JRA1 Beam Telescope DAQ and Trigger

David Cussans, 18th October 2006

University of BRISTOL

Outline

- Aims
- People
- Structure
- DAQ
- Trigger

University of BRISTOL

Aims

- To allow groups to use the EUDET beam telescope with the minimum of difficulty.
  - Modular
  - Simple Interface

People involved in JRA1 DAQ

- Uni Geneva (M. Pohl, D. Haas, E. Corrin)
  - Responsible for overall JRA1 DAQ framework.
  - Produced “Demonstrator DAQ” from MAPS & DEPFET DAQ
- “Strasbourg” (G. Claus et. al.)
  - MAPS DAQ (building Telescope)
- Bonn (H. Kruger), Manheim (P. Fisher)
  - DEPFET DAQ (Test users)
- Bristol/LCFI (D. Cussans)
  - Trigger hardware (Test users, CCD, ISIS)

University of BRISTOL

Integrating DUT into DAQ

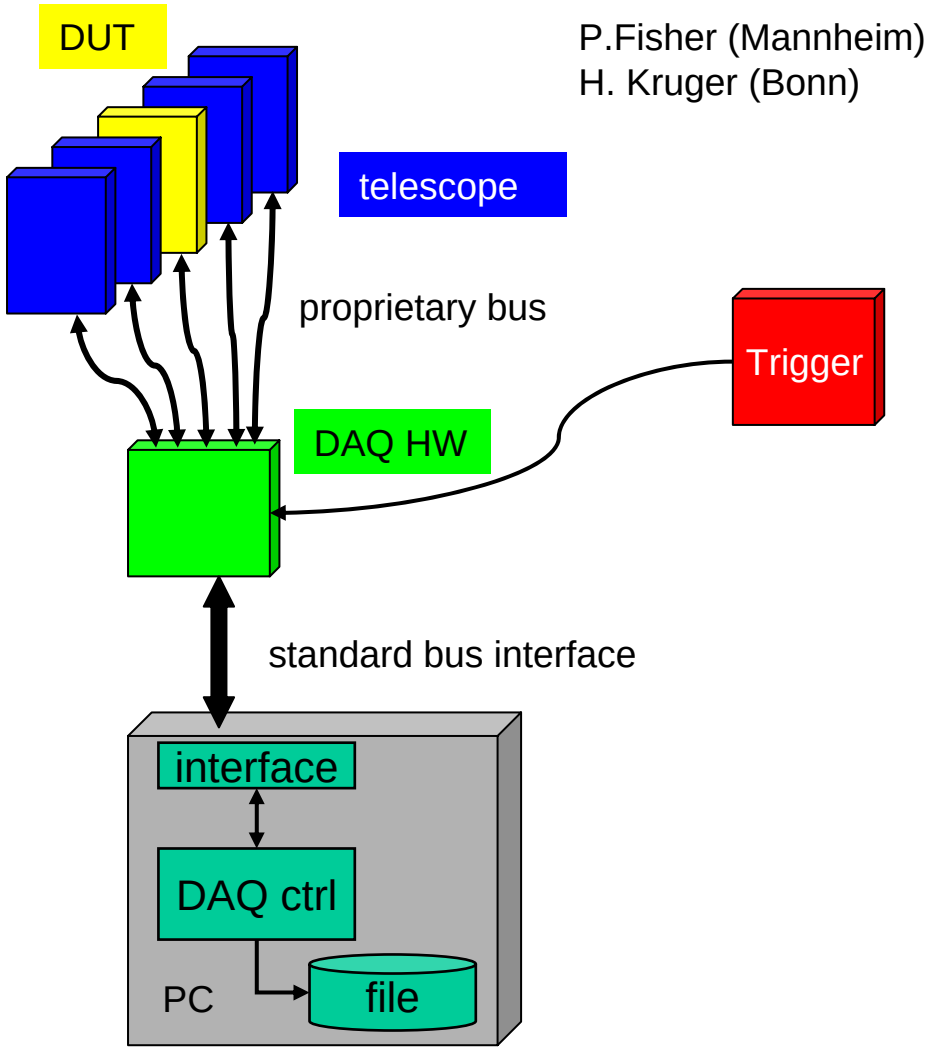
- How should the device under test be integrated with JRA1 beam telescope?
- Considerable thought given to this, since within the group there are tricky decisions to be made
  - Different groups with different detector technologies and different, pre-existing DAQ systems.
  - Nobody has a large pool of effort to re-write existing code.

Option1: Integration at 'hardware level'

- Use special purpose hardware interface to read out everything
- DUT users must comply to hardware specs
- Use one integrated DAQ software

Problems:

- all users must use special interface & DAQ
- Probably large overhead when using in lab



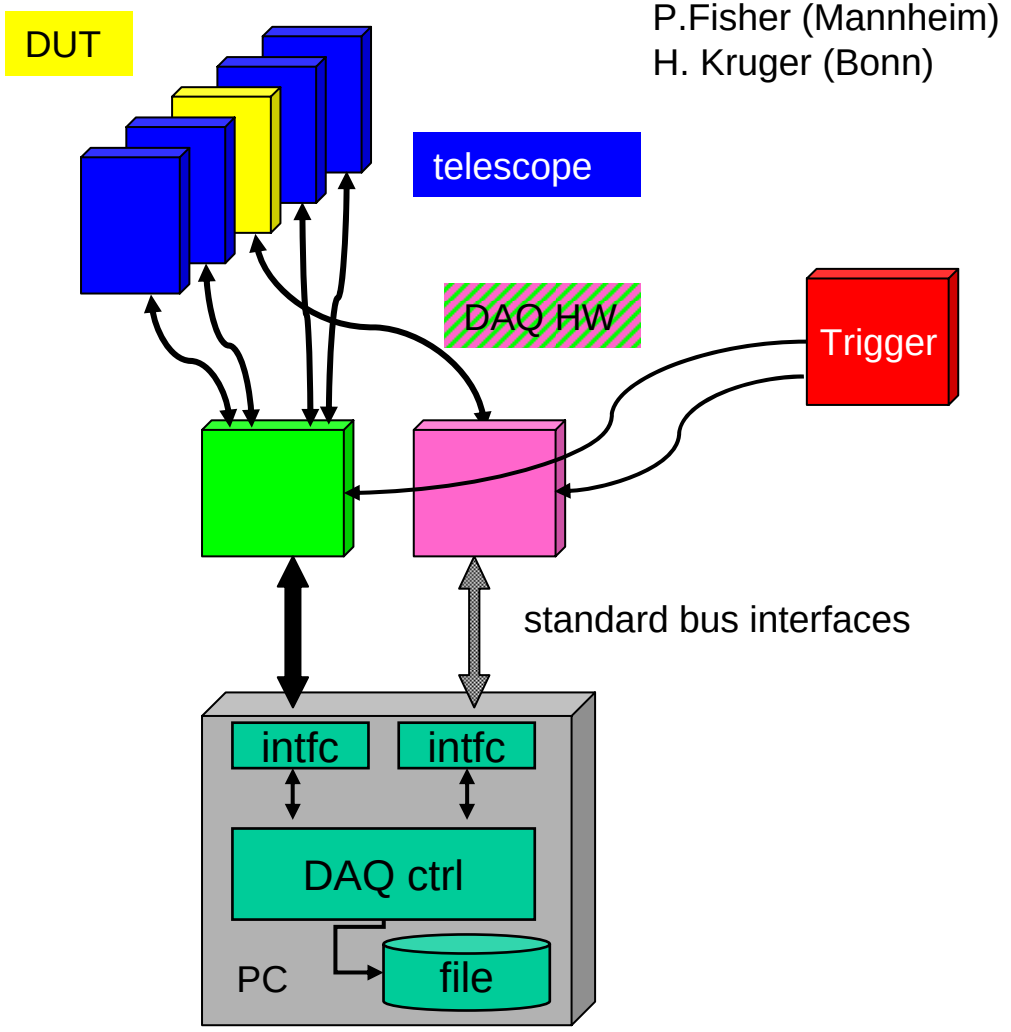
University of BRISTOL

Option 2: Integration at 'software level'

- DUTs provide their own DAQ hardware
- They can use 'any' PC interface
- Use one integrated DAQ software

Problems:

- 'The' DAQ PC must provide required h/ware interface.



University of BRISTOL

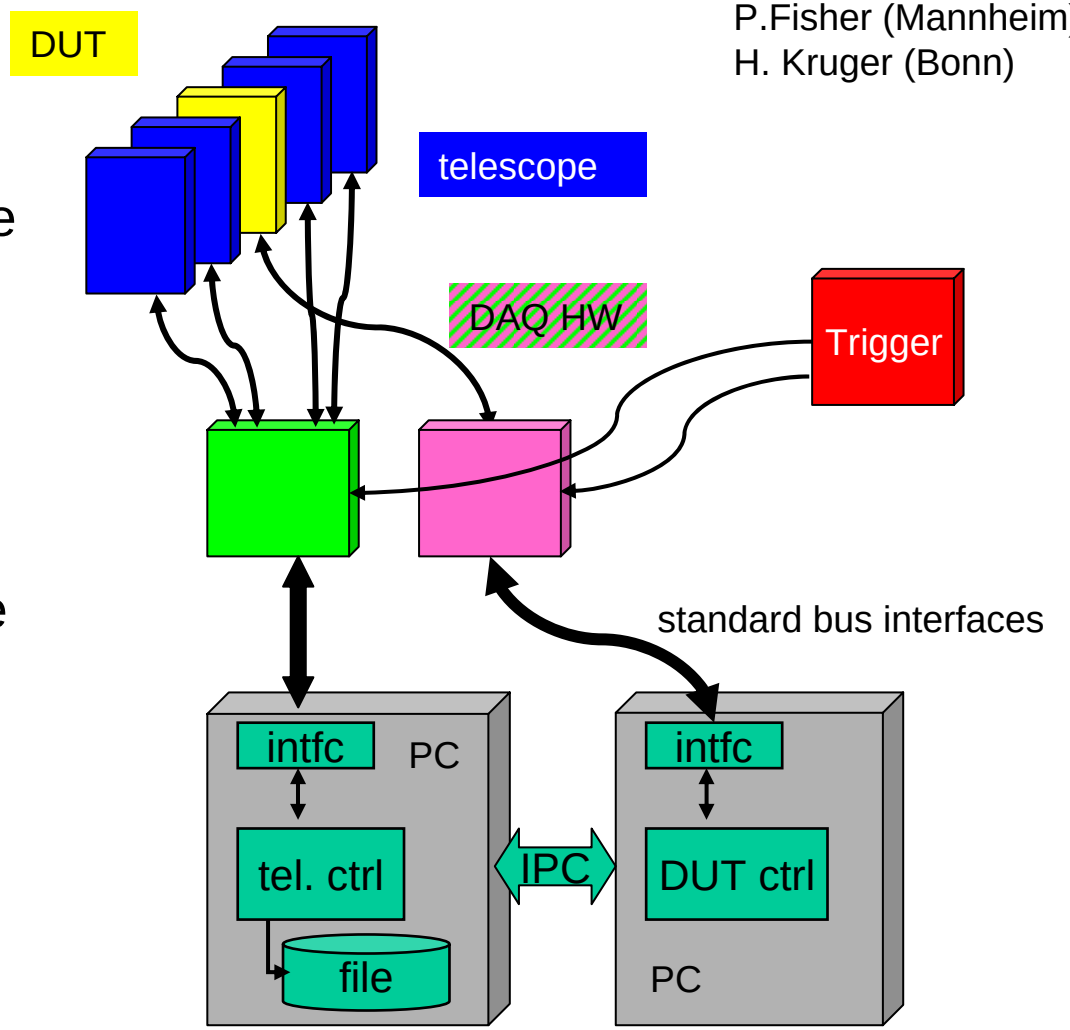
Option 3: Integration at 'data level'

- Use completely different hardware for the DUTs
  - Connect DUT to a separate PC
  - Readout Software is provided by the DUT user.
  - DUT sends only DATA to DAQ. This can be on same PC or via TCP/IP

Problems:

- How to make sure that devices are configured correctly at start of run.

P.Fisher (Mannheim)
H. Kruger (Bonn)



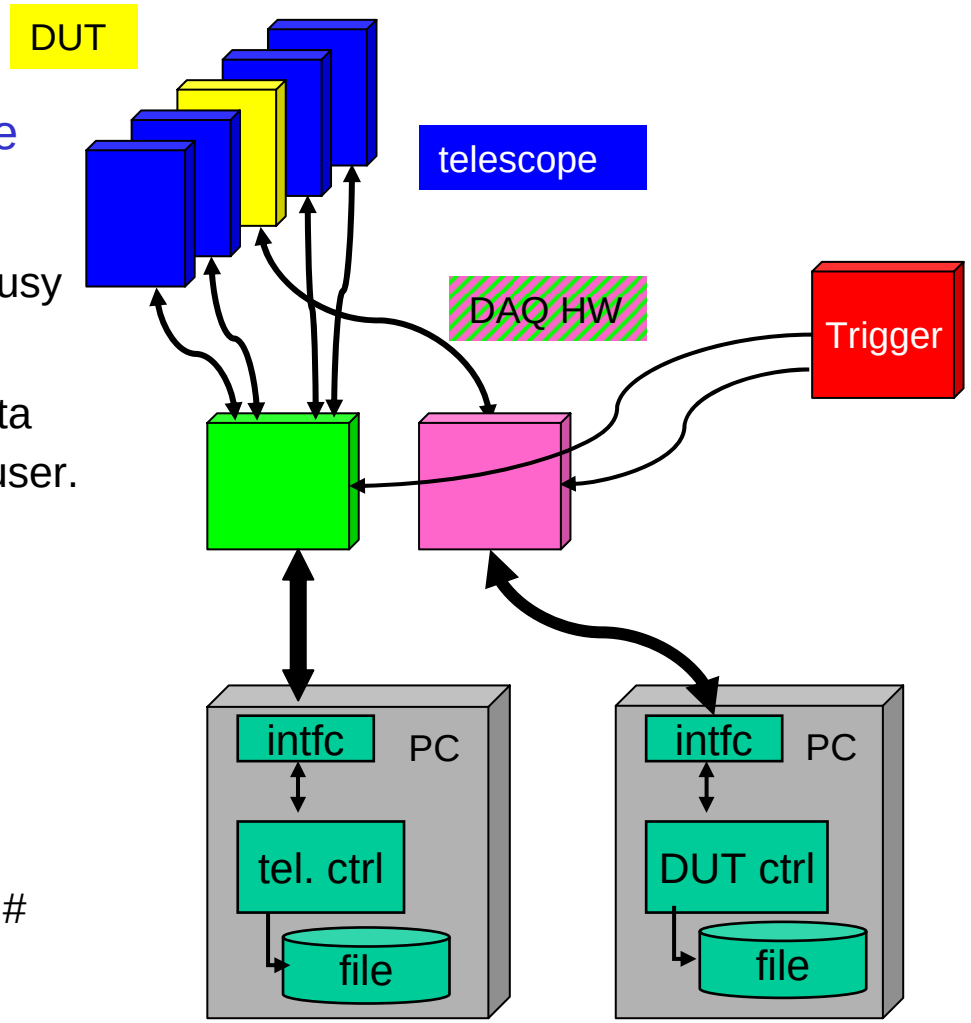
University of BRISTOL

Option 3: Integration at 'trigger level'

- Use completely different hardware and DAQ for the DUT
  - Synchronize only with Trigger, Busy and Reset signals.
  - Readout Software, DAQ and data storage is provided by the DUT user.
  - Events combined off-line.

Problems:

- Run control and configuration
- Online monitoring difficult.
- No way of detecting slippage between DUT and telescope evt #

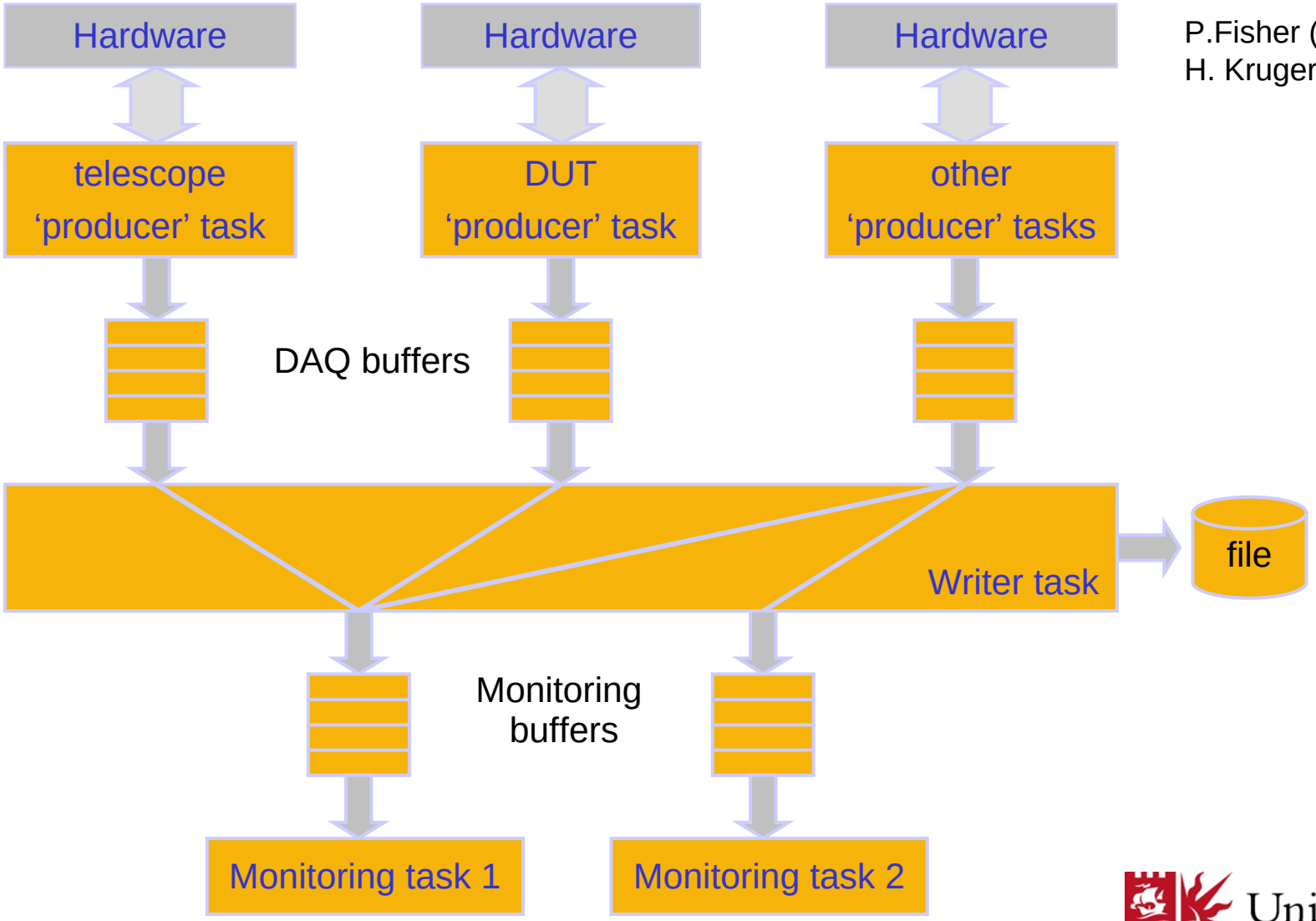


University of BRISTOL

Integration

- Decided to favour integration at the trigger level.
  - Integration at the hardware or s/ware level places too many burdens on DUT (requires rewrite of large amount existing code)
  - Integration at the trigger level leaves the DUT “flying blind” (no monitoring, so won’t spot mis-match between DUT and telescope event numbers until too late). However, a DUT can do this if they really want to…

DEPFET 'Mini-DAQ'



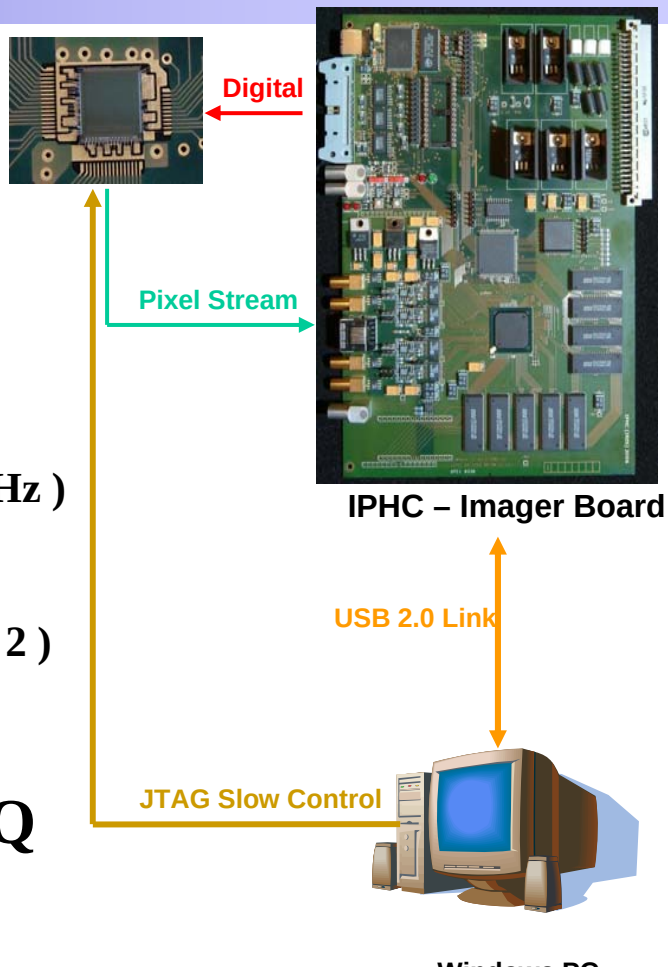
P.Fisher (Mannheim)
H. Kruger (Bonn)

University of BRISTOL

IPHC DAQ Proposal for MAPS Telescope Demonstrator

Hardware : Based on Imager board + PC

- MAPS Readout board developed at IPHC
  - For our beam Si-Strip Telescope upgrade (VME / USB)
- Data transfer to PC with USB 2.0 link
- Digital sequencer to control MAPS
- Analogue pixel stream acquisition (12 bits ADC, at up to 50 MHz)
- Can control MAPS with up to 1 Million pixels
- CDS calculation, Trigger handling on board Firmware (Virtex 2)
- On board zero suppression is foreseen (But not for June 2006)

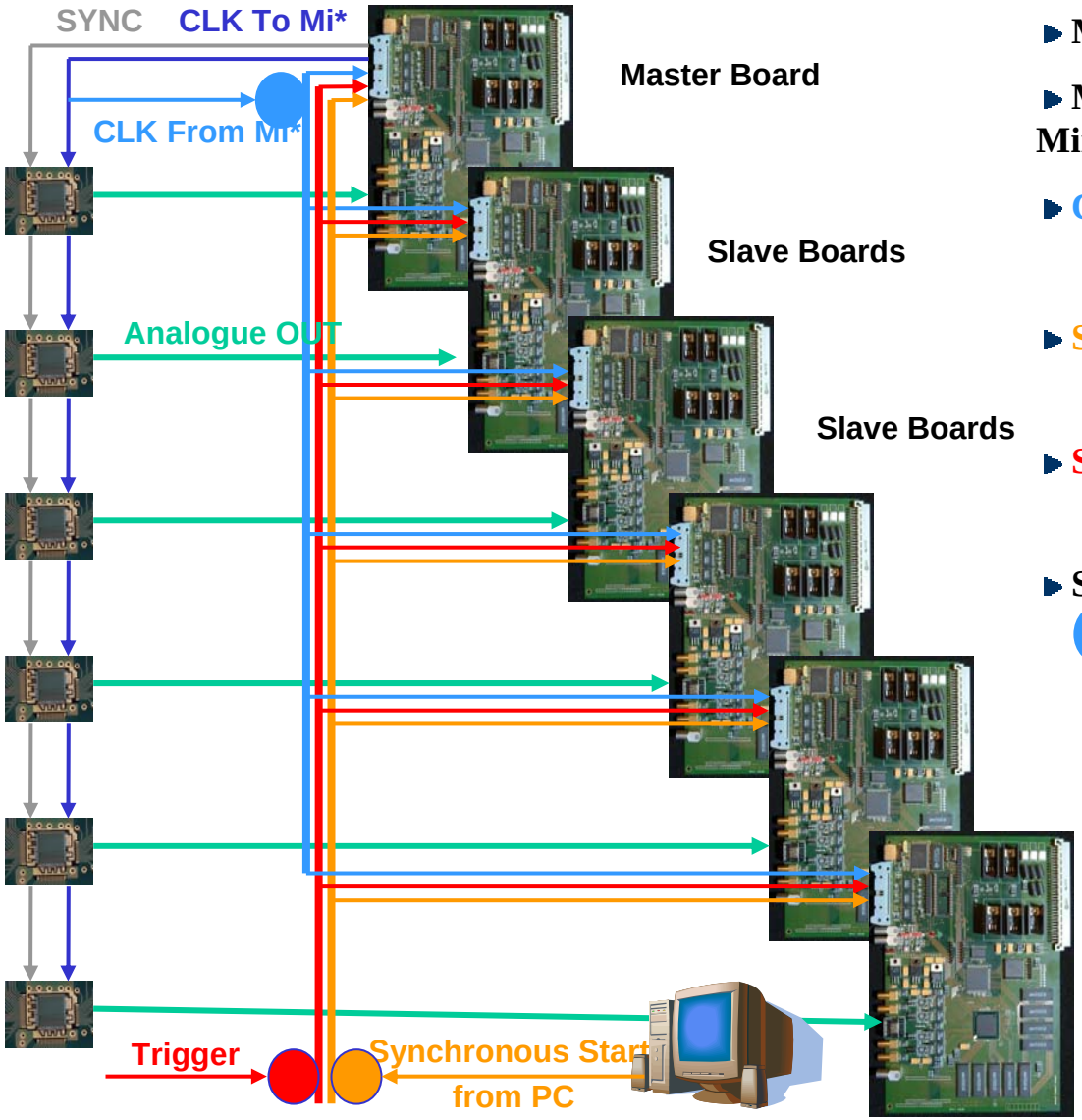


Software : Windows DAQ

- One PC can control up to 6 boards
- Event rate with 6 Planes of Mimo*3M (64 KPixels) 30 – 40 Hz (10 MHz - CDS)
- DAQ application : Stand Alone mode or Slave in JRA1 Global DAQ
- JTAG Slow Control is also provided to configure Mimo*3M (PC // Port)

University of BRISTOL

Preliminary " Integration " Test has been done



- Master / Slave board architecture
- Master provide CLK and SYNC to all Mimo*3M
- Clock go back to DAQ
  - Distributed to all boards (slaves + master)
- Synchronous start of all boards
  - From PC parallel port
- Synchronous stop of all boards
  - From telescope trigger
- Star distribution of all signals near boards

Simulation of 6 Mimo*3M RO

- Trigger and Hit : Pattern Generator
- Boards synchronization → OK
- Trigger handling → OK

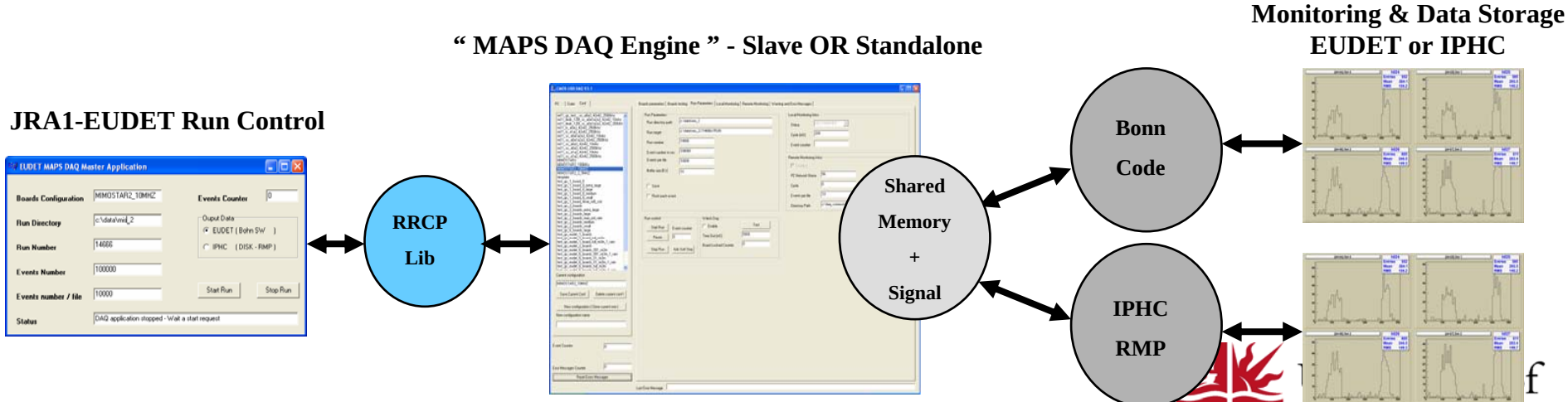
University of BRISTOL

Possible IPHC Software Integration in JRA1 Global DAQ

How to do it ?

- **A Master / Slave architecture**
  - **IPHC " DAQ Engine " Application is a slave**
  - **EUDET JRA1 Run Control is a Master**
- **Interface with two protocol**
  - **RRCP (Remote Run Control Protocol)**
  - **RMP (Remote Monitoring Protocol)**
- **Advantages**
  - **IPHC (Exist for Si-Strip Telescope) and JRA1-EUDET Data Monitoring and Data Storage strategies can be used**

The " DAQ Engine " can be Tested in our Si-Strip Telescope before integration in EUDET MAPS Telescope



Integration

- Emlyn Corrin at Geneva has taken Strasboug DAQ and Bonn/Manheim DAQ and integrated them into a “prototype JRA1 framework”
- Ready for beam tests (including trigger)

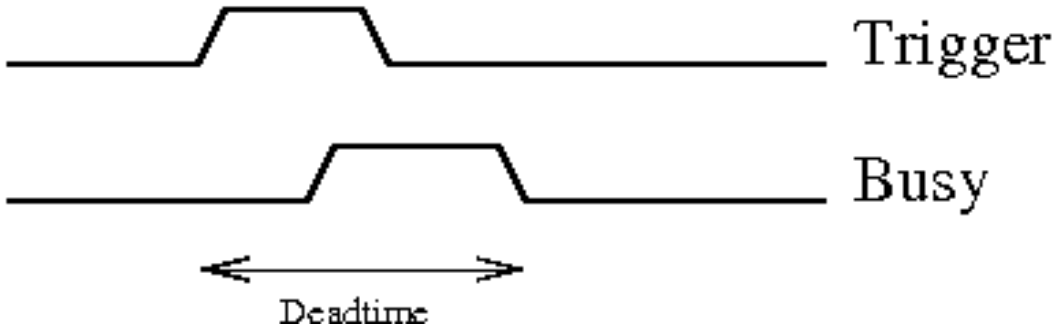
Trigger Hardware

- Simple and cheap (< €1500) enough to have in DUT home lab.
  - Also simple enough to be emulated “at home”
- Functionality
  - Receives NIM and/or PMT signals and passes trigger on to DUT and telescope front-end.
  - Vetoes further triggers until all devices under test drop “busy” signal.
  - Records timestamp for each trigger
  - Distributes trigger number (optional)

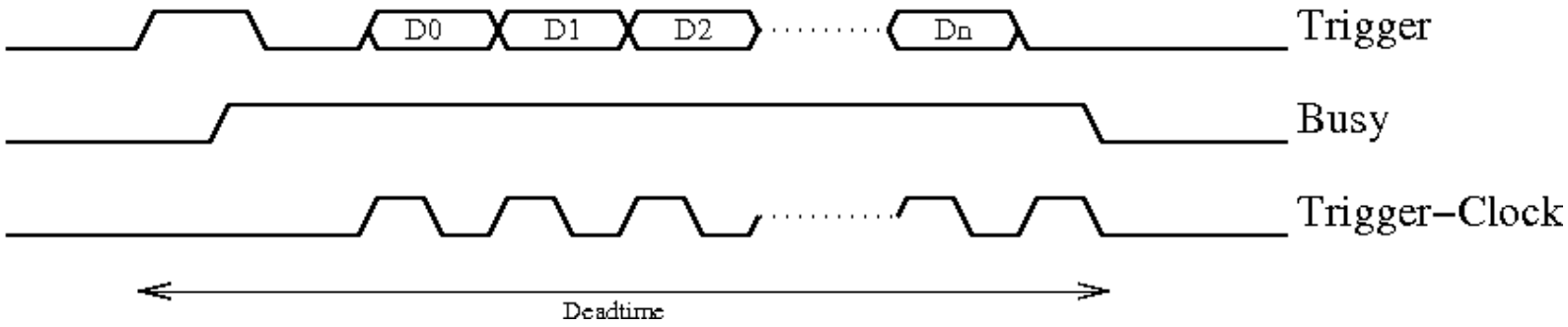
University of BRISTOL

Trigger Handshake

- Simple



- With trigger number



University of BRISTOL

“Trigger Logic Unit” (TLU)

- Four PMT/NIM inputs
- 15V outputs for powered PMT bases
- Six interfaces to DUT by LVDS on “RJ45” connectors
  - Two interfaces can be switched to TTL (NIM if sufficient interest) on Lemo-00
- Interface to host by USB 2.0
  - FPGA firmware downloaded at startup from host. (Firmware updates are simple)
  - Details http://svn.phy.bris.ac.uk/svn/uob-hep-pc017a/trunk/www/index.html

TLU Hardware Status

- Two test TLUs built. (one in Bristol, one in Geneva)
- Basic functionality tested at Geneva.
- Waiting for beam-tests.
- Three further units built.
  - Waiting for discriminator daughterboards
  - Delivered by end of ‘06
  - Cost: €1500 each

University of BRISTOL

Host Interface to TLU

- Commercial FPGA board (xc3s1000fg256) used as heart of initial implementation of TLU
  - Host access libraries in Linux and Windows
  - Alas, “closed source” so only compiled libraries can be distributed. Possibility of replacement using Bonn code.
  - Wrapper API written in “C” and C++
  - “SWIG” used to generate interface to Python and Perl scripting languages.

Future of TLU

- Accept inputs of (or generate internally) clock and beam related information (eg. Fake bunch-train-ID, bunch-ID) and fan-out to DUT
  - Can be prototyped with existing hardware
  - Keep existing RJ45 trigger/busy/reset interface. Add additional RJ45
- Clock out trigger number/timestamp/etc from TLU into DUT FIFO - reduce dead-time, allow more info.
- Evolve into a “Taggling Logic Unit” for use with triggerless DAQ (either telescope or DUT)

Summary

- JRA1 group aiming for a system that not only gives excellent tracking performance but is also easy to interface to.
- Fruitful debate and discussion continues into how best to partition the system and divide effort.
- Existing DAQ systems from MAPS and DEPFET sensor programs have been integrated into a prototype JRA1 framework and are ready for beam-test.
- Trigger hardware has been built and is ready for beam-test.