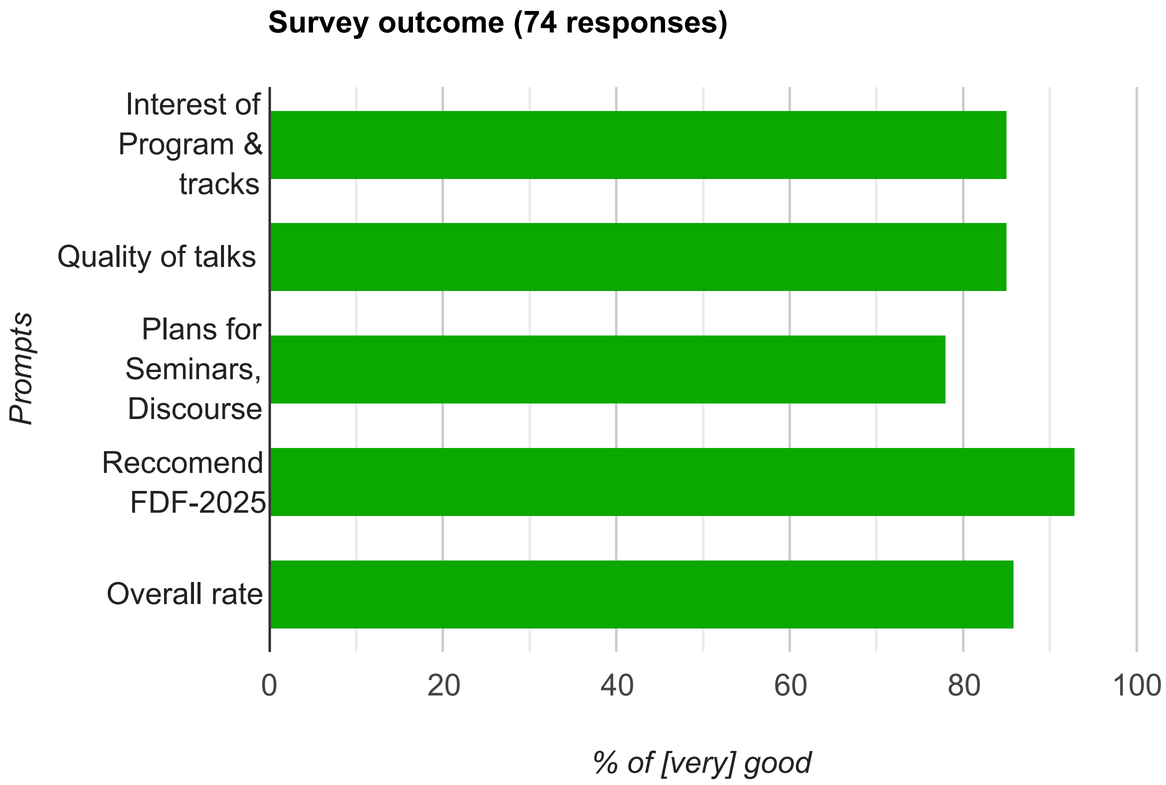
- **Many common issues, many solutions**
  - even within CERN alone
  - CERN OSPO helps going open-source (talk by J. Serrano)
- **Efforts to be consolidated**
  - Yet hard to re-adapt once [developed and] used to one’s own tool-kits
- **FDF** is chance to Discover and Collaborate
  - Teach and Learn from each other
  - Networking for future collaboration

FDF-2024 Audience

Registered participants: 197

- Backgrounds
  - @CERN : 137 – Others : 60
  - Academia : 160 – Industry: 37
- Attendance type
  - In-person : 131 (+ intruders!)
  - Webcast (peak): 114





Newsletter
Q&A Chat
CERN - 11-13 June 2024
FDF
PARTICLE PHYSICS UNIVERSITY OF BIRMINGHAM
CAEN
Tools for Discovery
CAEN
Tools for Discovery
CAEN
OPEN INNOVATION!

A first example
The BondMachine Project
CAEN
CAEN
OPEN INNOVATION!

- June 11-13, 2024
  - 1 full + 2 half-days
  - + CERN site visits (45p), Social dinner (79p)
- Kjell Johnsen Auditorium, CERN
  - Presentations
  - Room for Q&A
  - Room for coffee breaks, talk, and networking
- Embracing CERN values
  - Community, Knowledge share: share, collaborate, open-source
  - Work-everywhere, Vendor-agnistic
  - Excellence, Efficiency: avoid redoing, improving

home.cern