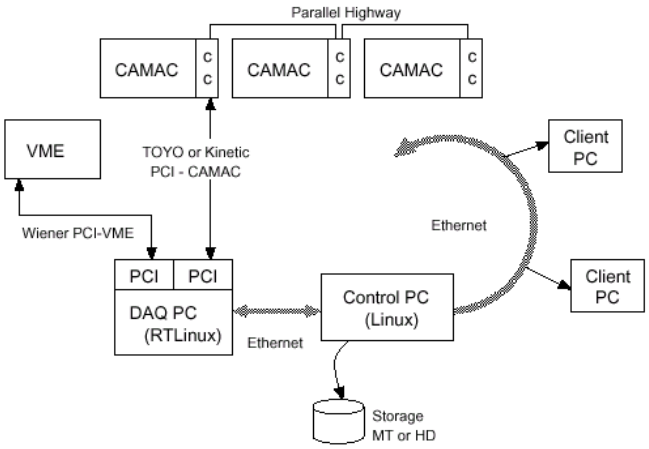
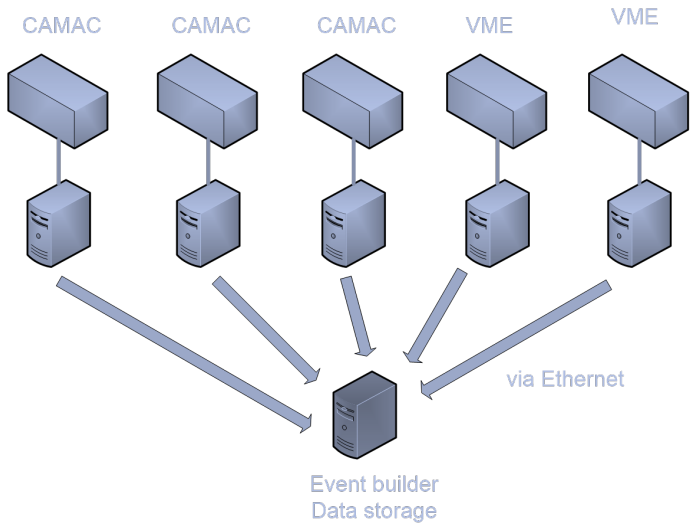
MOCO and MPV

Hidetada Baba
RIKEN Nishina Center

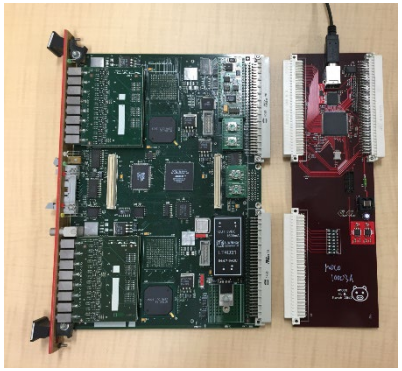
parallelized VME



～2006, RIPS DAQ
1 CPU, many crates



2007～now
1 controller per crate

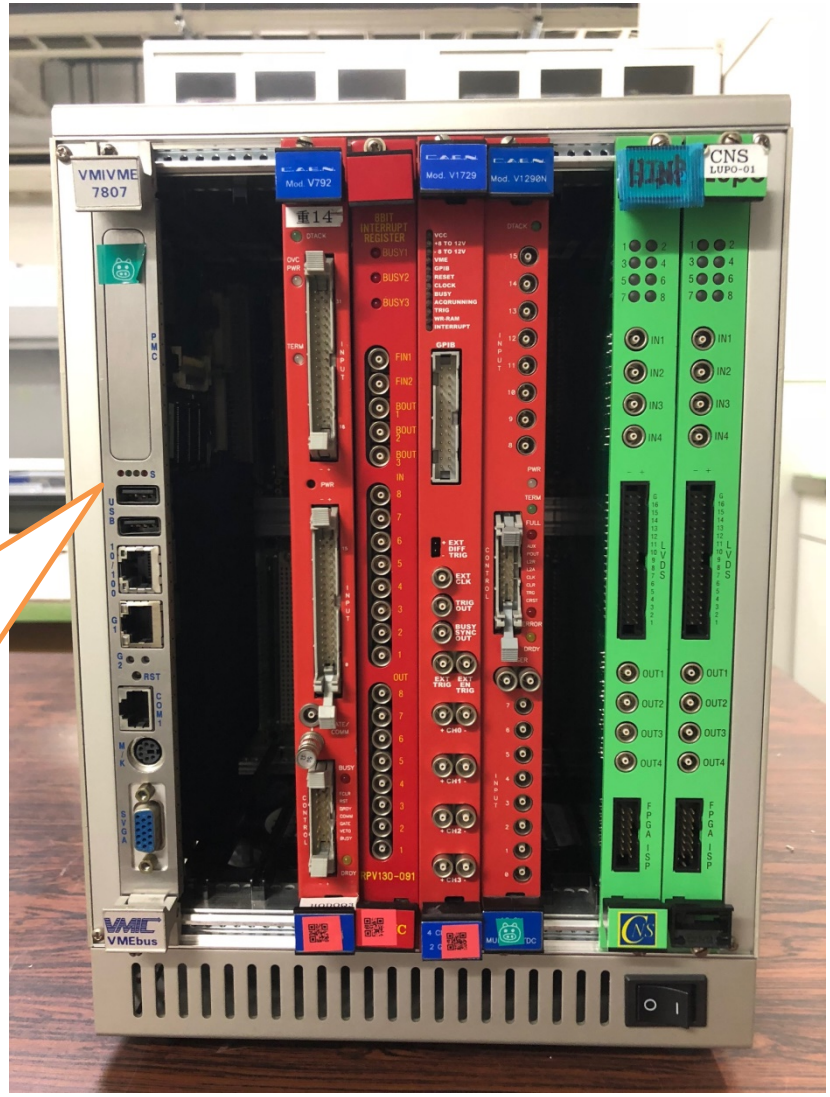


1 controller per module

VME

Controller =
CPU Board
$4,000

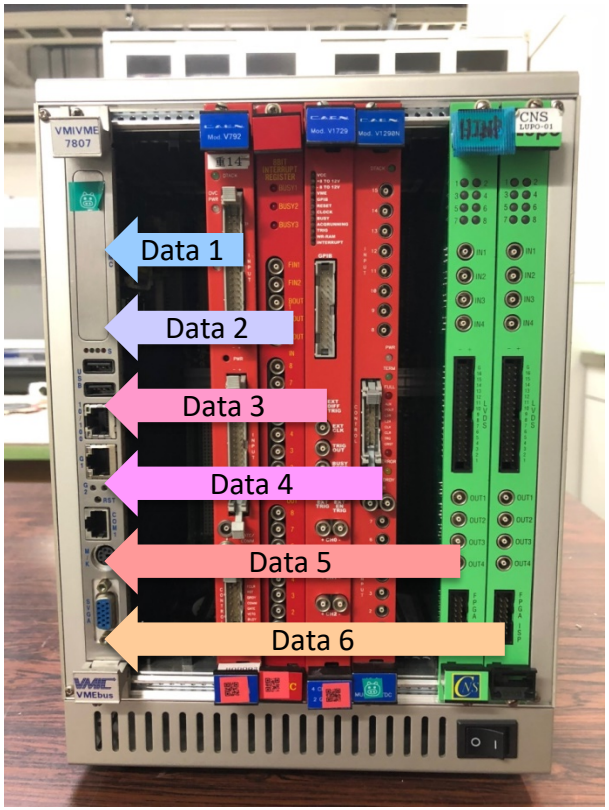
Linux OS



VME Readout Time (Dead Time)

Latency by OS	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6

～20us



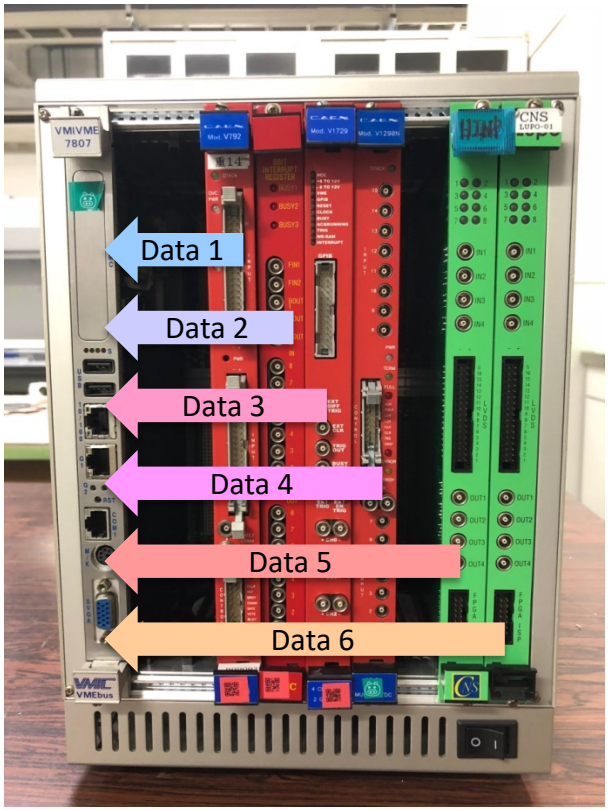
Data rate = 160Mbps

0.2 us x N word

VME Readout Time (Dead Time)

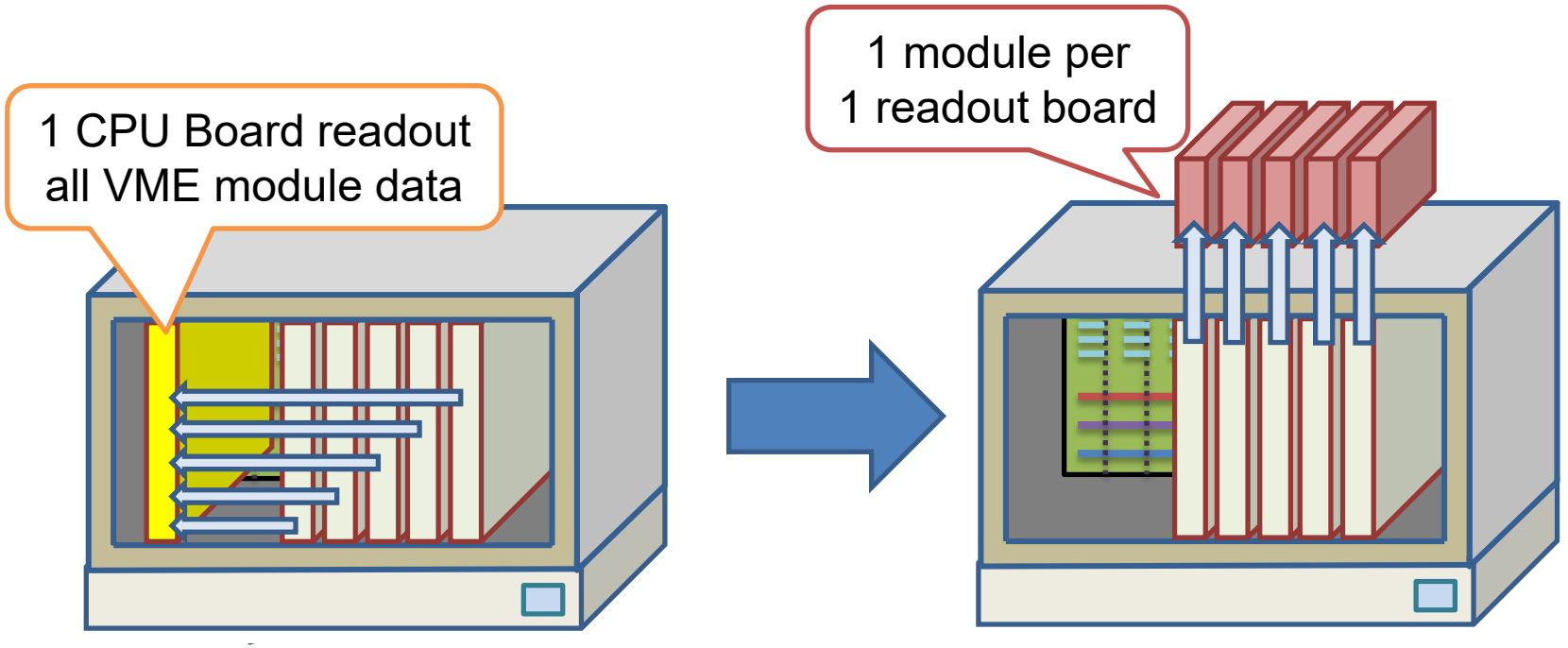
~20us

0.2us x (N_1+N_2+...) word + overhead (typ. **~100us**)



Data rate = 160Mbps

0.2 us x N word



Readout
～ 100us
(OS Latency ～20us)

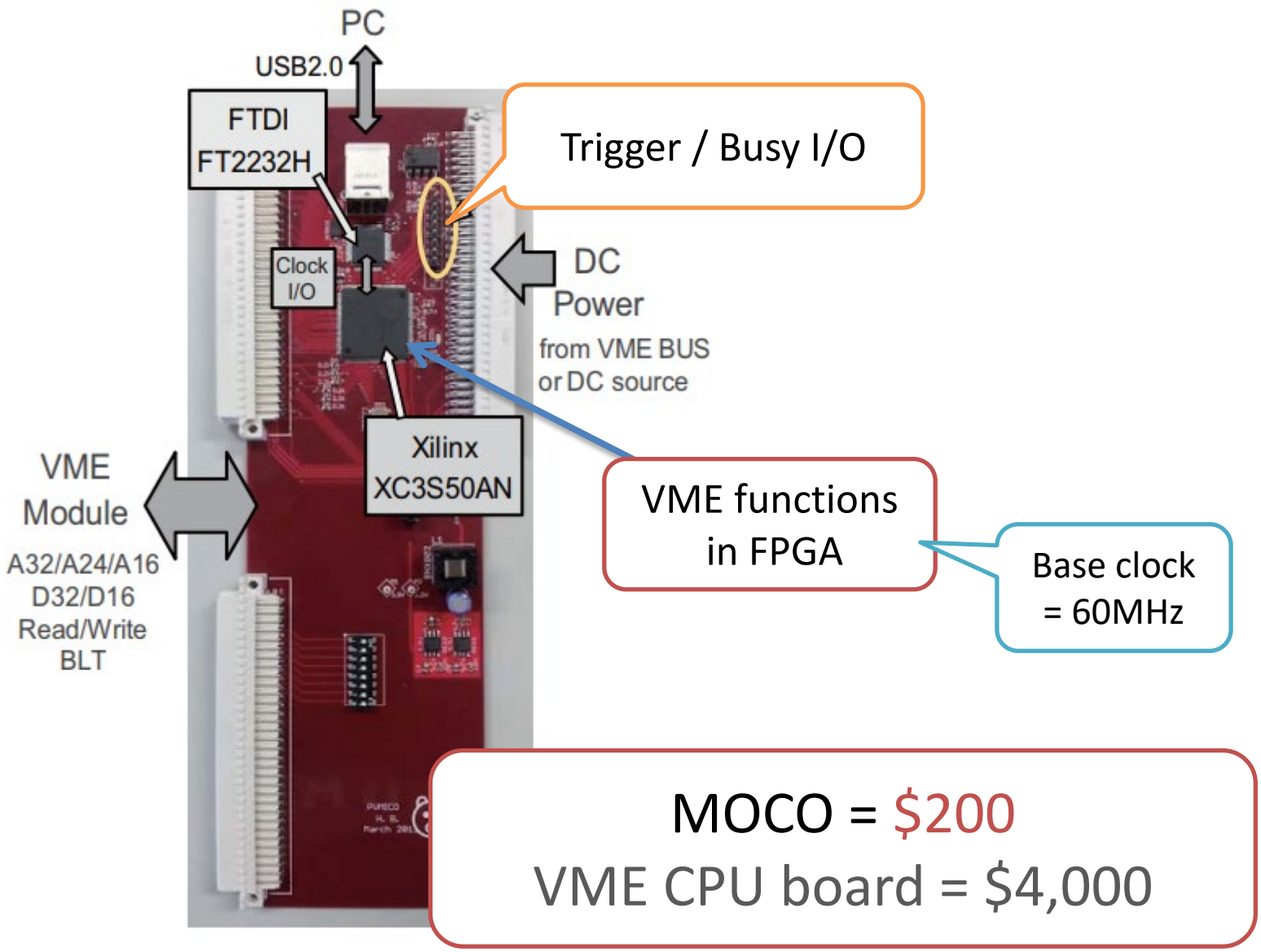
Parallelize!!

Readout
<< 20us
(no OS latency)

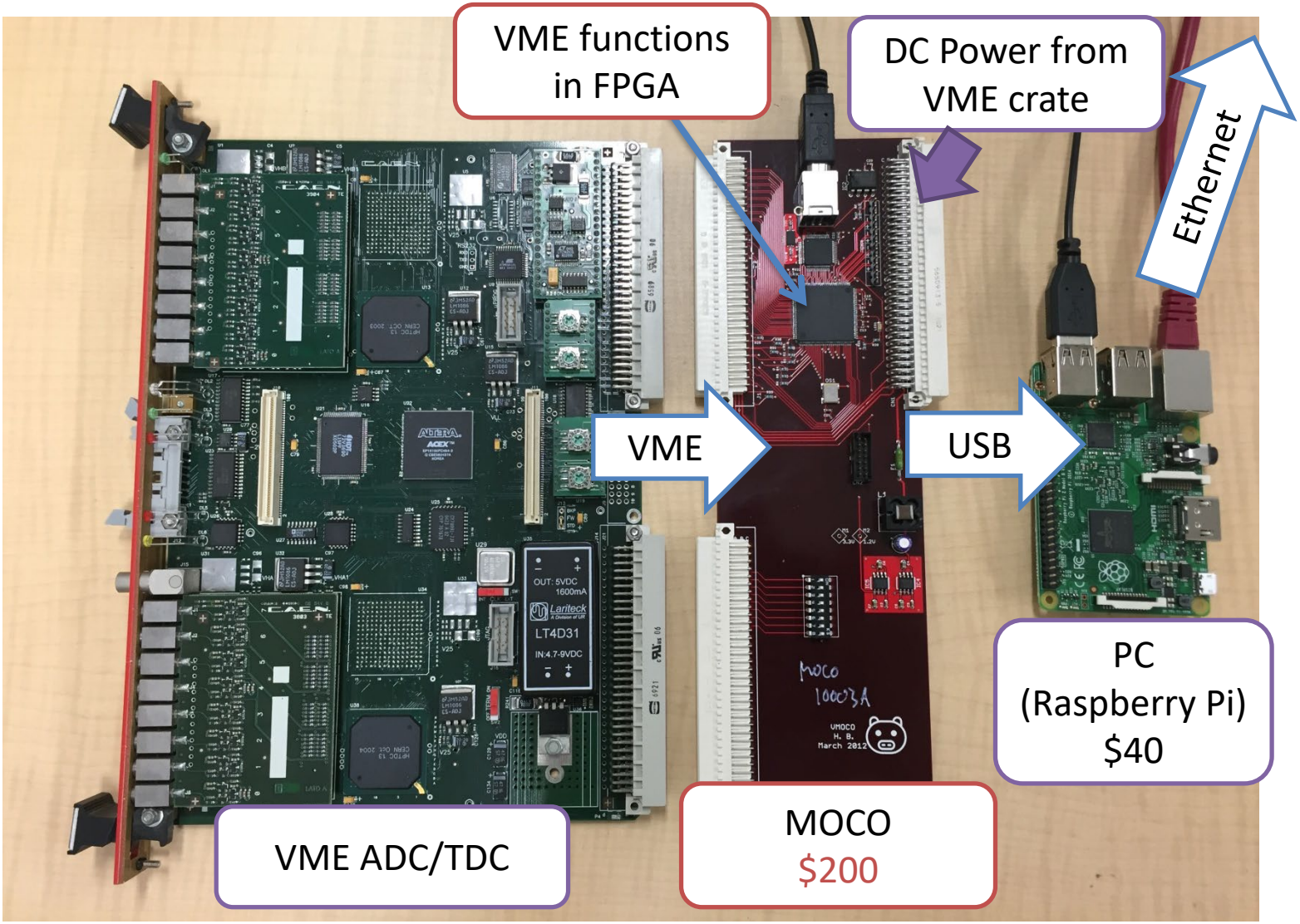
Module Parallel

Solution = FPGA based VME Controller

2011～



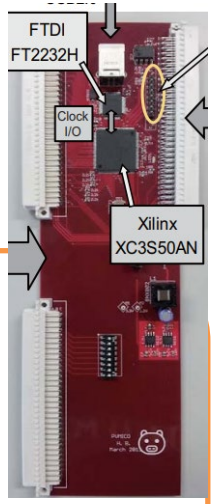
MOCO (Mountable Controller) 2011～



Performance of MOCO

	Standard VME	MOCO
Controller price	$4,000 / crate	$200 / module
Interrupt Latency	> 20 us	0.02 us (1 clock cycle)
Data rate per controller	160Mbps (0.2us/word)	160Mbps (0.2us/word)
Data rate per crate	160Mbps	160Mbps x N (N = Num. of MOCO)

Extension of VME



High throughput

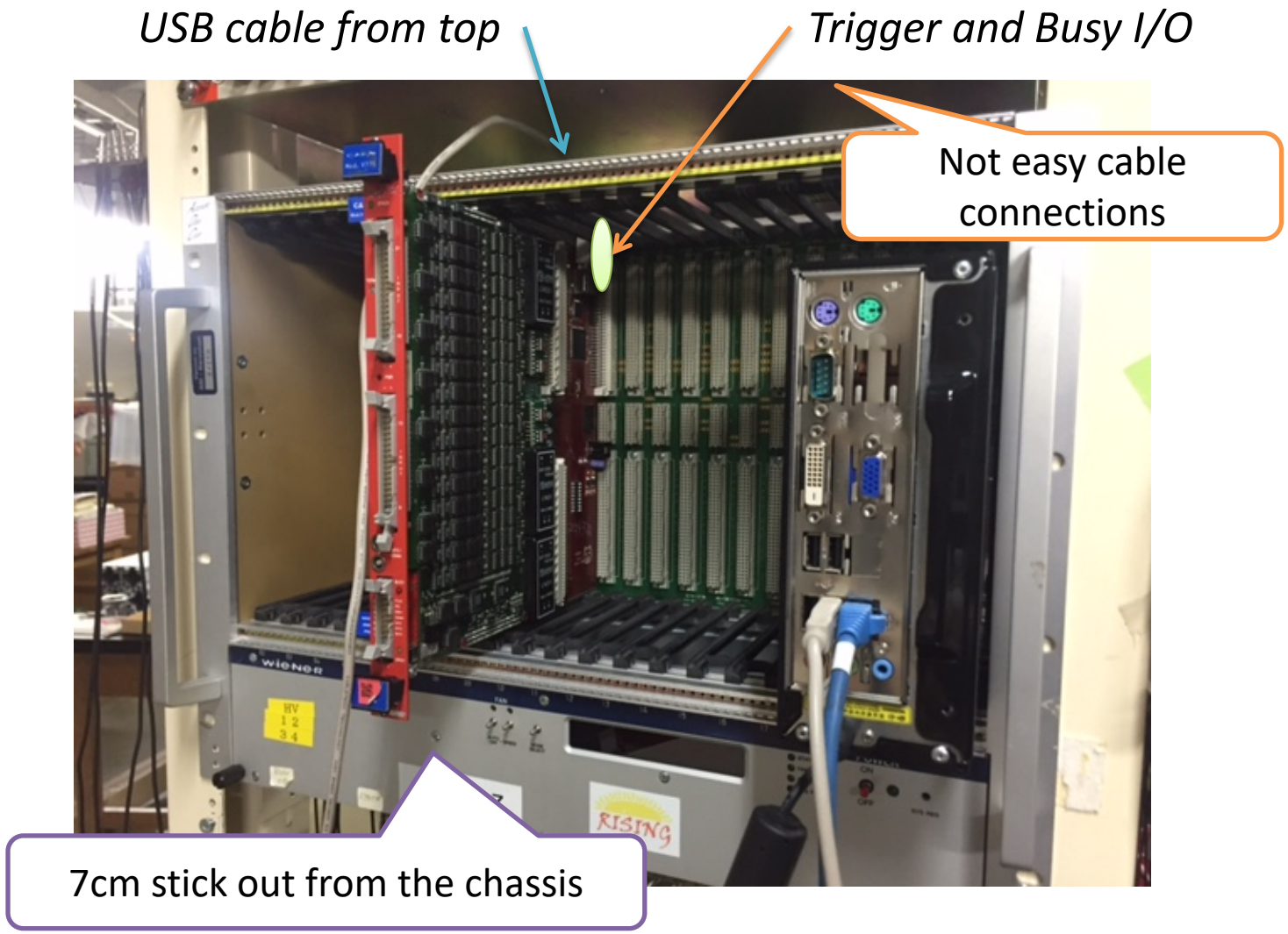
+ Cost efficient

+ Compatible with legacy VME

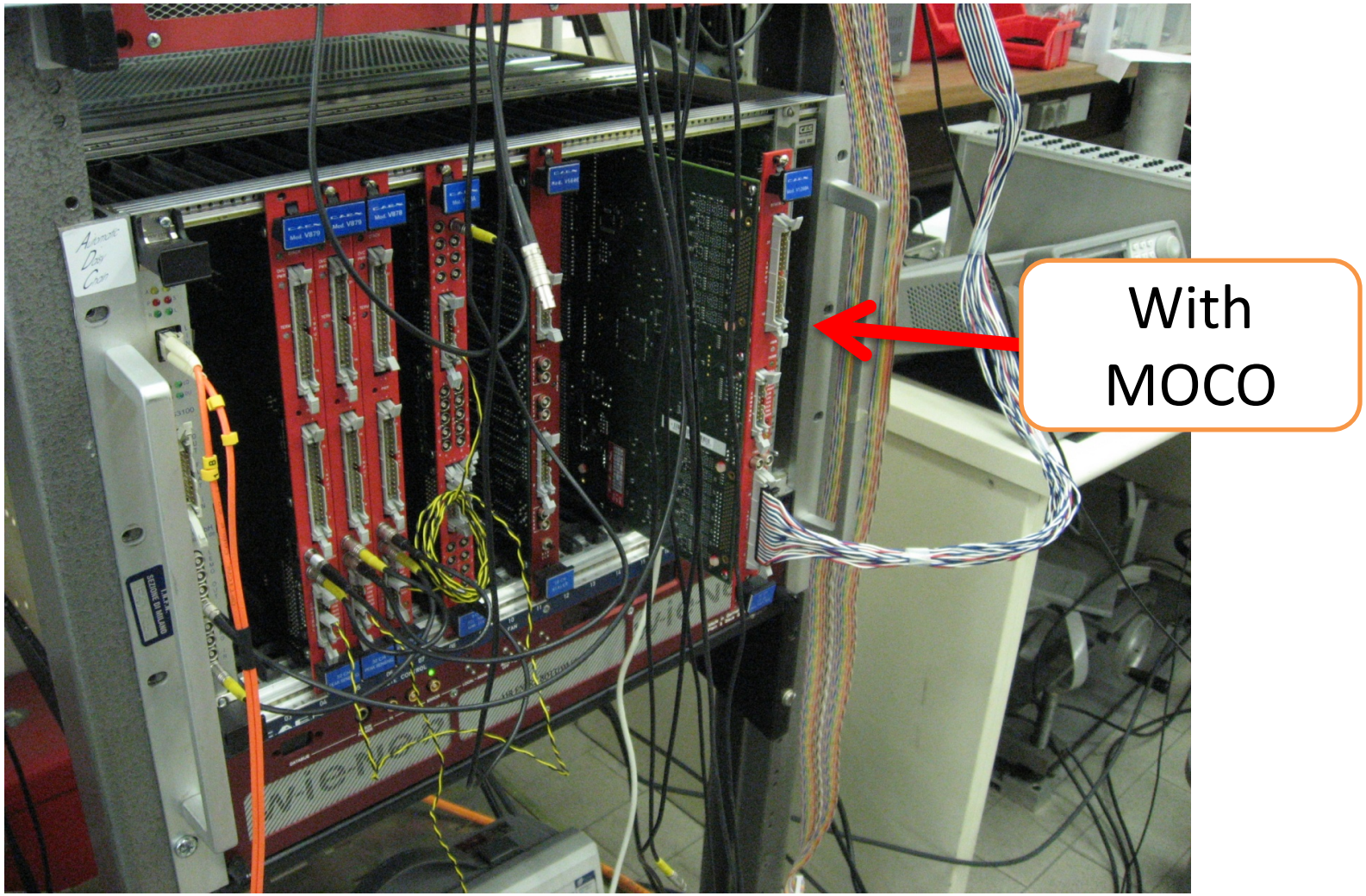
~~+ Easy-to-Use~~

~~+ Robust~~

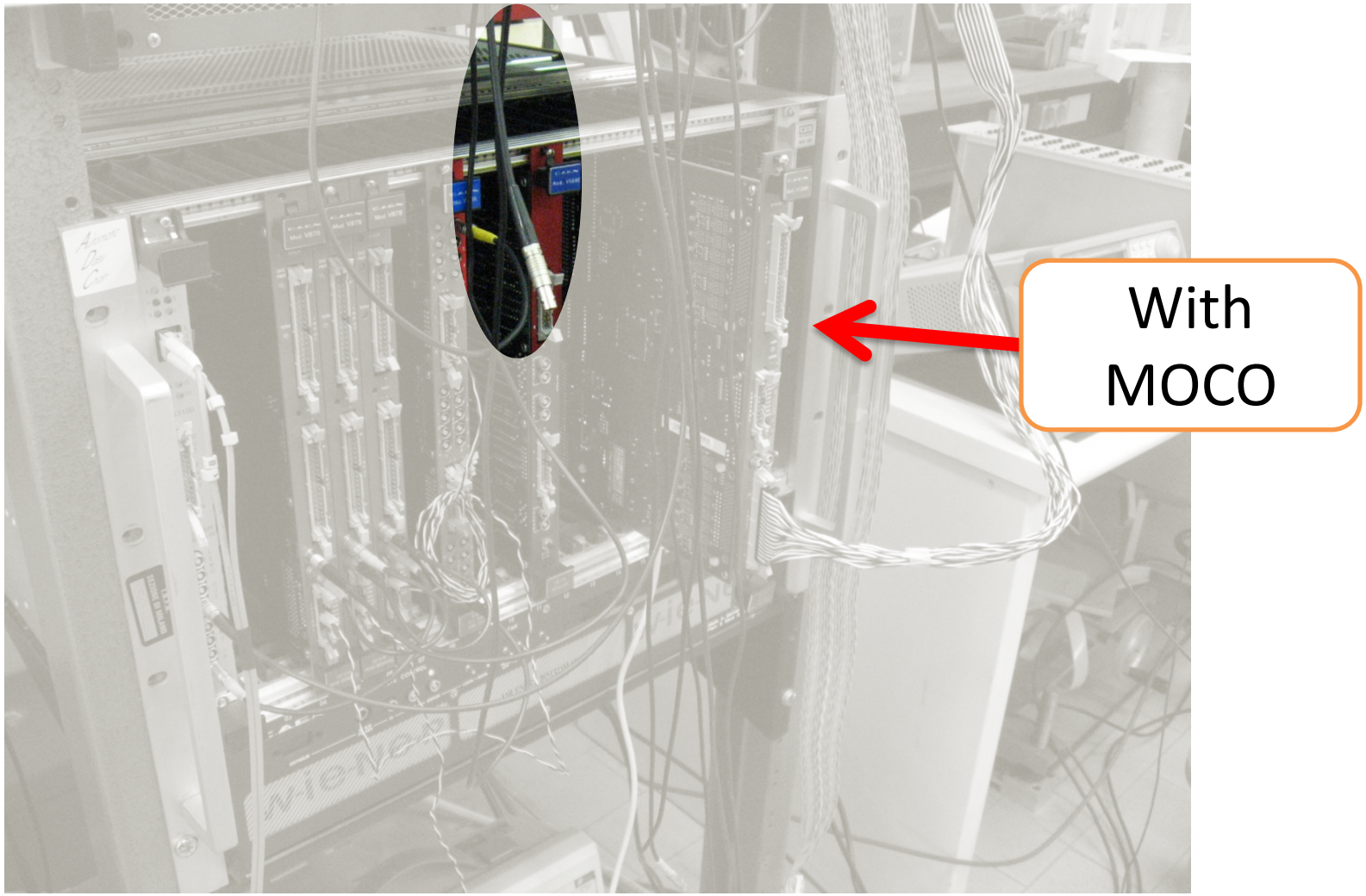
Standard VME with MOCO



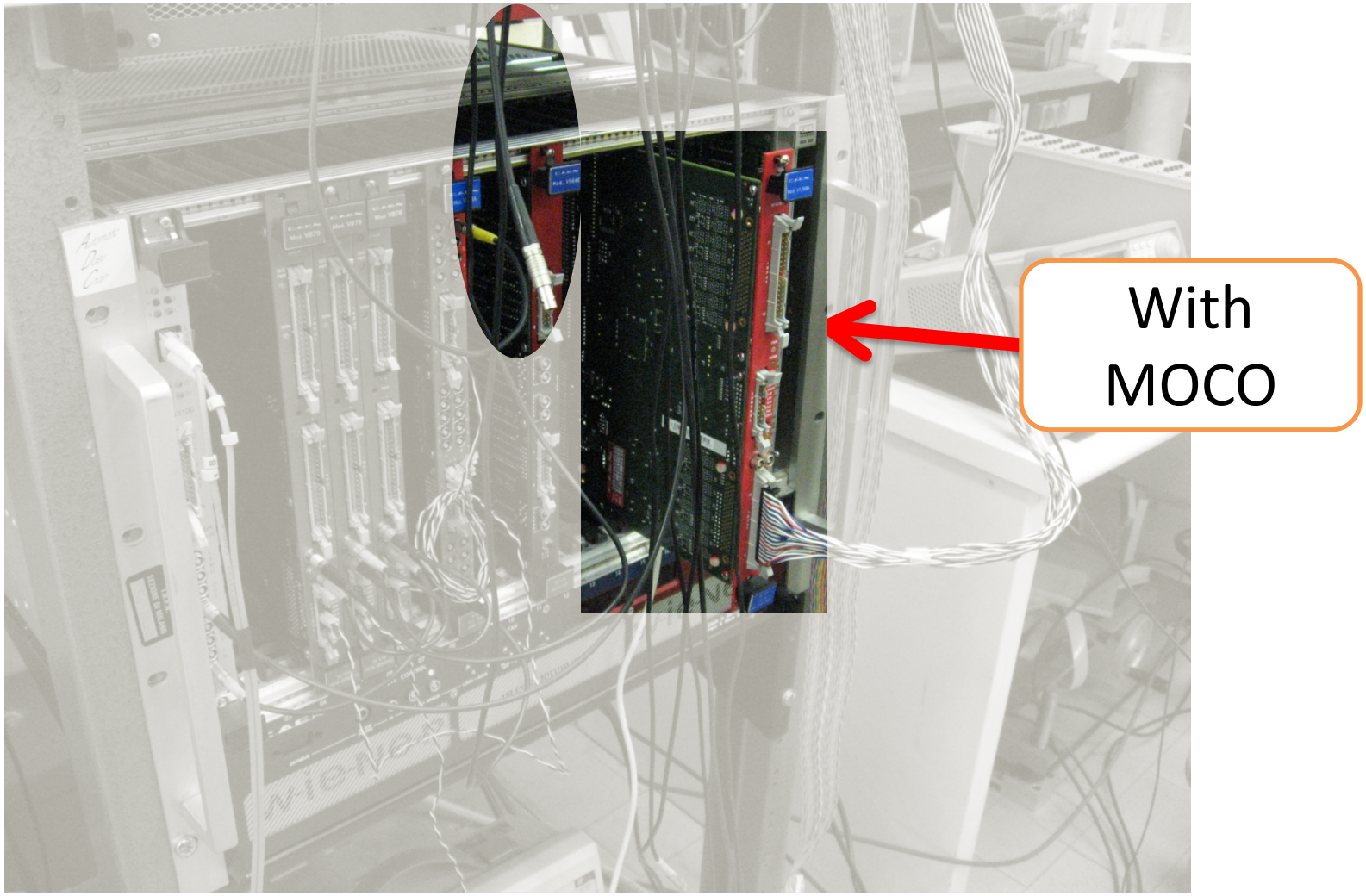
Standard VME with MOCO



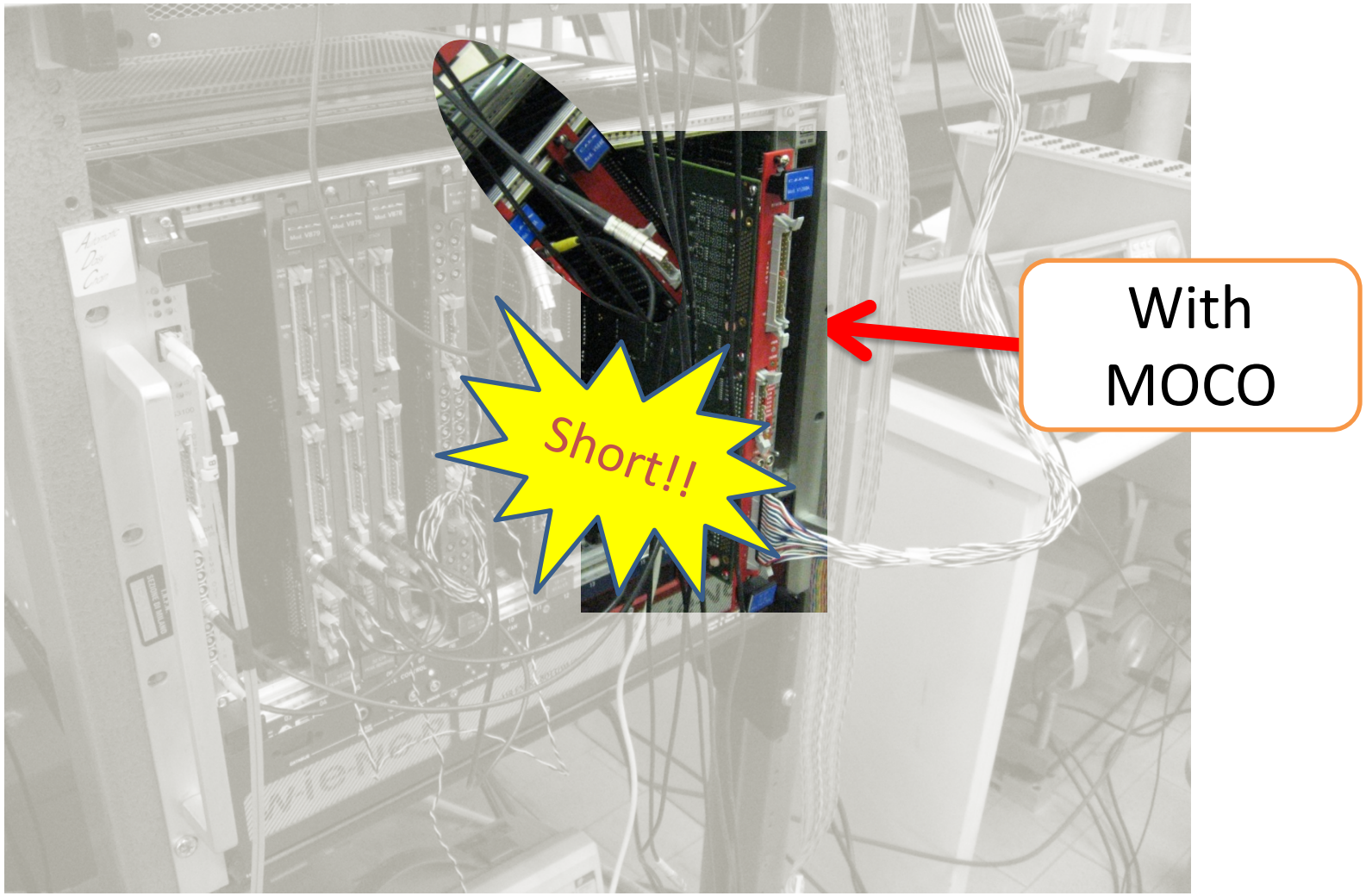
Standard VME with MOCO



Standard VME with MOCO



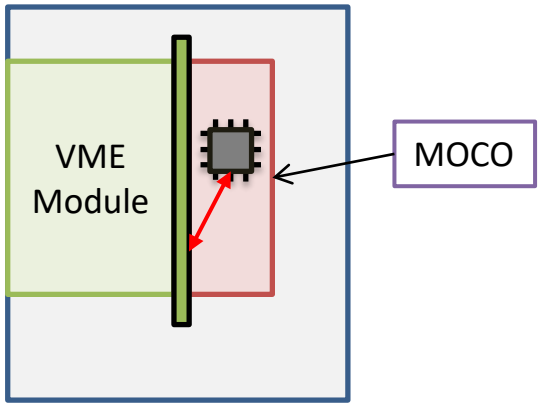
Standard VME with MOCO



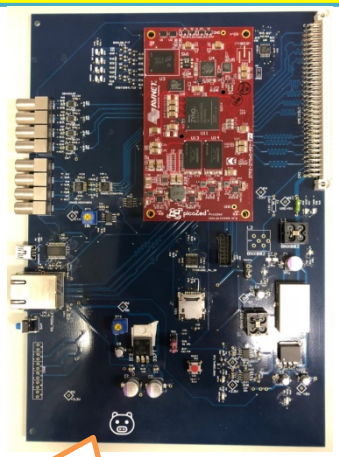
New development

- MPV
  - MOCO with Parallelized VME

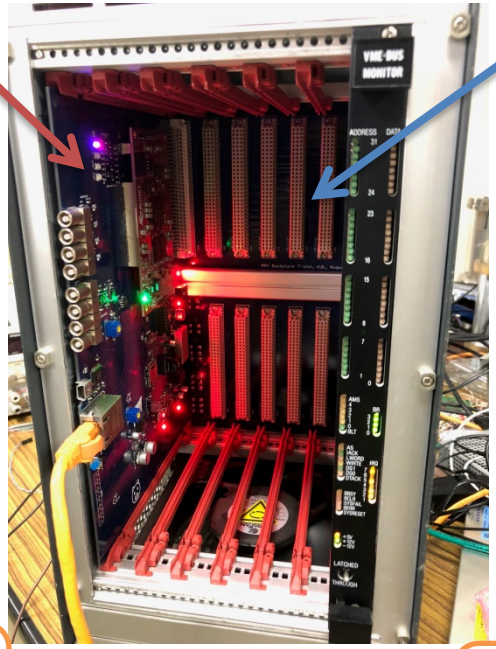
Side view



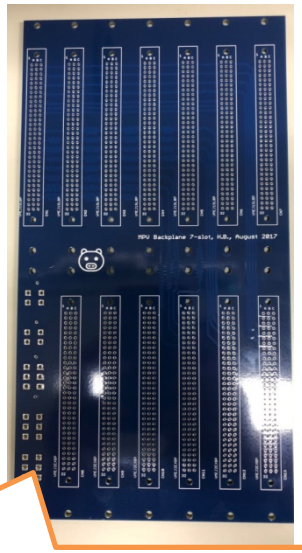
MPV Controller



Merge data from MOCO, and trigger/busy management

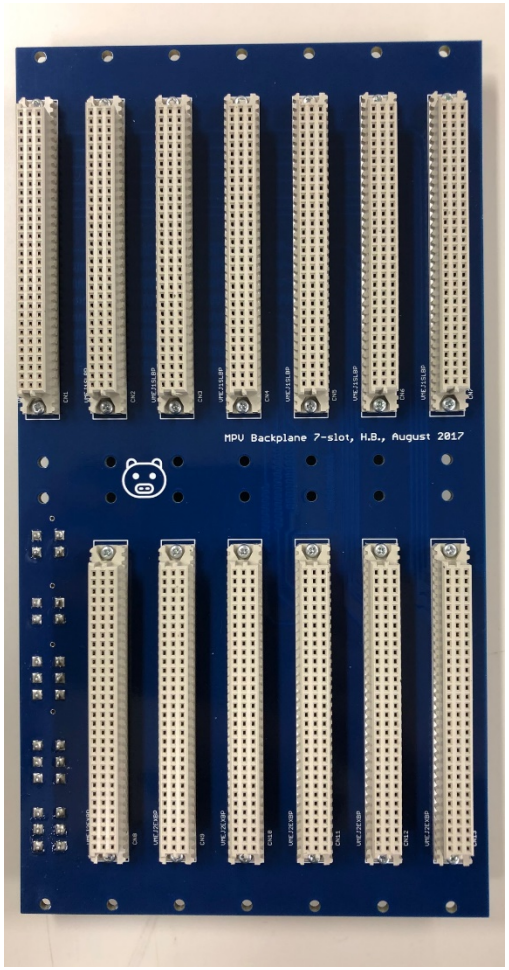


MPV Backplane



Communication lines between controller and MOCO

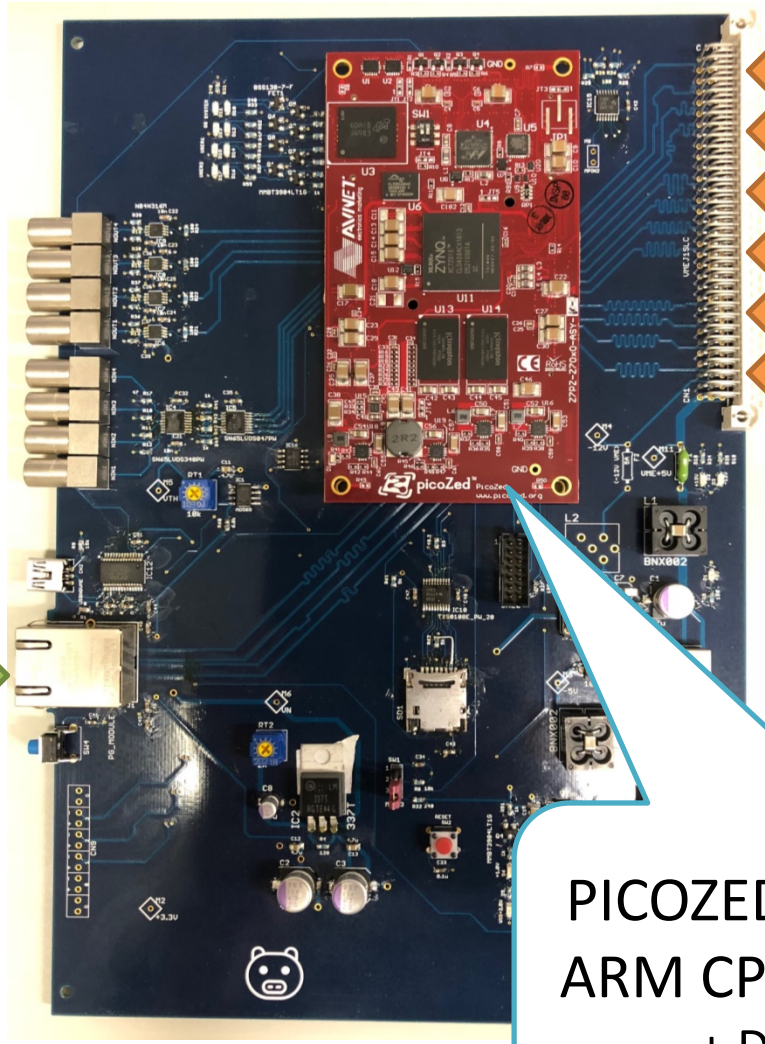
Backplane



Dimension is compatible with Schroff VME backplane

MPV Controller

LVDS I/O (4 x 6 Lines)



- MOCO 1
- MOCO 2
- MOCO 3
- MOCO 4
- MOCO 5
- MOCO 6

NIM I/O
Trigger,
Busy, etc

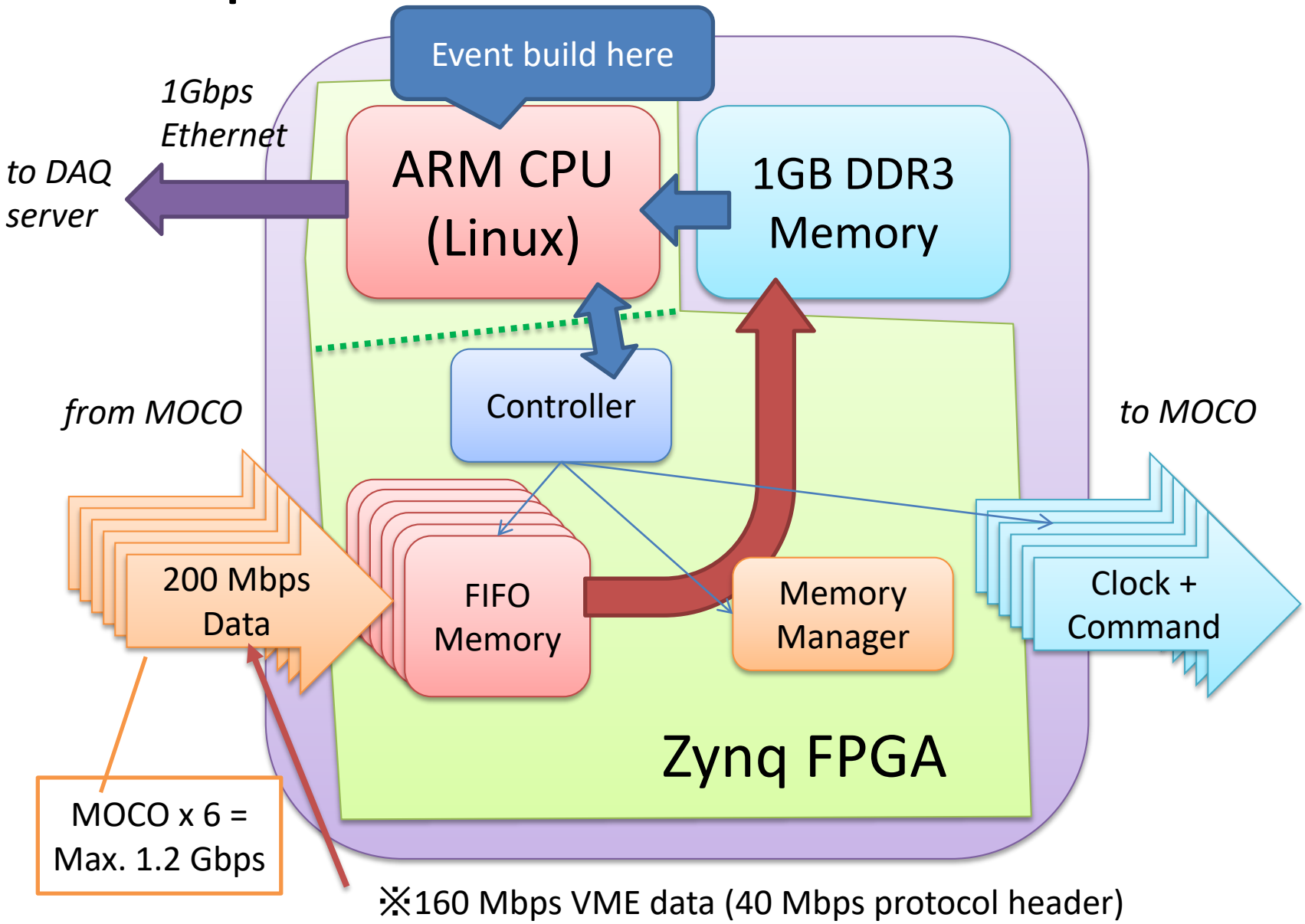
1Gbps
Ethernet

Clock, Trigger, Busy, Data, Command are communicated through the Backplane

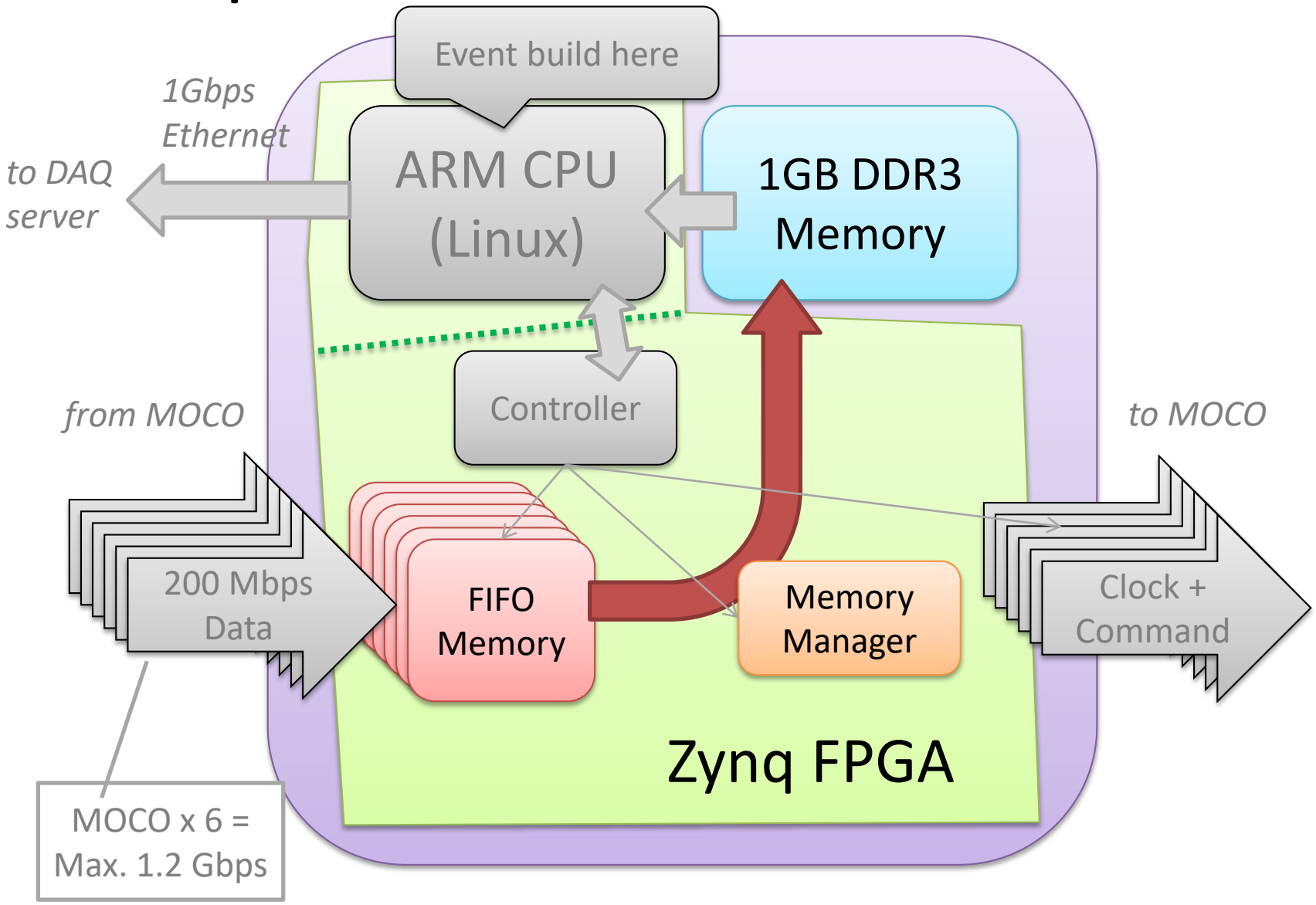
easy-to-use

PICOZED 7010 (Xilinx Zynq)
ARM CPU (Linux OS)+ FPGA
+ DDR, eMMC, etc…

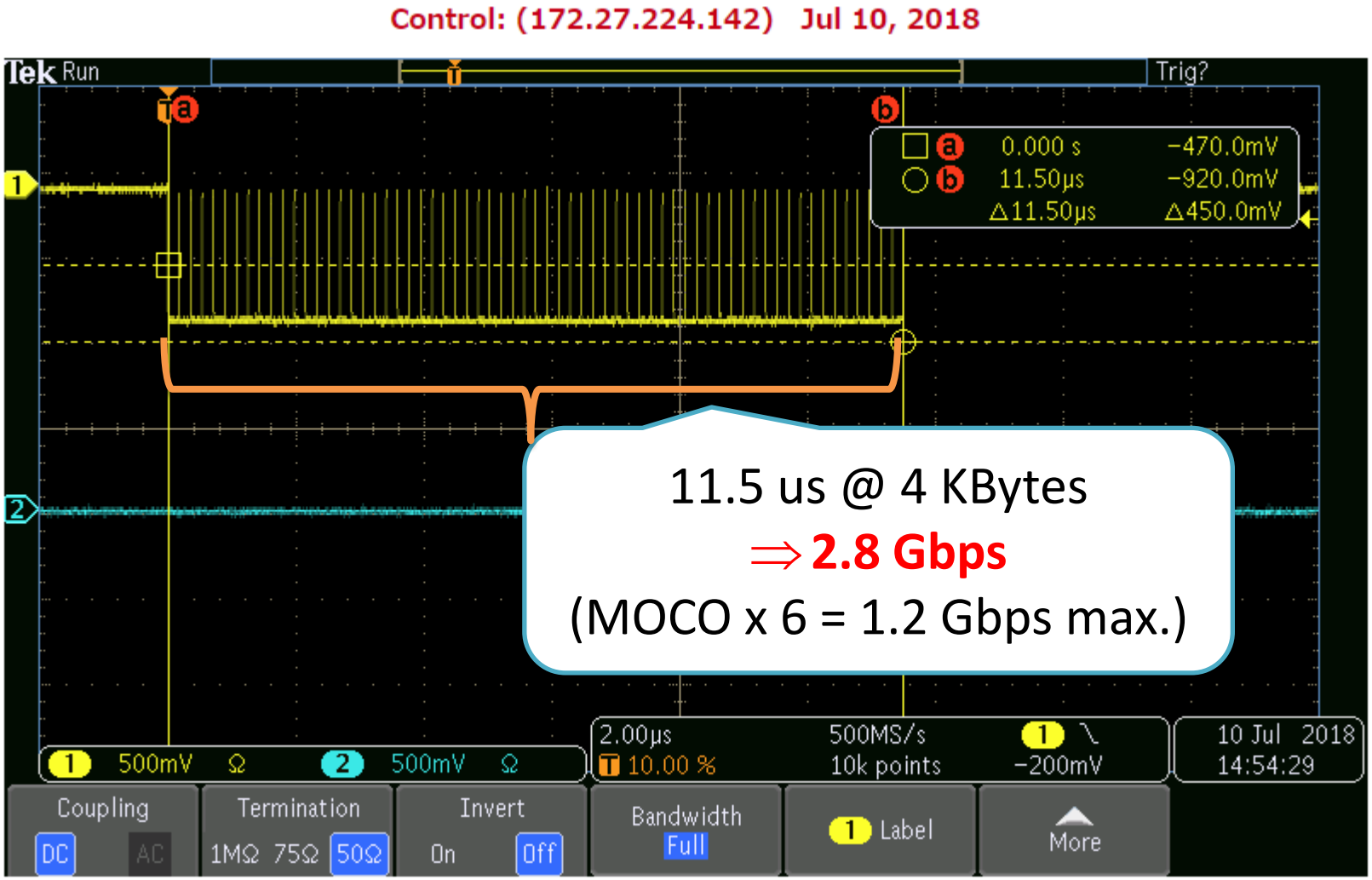
Data path in MPV Controller



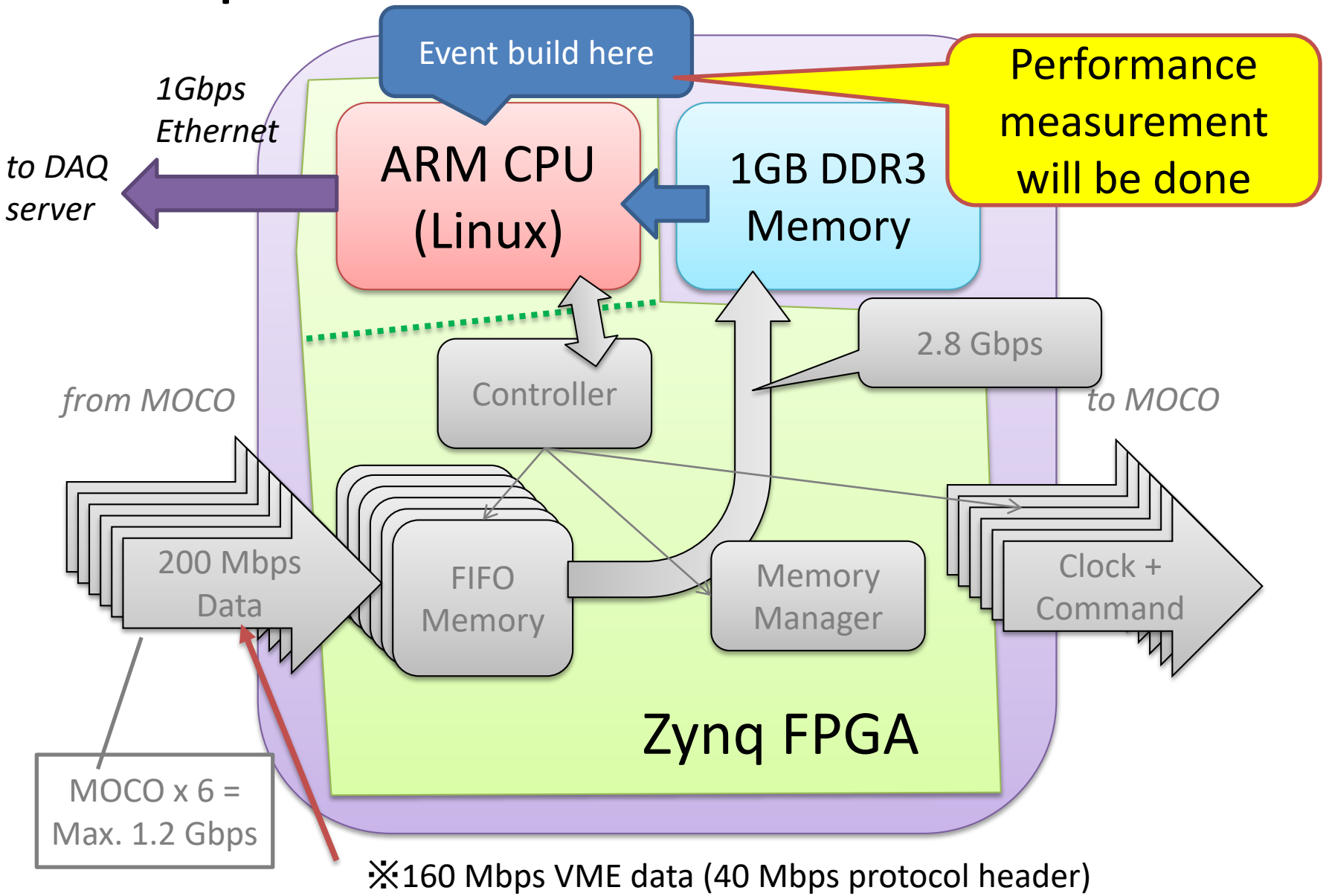
Data path in MPV Controller



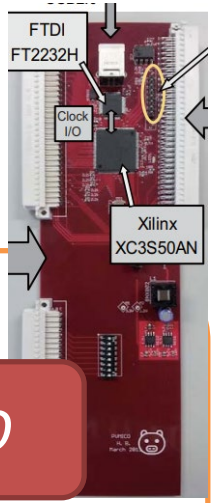
FPGA -> DDR for Linux



Data path in MPV Controller



Conclusion



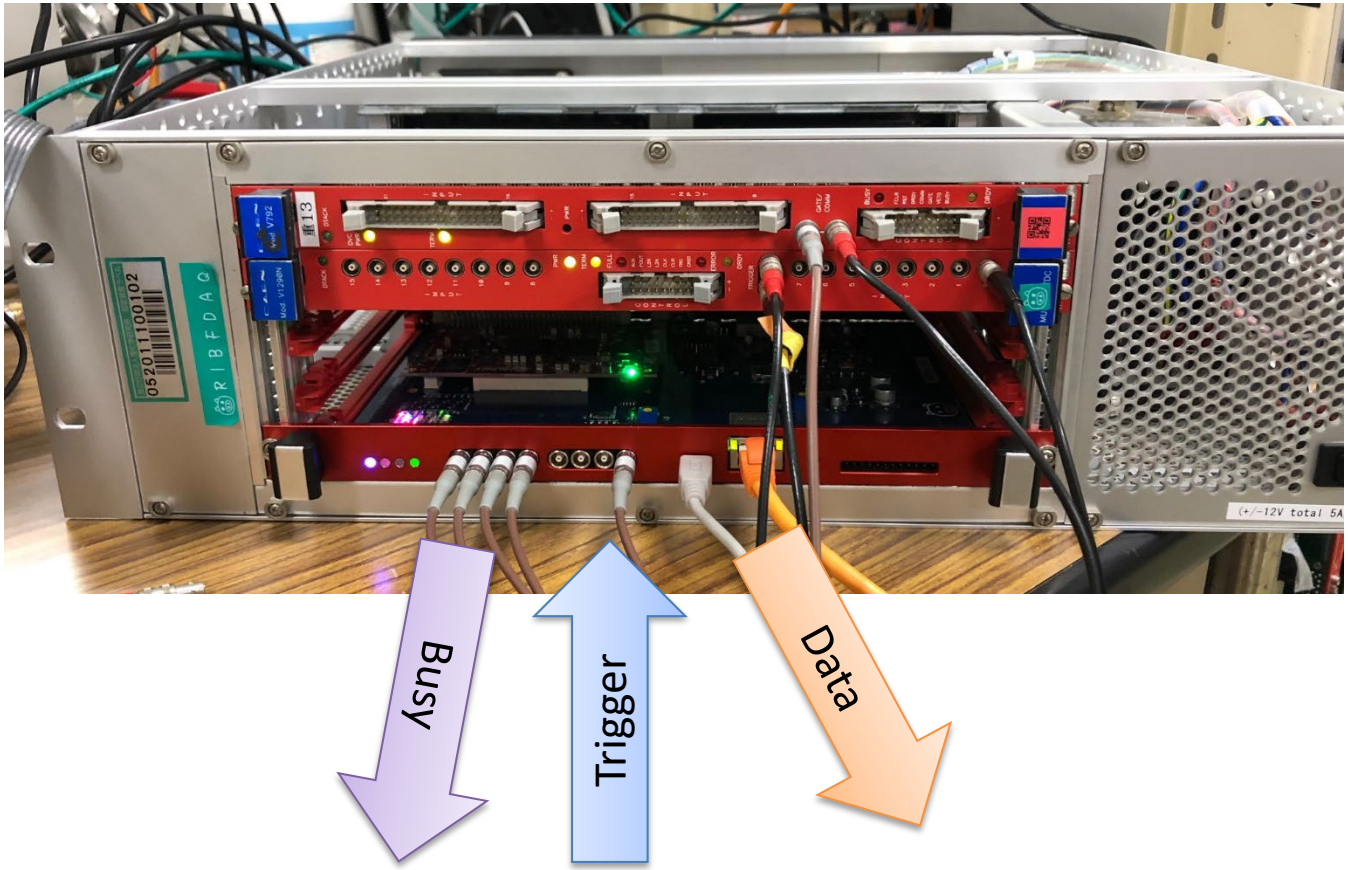
High throughput

MOCO

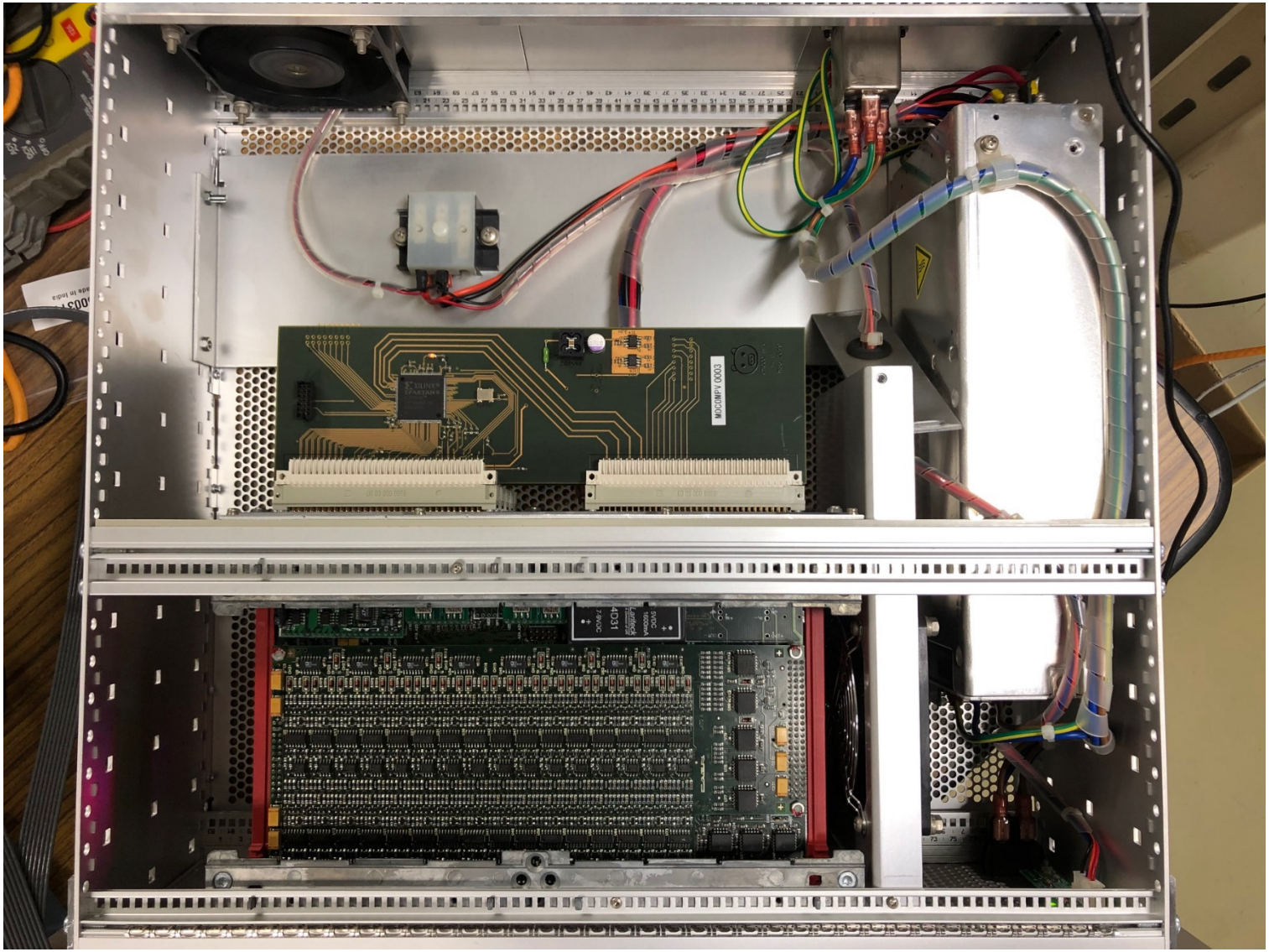
+ Cost efficient
+ Compatible with legacy VME
+ Easy-to-Use
+ Robust

MPV
Parallel readout extension of VME

MPV



MPV



Control = Web browser

web server is running on MPV controller

MPV Controller Index (mpv06)

index.cgi / 190919

Refresh

Flash memory = Read Only

Write OK | Read Only

MPV Status

```
*** MPV Status ***
Firmware :  1.25
MOCO Status :
 MOCO0 :   Enable  Ready TestOK
 MOCO1 :   Enable  Ready TestOK
 MOCO2 :   Enable  Ready TestOK
 MOCO3 :   Enable  Ready TestOK
 MOCO4 :   Enable  Ready TestOK
 MOCO5 :   Enable  Ready TestOK
Trigger Select : in0
Busy Select : moco0 moco1 moco2 moco3 moco4 moco5
Level output : 0x0000
NIM Out 0 : ungated
NIM Out 1 : gated
NIM Out 2 : clk10k
NIM Out 3 : eob
```

EFN

Process EFN Set

babildes 66 Set

babies

- MOCO0 SegmentID : 0-3-1-8-0 modify (C32)
- MOCO1 SegmentID : 0-3-3-8-0 modify (C32)
- MOCO2 SegmentID : 0-3-4-8-0 modify (C32)
- MOCO3 SegmentID : 0-3-5-8-0 modify (C32)
- MOCO4 SegmentID : 0-3-6-8-0 modify (C32)
- MOCO5 SegmentID : 0-3-2-21-0 modify (V7XX)

babildes mode

change setting

MOCO Setting Index (MOCO 5)

moco.cgi / 190919

back to index

Refresh

Flash memory = Read Only

Module

SegmentID : 0-3-2-21-0 (V7XX)

		Current	New
Device	BIGRIPS	0	0
Focal	F3	3	3
Detector	PLAQ	2	2
Module	V7XX	21	V7XX
EF type	Normal	0	Normal

Update

See SegID Web Page

Initialize parameters

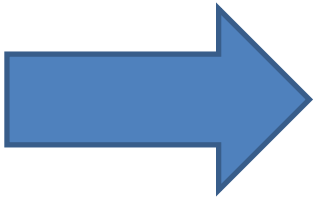
Set

formv7xx.sh / 190903

Parameter	Set	Current
Geometry	0	0
IPED/FSR	250	250
V785 case, should be brank		
Threshold (ch0)	0	0
Threshold (ch1)	0	0
Threshold (ch2)	0	0
Threshold (ch3)	0	0
Threshold (ch4)	0	0

Commissioning at CAT+ESPRI experiment November 2019

- F3, F5, F7, F8, B3F
  - 5 systems
  - 4 = replace standard VME
  - 1 = new crate

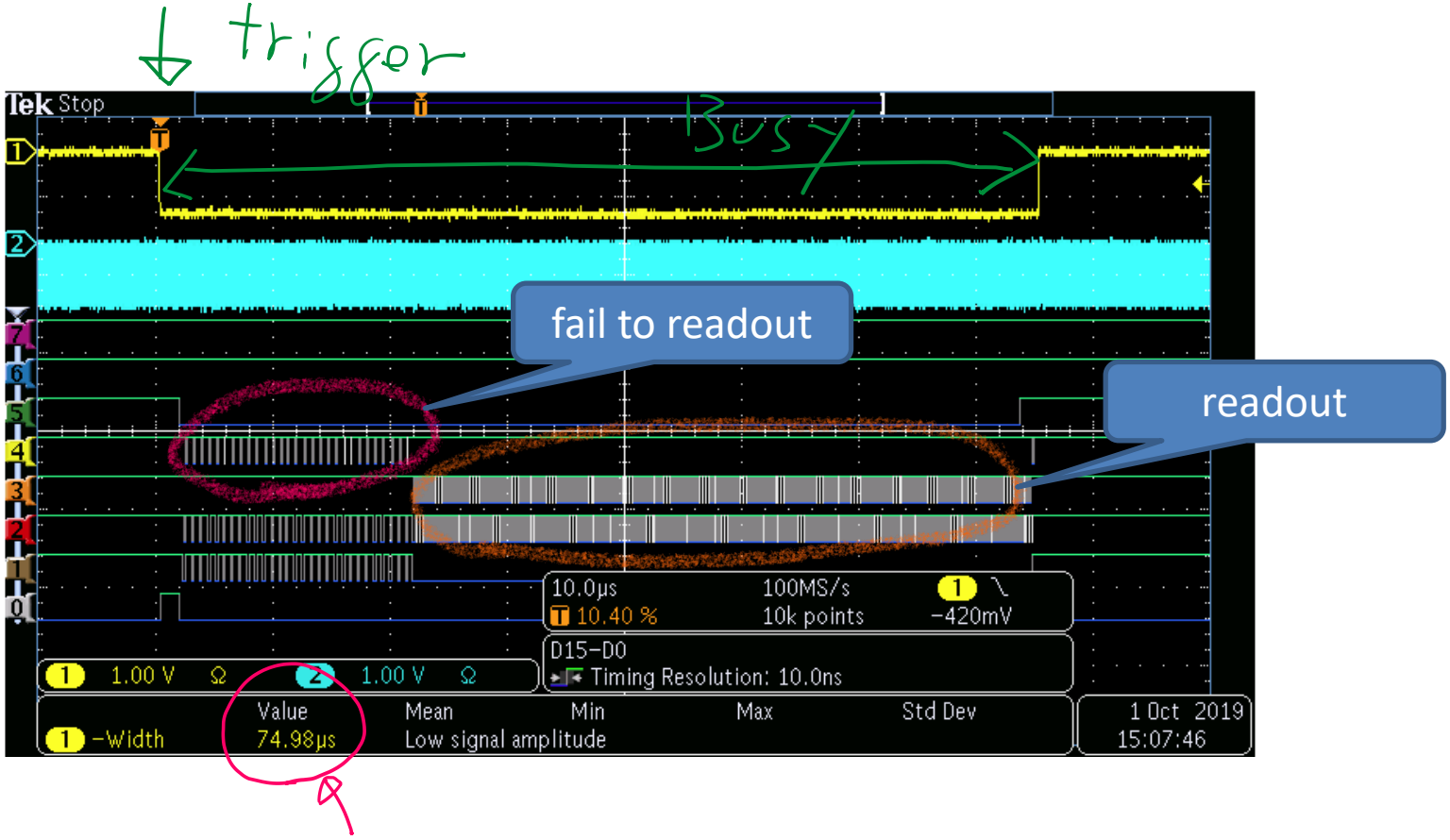


Stably worked

Performance

- 0.1us overhead
  - trigger distribution (8bit command packet)
  - end of data flag (2bit flag)
- 0.2us / 32bit data readout
  - 16 bit data + 4bit flags (2 lines parallel)
  - (if VME module respond within 0.2us)
- 100 word readout
  - 20.1 us / event (for readout)
- V792 case (34 word readout)
  - 1us gate + 7us conversion + 7us readout = 15us
  - Negligible interrupt latency (VME CPU = ～30us)
- Maximum data throughput
  - to be measured (limitations = CPU power)

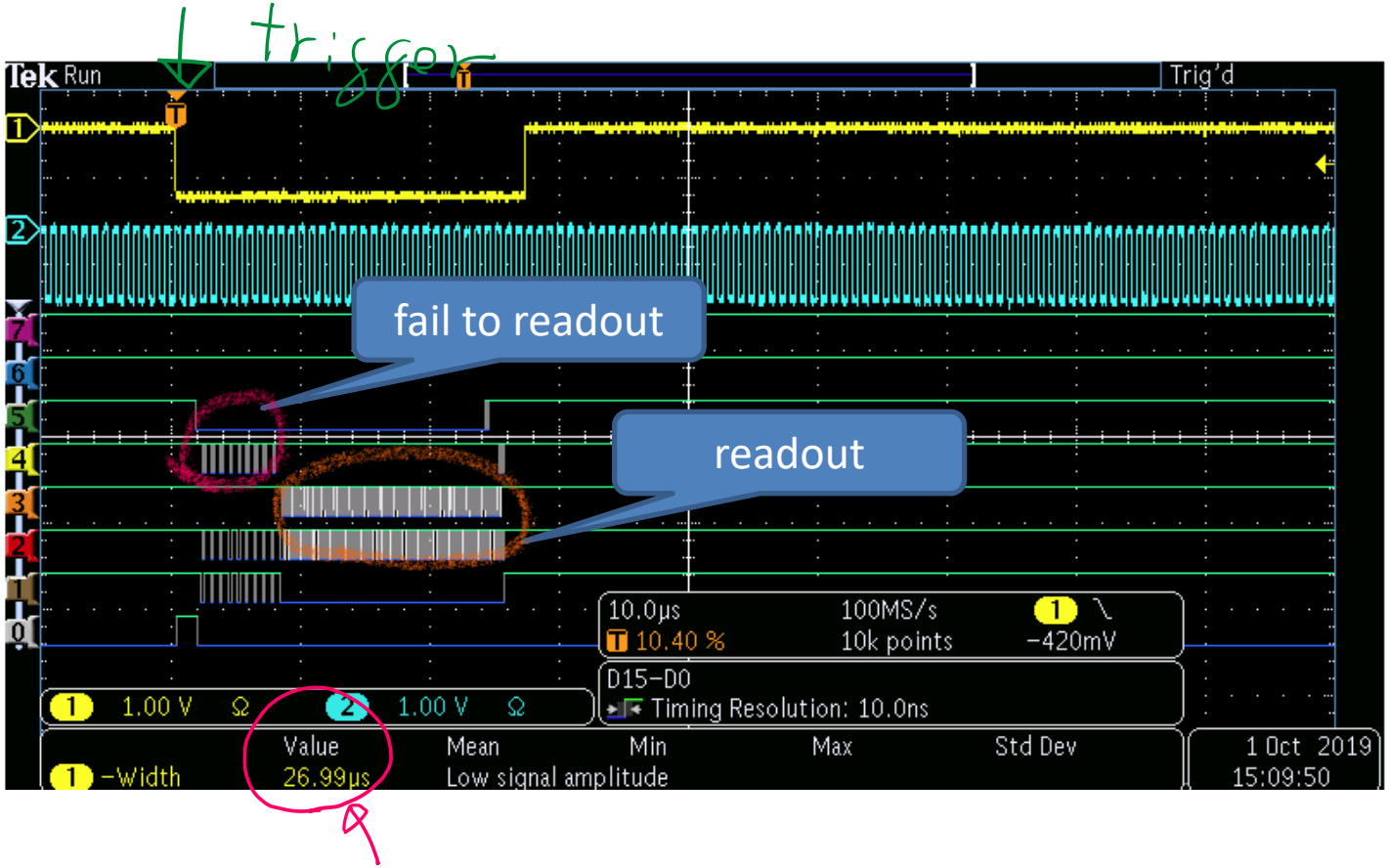
V1190/V1290 problem



214 word readout = 75us

→ 0.35 us/word

V1190/V1290 problem



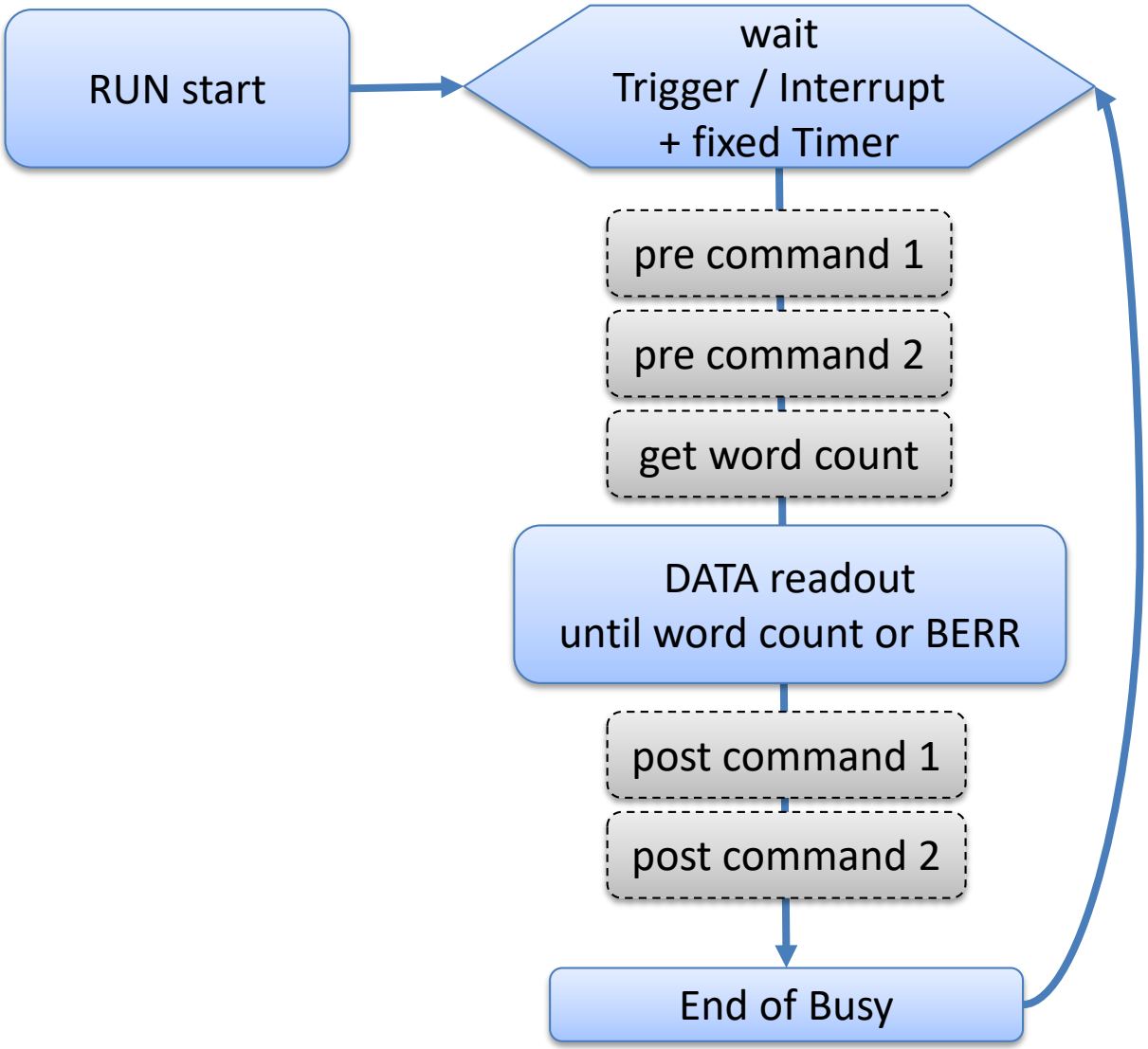
76 word readout = 27us

→ 0.35 us/word

V1190/V1290 problem

- MPV (MOCO) works 0.2u / word
- but V1190 / V1290 has
  - 0.15 us /word additional overhead to be ready to readout (0.35us / word)
- V1190 returns interrupt signal
  - Usually it is “Ready for readout”, but V1190 is not
  - VME CPU case, we always had 30us overhead by OS
    - We didn’t realize this problem (word size is < 200)

MOCO readout



limitations on MOCO

- Support VME modules
  - V785 series, V1190 series, MADC series
  - SIS3820, LUPO etc.
  - Block transfer from single address
  - N x Single transfer from single address
- other VME modules
  - Dedicated firmware on MOCO is required
  - AMSK TDC, SIS FADC
- Resource of MOCO's FPGA (Spartan 3AN 50) is very limited
  - Upgrade MOCO's FPGA is an option
  - Online MOCO firmware update will be available
    - Now, manual cable connection is required
    - (MPV controller can be updated online)

Production Cost

- 1 full MPV = 6 slot
- Cost per 6 VME module
  - MPC Controller — 60k Yen
  - MOCO — 20k x 6 = 120k Yen
  - Backplane — 20k Yen
  - VME shelf for MPV — 250k Yen
  - *Total* — *450k Yen ($4500)*
    - *Exclude VME shelf* — *200k Yen*