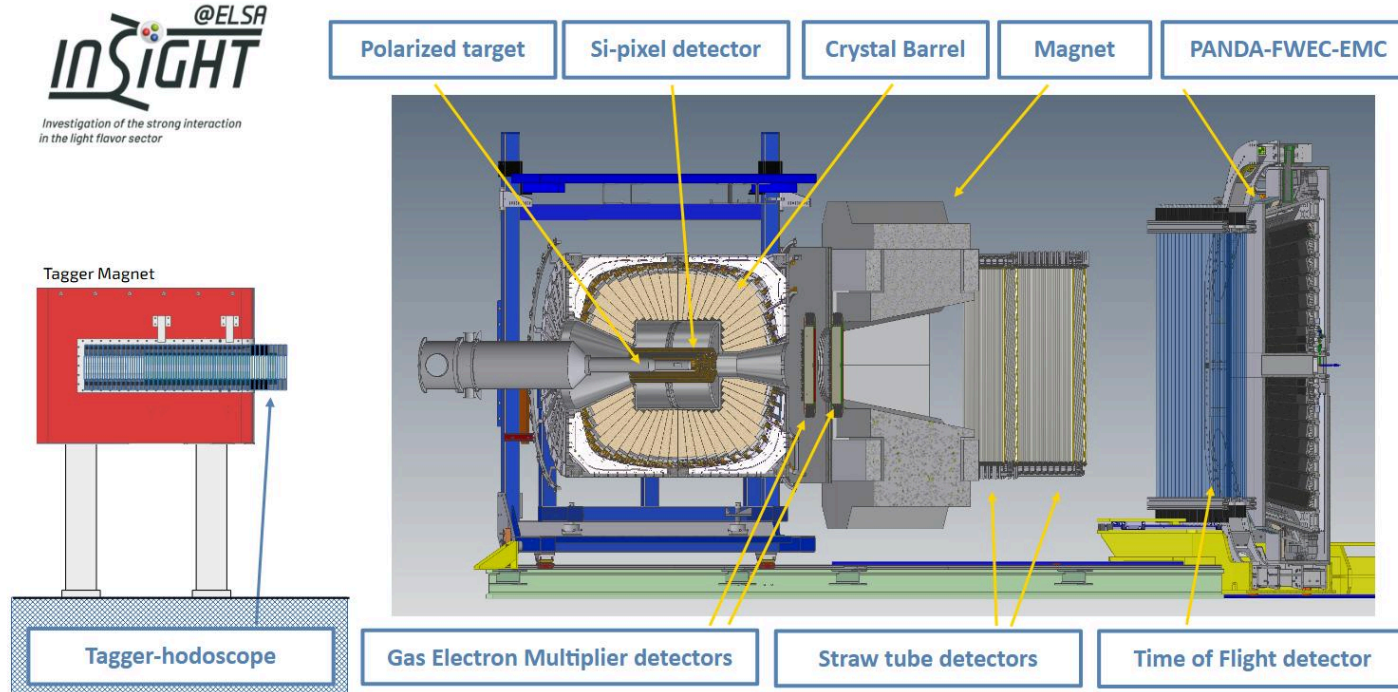


A Modular Firmware Framework for the INSIGHT Experiment with HDLMake

Martin Bieker, Benedikt Otto

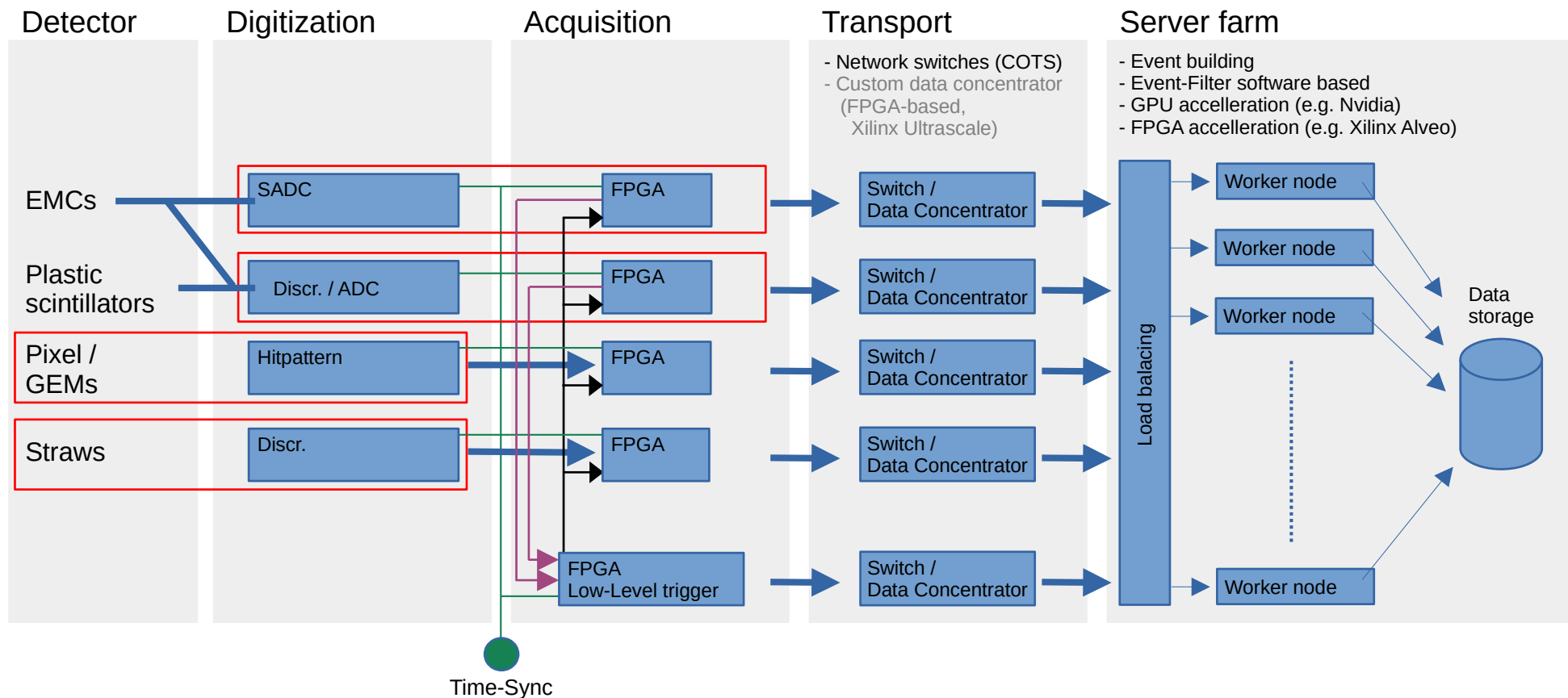
Electronics & Applications Seminar, December 1st, 2025

Insight Experiment @ ELSA



- Successor of CB-ELSA
- Photoproduction on polarized target
- High resolution photon measurements
- Precise charged particle detection

INSIGHT DAQ Pipeline



INSIGHT DAQ Pipeline

Trigger-less, streaming DAQ

- Free-running subdetector
- Send every hit to back-end
- Software trigger/ event filter on server farm
- Precise timing

DAQ Philosophy

Less custom – more COTS

- Custom electronics only on front-end
- FPGA-based acquisition
- Move to industry standard hardware (Switches) and protocols (IP/UDP) ASAP

FPGAs

Interface between custom and commercial worlds

INSIGHT DAQ Pipeline

Trigger-less, streaming DAQ

- Free-running subdetector
- Send every hit to back-end
- Software trigger/ event filter on server farm
- Precise timing

DAQ Philosophy

Less custom – more COTS

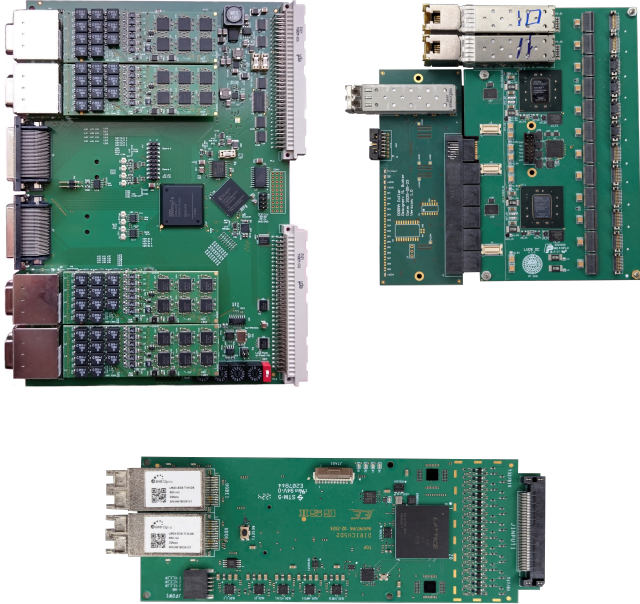
- Custom electronics only on front-end
- FPGA-based acquisition
- Move to industry standard hardware (Switches) and protocols (IP/UDP) ASAP

FPGAs

Interface between custom and commercial worlds

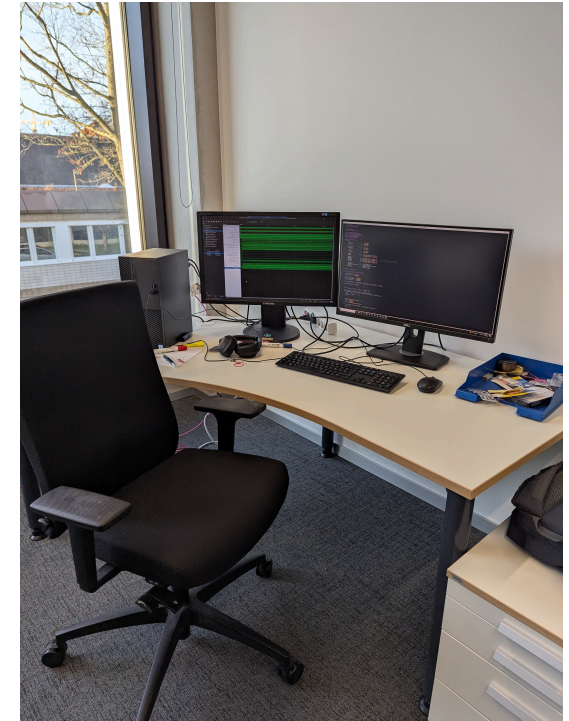
In this talk:
Firmware
=
Gateway for
FPGAs

DAQ Integration – Challenges



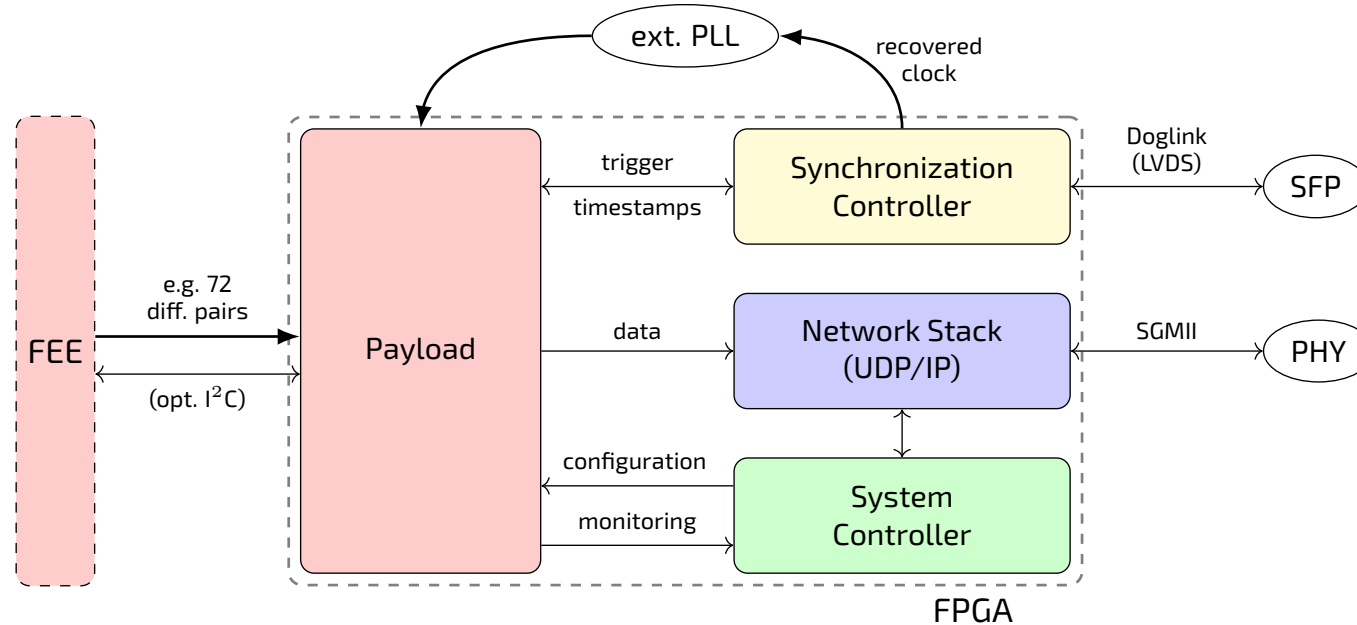
Diverse hardware

Ambitious schedule



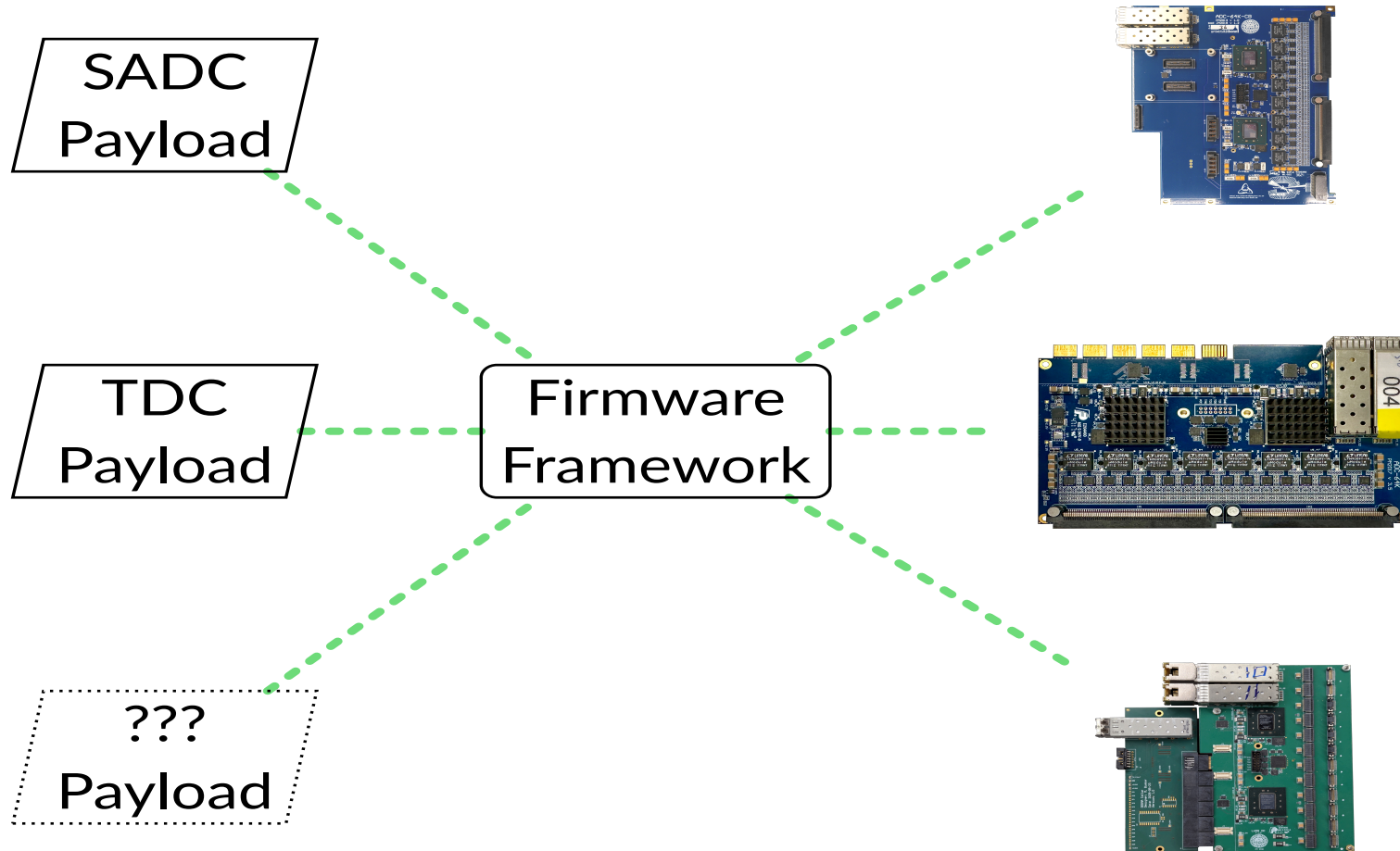
Limited person power

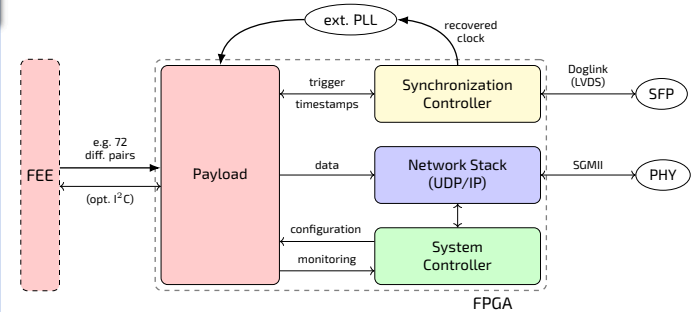
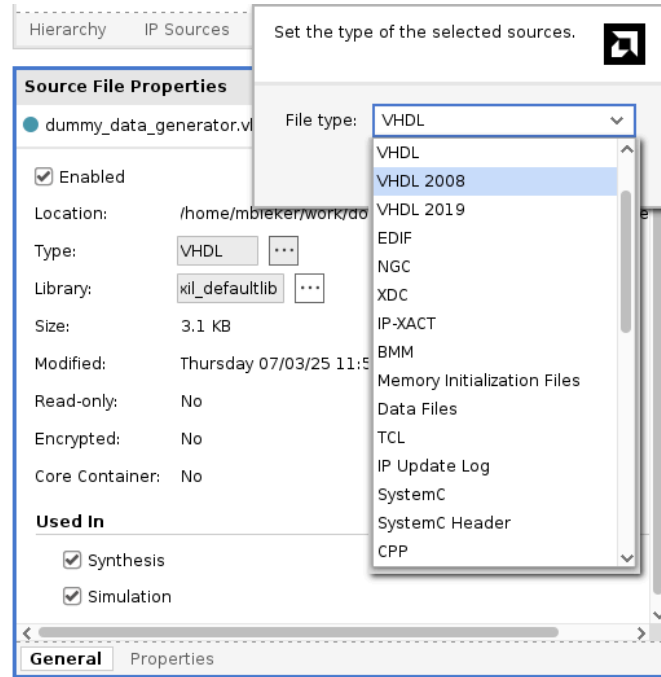
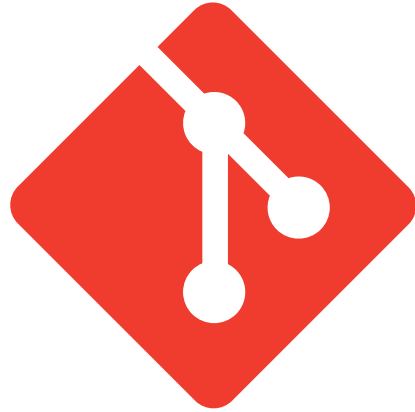
Firmware Framework



- Modular structure
- Re-use of common functional blocks
- Payload for subdetector specific logic
- Based on SADC firmware (CB & FWEC)

Multi-Function & multi-hardware



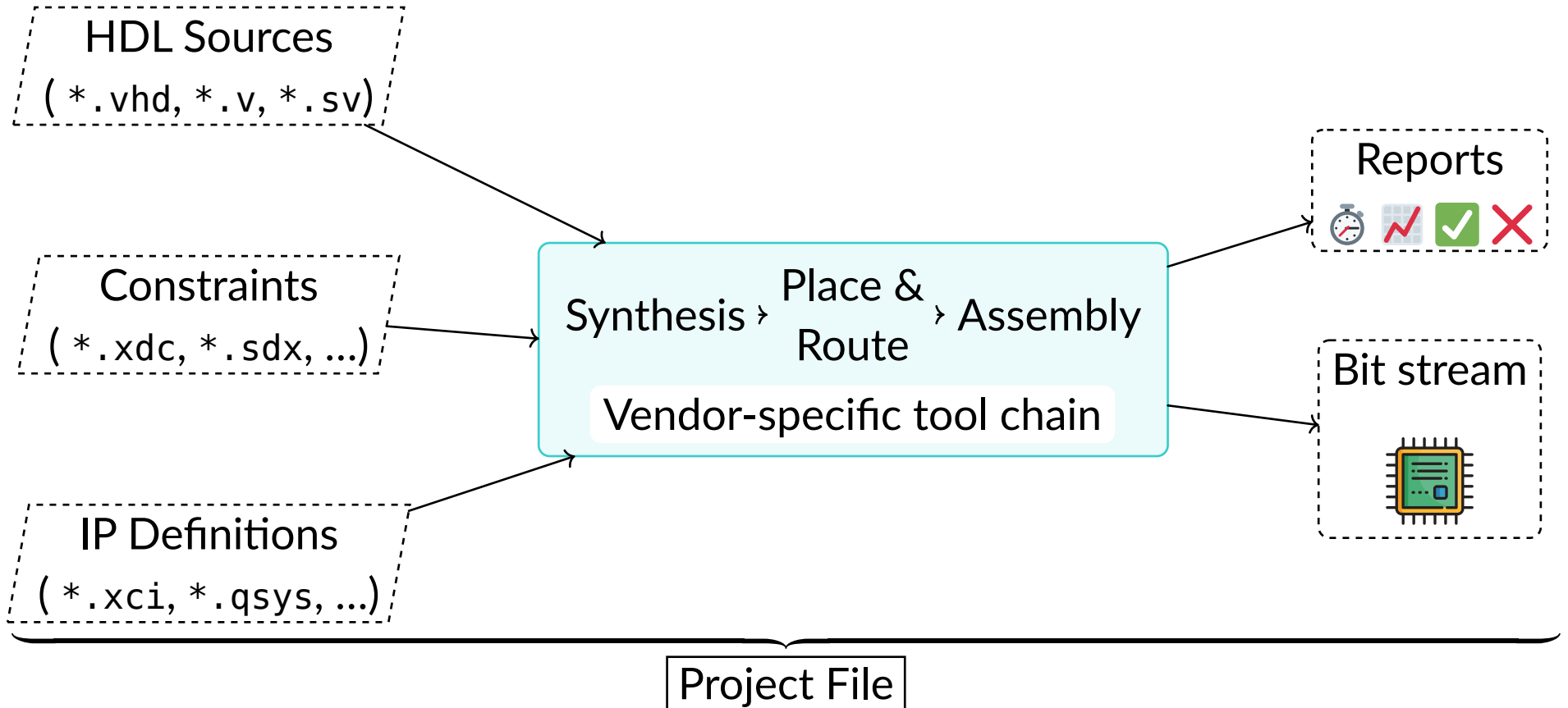


Version control

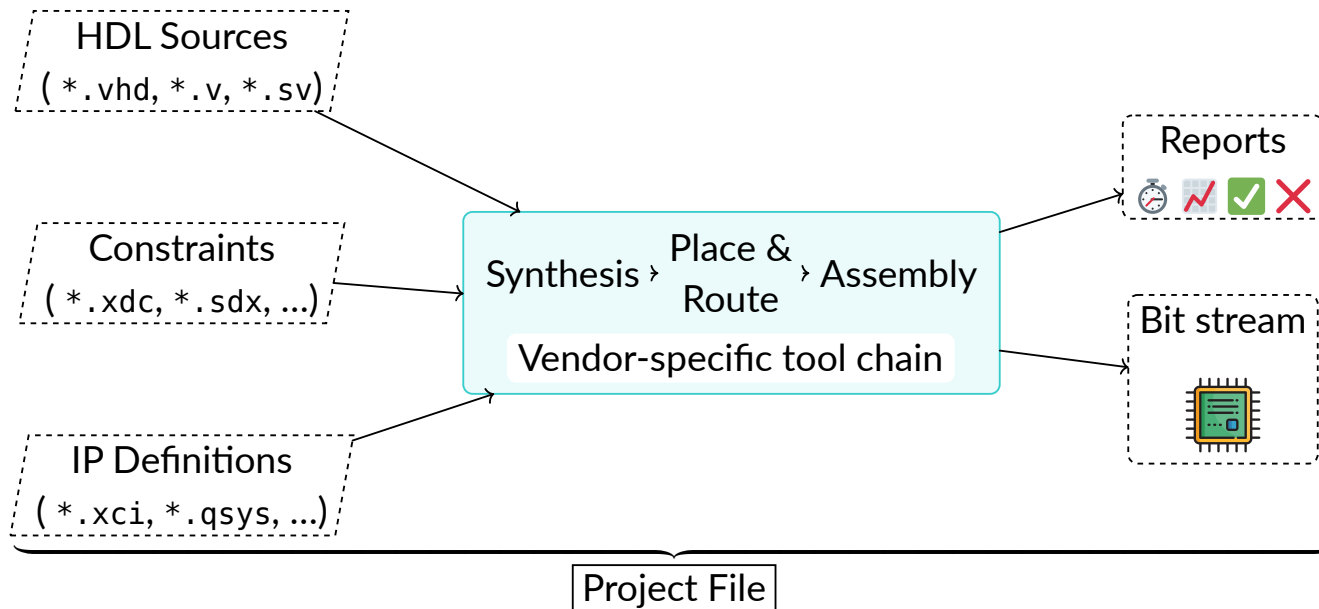
Manual setup

Support for modular projects

Firmware compilation flow



Version Control



Differentiate:

artefacts

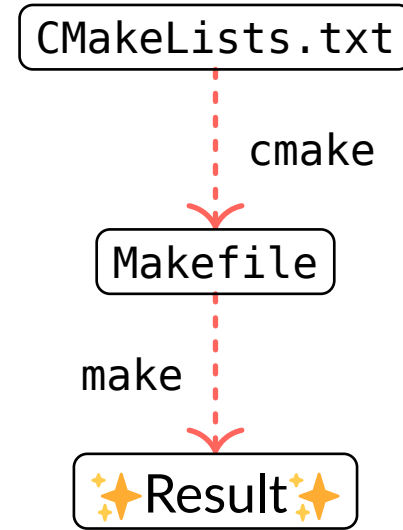
- Created by the designer
- Read-only during compilation
- Version Control: ✓

build products

- Generated by the tool
- Created modified during run
- Version Control: ✗

A typical software flow

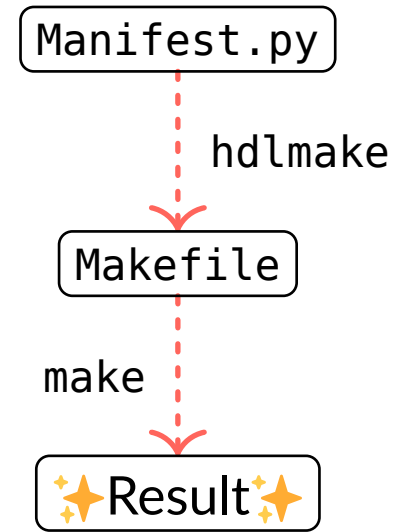
e.g. Cmake



HDLMake¹ – “CMake for Hardware”

An open-source tool for

- Makefile generation for synthesis and simulation
- HDL parser and dependency solver
- Modular project structure
- Version control support



¹ <https://ohwr.gitlab.io/project/hdl-make/>

Manifest.py

contains project definition with

- Synthesis tools
- Configuration
- Source files
- Recursive import of sub-modules

in Python

```
target = "xilinx"
action = "synthesis"

syn_device = "xc7k160t"
syn_grade = "-1"
syn_package = "fbg484"

syn_top = f"TOP_{board}"
syn_project = "demo_project"
syn_tool = "vivado"

modules = {
    "local" : [ "src", "ip" ]
}
```

Manifest.py

contains project definition with

- Synthesis tools
- Configuration
- Source files
- Recursive import of sub-modules

in Python

```
files = [  
    f'TOP_{board}.vhd'  
]  
modules = {  
    'local': [  
        'axis',  
        'boards',  
        'clock',  
        'eventsender',  
        'extern',  
        'network',  
        'payload',  
        'reset',  
        'synchronization',  
        ...,  
    ],  
}
```

Manifest.py

contains project definition with

- Synthesis tools
- Configuration
- Source files
- Recursive import of sub-modules

in Python

```
# This defines this file as  
# a valid endpoint for the payload  
PAYLOAD = True  
  
import os  
  
# include all vhd files  
files = [  
    file for file in os.listdir('.')  
        if file.endswith('.vhd')  
]
```

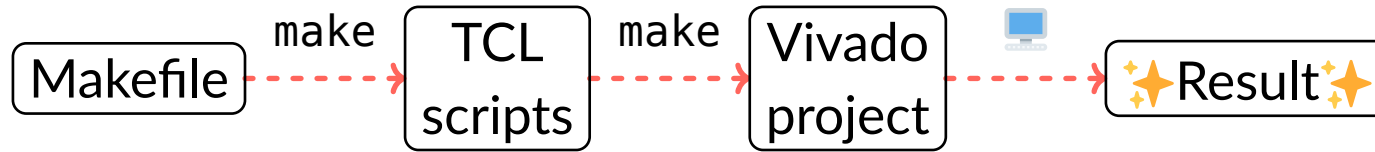
User interface

```
→ firmware git:(multichannel-wu) x python3 prepare_build.py --help
usage: prepare_build.py [-h] [--target {project,non-project}]
                        [--board {CBSADC11,CBSADC12,CBSADC10,LVDS_DC,PANDA}]
                        [--payload {CB,tdl_tdc}] [--store | --no-store]
                        [--persist] [--message MESSAGE]

options:
  -h, --help            show this help message and exit
  --target {project,non-project}
  --board {CBSADC11,CBSADC12,CBSADC10,LVDS_DC,PANDA} --payload {CB,tdl_tdc}
  --store, --no-store
  --persist
  --message MESSAGE
```

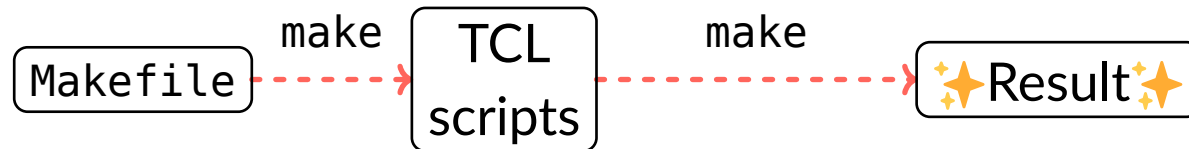
Project-less workflow in Vivado

Plain **hdlmake** generates Vivado project:



HDLMake extension

developed by F. Feldbauer (RUB)¹



¹  [git: https://gitlab.ep1.rub.de/florian/hdlmake_nonproj](https://gitlab.ep1.rub.de/florian/hdlmake_nonproj)

Workflow – Getting started ...

```
git clone --recursive <my-project-url>
cd <my_project>

#Setup HDLMake
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt

#Create Makefile
python3 prepare_build.py --board LVDS_DC --payload template --target non-project --message "Test
project"

#let's go
make synth_ips
make
```

... some time later

```
#####  
# Timing report #  
#####  
  
      WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS  
Failing Endpoints  THS Total Endpoints      WPWS(ns)      TPWS(ns)  TPWS  Failing Endpoints  TPWS Total  
Endpoints  
      0.870      0.000      0      19413      0.041      0.000  
0      19413      0.000      0.000      0      7844  
All user specified timing constraints are met.
```

... some time later

```
#####  
# Error report #  
#####  
  
Synthesis  
524 Infos, 8877 Warnings, 50 Critical Warnings and 0 Errors encountered.  
  
Place & Route  
34 Infos, 0 Warnings, 1 Critical Warnings and 0 Errors encountered.  
30 Infos, 1 Warnings, 3 Critical Warnings and 0 Errors encountered.  
13 Infos, 0 Warnings, 3 Critical Warnings and 0 Errors encountered.  
17 Infos, 0 Warnings, 1 Critical Warnings and 0 Errors encountered.  
29 Infos, 0 Warnings, 4 Critical Warnings and 0 Errors encountered.  
  
touch bitstream
```

Missing something?

Vivado GUI provides useful features

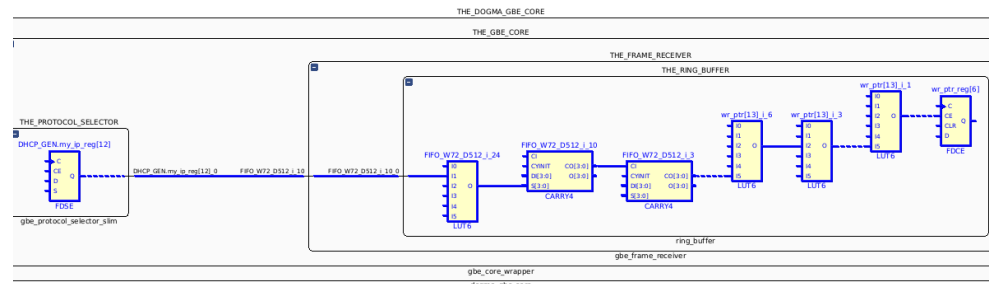
- Reporting (timing, resources, etc.)
- Interactive schematics
- Debugging tools

Solution:

- Automatically creating checkpoints after each compilation step
- Separate for each run
- **RunDB** with important information in sqlite

The screenshot shows the Vivado Project Summary window for 'Path 41 - dogma_hub1_dcm_timing_summary_routed'. The 'Summary' tab is active, displaying various timing metrics. Below the summary, the 'Source Clock Path' and 'Data Path' are detailed in a table format.

Delay Type	Incr (ns)	Path ...	Location	Cell Pin	Cell	Netlist Resources
Delay Type						
(clock clk125 rise edge)	(r) 0.000	0.000				
GTPE2_CHANNEL	(r) 0.000	0.000	Site: GTPE...ANNEL_X0Y0	TXOUTCLK	gtpe2_i (GTPE2_CHANNEL)	THE_DOGMA_GBE_CORE/pcs_pma
net (f0w1, routed)	1.209	1.209				
BUFG (Prop, bufg1_0)	(r) 0.081	1.290	Site: BUFGCTRL_X0Y17	0	bufg1_boutclk (BUFG)	THE_DOGMA_GBE_CORE/core_clo
net (f0w1, routed)	1.392	2.681				
MMCM2_ADV (Prop, mmcm2_adv_clkout0)	(r) 0.077	2.758	Site: MMCM2_ADV_X1Y0	CLKOUT0	mmcm2_adv_inst (MMCM2_ADV)	THE_DOGMA_GBE_CORE/core_clo
net (f0w1, routed)	1.421	4.179				
BUFG (Prop, bufg1_0)	(r) 0.081	4.260	Site: BUFGCTRL_X0Y4	0	bufg1_userclk2 (BUFG)	THE_DOGMA_GBE_CORE/core_clo
net (f0w7378, routed)	1.372	5.632				
FDRE			Site: SUCE_X49Y8	D	rc_data_local_reg[7] (FDRE)	THE_DOGMA_GBE_CORE/THE_GBE
Data Path						
Delay Type						
(clock clk125 rise edge)	(r) 0.379	6.011	Site: SUCE_X49Y8	Q	rc_data_local_reg[7] (FDRE)	THE_DOGMA_GBE_CORE/THE_GBE_CORE/THE_GBE_MAIN_CTRL/RC_D



IP cores – the good, the bad, the ugly

- Problem with excessive use of IP cores
 - Breaks composability
 - Complicates simulation / testing
 - Encrypted HDL prevent automatic dependency resolution

More useful abstractions

- HDL libraries and modules
(Parameterized Macros, OpenCores)
- Device primitives



Hard IP (transceivers etc.)

Soft IP (e.g. network cores)

“8bit fifo”, “10b-fifo”

Conclusion

- FPGAs play an important role in the INSIGHT DAQ pipeline
- Integration of many sub-detectors is necessary
- Modular firmware framework based on SADC firmware
- HDLMake has made our life easier
 - Proper version control
 - Easy setup for new designers
 - Code re-use

Mom can we have  python

No son we've got scripting at home



Comments Experiences

Discussion Suggestions

Ideas Questions

Backup

Supported Tools

Tool	Synthesis	Simulation
Xilinx ISE	Yes	n.a.
Xilinx PlanAhead	Yes	No
Xilinx Vivado	Yes	Yes
Intel Quartus	Yes	n.a.
Microsemi (Actel) Libero	Yes	n.a.
Lattice Semi. Diamond	Yes	n.a.
Gowin EDA	Yes	n.a.
Xilinx ISim	n.a.	Yes
Mentor Graphics Modelsim	n.a.	Yes
Aldec Active-HDL	n.a.	Yes
Project IceStorm	Yes	n.a.
Icarus Verilog	n.a.	Yes

Source: <https://ohwr.gitlab.io/project/hdl-make/>

RunDB

```
CREATE TABLE IF NOT EXISTS params (  
    datecode TEXT,  
    timestamp INTEGER,  
    git_hash TEXT,  
    git_dirty BOOLEAN,  
    board TEXT,  
    mac TEXT,  
    ip TEXT,  
    udp_port INTEGER,  
    payload TEXT,  
    target TEXT,  
    diff TEXT,  
    message TEXT,  
    vivado_version TEXT  
)
```