



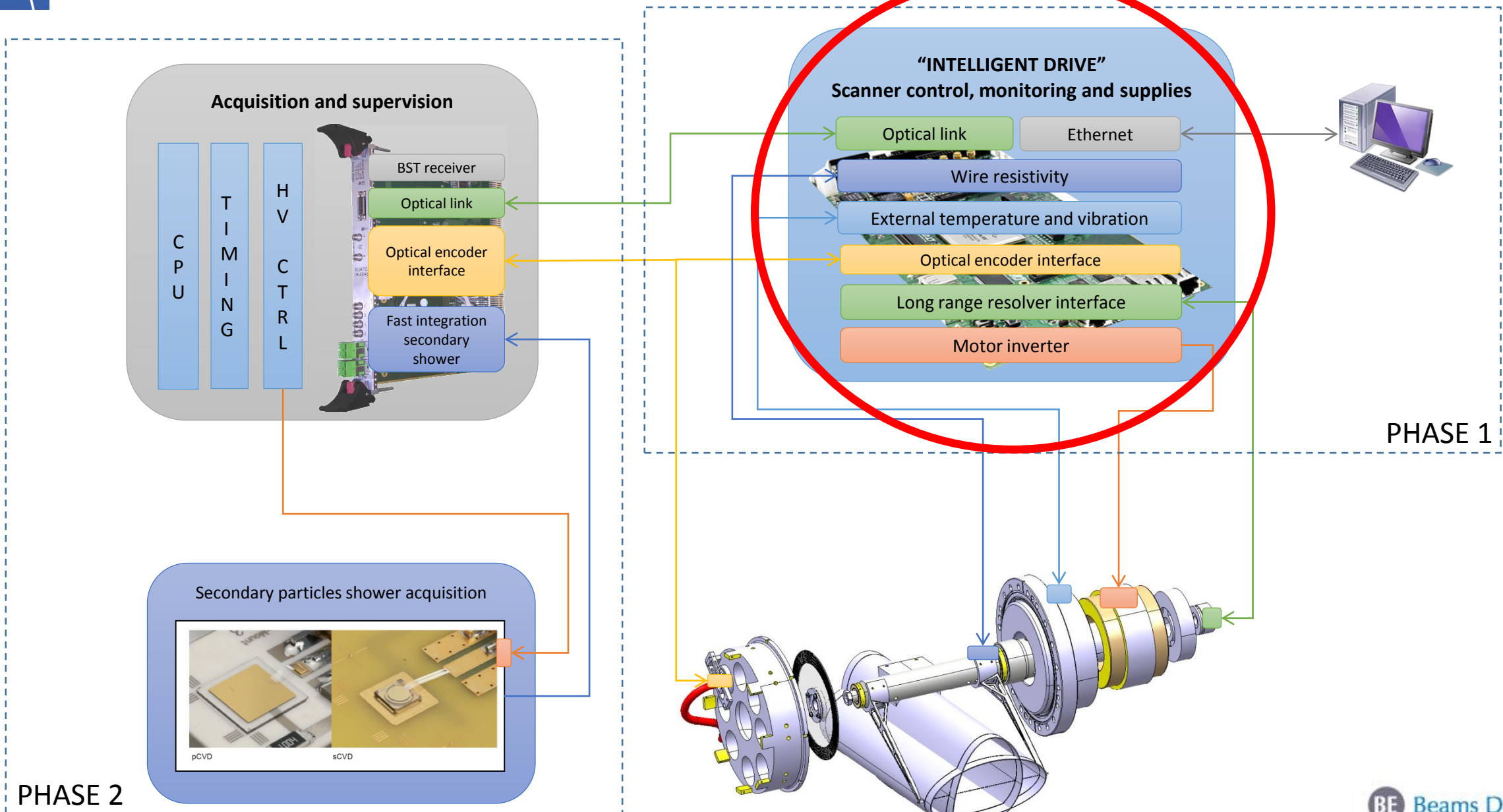
# BWS motion control electronics

BI technical board – 09.06.2016

J. Emery & P. Andersson



# Overview: Control and Acquisition electronics



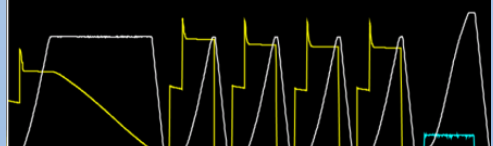


# Overview: External systems connections

## General Machine Timing (GMT)

Low jitter < 1ns, Granularity: 1ms

SPS-PAGE1 Current user: SFTLONG2 13-05-10 09:44:49  
SC 62290 (998P, 46.8s) FT: 500 ms 13-05-10 09:44:50



Target	I/E11	MUL	%SYM	Experiment
T2	25.6	10	94 a	H2/H4
T4	32.2	9	92 a	H6/H8
T6	126.7	1	90 a	COMPASS
T10	0.0	0	0	

CNGS T40.1	138.5	Ok (0)	Comments (12-05-10 17:32)	
CNGS T40.2	144.0	Ok (0)	Phone: 77500 or 70475	
LHC	0.0 E10	Dest: CNGS		

User	Injected	Flat Top
LHCFAST1	107.2 E8	107.2 E8

## Beam Synchronous timing (BST)

Bunch synchronisation  
(25 ns accurate clock)

Revolution frequency synchro  
Triggers: scan start, post-mortem  
Granularity: 89us (LHC), low jitter < 1ns

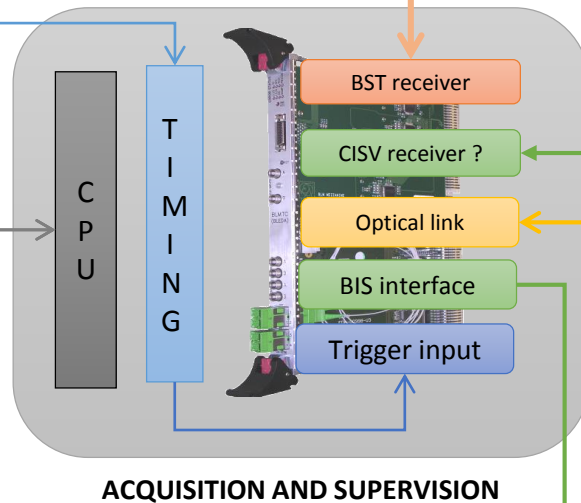
## Beam Energy and Intensity

CISV?  
BCT?

Settings

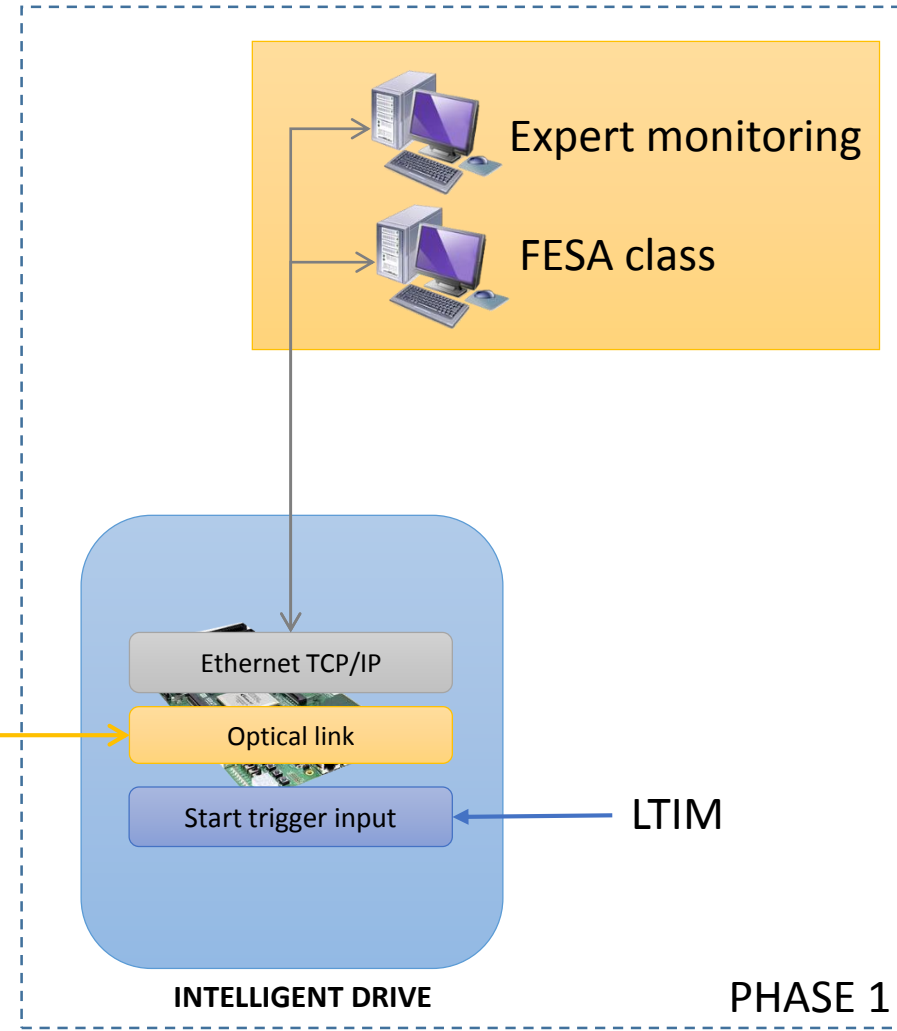
Control room (CCC)

Logging storage  
Long term storage for offline analysis



PHASE 2

Beam Interlock System (BIS)



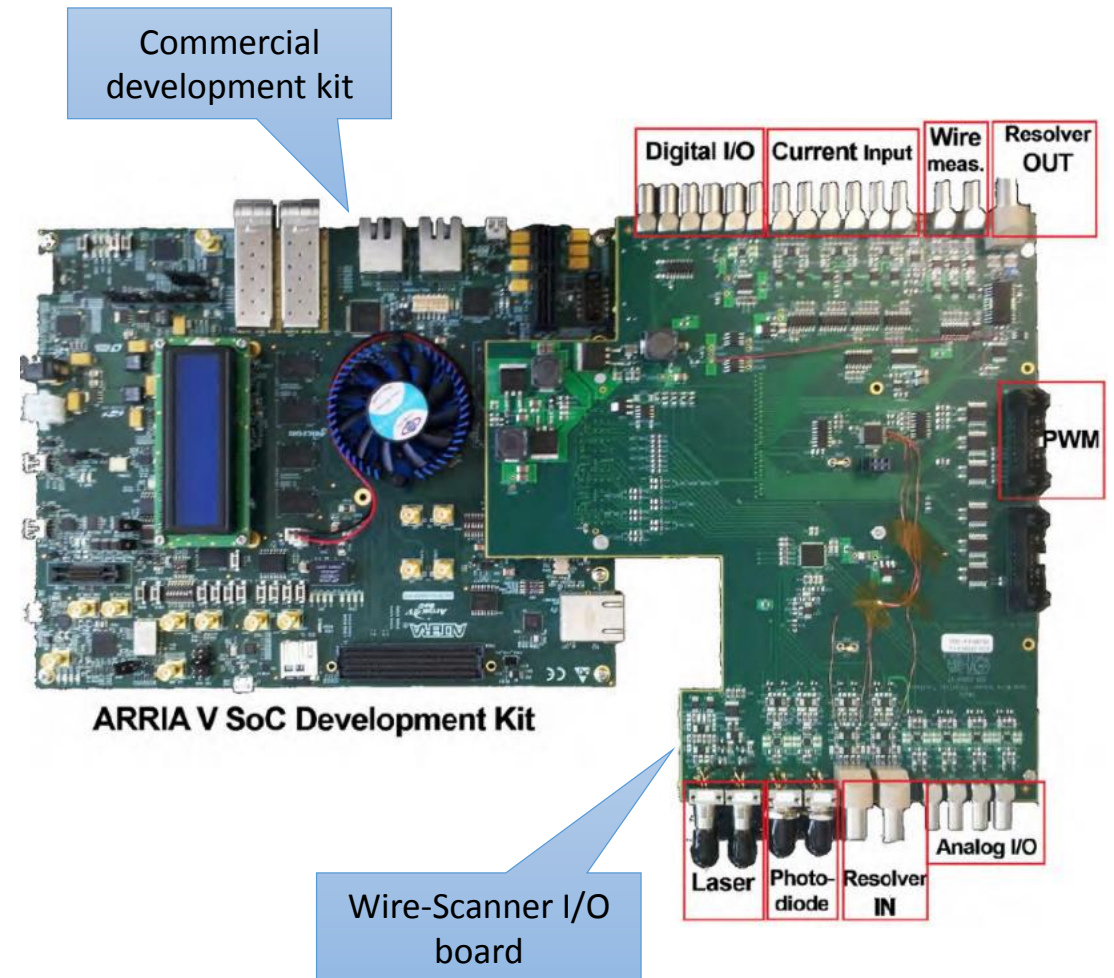


# Context of this presentation

- Beam Wire Scanner meeting on the 19.05.2016 to expose the options for the digital system of the “Intelligent drive”
- These slides are extracted for this presentation
- It was decided to go for VME board + Mezzanine solution as other BI projects use this solution

# Three options for the digital platform

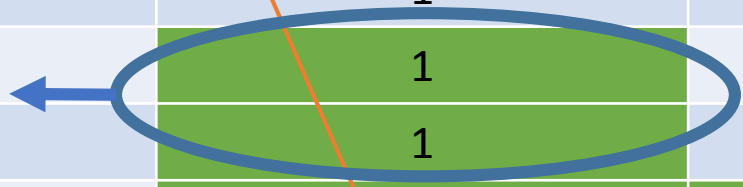
- Option 1:
  - Standalone VME board + custom mezzanine
  - Use of the VFC in standalone mode
  - Dedicated analog/optical mezzanine
- Option 2:
  - Combined analogue-digital board
  - Use the FPGA reference design (Altera)
  - Add dedicated analog/optical circuits
  - Combine the 2 boards we have today
- Option 3: Starter-kit + mezzanine
  - Use Arria V SoC Dev kit
  - Dedicated analog/optical mezzanine





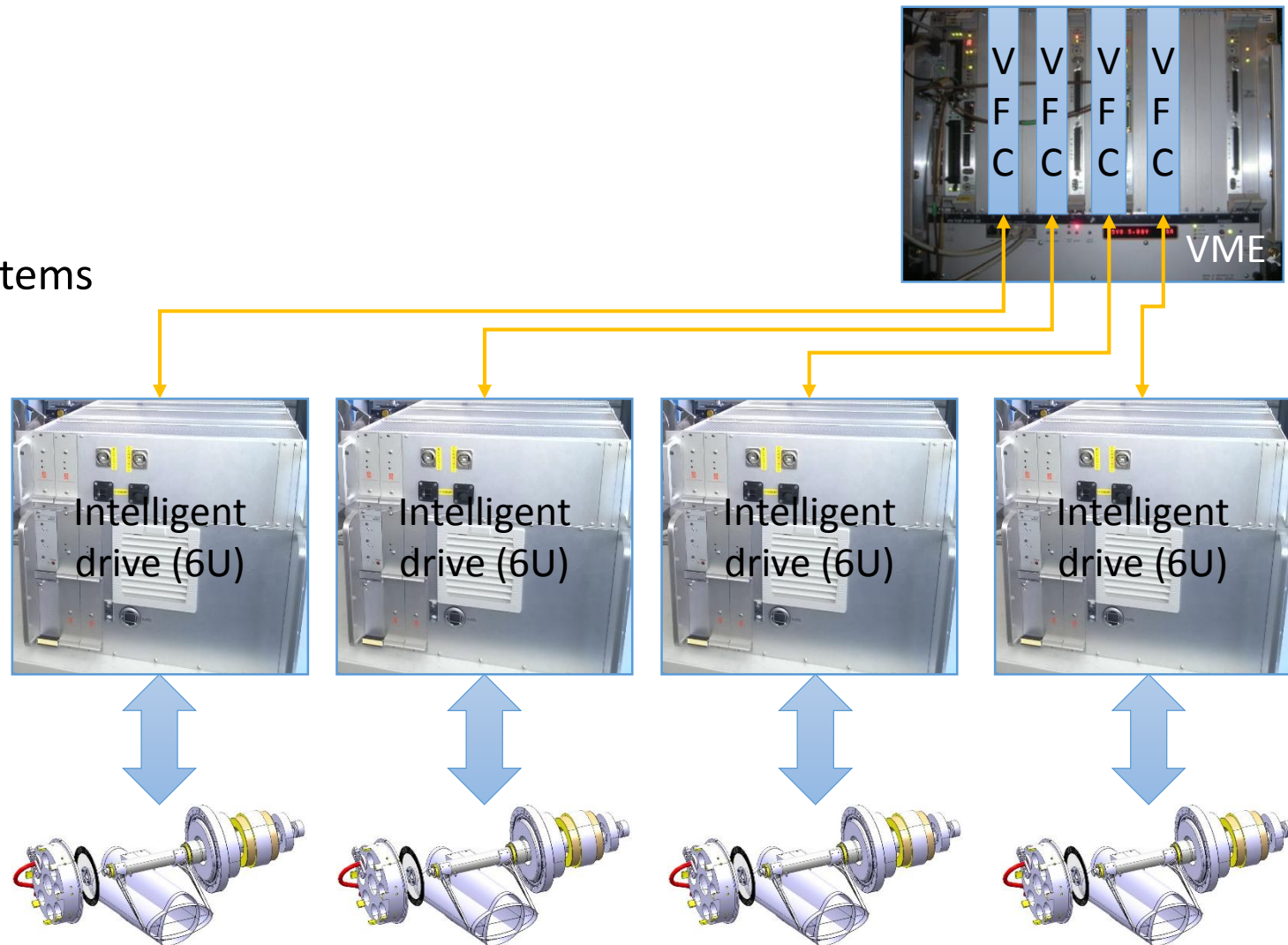
# Summary table

Criteria	Option 1: VFC modified	Option 2: Custom	Option 3: DevKit
Board size	3	1	2
Powering	3	1	2
EMI	3	1	2
FPGA resources	2	1	1
Board interfaces	1	2	2
External Memory	2	1	1
Testability	1	1	1
Code reuse	1	1	1
Methodology	2	1	1
FW readiness	3	2	1
HW readiness	2	3	1
Hardware design effort	2	3	1
Firmware design effort	3	2	1



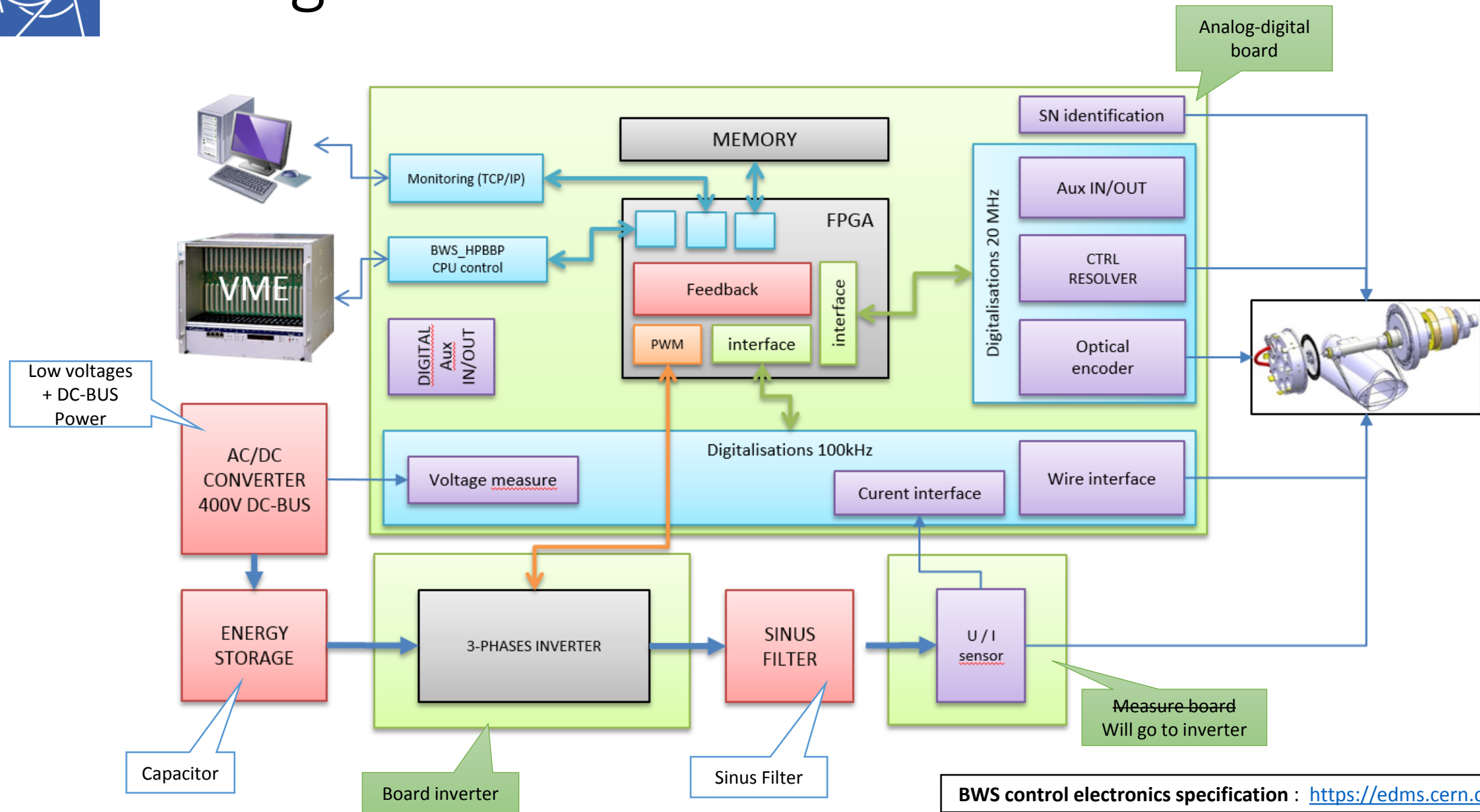
# Scanner control architecture

- One intelligent drive (ID) per scanner
  - avoids multiplexing
  - constant monitoring/control
  - allow parallel scans
  - But imply more control and acquisition systems
- Deported processing from VME to ID
- Local monitoring and fault diagnostics
- One VME crate for multiple scanners
  - Number depends on CPU-Memory load
- So we try to minimize the ID size
- Will still require more space than current installation:  
(PSB: 3 racks instead of 1)





# Intelligent drive architecture



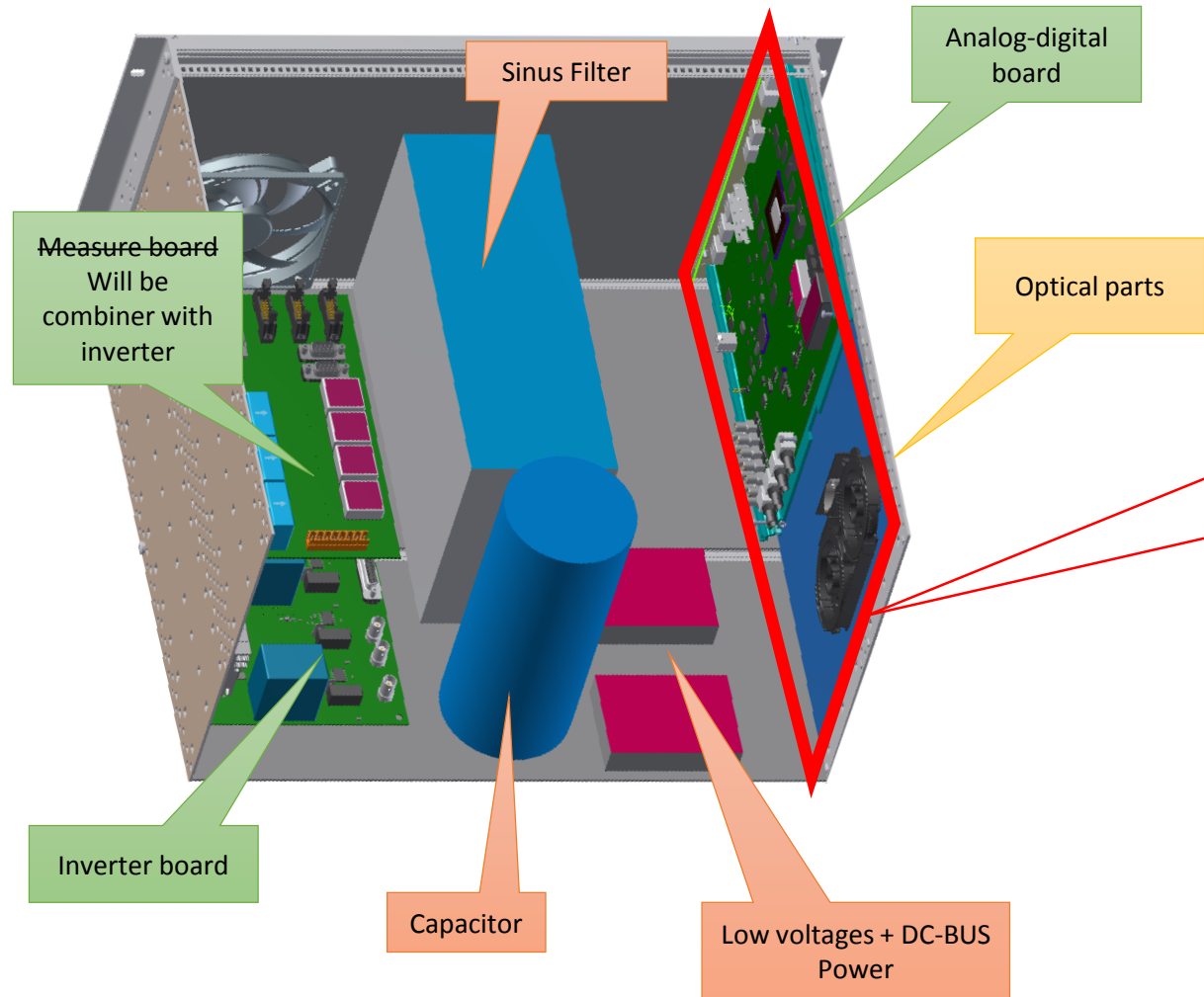
BWS control electronics specification : <https://edms.cern.ch/document/1318827>





# Space availability for the board in ID

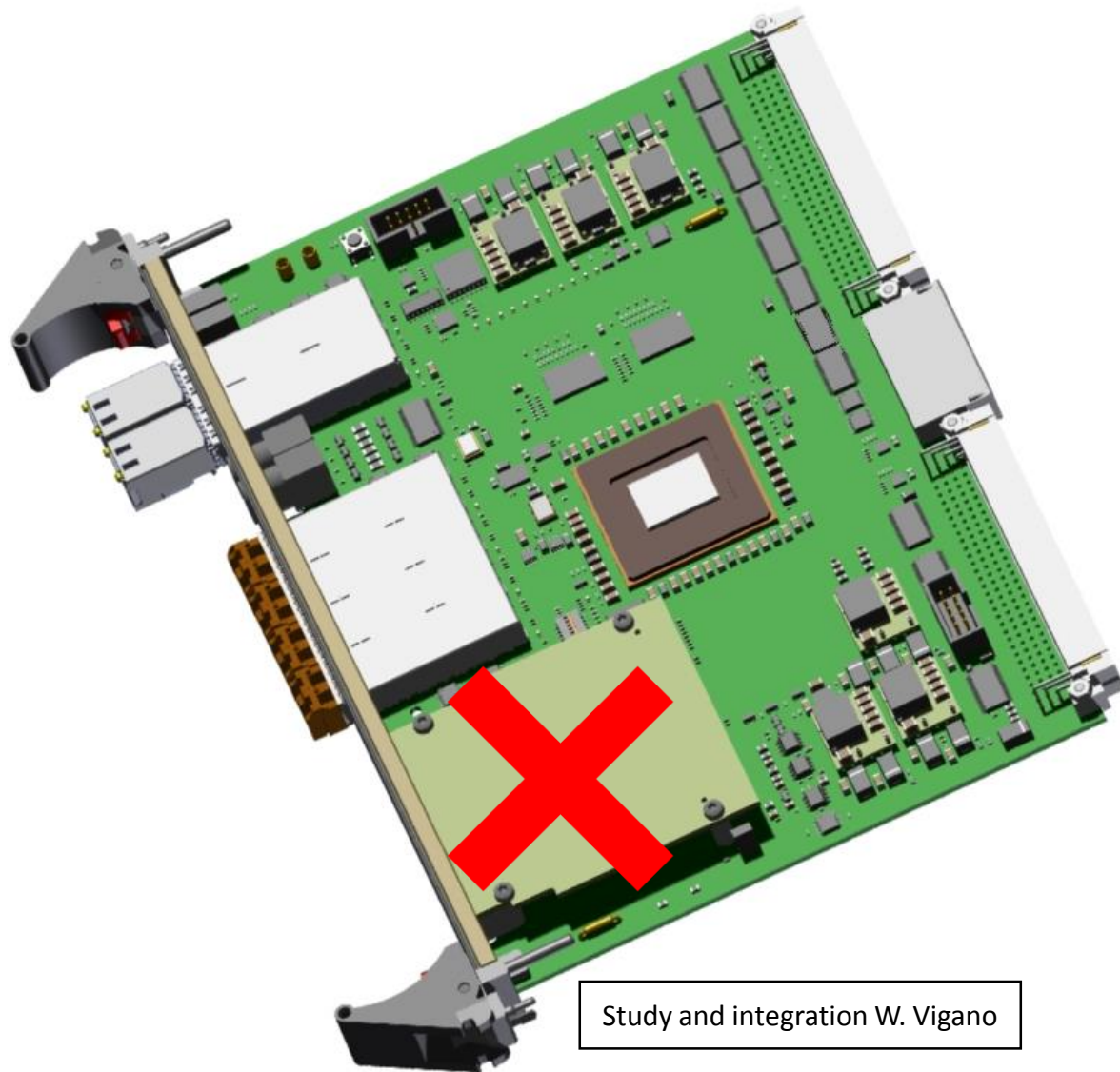
- Board realisation
- 1) Boards size
  - 2) Powering requirements
  - 3) Cabling and EMI protection



Space available for:  
analog, digital and optical

Made possible by reducing:  
Inverter size  
Power supplies types  
internal arrangement  
(see Patrik's presentation)

# Option 1: VFC + MEZZANINE

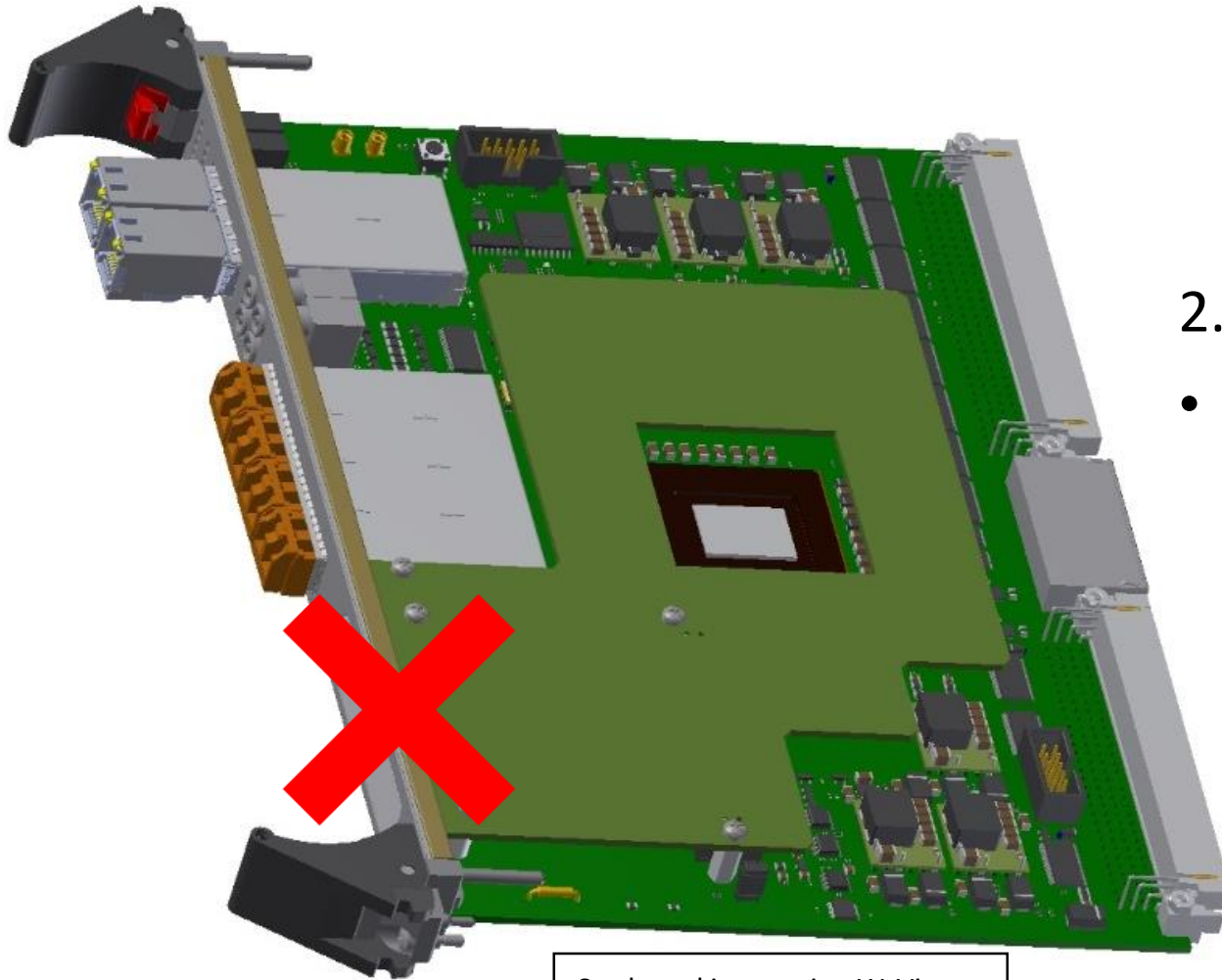


Study and integration W. Vigano

1. Standard VFC + standard mezzanine
  - Missing board space for components

# Option 1: VFC + MEZZANINE

- 1) Boards size
- 2) Powering requirements
- 3) Cabling and EMI protection



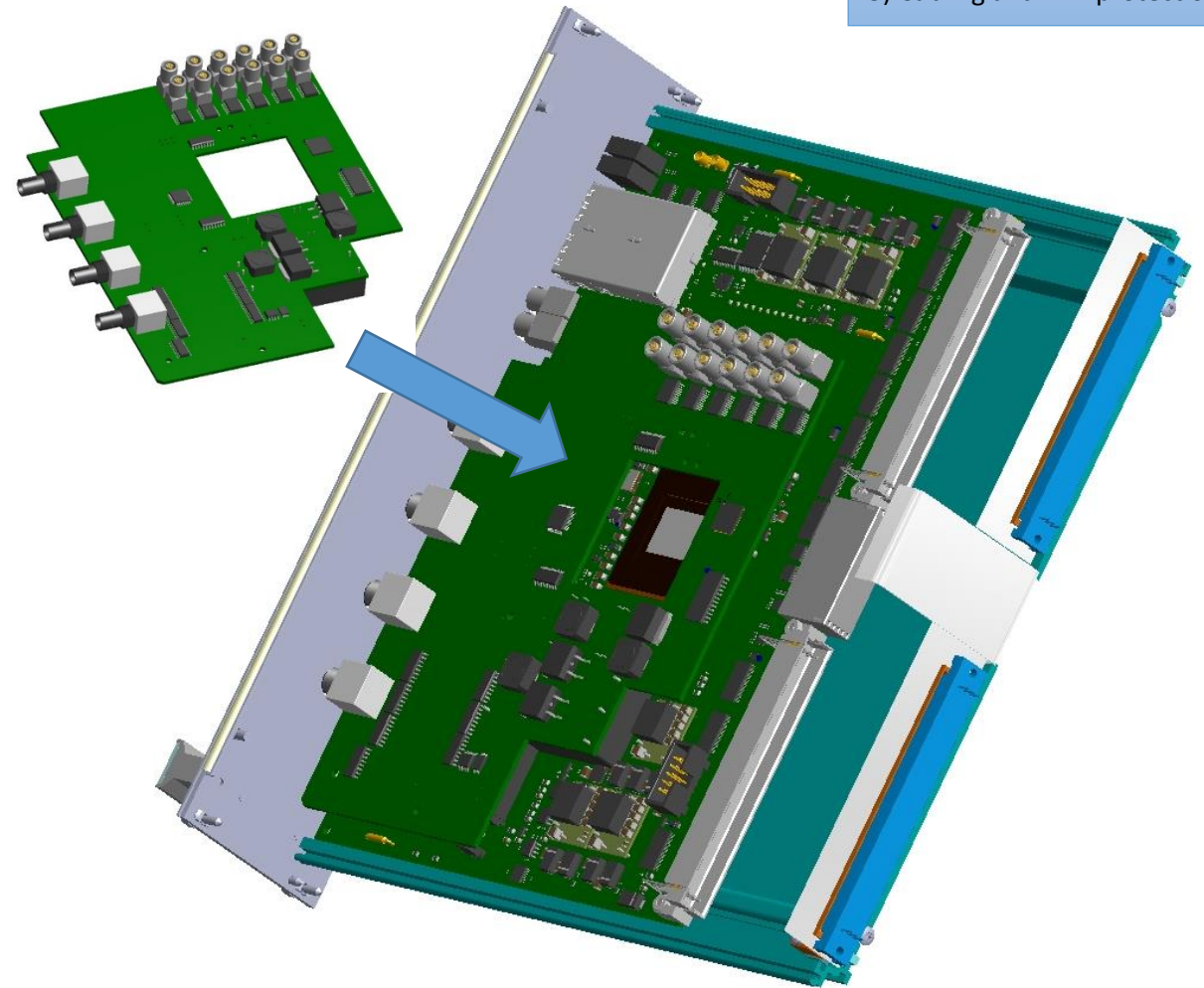
Study and integration W. Vigano

2. Standard VFC + maximized mezzanine
- Missing space for the optical components

# Option 1: VFC + MEZZANINE

## 3. Customized VFC + maximized mezzanine

- Remove the lower SPF cages
- Check power supply VADJ
- VFC in standalone mode
- Stack-up of VMC and mezzanine
- Dedicated analogue/optical mezzanine
- None standard FMC mezzanine shape due to the numerous components
- VME connectors for powering the boards

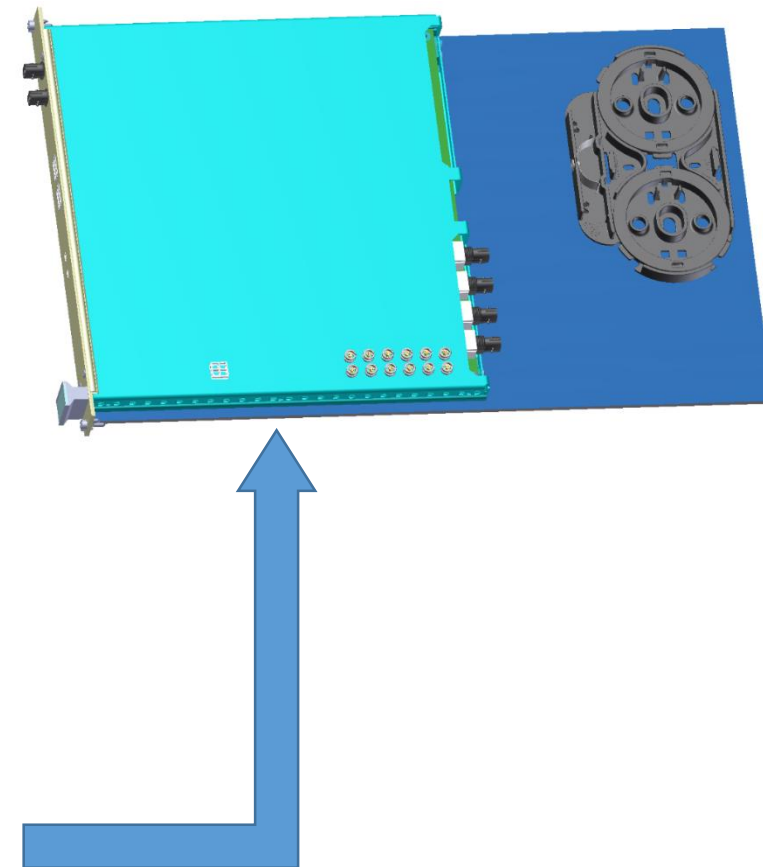
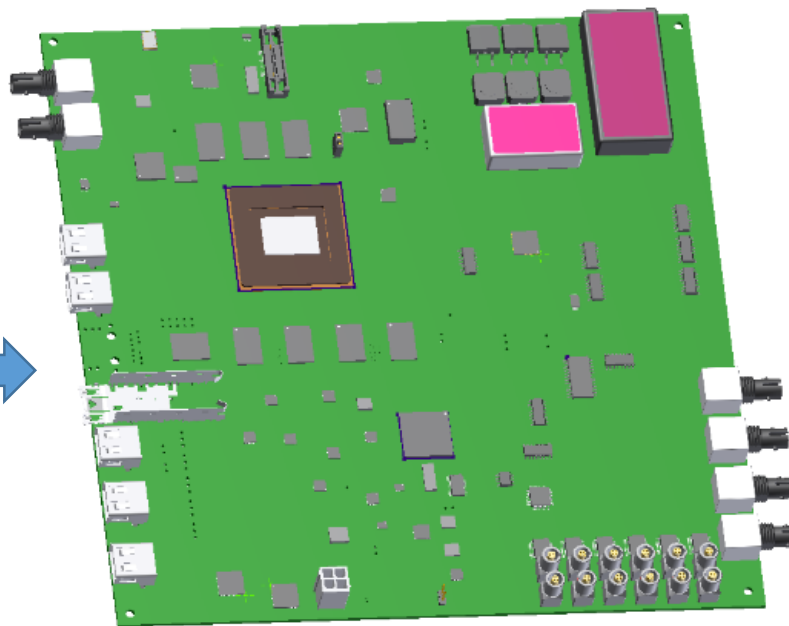
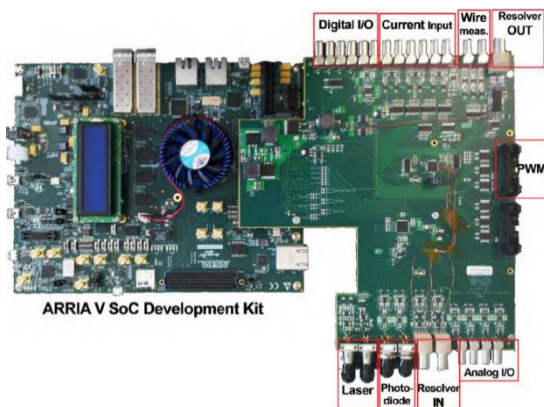


# Option 2: Combined analog-digital board

## Board realisation

- 1) Boards size
- 2) Powering requirements
- 3) Cabling and EMI protection

- Combination of the 2 boards we have today
- Use the FPGA reference design (Altera)

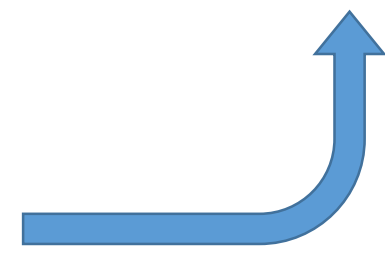
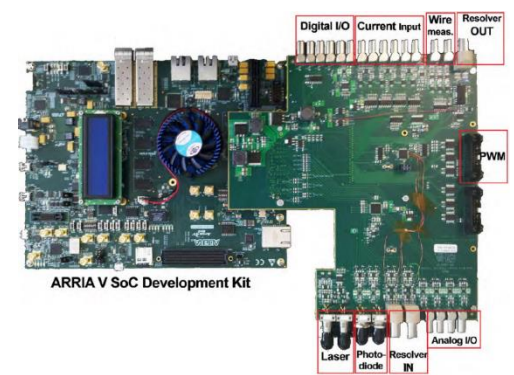
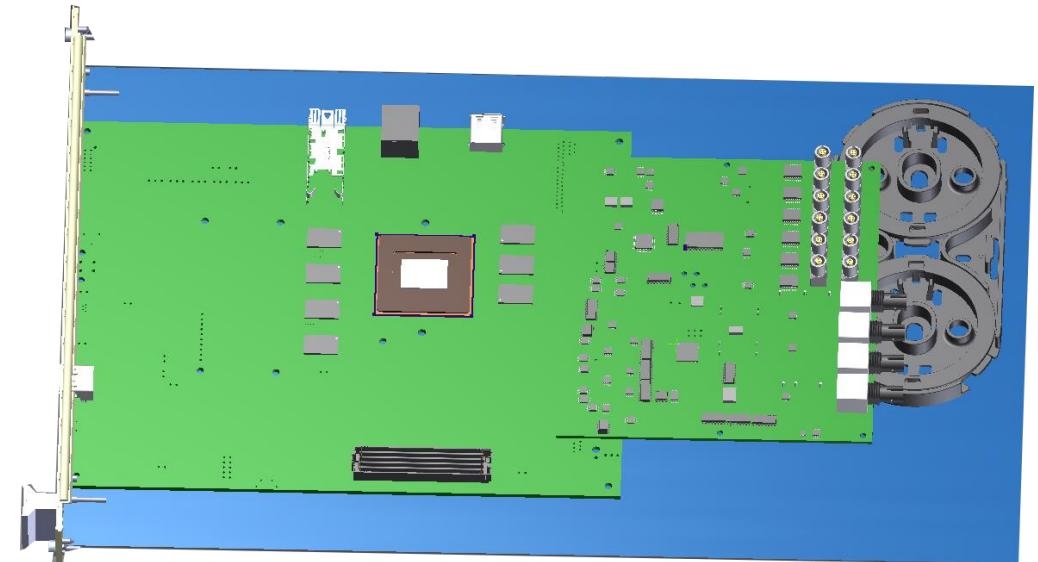


Boards combinations & integration: P. Andersson

# Option 3: Starter-kit + mezzanine

- 1) Boards size
- 2) Powering requirements
- 3) Cabling and EMI protection

- Same configuration as today ✓
- FPGA platform Arria V SoC Dev kit
- Modification of the wire-scanner mezzanine to fit the box



Boards combinations & integration: P. Andersson



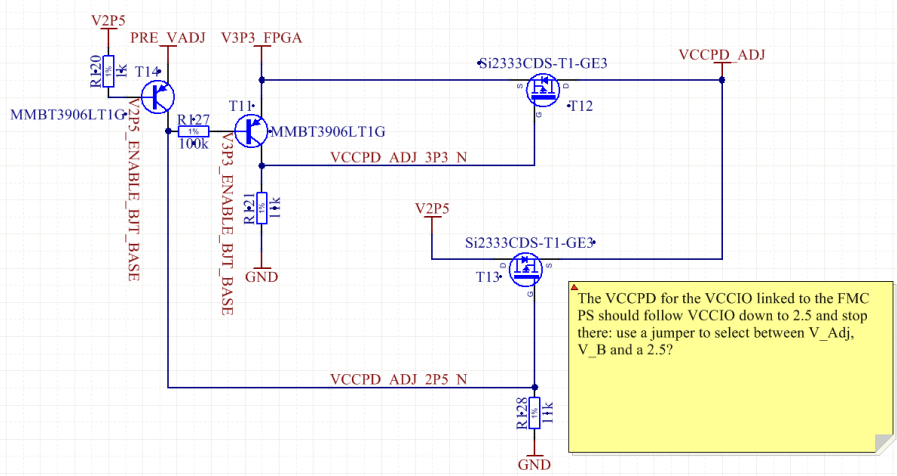
# Powering requirements

- Board realisation**
- 1) Boards size
  - 2) Powering requirements
  - 3) Cabling and EMI protection

VFC powering the BWS analog board:

- 12 [V] -> OK
- 3.3 [V] -> OK
- 1.8 [V] -> to be checked

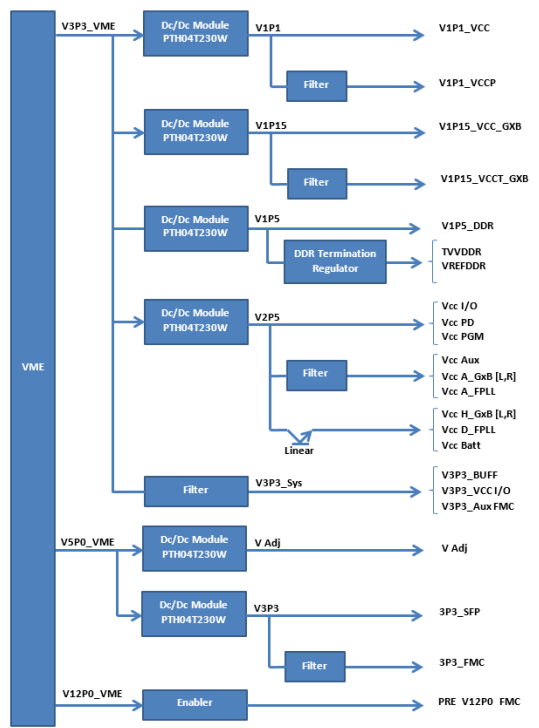
1.8[V] is needed to operate the fast ADC, the whole board is using this voltage.



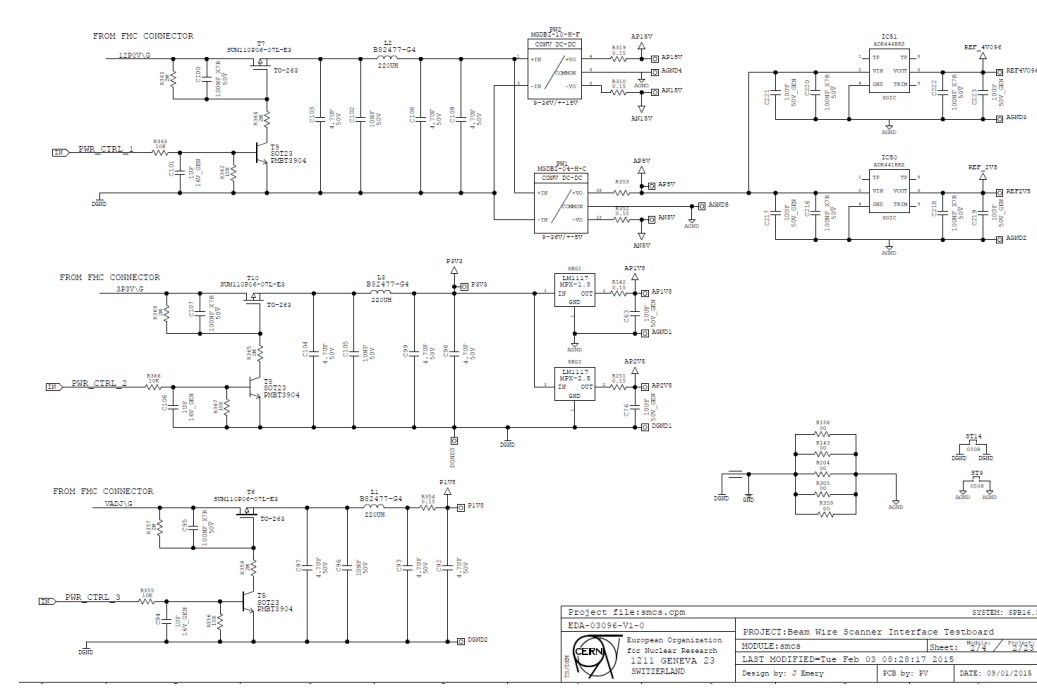
▲ The VCCPD for the VCCIO linked to the FMC PS should follow VCCIO down to 2.5 and stop there: use a jumper to select between V\_Adj, V\_B and a 2.5?

VFC Vadj. Powering schematics

## VFC power scheme



## BWS analog powering



⇒ Confirmation from Andrea, 1.8V is possible  
 ⇒ Is this functionality already tested on the board for another project?



# EMI susceptibility

Board realisation

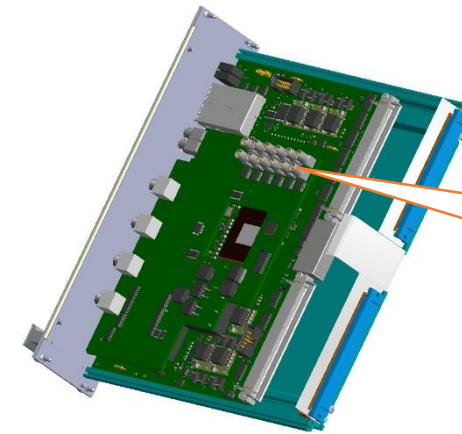
- 1) Boards size
- 2) Powering requirements
- 3) Cabling and EMI protection

## Cablings:

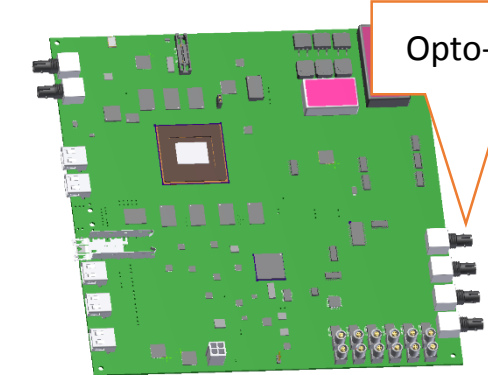
- Similar philosophy of the cablings for all 3 solutions (use of top connectors)
- Other connections slightly worst on the DevKit since uses all sides

## EMI interferences:

- All 3 options will use same box => same shielding from external sources
- Only remains perturbations between analog and digital part



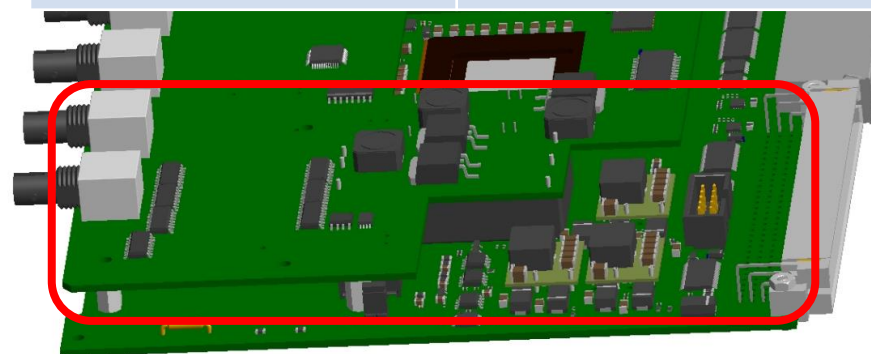
Connections to the inverter & measure



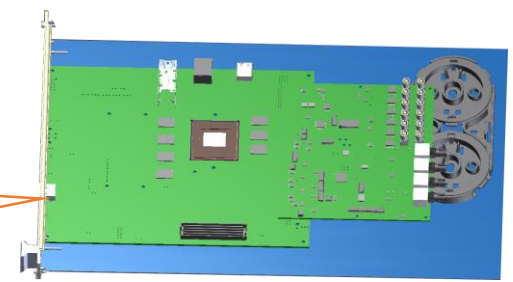
Opto-electronics

Options	Electrical coupling
1. VFC	-
2. Custom	+
3. DevKit	-

=> We are preparing a new mezzanine which will sit next to the VFC to overcome potential issue



Ethernet – optical link







# Digital architecture related criteria

## Digital architecture

- 1) FPGA internal resources
- 2) Board interconnects
- 3) External memory

- FPGA logic elements:
  - ok for today's implementation: with the 3 options
  - Future implementation: Depends on processing complexity
- More flexibility using ARM CPU
- External memory potential limitation
- VFC TCP/IP Data transfer will probably be 4 to 10x slower than today (400 Mbits, 100 to 40 Mbits)

	Custom design Analog-Digital	VFC
Code status (for 2016)	95%	50% (6 months)
Evolvitivity		Neutral
Use of IP developed for VFC	Same FPGA & transceiver	
FPGA use as today ALM [%]	14	21
Memory [%]	10	16
DSP Blocks [%]	13	16
FPGA type	ARRIA V - SOC	ARRIA V
Type	5ASXFB3H4F40C5N*	5AGXMB1G4F40C4N
Nbr Gates	362K	300K
ALM (adaptive logic module)	136880	113208
Memory (M10k)	17,260	15100
DSP Blocks	1045	920
Logic use for soft CPUs	2x ARM processor at 1GHz	Software NIOS II at 200 MHz**
Transfert TCP/IP (as today)	>400 Mbits tested point-to-point	40 -100 Mbits max
memory controller	3 hard memory controllers	2 hard memory controllers
Processor side	2x 256 x 16bits + 1 x 256 x 16bits ECC	-
DDR3 SDRAM type	MT41K256M16HA-125:E	MT41K512M16HA-125:E
Organisation	256 M x 16	512 M x 16
FPGA side	4x 256M x 16bits	2x 512M x 16 bits
memory total in bytes	2048 Mbytes	2048 Mbytes
shared with program RAM	no	yes
Nbr of measurement saved (worst case SPS)	2048/336 = 6	2048/336= 6
Maximum theoretical transfert	4 x 1600 Mword = 12.8 Gbyte	2 x 1600 Mword = 6.4 Gbyte
Implemented interface tested	4 x 800 Mword = 6.4 Gbyte/s	Extrapolation: 2x800/4 = 0.8 Gbit/s
SPS: 336 Mbytes burst read time	0.0525	0.105
PSB: 190 Mbyte burst read time	0.0296875	0.059375

With NIOS Softcores

Many parts missing

using NIOS Softcores

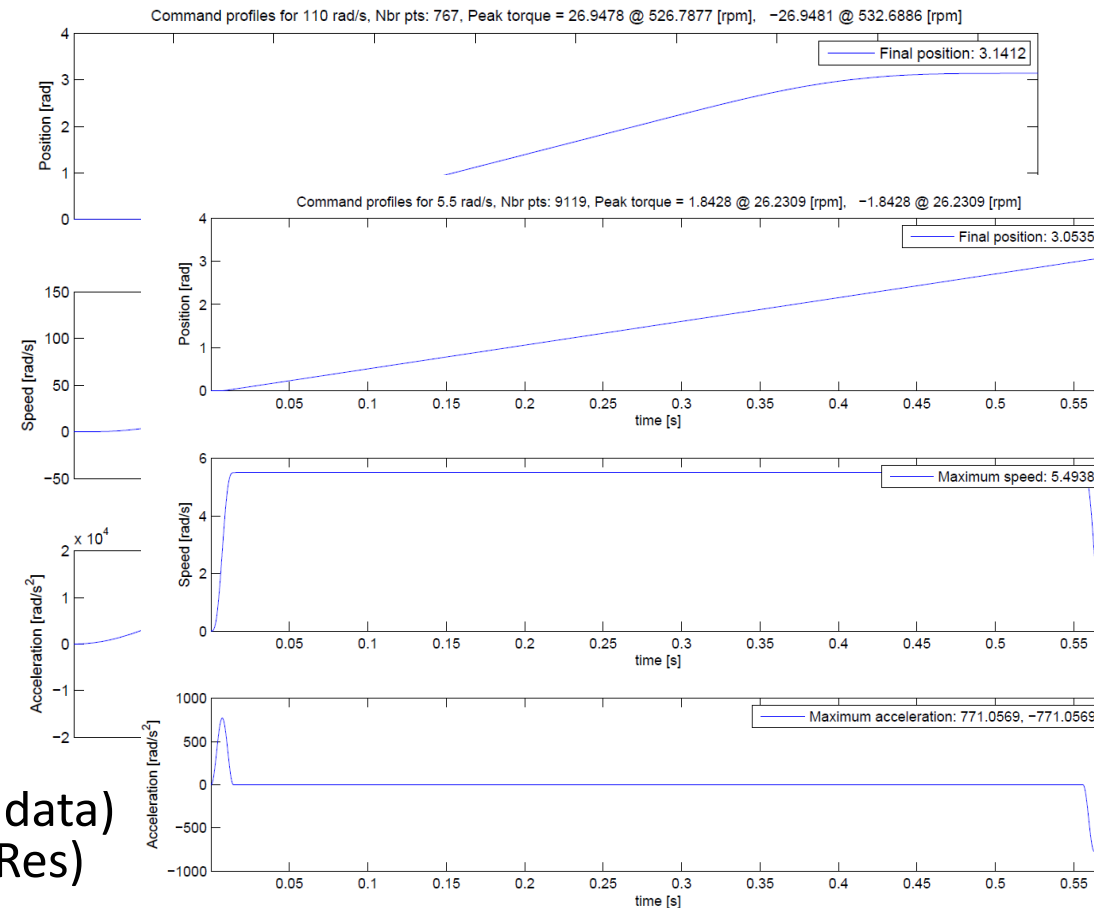
Details next slides

\*As today on the started kit

\*\*Altera "Nios II Performance Benchmarks" 16.12.2015

## Depends on the use cases

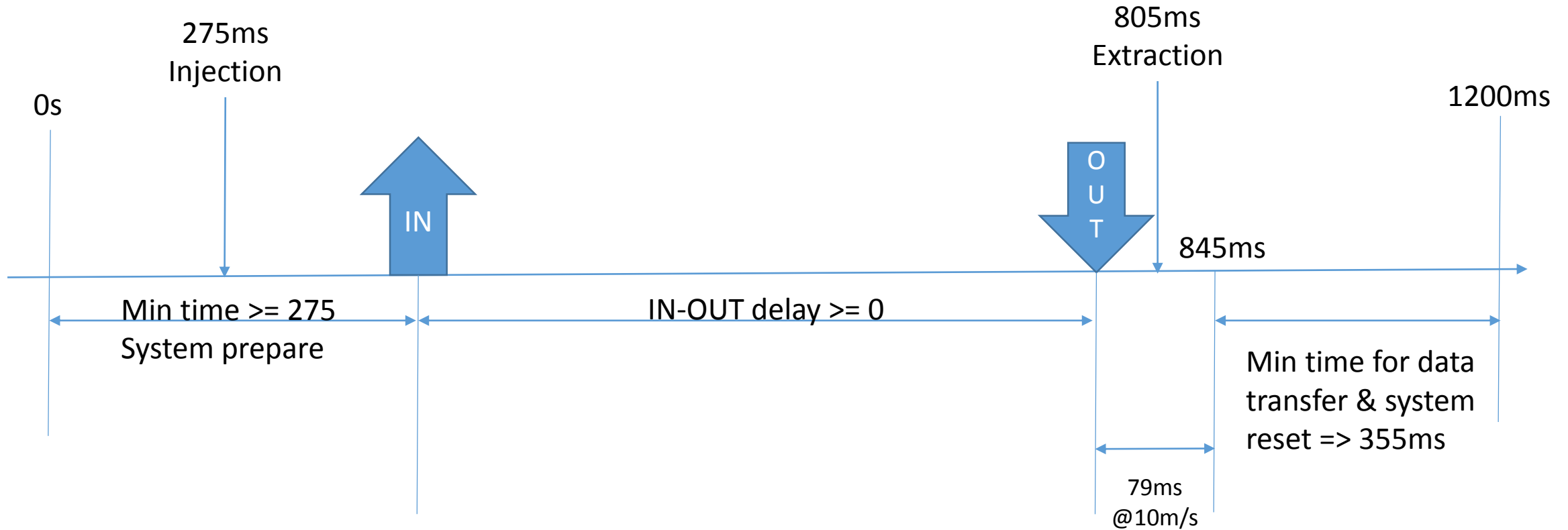
- Scans duration change a lot with speeds  
 20 [m/s] -> 48 [ms] (767 pts)  
 1 [m/s] -> 570 [ms] (9119 pts)
- Time between IN and OUT  
 (We will limit this time to about 1s)
- Number of scans per user:  
 min. 2 if we limit INOUT time to get same functionality  
 max. determined by memory depth,  
 mode of operation (expert/op),  
 required repetition rate (can we fix it?)
- For SPS:  
 With time between IN and OUT of 1s  
 Worst Case:  $2048/336 = 6$  scans (full record OPS and resolver data)  
 Best Case:  $2048/5 = 409$  scans (no offline processing of OPS/Res)
- VFC or for custom options have the same memory depth (2048 Mbytes)





# PSB basic period, 1.2s, one BWS cycle

- Digital architecture
- 1) FPGA internal resources
  - 2) Board interconnects
  - 3) External memory



355ms to transfer 101 Mbytes  $\Rightarrow 298$  Mbit/s  
Not possible with TCP/IP and VFC  
Needs full implementation using VME (Phase 2)

# Memory depth for the PSB

Expert mode  
(detailed data)

Tangential speed [m/s]	Angular speed (PSB)	movement duration [s]	max. INOUT	feedback + wire data [Mbits]	Optical encoder [Mbits]	resolver raw [Mbits]	total [Mbits]	[Mbyte]
20	133	0.04	0.51	20.33	360.11	360.11	740.54	92.57
15	100	0.053	0.504	21.02	372.31	372.31	765.65	95.71
10	67	0.0785	0.491	22.33	395.51	395.51	813.34	101.67
1	6.7	0.485	0.245	41.86	741.58	741.58	1525.02	190.63
20	133	0.04	0.51	20.33	0.58	0.29	21.20	2.65
15	100	0.053	0.504	21.02	0.60	0.30	21.92	2.74
10	67	0.0785	0.491	22.33	0.64	0.32	23.28	2.91
1	6.7	0.485	0.245	41.86	1.20	0.60	43.66	5.46

Operational mode  
(OPS & RESOLVER processed)

- 1) One measurement cycle (one IN, one OUT)
- 2) Data recorded continuously between in and out movement
- 3) INOUT time calculated for IN=275ms, OUT=805ms
- 4) **Expert mode** => Motion data and raw encoders data storage  
Will be used until we have the optical position sensor digitalised in the VME
- 5) **Op Mode** => Motion data and processed encoders data storage

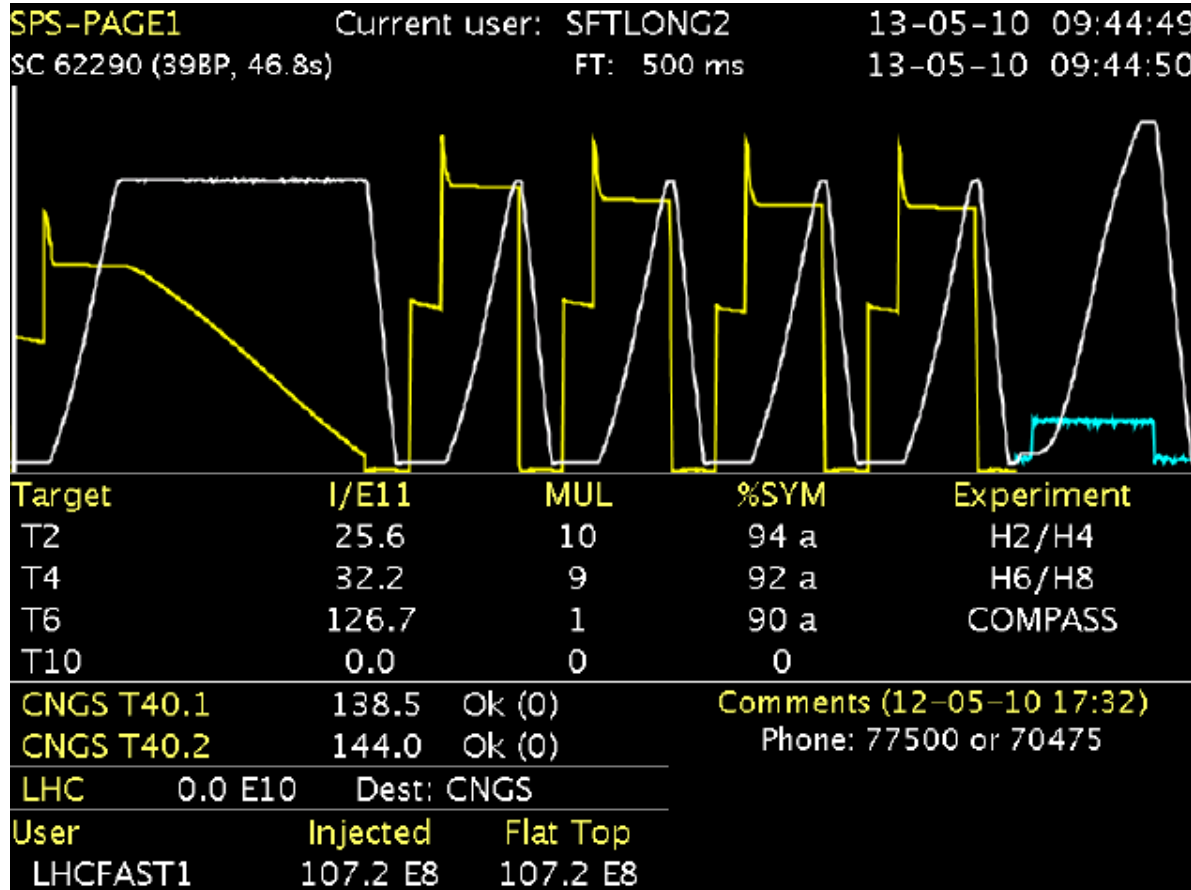
IN-OUT delay used for the calculation:  
 20 m/s => 805-20-275=510ms  
 15m/s => 805-53/2-275=504ms  
 10m/s => 805-79/2-275=491ms  
 1m/s => 805-570/2-275=245ms



# SPS Supercycle

## multiple BWS cycles per user

Digital architecture  
 1) FPGA internal resources  
 2) Board interconnects  
 3) External memory



- Multiple users with different durations 1.2s to >20s
- Challenge arises when managing multiple scanning cycle per user
- Clear use cases must be given by OP/ABP to calculate the among of produced data and what data reduction we need to apply:
  - at the FPGA levels (ID and/or AS)
  - at the VME - CPU levels



# Memory depth for the SPS

Digital architecture  
 1) FPGA internal resources  
 2) Board interconnects  
 3) External memory

Expert mode  
 (detailed data)

Tangential speed [m/s]	Angular speed (SPS)	movement duration [s]	INOUT	feedback + wire data [Mbits]	Optical encoder [Mbits]	resolver raw [Mbits]	total [Mbits]	[Mbyte]
20	110	0.048	1	37.76	668.95	668.95	1375.65	171.96
15	82	0.064	1	38.87	688.48	688.48	1415.82	176.98
10	55	0.0933	1	40.88	724.24	724.24	1489.37	186.17
1	5.5	0.57	1	73.73	1306.15	1306.15	2686.04	335.75
20	110	0.048	1	37.76	1.08	0.54	39.38	4.92
15	82	0.064	1	38.87	1.11	0.56	40.53	5.07
10	55	0.0933	1	40.88	1.17	0.58	42.64	5.33
1	5.5	0.57	1	73.73	2.11	1.05	76.89	9.61

Tangential speed [m/s]	Angular speed (SPS)	movement duration [s]	INOUT	feedback + wire data [Mbits]	Optical encoder [Mbits]	resolver raw [Mbits]	total [Mbits]	[Mbyte]
20	110	0.048	10	347.86	6162.11	6162.11	12672.08	1584.01
15	82	0.064	10	348.96	6181.64	6181.64	12712.24	1589.03
10	55	0.0933	10	350.98	6217.41	6217.41	12785.80	1598.22
1	5.5	0.57	10	383.83	6799.32	6799.32	13982.47	1747.81
20	110	0.048	10	347.86	9.94	4.97	362.77	45.35
15	82	0.064	10	348.96	9.97	4.99	363.92	45.49
10	55	0.0933	10	350.98	10.03	5.01	366.02	45.75
1	5.5	0.57	10	383.83	10.97	5.48	400.28	50.04

Operational mode  
 (OPS & RESOLVER processed)

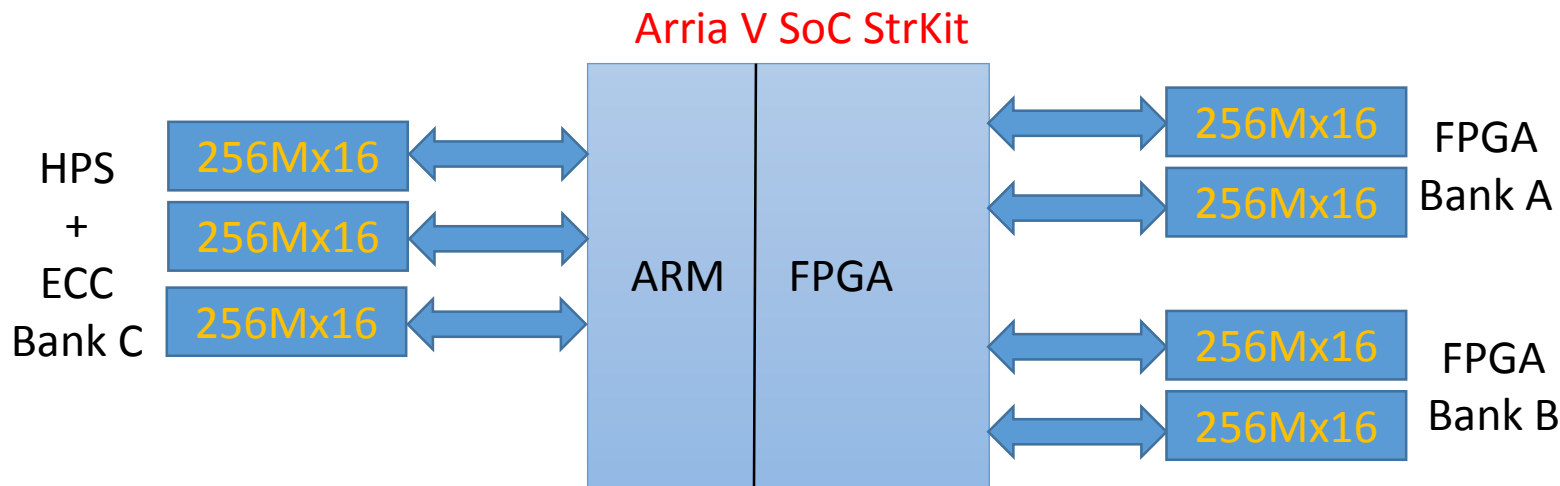
- 1) Only one measurement cycle (one IN, one OUT)
- 2) Data recorded continuously between in and out movement
- 3) Large data grow due to INOUT delay => Can we limit INOUT to a maximum of 1s and do multi-scans per cycle?
- 4) **Expert mode** => Motion data and raw encoders data storage  
Will be used until we have the optical position sensor digitalised in the VME
- 5) **Op Mode** => Motion data and processed encoders data storage



# External memory access organisation

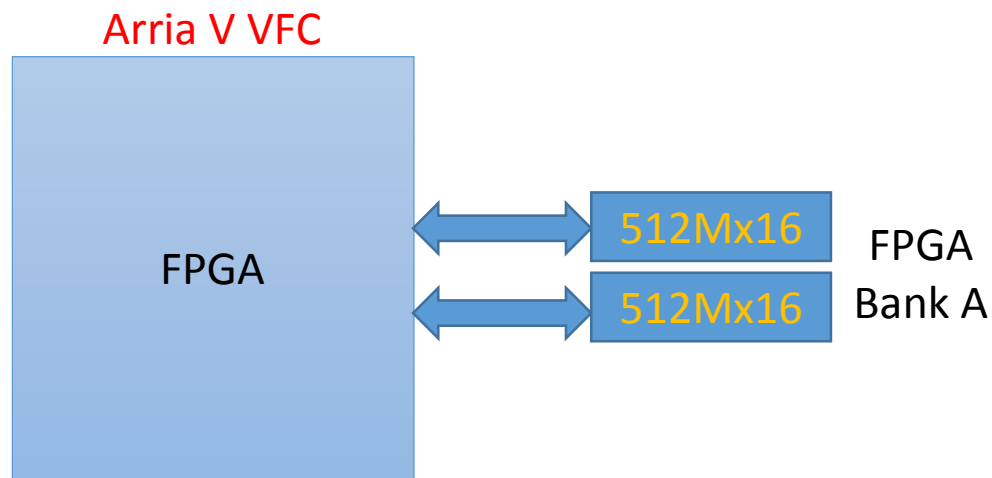
- Digital architecture
- 1) FPGA internal resources
  - 2) Board interconnects
  - 3) External memory

Is the external memories connections could be a limitation?



Theoretical transfer: 12.8 Gbit/s  
 Implemented: 6.4 Gbit/s

Bank A: Fast ADC => 4x20MHzx16bits => 1.28 Gbit/s  
Bank B: Other ADC => 70\*16k\*32bits => 36.12 Mbits/s  
Bank C: ARM dualcore ram memory



Theoretical transfer: 6.4 Gbit/s  
 Implemented?: 3.2 Gbit/s

Bank A:

- 4x20MHzx16bits => 1.28 Gbit/s
- 70\*16k\*32bits => 36.12 Mbits/s
- Softcore (Nios) RAM?

Efficiency reduction due to arbiter + R/W and Random access:

- write to multiple memory location (20M & 16K): ? 25%

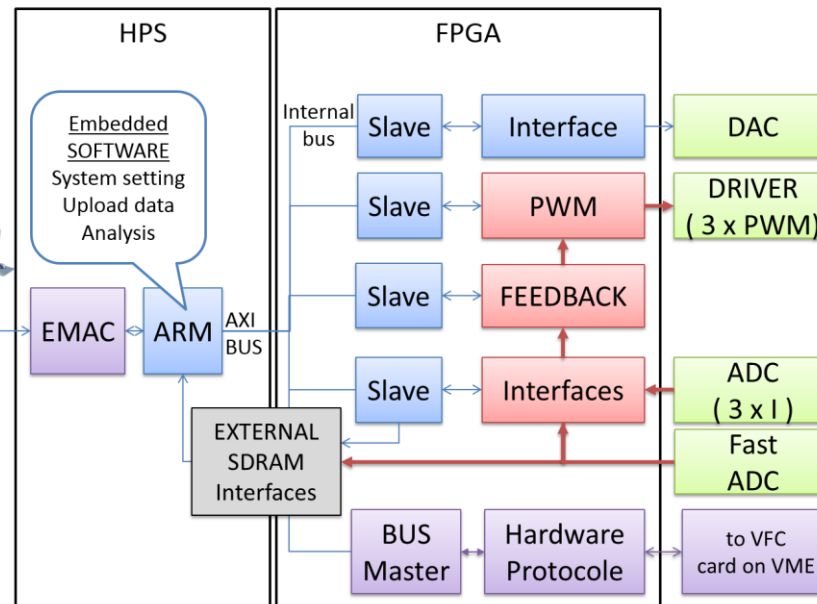
3.2 Gbit/s / 4 => 0.8 Gbit/s => **Potential limitation for OPS + Resolver**

Is the FPGA selection will determine the system and Firmware testability?

- Simulation level:  
Most of the final code written in VHDL => Verification (VHDL, SystemVerilog, Simulink) on simulator for all options.
- Component level:  
JTAG (and JTAG link) probing will be available on all options in the lab and on fields prototypes.
- System level: lab debugging and field validations:  
Same method could be used (TCP/IP access to large internal data with expert application), transfer rate will vary.

**Expert mode**  
detailed observation of large sets of data

**Operational mode**  
Possible with fast TCP/IP using ARM cores



## Hardware link to VME

- Transport and integrity of data
- Transparent links between SoC domain
- memory mapping between the FPGAs
- JTAG link between FPGAs

Implementation  
Sep 2016-January 2017

=> THIS COULD BE REUSED FOR OTHER PROJECTS



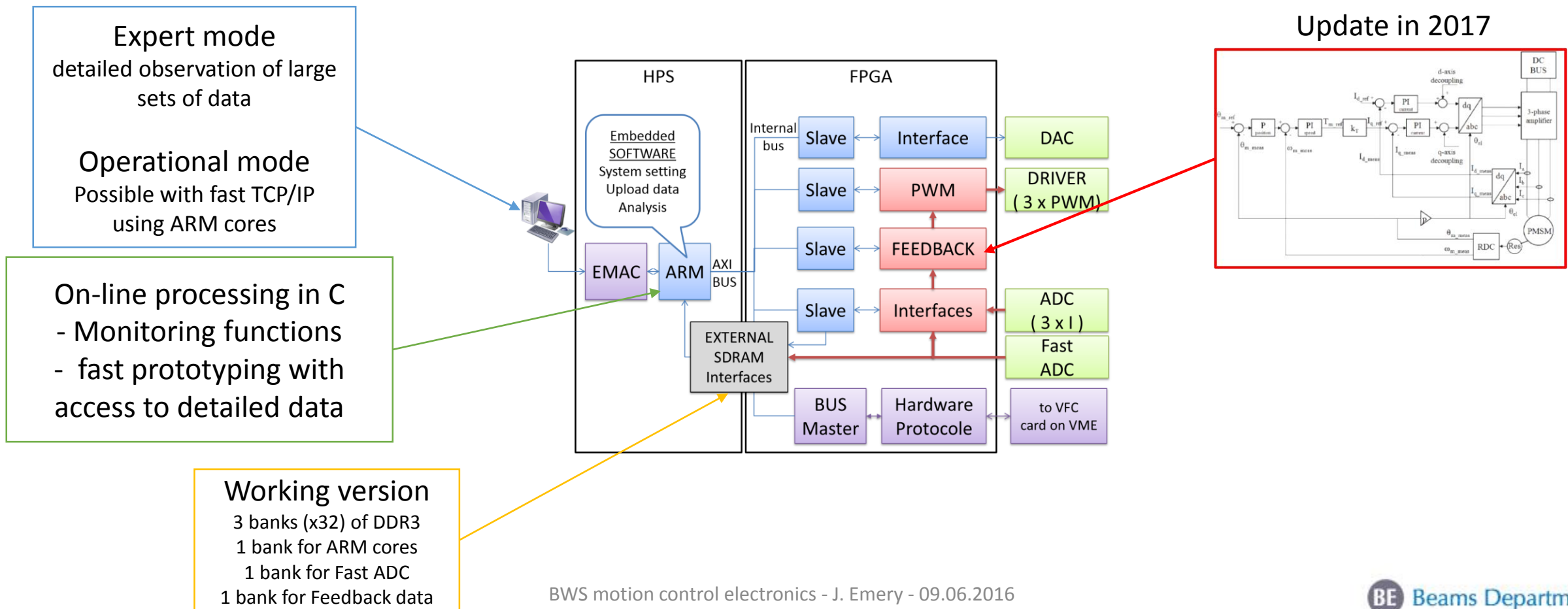
# Code reuse between options?

- 1) Testability
- 2) Code reuse
- 2) Methodology

Is the FPGA selection will determine code reusability?

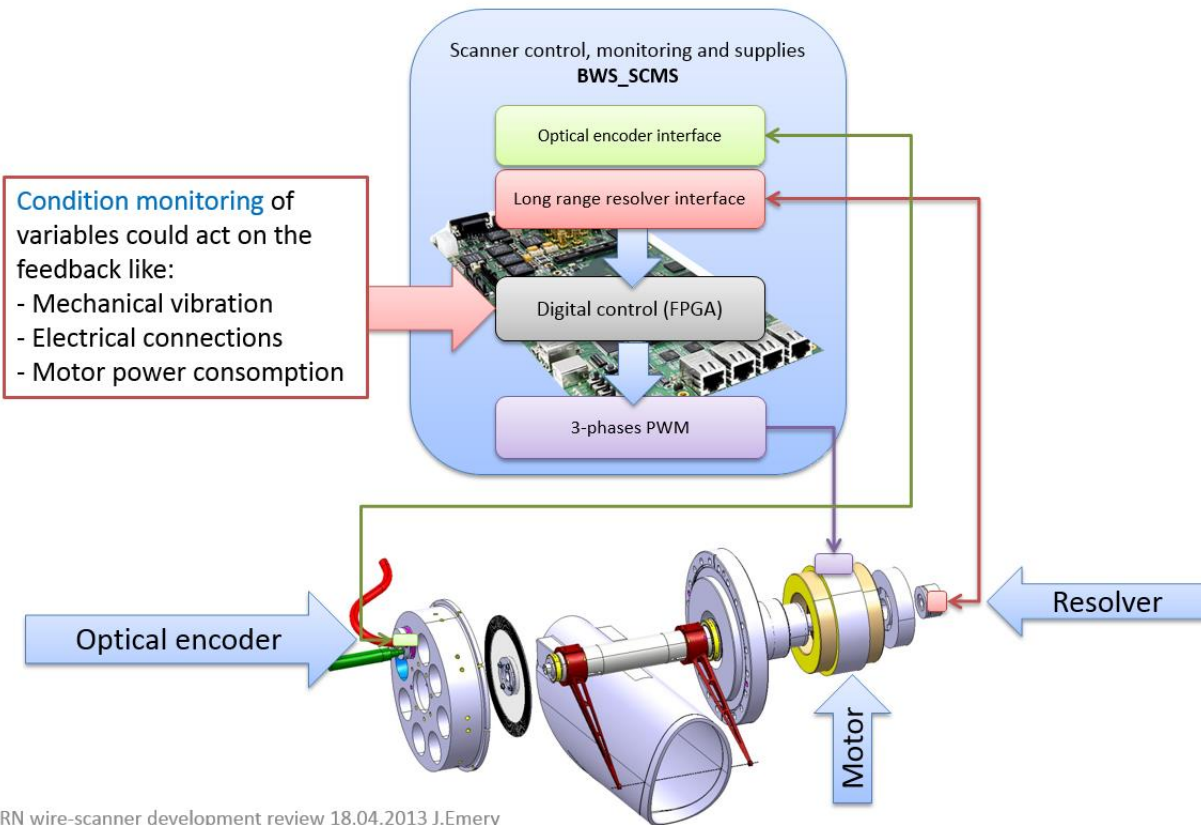
- **Yes partially**, because we have already large working code (option 2 and 3)
- **No**, because all main functionalities will be in VHDL (reusable for all options)
- **Not really**, not much reuse of existing VFC code for all options (no need of VME, BST, etc ...)

Ok for 2016  
Update in 2017



# Condition monitoring: Survey all system variables

## Future Actuator and Control unit



- Condition monitoring and decision in real time.
- To react to unexpected even during a movement
- Large number of parameters to take into account
- Target reaction time within one feedback period 62.5us:  
12k instructions Nios  
62k instructions ARM



# Challenges of the ID processing

Design process

- 1) Testability
- 2) Code reuse
- 2) Methodology

- **Position, speed and torque precise controlling**  
fully written in VHDL  
first version operational for 2016  
Second version foreseen in 2017 (improve precision and flexibility)
- **On-line data processing and fault detection**  
Will be used for survey system conditions (mechanical, electrical, controls)  
Prototyping foreseen in Simulink/MatLab and C in the drive  
Implemented in VHDL for critical ones, leave
- **Special functionalities**  
Processing/Area to foresee for future functionalities:  
Tails measurements procedure, Delayed multi-scans (reconstruct small beams),  
vibrations on-line compensation.

=> I will start detailed work on this subject in September 2016 (MSE)

# Additional slides



# Design related criteria

## Design process

- 1) Testability
- 2) Code reuse
- 2) Methodology

Is the FPGA selection will drive design methodology?

- Yes, hardware processors can allow on-line prototype of processing to run in real time.

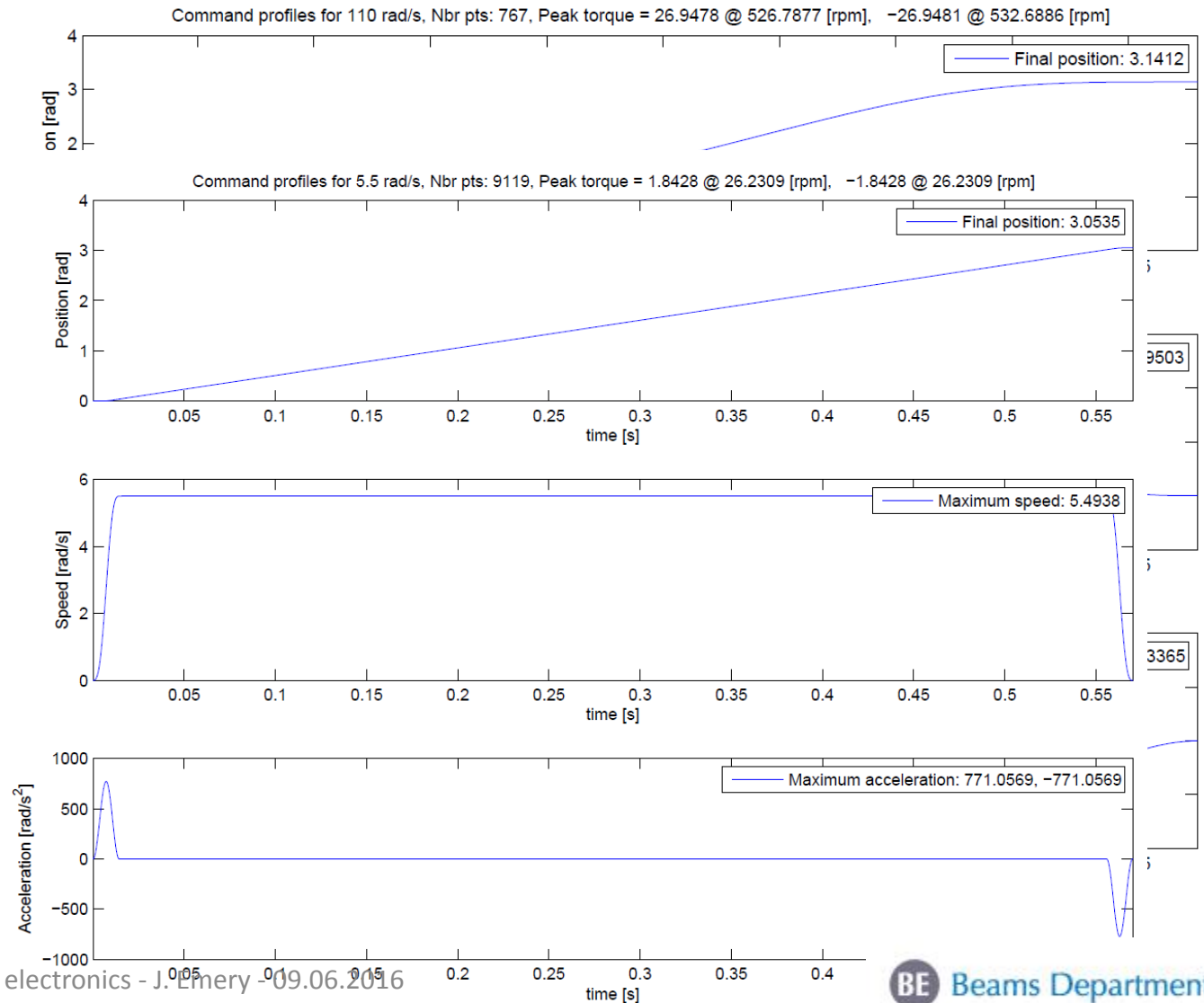
Methodology:

- 1) Prototyping data processing in MatLab on existing raw data
- 2) Simulink modelling of the algorithms and test on Dspace
- 3) Implementation in C => Fast to write in C and test on real system  
Needs fast processing units  
Needs memory access to data being recorded
- 4) Final version must be in VHDL:
  - Parallel processing independent to any OS or other running tasks
  - Powerful verification in simulation
  - Powerful tools to do verification on the FPGA
  - But: Long development & verification time



# Profiles: online calculation vs pre-calculated

- Operate at different top speed  
20 [m/s] -> 48 [ms] (767 pts)  
1 [m/s] -> 570 [ms] (9119 pts)
  - Today pre-calculated into 3 tables included in the FPGA as ROM (safe).
  - Needs 3 tables for each preset
  - Alternative:  
Online calculation based on system properties. (3 parameters to play with: Jmax, duration, ratio acc/cst speed).
- Optimised iterative Algorithm to be written in C and monitored by FPGA



# Feedback implementation in VHDL

