Status SPIDR4: A Timepix4 read out system

SPIDR4 development team:

Bas van der Heijden

Guido Visser

Charles Ietswaard

Vincent van Beveren

Henk Boterenbrood

Berend Munneke

Martin van Beuzekom

Martin Fransen

Kiavash Matin

Bram Bouwens

Jord Prangsma

Steven Tan

Walter van Bodegom

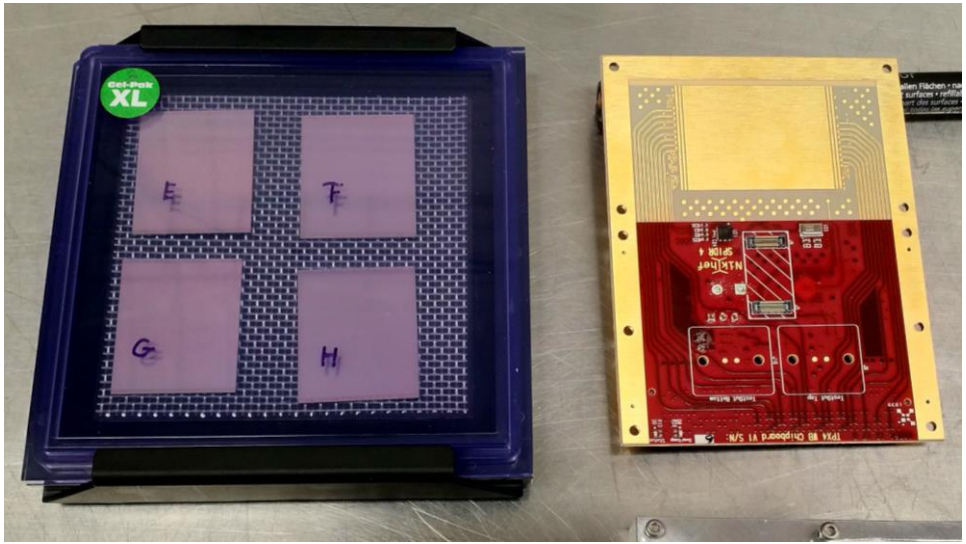
Nikhef

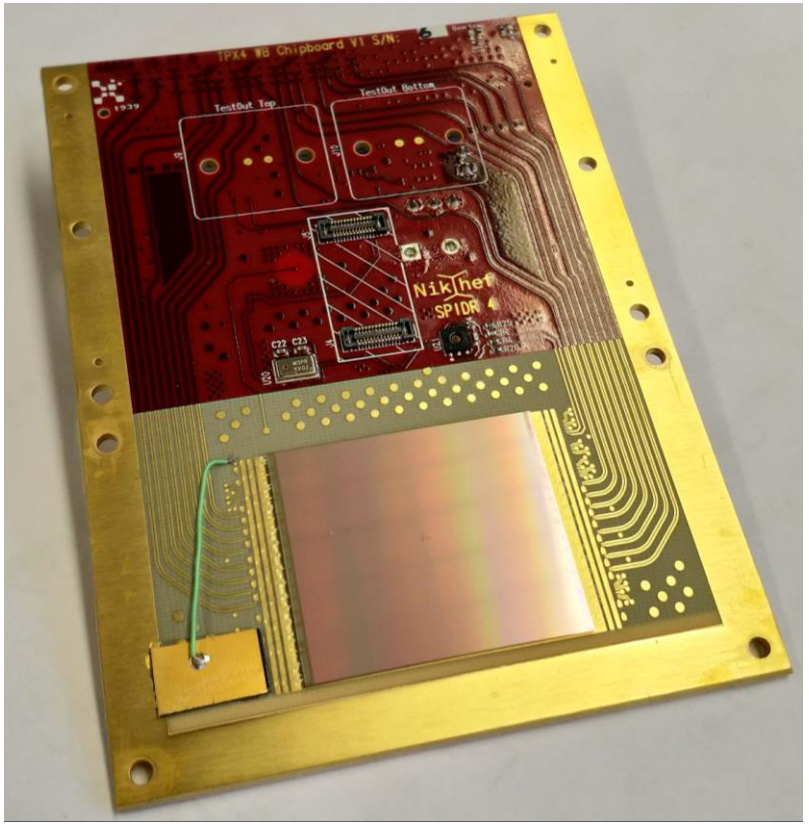
AMSTERDAM SCIENTIFIC INSTRUMENTS

Timepix4

			Timepix4 (summer 2019)
Technology			65nm – 10 metal
Pixel Size			55 x 55 µm
Pixel arrangement			**4-side buttable** **512 x 448**
Sensitive area			**6.94 cm²**
Readout Modes	Data driven (Tracking)	Mode	TOT and TOA
		Event Packet	64-bit
		Max rate	3.58×10^6 hits/mm²/s
		Max Pix rate	**10.8 KHz/pixel**
	Frame based (Imaging)	Mode	CRW: PC (8 or 16-bit)
		Frame	Full Frame (without pixel addr)
		Max count rate	~5 x 10^9 hits/mm²/s
TOT energy resolution			**< 1Kev**
Time resolution			**~200ps**
Readout bandwidth			≤163.84 Gbps (16x @10.24 Gbps)
Target global minimum threshold			<500 e^-

Timepix4





Timepix4 testing is ongoing work

Design ‘philosophy’ Timepix4 read out system

Timepix4 is a versatile chip! → versatile applications!

- particle tracking, X-ray, ion mass spectroscopy, etc..

TPX4 will often be in harsh or ‘strange’ environment

- Vacuum, low/(high?) temperature, B field, radiation, etc..

→ Keep the ‘bulk electronics’ away from the TPX4 chip

- To protect electronics
- To allow for ‘compact’ detectors

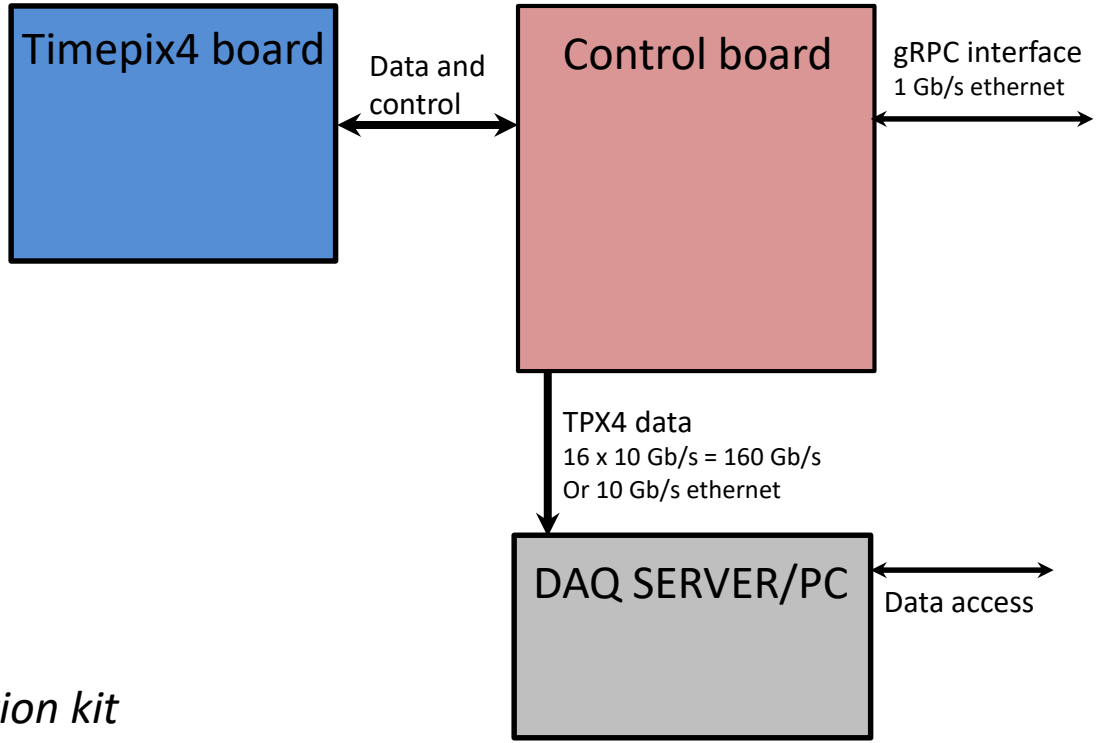
→ ‘eye on a wire’. TPX4 on a minimalist chip board
→ Modular design for flexibility and scalability
→ ‘Big boys’ DAQ system to test a ‘big boys’ chip to its limits
→ Scalable and STILL benefit from the TPX4 200 ps time bins

We want the whole shebang

SPIDR4 (Speedy PIxel Detector Readout (version 4))

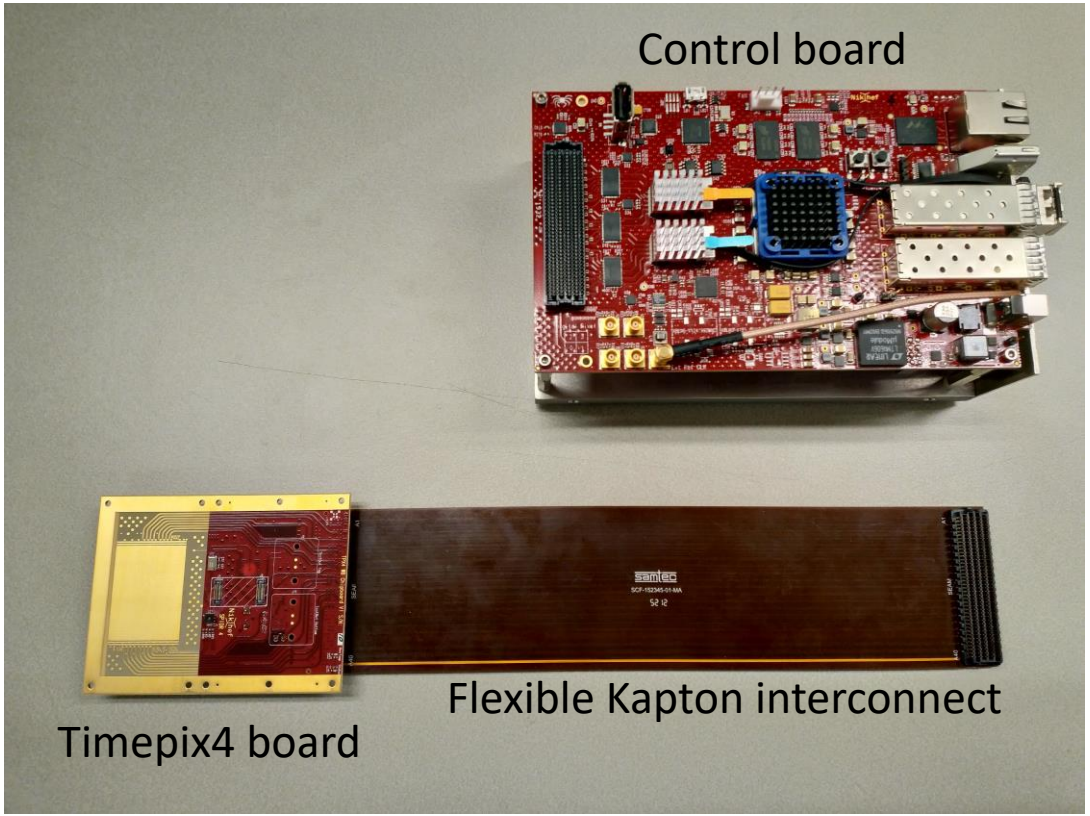
The components:

- Timepix4 board
  - Hosting TPX4
- Control board
  - TPX4 Control
  - Timing
  - Transfer TPX4 Data to DAQ
- DAQ system
  - 16 x 10 Gb/s =160 Gb/s data input
  - Data access tools



Consider it a Timepix4 evaluation kit

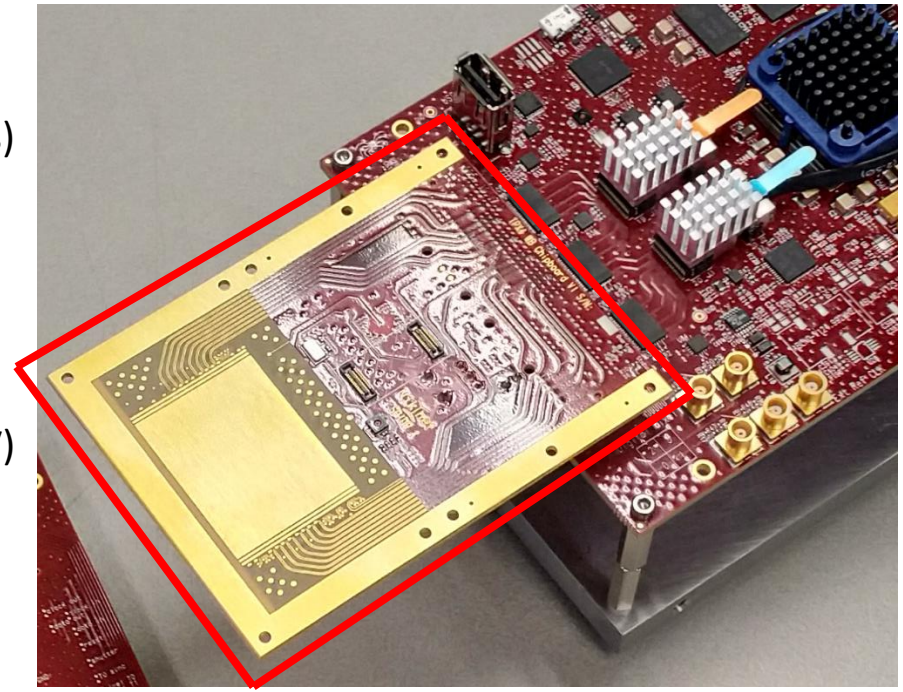
SPIDR4 (Speedy PIxel Detector Readout (version 4))



DAQ SERVER

TPX4 chip carrier board

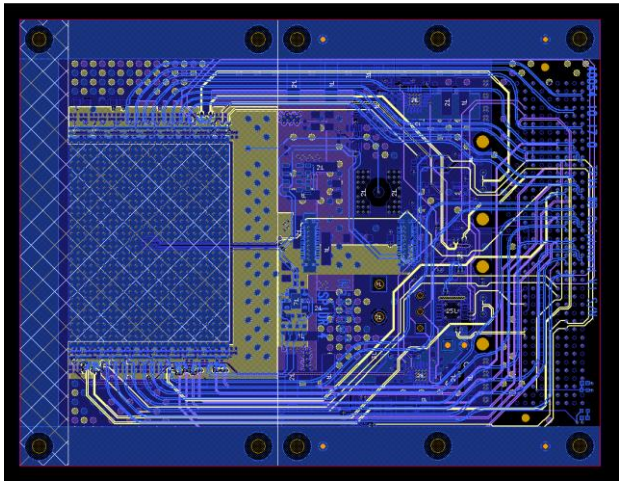
- Wire bond version
- Full bandwidth TPX4 readout (160 Gb/s)
- Power regulators
- Sensors
  - Power (I,V)
  - B field
  - Temperature
  - Humidity
- External bias voltage input (up to 400 V)

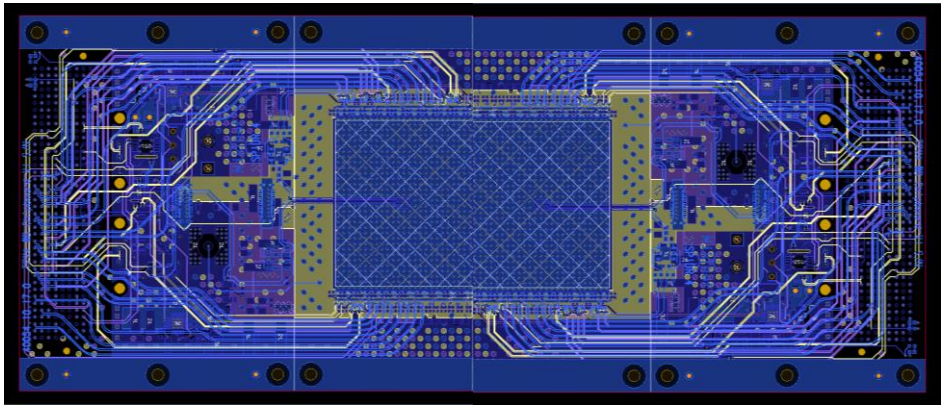


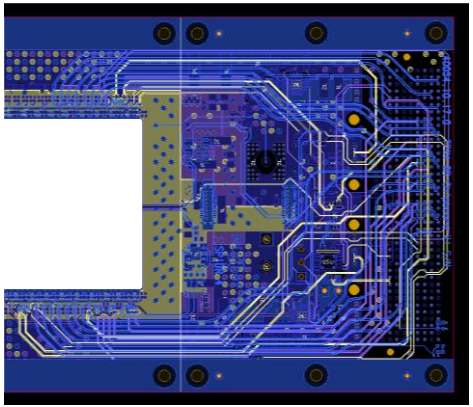
TPX4 chip carrier board

- One side for tiling
- No traces behind TPX4 chip

→Can remove PCB to reduce cross-section

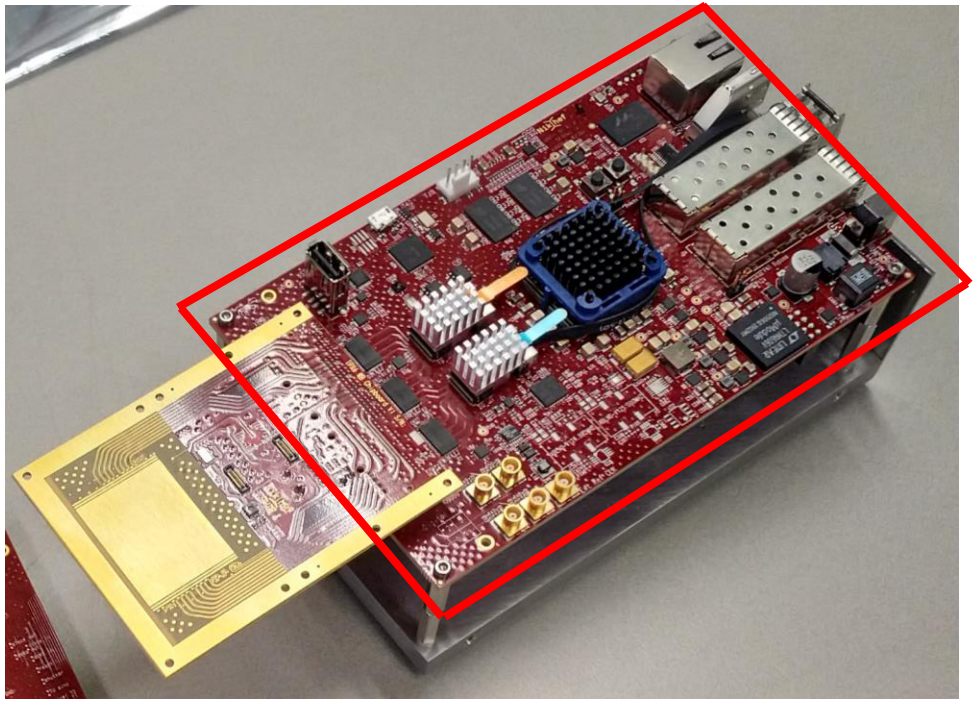






Control board

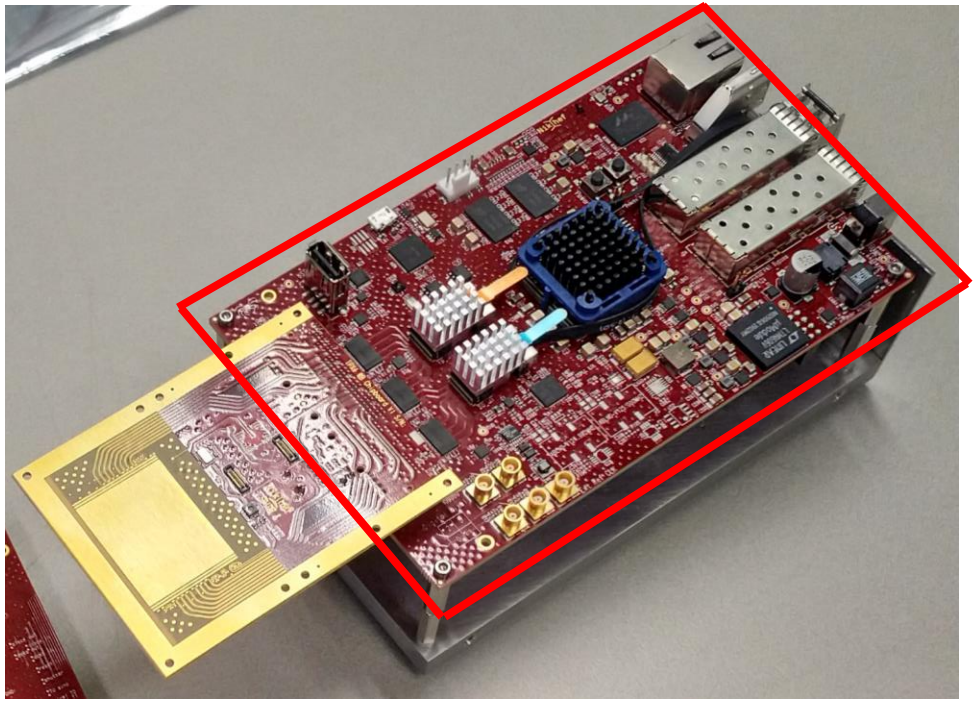
- PetaLinux on Zynq XC7Z030/35
- Control interface:
  - gRPC (Remote Procedure Call)
  - 1 Gb/s ethernet
- gRPC interface libraries for:
  - C++
  - Python (2.X, 3.X)
  - Java
  - …
- Configuration TPX4 (up to 12 chips)
  - DAC's
  - THL
  - Mode
  - Data out (SPI/fast links, link config)
- Configuration peripherals
  - Power
  - Timing



Control board

- Monitoring environment
  - TPX4 V, I, temp, B, RH, …
  - Control board
- Timing
  - On board
  - External
    - Conventional master timing
    - White Rabbit (<100ps)
- Up to 24 x 10 Gb/s electric to optical

(for up to 12 TPX4 chips)



SPIDR4 DAQ

DAQ options with SPIDR4:

- Full TPX4 output! 16 x 10 Gb/s, over PCIe
- 10 Gb/s ethernet
- SPI, through control interface

High performance DAQ system:

- Commercial Virtex Ultrascale FPGA board
- 32 x gen3 PCIe
- Server with 64 GB memory

(~ one sec TPX4 data at full speed)

SPIDR4 data

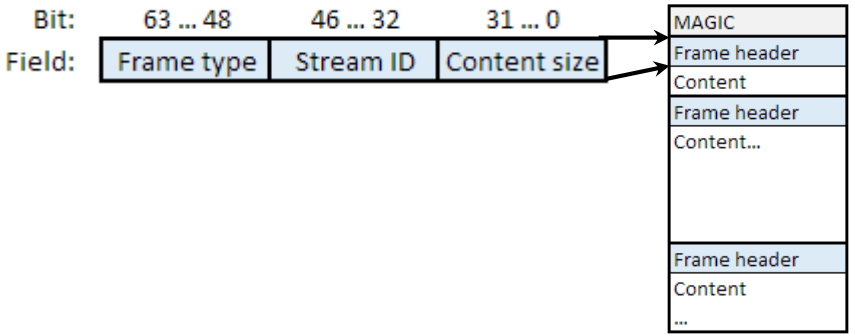
Tools (command line):

- *sdaq*: acquire TPX4 data
  - DAQ server, PCIe, 16 x 10 Gb/s
  - Control board, 10 or 20 Gb/s
  - Control board, slow control < 40 Mb/s
- *sinspect*: scan through data file
  - Collect statistics
  - Display histogram
- *tpx4item*: read write TPX4 registers or bitfields

(by name or register address)

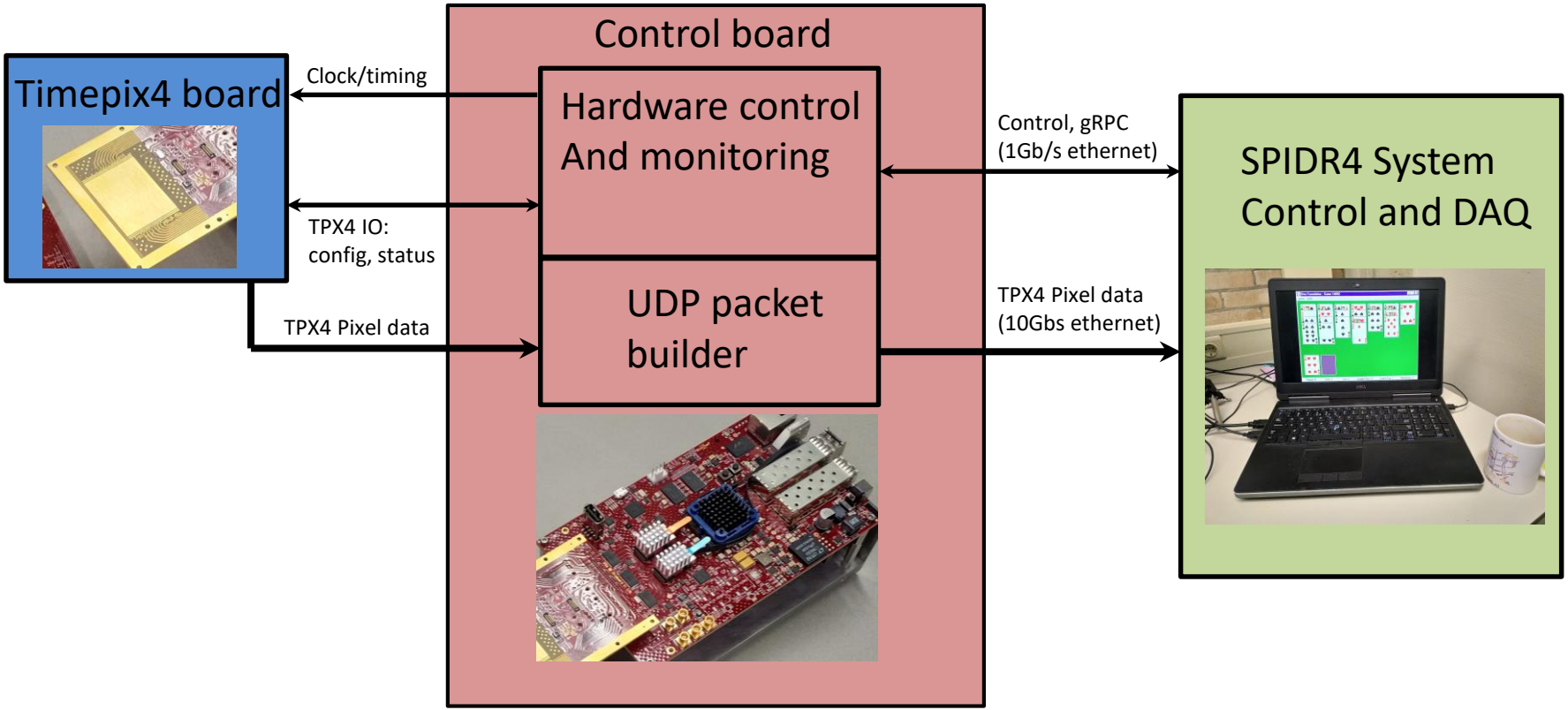
```
martin@martin-Precision-7510:/media/martin/DataNVME/SPIDR4data$ sdaq -t 30 -f 64000 -b 4096 192.168.100.1:43210 tst
(file size capped to 16GB)
Write SPIDR4 data to file
SpidrReceiver: Failed to allocate from /dev/cmem_rcc; now allocating buffer on heap
SpidrIpReceiver43210 @ 192.168.100.1:43210
Started SPIDR4 receiver 0 (buffersize=4096MiB)
Started FileWriter 0 (filesize=16384MiB)
 Secs | Port | Recvd[MB/s] | File[MB/s] | Tot[(M)B] | Rec[(M)B] | Buf[%]| Wrap
------|------|-------------|------------|-----------|-----------|-------|------
    1  43210       1034.0       1018.5       1034.0       1018.5       0       0
    2  43210       1023.7       1035.2       2057.7       2053.7       0       0
    3  43210       1023.4       1006.2       3081.2       3059.9       0       0
    4  43210       1023.8       1024.1       4104.9       4084.0       0       0
    5  43210       1023.7       1032.1       5128.6       5116.0       0       1
    6  43210       1023.4       1035.8       6152.0       6151.8       0       1
    7  43210        996.5        976.3       7148.5       7128.1       0       1
    8  43210        945.3        951.2       8093.8       8079.2       0       1
    9  43210       1015.9       1021.5       9109.7       9100.8       0       2
   10  43210       1023.9       1025.6      10133.6      10126.4       0       2
   11  43210       1023.6       1015.1      11157.2      11141.4       0       2
   12  43210       1023.8       1025.8      12181.1      12167.2       0       2
   13  43210       1023.8       1025.0      13204.9      13192.2       0       3
```

- One data format, independent from DAQ path
- Header includes TPX4 settings

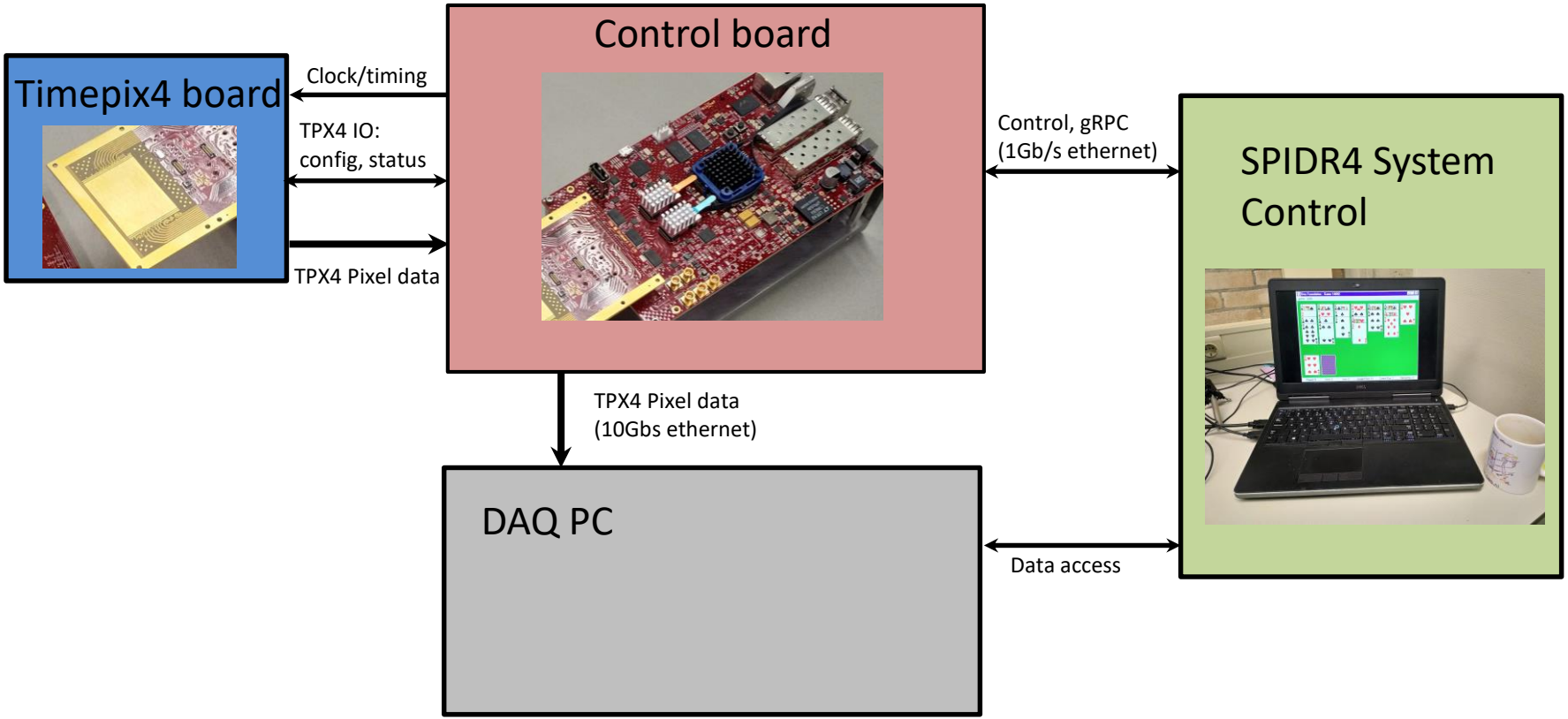


SPIDR4: a few scenarios

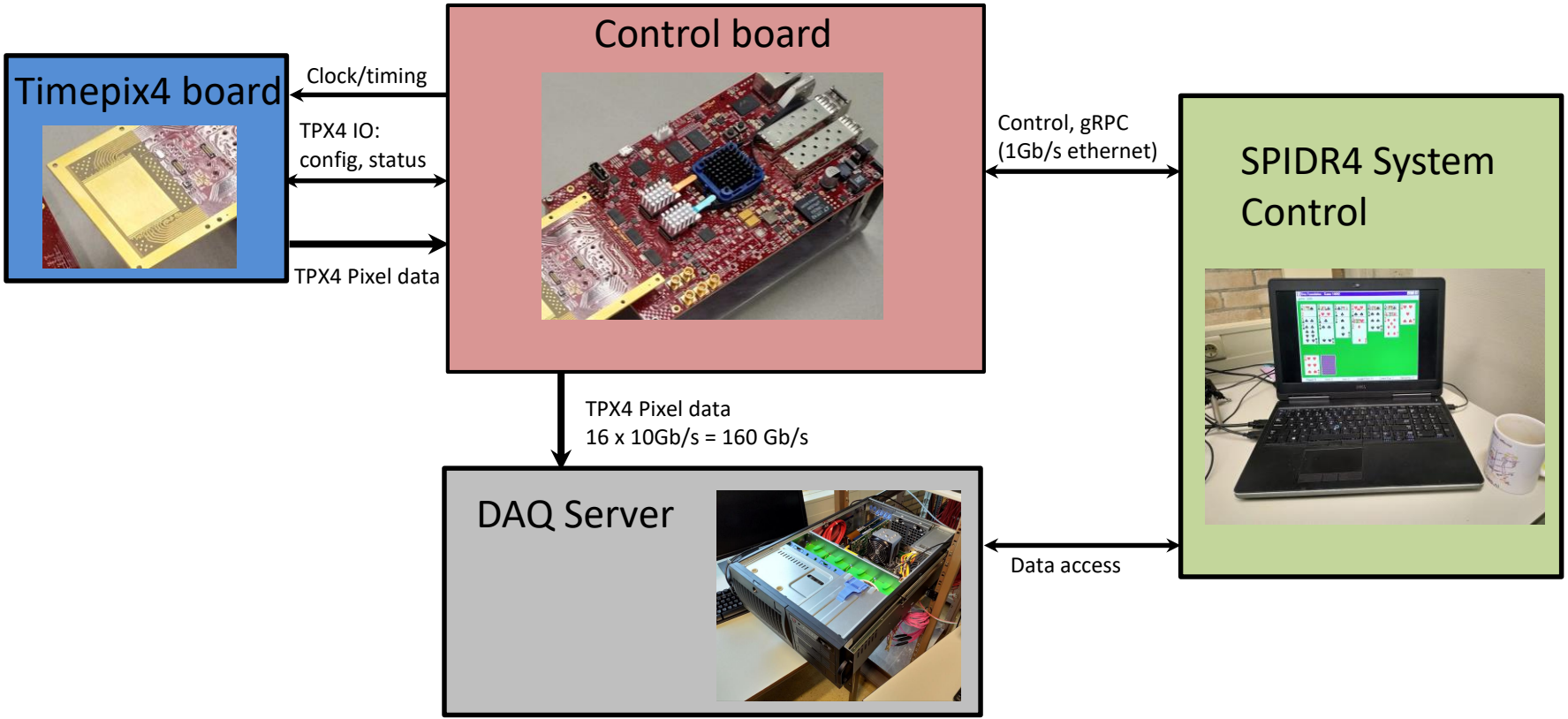
SPIDR4: 10 Gb/s DAQ, simple version



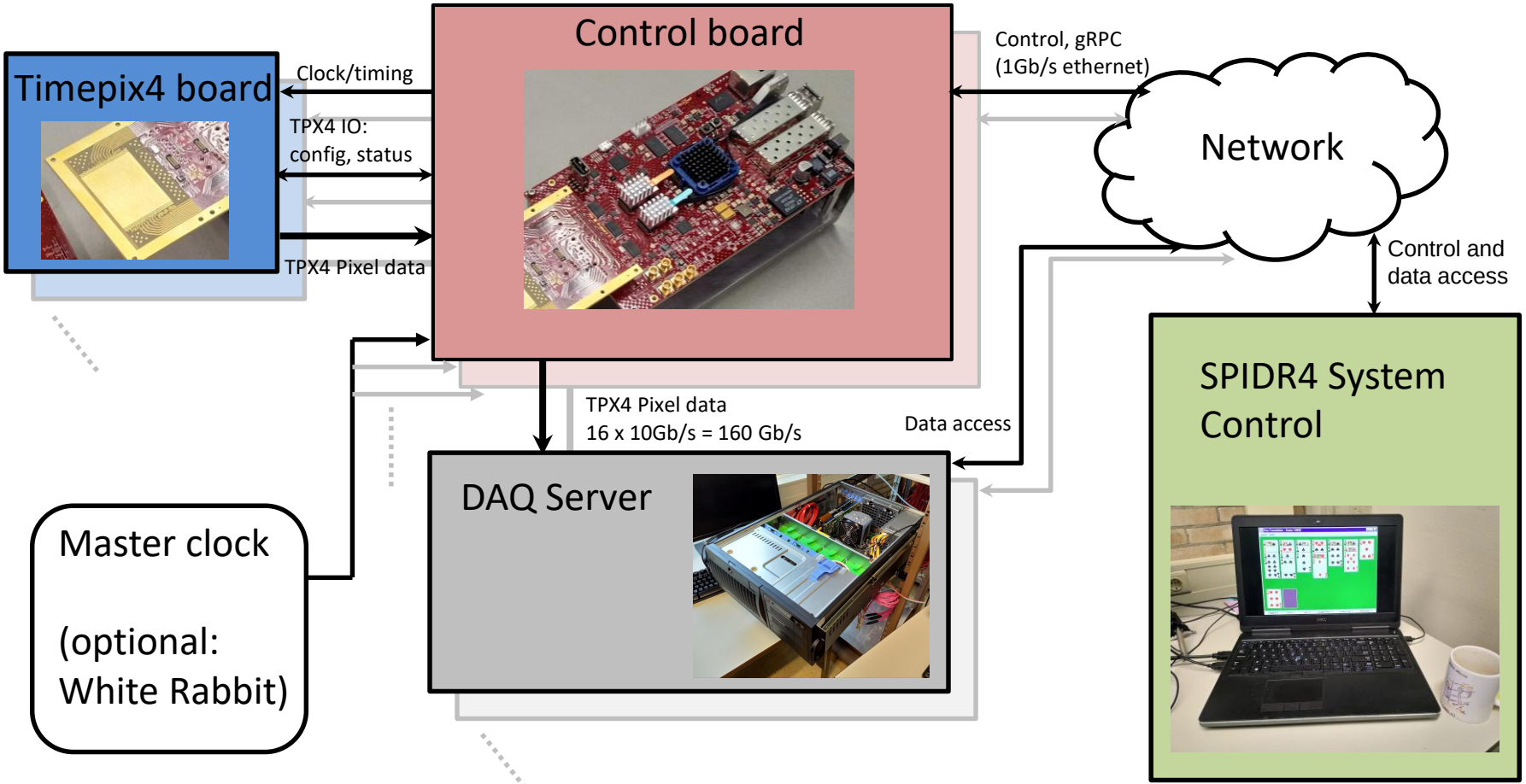
SPIDR4: 10 Gb/s DAQ, dedicated DAQ PC



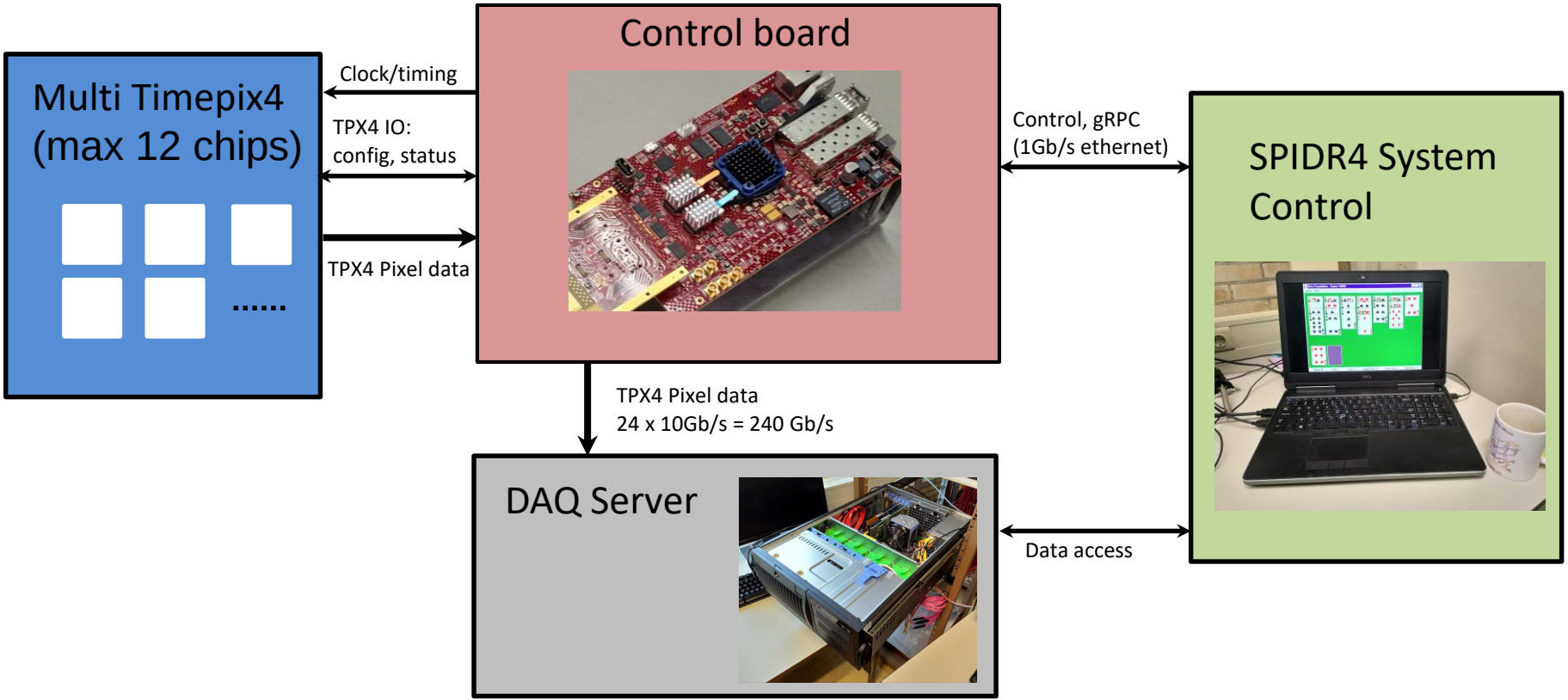
SPIDR4: High performance DAQ



SPIDR4: High performance DAQ, multiple synchronous systems



SPIDR4: Multi chip, 2 x 10 Gb/s per TPX4 chip

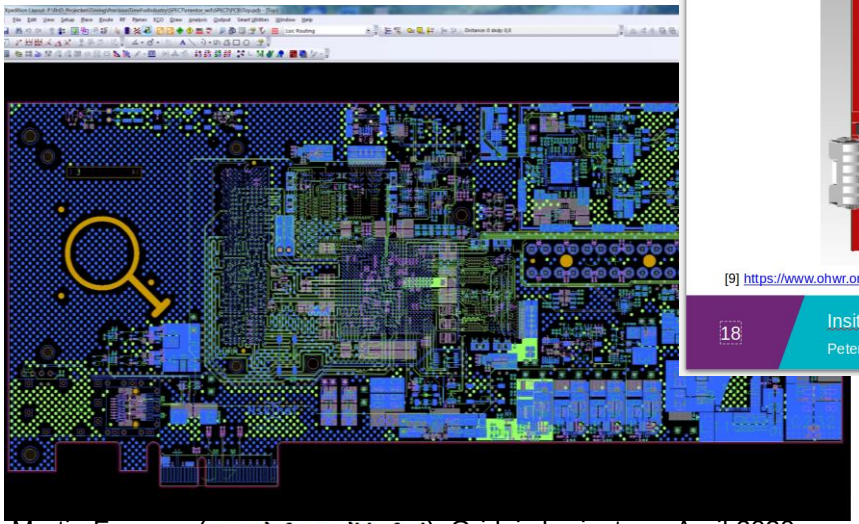


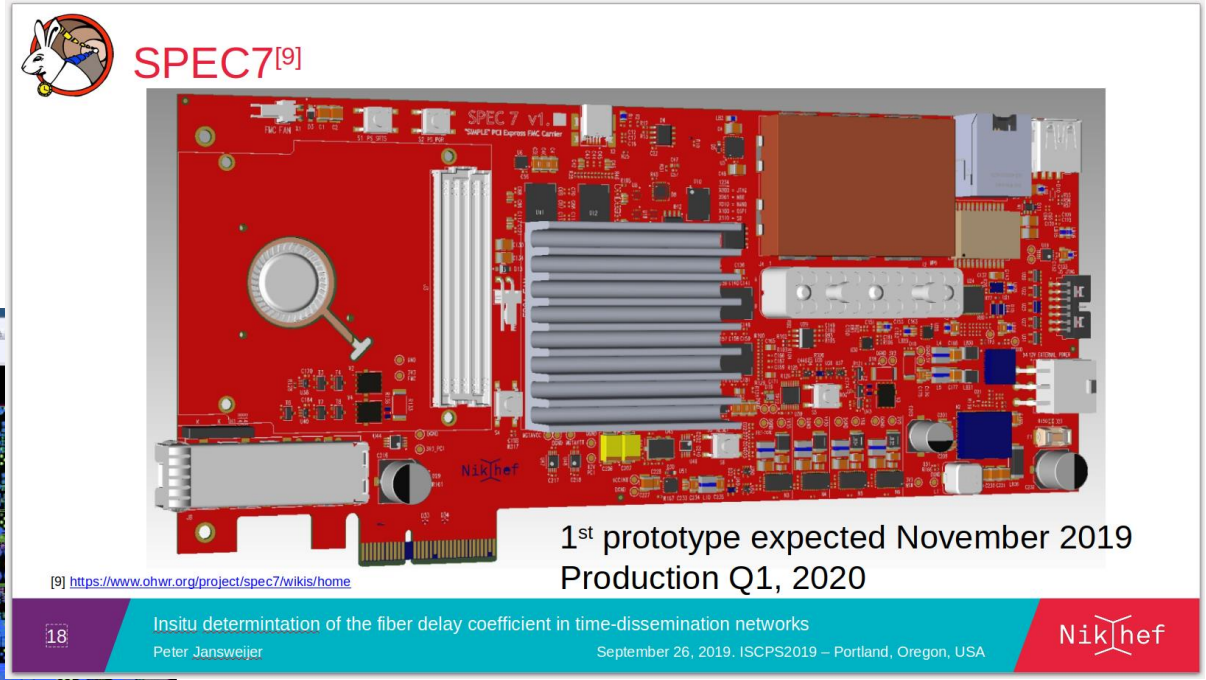
Thanks for your attention,

Questions?

White rabbit synchronization

- Synchronization using PTP
- To sync large TPX 4 based systems
- Improved White Rabbit at Nikhef
  - <100 ps absolute timing (was <1ns)
  - <10 ps stability





https://www.ohwr.org/project/spec7/wikis/home

https://www.ohwr.org/project/white-rabbit/wikis/home

Clock and data path break out PCB

Test signal integrity from clock and data lines

- ‘Bulls eye’ system
- Access to each data lane