Controls

Accelerator Control Systems, Design and Operation

Kazuro Furukawa

High Energy Accelerator Research Organization (KEK)

KEKB and Linac Control Groups

< kazuro.furukawa @ kek.jp >
<http://www-linac.kek.jp/linac/>
<http://www-linac.kek.jp/cont/>

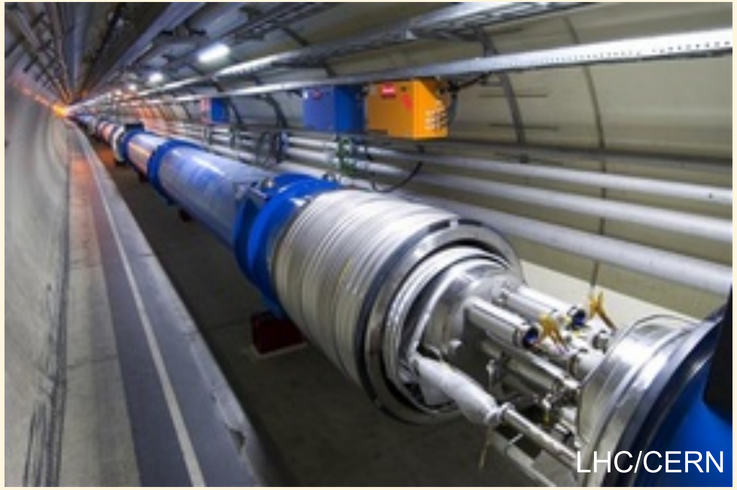
Contents

- Overview
- Accelerator Controls
- Available Technologies
- Reliability
- Example
  - Past KEKB
  - SuperKEKB
  - Operation
- Conclusion

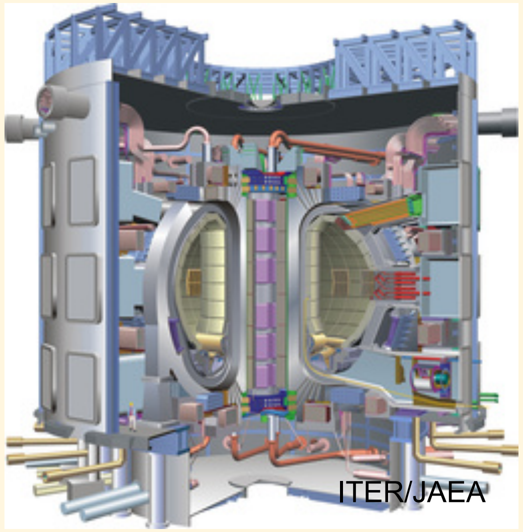
Overview

Large Experimental Physics and Controls

- **Experimental Apparatus run by tens to thousands of researchers**
  - High energy physics, Optical and radio observatory, Plasma fusion, Gravitational wave interferometer, etc.
  - Tend to be designed, constructed, and run internationally
  - CERN/LHC at Geneva, ITER at Cadarache, ALMA at Chile, SuperKEKB at Tsukuba, ILC developments all over the world, etc.


LHC/CERN

ALMA/NAOJ

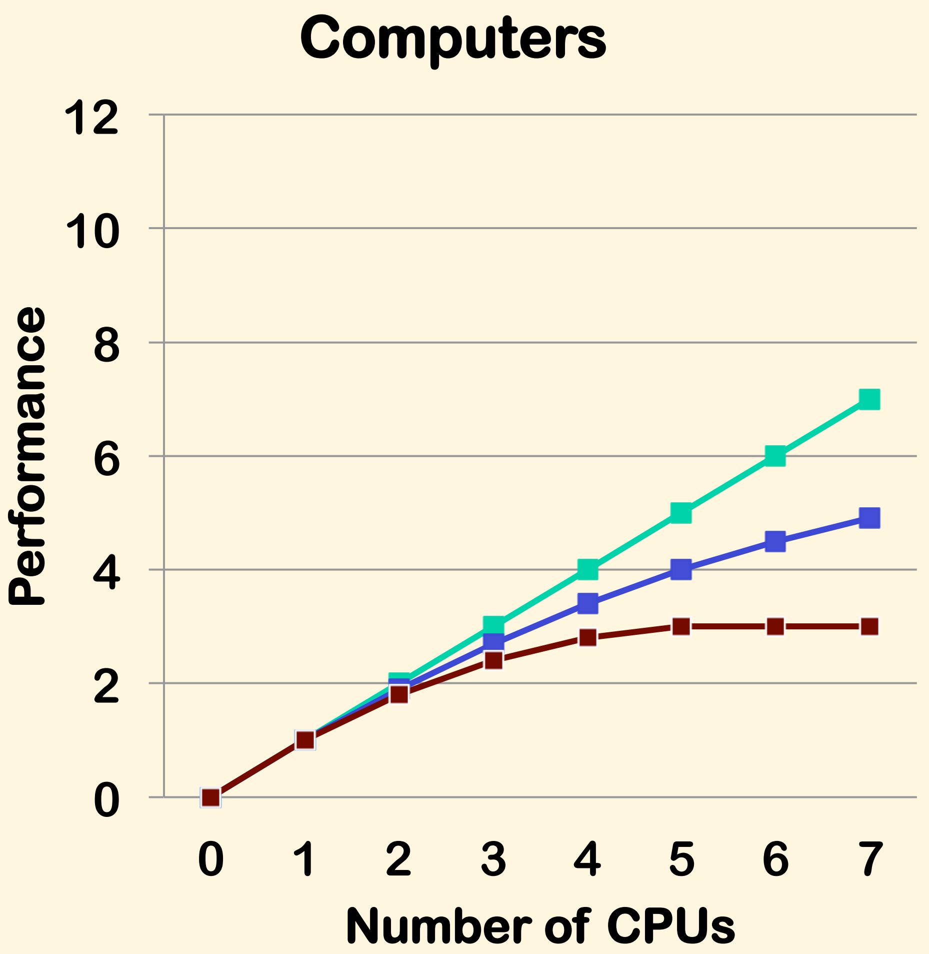

ITER/JAEA

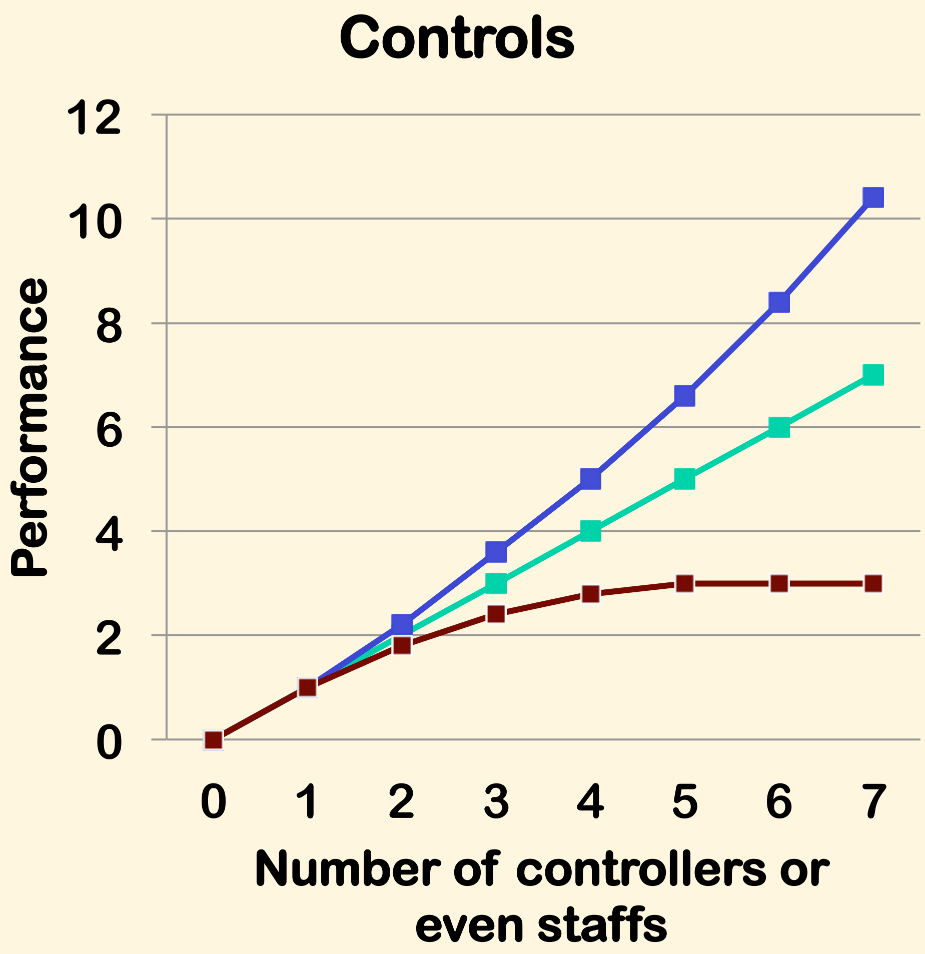
Conferences

- Exchange collaboration ideas internationally
  - Collaboration is important because we don't want to reinvent the wheel
  - Asia, Europe and North America regions
- ICALEPCS, International conference on accelerators and large experimental physics control systems
- PCaPAC, International workshop on personal computers and particle accelerator controls
- WAO, International workshop on accelerator operation
- IBIC, International beam instrumentation conference
- IPAC, International particle accelerator conference

Performance achievement

If the system is small single controller or single person can cover it. However, … We need global optimization instead of local optimization





Challenges

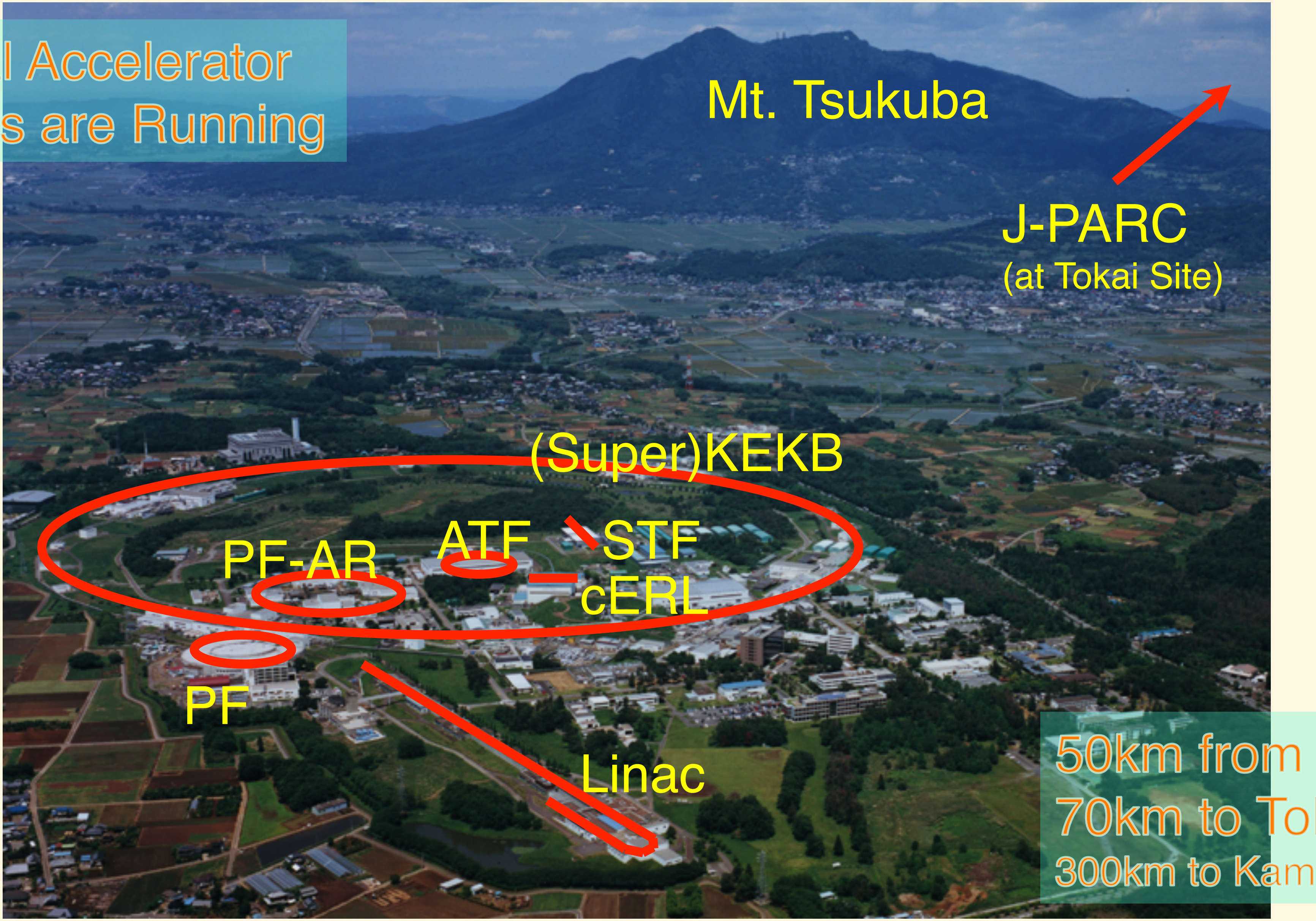
- Number of components
- Number of kinds of components
- Collection of user requirements
  - with some knowledge of accelerator
  - collaboration with equipment and physics groups
- Efficient use of industrial solutions
  - Extreme is commercial accelerator
- Many interactions with all groups
  - We can enjoy controls

Challenges (2)

- **Online simulation capabilities**
  - **Rapid prototyping and rapid turnaround are important now**
  - **Acceptance of changing requirement**
- **Daily maintenance**
  - **Routine automated procedures**
- **Performance analysis**
  - **Analysis of data history archives**
    - **Correlations, trends, sudden changes, etc.**
  - **Monitoring from offices**
- **Protection systems, personnel, machine, etc.**
- **Above procedures have to be carried at both equipment and operation layers**

Accelerators at KEK

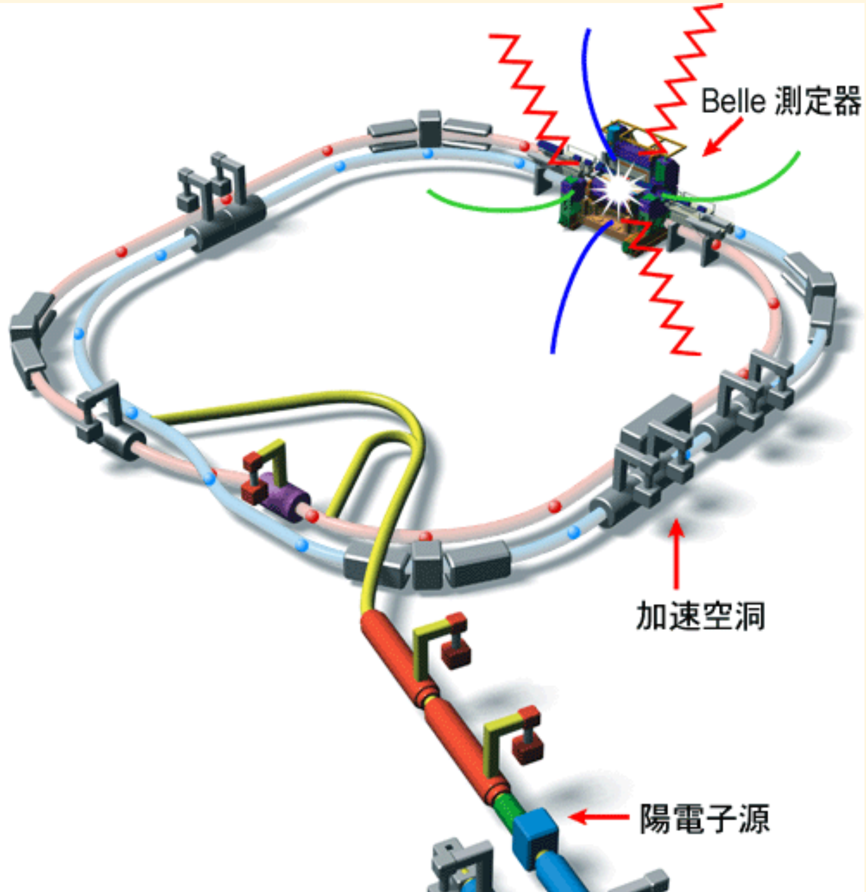
Several Accelerator Projects are Running

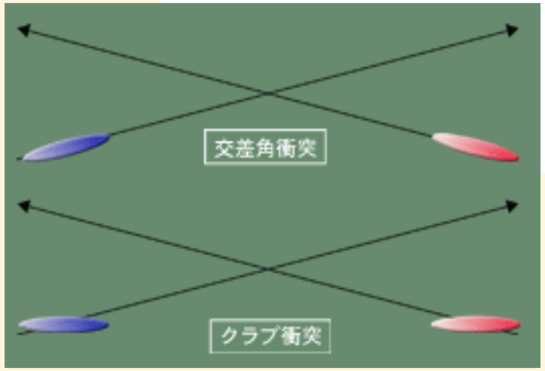


50km from Tokyo
70km to Tokai
300km to Kamioka

KEKB and Linac

- **KEKB B-factory: Electron/Positron Asymmetric Collider for CP-violation Study**
  - **~3km Dual-rings: Electron(8GeV - 1.4A) / Positron(3.5GeV - 1.8A)**
    - **Stable and Robust Operation**
    - **Many Active Operation Parameters**
    - **Importance of Controls**
- **Linac:**
  - **~600m, 50Hz**
    - **8GeV 2nC Electron, 3.5GeV 1.2nC Positron**
    - **Beam switchings, KEKB, PF, PF-AR rings**
- **Achieved world luminosity record**





Increase of Luminosity with Crab Cavities

Control Systems at KEK (1)

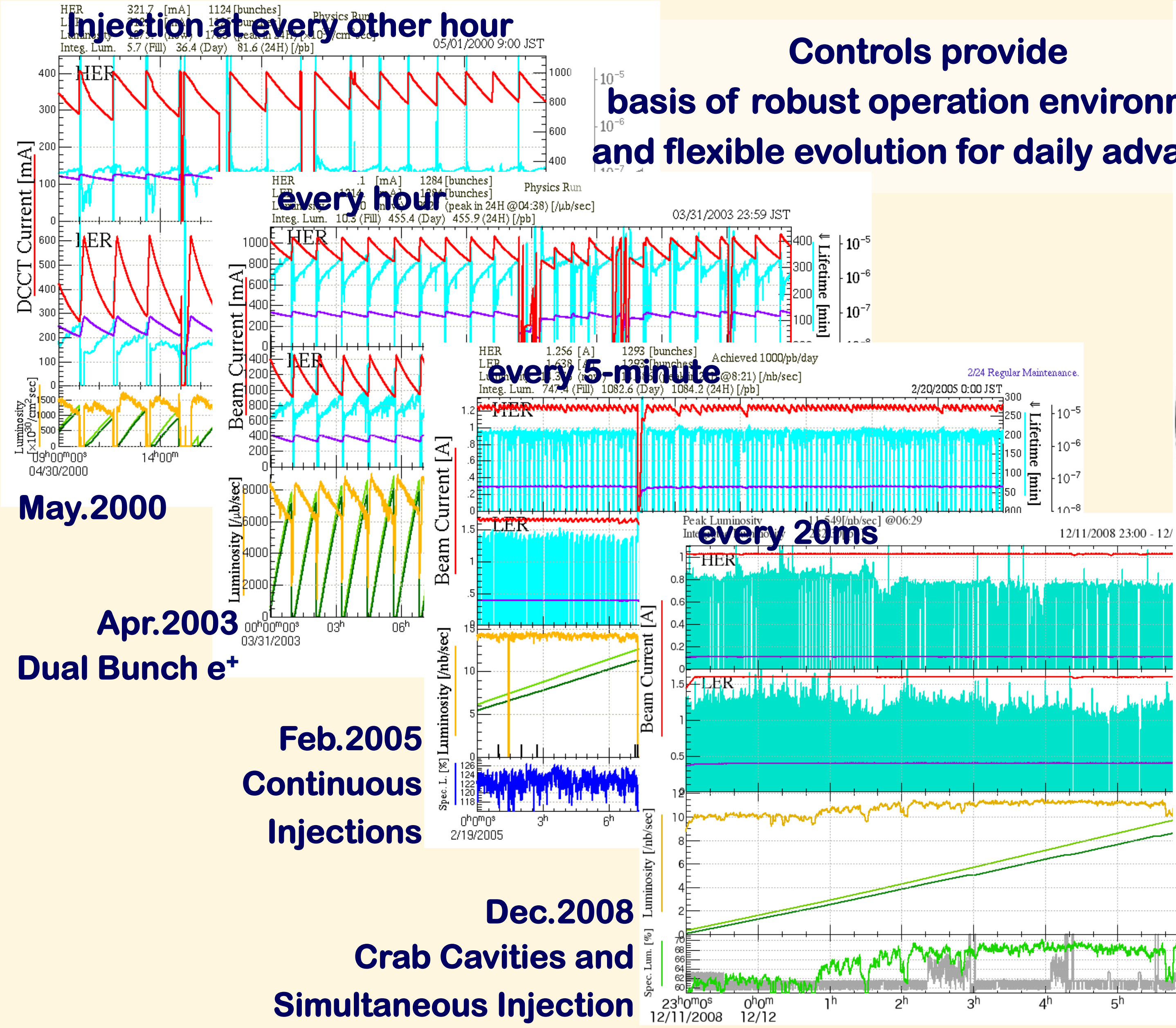
- **Sharing resources as much as possible**
- **SuperKEKB (B-factory for flavor physics study)**
  - **Will inherit resources from KEKB (and TRISTAN)**
    - **Upgrade 2011-2014 is on-going**
- **Linac (electron/positron)**
  - **Inject beams to 4 rings, (Super)KEKB, PF, PF-AR**
    - **Pulse-to-pulse (50Hz) beam modulation (2.5GeV ~ 8GeV, e–/e+)**
- **PF (Photon Factory)**
  - **Moved to EPICS environment, serve large number of users**
    - **Mainly with Linux-VME**
- **PF-AR (Photon Factory Advanced Ring)**
  - **Mostly the same environment as KEKB**
    - **Many CAMAC installations**

Control Systems at KEK (2)

- **J-PARC (Proton Accelerator Research Center)**
  - **Inherit success at KEKB and linac**
    - **Many network-based EPICS devices**
- **ATF (Accelerator Test Facility)**
    - **Vista Controls environment with CAMAC**
    - **Linux and socket environment with some EPICS devices**
- **STF (Superconducting RF Test Facility)**
  - **Test facility for ILC**
    - **EPICS with Linux, ATCA test, PLC, …**
- **cERL (Compact ERL)**
  - **Being built for ERL development**
    - **May share the resources with other accelerators**

KEKB Operation Improvement (base of SuperKEKB)

Controls provide
basis of robust operation environment
and flexible evolution for daily advances

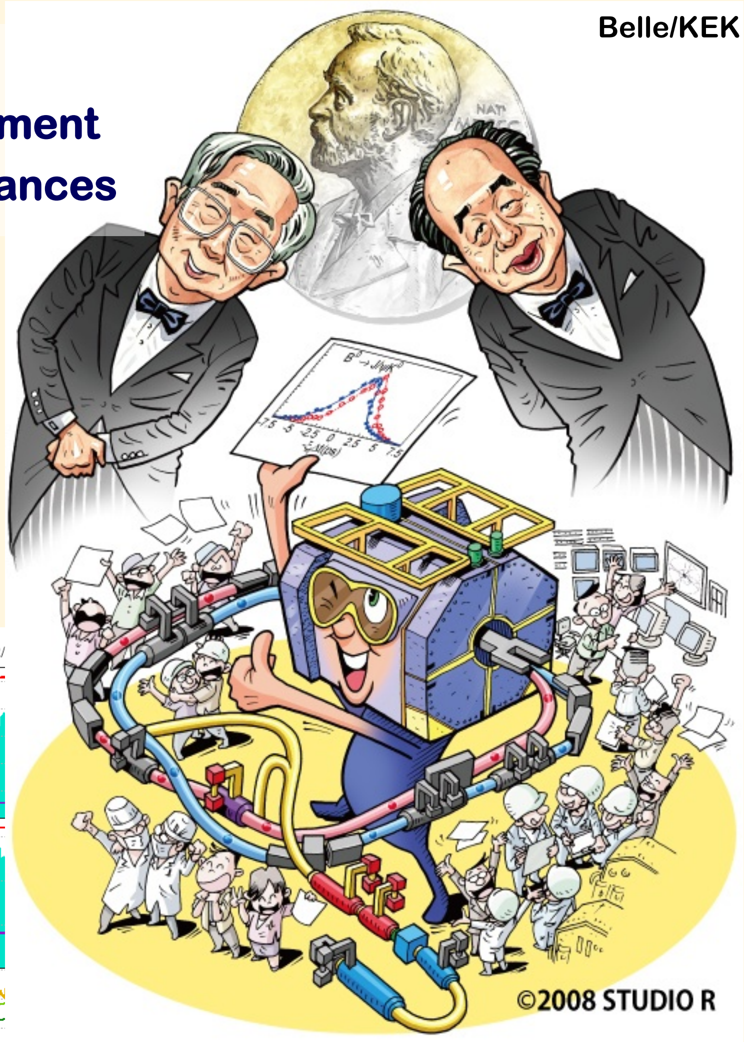


May.2000

Apr.2003
Dual Bunch e⁺

Feb.2005
Continuous
Injections

Dec.2008
Crab Cavities and
Simultaneous Injection

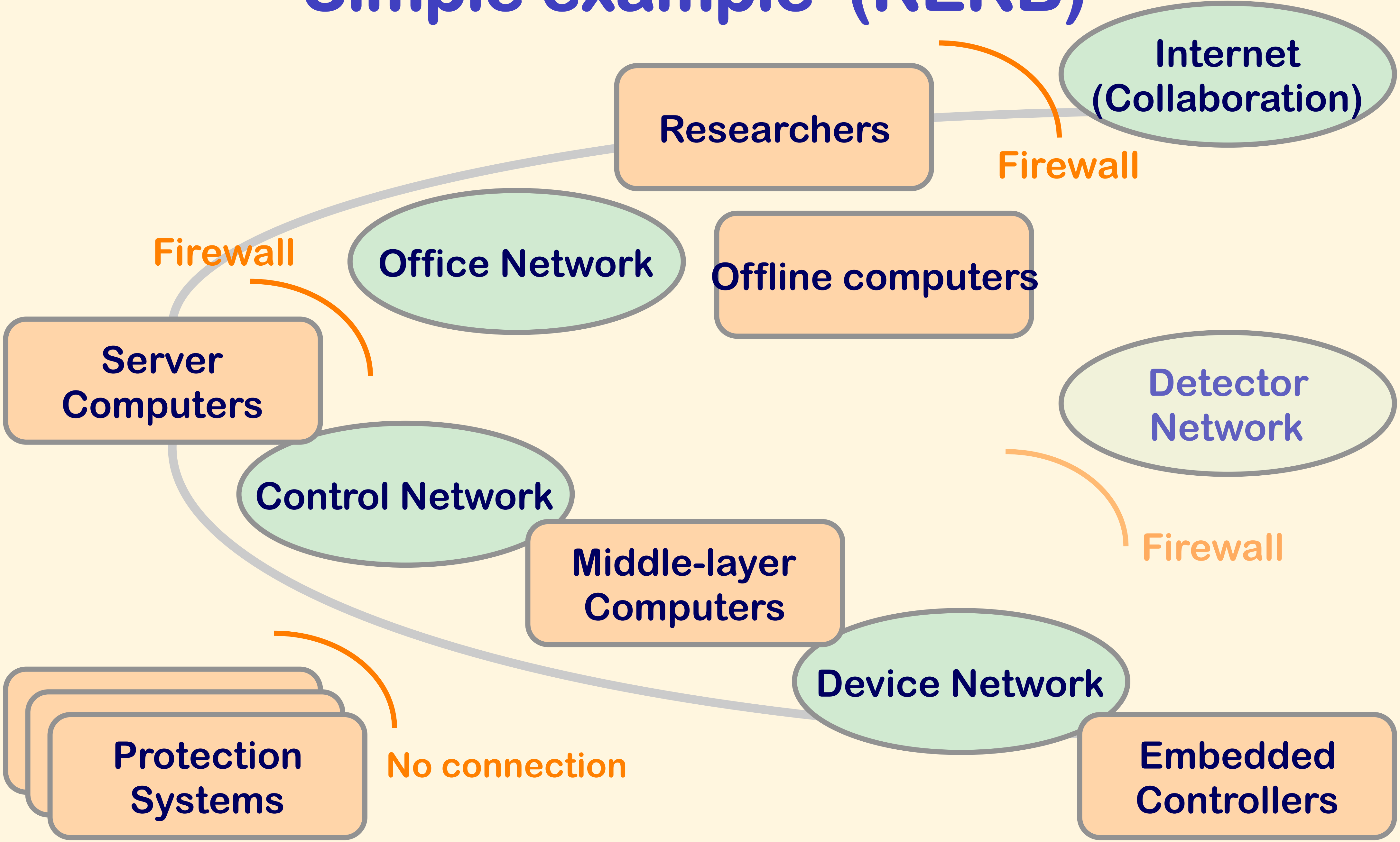


Important to follow
changing demands
flexibly

Size of Middle/Large Accelerator (KEKB)

- Composite of many devices
  - Thousands of magnets
  - Tens of microwave sources
  - Thousands of vacuum components
  - Thousands of beam instrumentations
  - Hundreds of researchers/engineers
    - Accelerator and detector share the tasks
    - Accelerator itself is a physics research field
    - Collaboration with other fields like large scale superconducting system and vacuum system
- Controls
  - Thousands of small form-factor embedded controllers
  - Hundreds of middle-layer computers
  - Tens of server computers
  - IP-based network
- 10-100 times larger at the next generation machine like ILC

Simple example (KEKB)



Accelerator Controls

Accelerator Controls

Objectives

- Control requirements are decided only after requirements of equipment are decided
  - Final objective is physics experiment achievement
- Requirements often change during the operation
  - Flexibility as well as robustness is required
  - Realization of bright new idea is important, while many of them are thrown away later. Speed!
- Unfortunately, no general purpose control system yet
  - We should design something to meet our accelerator needs
    - At the same time we have many components that can be shared
    - Effective sharing can be achieved via collaboration

Some History

- **Discussion on accelerator controls**
  - **Conference of ICALEPCS started in 1987**
    - **Industrial standards like X-Window and VME, success of NODAL interpreter at SPS/CERN**
      - **Before, it was said that there were no sharable components between institutes**
    - **More sharable, reusable control components (Eco!)**
  - **NODAL deployment at TRISTAN/KEK**
  - **Industrial Micro-Computer and VAX/VMS at SLC/SLAC**
  - **Standard model (of accelerator controls)**
    - **Field-network + VME + Unix + X11**
  - **Hope for sharable control system**
    - **Defining class object to express whole accelerator (?)**
      - **Practically impossible, then stacked**
  - **General purpose control API from specific institutes**
    - **ncRPC/LEP/CERN, TACL/CEBAF/TJNAF, ACNET/FNAL/Tevatron, etc**
    - **EPICS was employed at SSC, then EPICS spreads APS, CEBAF, BESSY, …**
  - **Many absorb industrial advances**
    - **More computer aided development possible**
    - **CICERO (CORBA)/CERN, TANGO (CORBA), JCOP (CORBA+Java)/CERN, …**
    - **Windows (COM)/Microsoft, …**

No Complete Sharable Control System

- The sense of balance required to select and adopt available technologies

Object-oriented vs. Channel-oriented

- Object-oriented technology
  - More support benefits from software engineering
  - Extendable, clearer definitions
  - Different people have different ideas on control objects
- Channel-oriented technology
  - Flat (one-layer structure), simple, scalable
  - Not much support from software engineering
  - Easy to make gateways
    - EPICS is basically channel-oriented

More balances

- Compiled language vs. interpretive language
  - Two level languages
    - Interpretive language for rapid prototyping
    - Compiled language for established algorithms
  - After too much success of NODAL
    - Many un-manageable programs
  - Compiled languages programmed by expert
    - Documentation, maintenance, policy-driven
    - Manageable, then reliable
  - Interpretive/scripting languages
    - Rapid development
      - Realization of novel ideas in hours
    - Everyone attends the construction of operation environment
    - Another level of management/maintenance required

More balances for technology choice

- **Best & aggressive vs. moderate & conservative**
  - **New technology is attractive**
    - **But can be a "fad"**
    - **Can we justify the choice?**
  - **For longer life-span, which is better?**
    - **Life of accelerator is often very long compared with**
      - **User facilities**
      - **Commercially available software/communication technologies**
    - **Operational performance continuously advances**
  - **Accumulation of operation knowledge base**
    - **Stored mainly as software and database in the control system**
      - **Beam stabilization algorithms, hardware startup procedures, etc**
  - **It is valuable treasure**
    - **There should be mechanism to keep such resources**
      - **With longer life-span**

‘Standard’ balances

- International vs. de-facto standards
  - International organizations pursue ideal solutions
    - Sometimes they don’t become de-facto standards
    - Selection of one of many standards is difficult
  - Watching the market
    - TCP/IP network, Unix/Windows OSes, VME, MicroTCA, etc
  - Advantages of de-facto standards
    - Economical advantage to select products out of markets
    - Save man-power avoiding proprietary development
    - Other institutes may choose the same standard
    - Solutions will be provided for the old standard in the next generation
    - As a whole, it is good for long life-span

Available Technologies

PLC

- **Programmable Logic Controllers (PLC)**
  - **Rule-based algorithms can be well-adopted for simple controls**
  - **IP network for the both controls and management were preferable**
    - **Especially at KEK/Linac which has a policy of IP only field network**
  - **~150 PLCs at Linac since 1993, and also many at J-PARC**
  - **Isolated/separated development becomes easy**
    - **Outsourcing oriented**
  - **Equipment developer oriented**
    - **Many maintenance capabilities were implemented**
  - **IEC61131-3 Standards**
    - **5 languages, with emphasis on naming**
    - **Effort to make common development environment**
    - **Should be paid more attention**
  - **Redundancy sometimes possible**
  - **Embedded Linux/EPICS possible**
    - **Efficient reuse of resources, cheap, quick, etc.**

Network with only IP/Ethernet

- **The policy chosen when we upgrade Linac in 1993**
  - **Make network management simpler**
    - **Faster switches, routing, network-booting, etc.**
  - **Avoid Hardware failure and analysis effort with old field network**
    - **Home-grown field networks need much dedicated man-power**
  - **Cost for optical Ethernet went down at around 1995**
    - **Linac has high-power modulator stations, noise source**
  - **Nowadays many facilities have this policy with GbE**
    - **J-PARC controls basically followed this**
  - **More and more intelligent network devices**
    - **ex. Oscilloscopes with Windows/3GHz-Pentium built-in**
    - **Even EPICS IOC, MATLAB, or others can be embedded**
  - **Network components can be replaced one-by-one**
  - **Security consideration will be more and more important**

FPGA

- Another "everywhere" after IP network
  - Digital circuit and software can be embedded into one chip
    - Even CPU core, Linux, and EPICS are embedded
    - Flexible and robust, wonderful platform for local controls
      - Sometime terrible source of bugs
  - Nano-second level timing
  - More and more gates, memory, pins, etc
  - More software support needed
  - Open hardware initiative may enhance the benefits

ATCA and μTCA

- **Advanced telecommunications computing architecture**
  - Accommodate many 100ohm serial buses
  - GbE or PCI-express, 10GbE, etc
  - Typically 14slots in 19” width and 12-unit height
  - Shelf manager manages healthiness of the system
    - through Intelligent Platform Management Interface (IPMI)
  - Many reliability improving facilities, redundancy, hot-swap, etc.
- **MicroTCA**
  - More recently defined in 2006, based on AdvancedMC Mezzanine Card defined in ATCA
  - Begin to acquire many facilities from ATCA
- **Future accelerators**
  - Development started for ILC, XFEL/DESY, and detectors

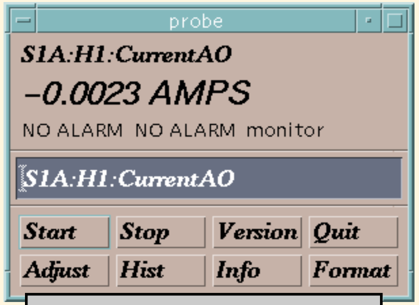
EPICS

- EPICS was chosen at many institutes, as they expected sharable software tools
  - As sharable circumstances increase, sharable software expands
- Started at LANL, one of two general purpose control software packages
  - EPICS was chosen at ANL/APS among others, and it was extended
  - Chosen at CEBAF/Jlab, DESY, etc, then chosen at SSC
  - Spread to light sources, HEP including KEKB, Fusion experiments, Observatory, etc
  - Contains all or most of basic facilities which are required in controls
  - Technique not to loose performance even with flexibility and expandability
- Open Source
  - Freeware for research institutes before 2003
  - Then after 2003 Open Source, open even to companies
- Development collaboration
  - Core group and coderathon
  - Collaboration Meeting at least twice a year
  - Mailing List
  - Web at APS <http://www.aps.anl.gov/epics/>
- Recently
  - ILC and ITER also utilize EPICS as interim base
  - Even commercial products which supports EPICS

EPICS software components

- 'Base' normally provided by manager, but can be tested by you easily
  - Retrieve EPICS Base (ex. baseR3.14.12.tar.gz)
  - Expand it
  - set EPICS_BASE=`pwd`
  - setenv EPICS_HOST_ARCH `startup/EpicsHostArch.pl`
  - make (Basic software including libraries are built)
  - $EPICS_BASE/bin/$EPICS_HOST_ARCH/makeBaseApp.pl -t example myapp
  - $EPICS_BASE/bin/$EPICS_HOST_ARCH/makeBaseApp.pl -l -t example myapp
  - make (Example application programs are built)
  - bin/$EPICS_HOST_ARCH/myapp st.cmd
    - For example
      - <http://www-linac.kek.jp/cont/epics/linux/>
- Extensions or clients are developed at world wide institutes
  - Feedbacks from mailing list or collaboration meetings refine the software
  - Each institute chooses some of the available software and integrate

Channel Access Commands



S1A:H1:CurrentAO

-0.0023 AMPS

NO ALARM NO ALARM monitor

S1A:H1:CurrentAO

Channel Access Client

CA Client

CA Server

Channel Access Server

Process Variables:

S1A:H1:CurrentAO

S1:P1:x

S1:P1:y

S1:G1:vacuum

"connection request" or "search request"

Who has a PV named "S1A:H1:CurrentAO"?

I do.

"get" or "caget"

What is its value?

25.5 AMPS

"put" or "caput"

Change its value to 30.5

OK, it is now 30.5

"put complete"

or

30.5 is too high. It is now set to the maximum value of 27.5.

or

You are not authorized to change this value

"set a monitor"

Notify me when the value changes

It is now 20.5 AMPS

It is now 10.5 AMPS

It is now 0.0023 AMPS

"post an event" or "post a monitor"

EPICS

- Now is a kind standard, but …
- Object-oriented design support is limited now
  - Naming scheme, and/or design of new record
  - More software-engineering support favored
    - Several different efforts to provide better environment
      - Java IOC (M. Kraimer), Control system studio (M. Clausen), Data access (R. Lange)
- Security mechanisms
  - User, Host-based protection available
  - More security
    - Dynamic controls of security
    - Access logging/auditing, authentication, transaction mechanism
- Dynamic configuration of database
  - Dynamic creation / loading of records
  - Dynamic removal of records
    - Maybe some part of the codes can be shared with redundant-IOC project

Magnet Controls

- It is typical controls and still many things to do
- Many magnets and many power supplies
    - No one-to-one correspondence. One-to-many, several-to-one
  - Which hardware interface to use?
- Procedures
  - Interlock status, on/off, analog with some precision, etc
  - Energy, kick - field - current conversions
    - How to represent those conversion curves
  - Timing synchronous operation
    - for tune change, orbit correction, etc.
  - Standardization

Event System

Old Timing System at Linac

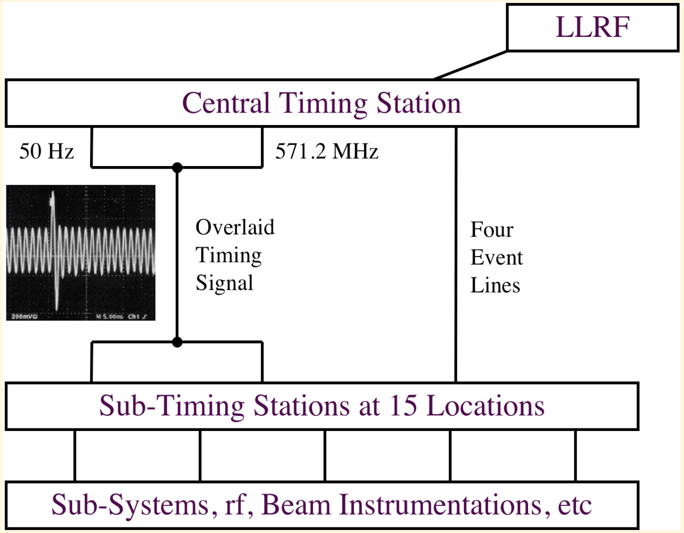
- Provides ~3pico-second Timings to ~150 Devices
- Slow (0.1~100 sec) global controls
- VME(x6) and CAMAC(x10)

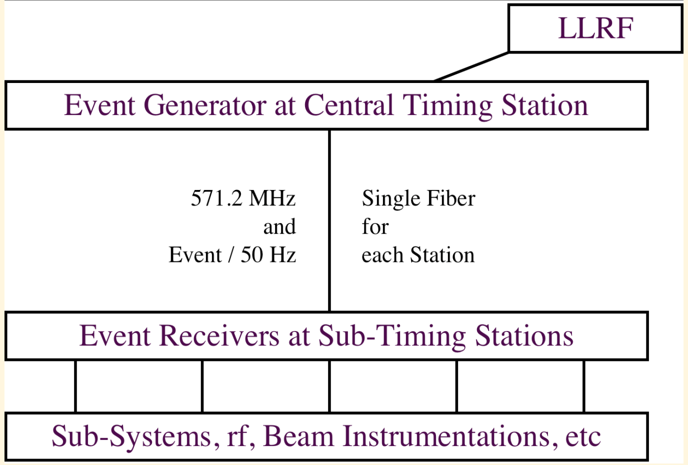
MRF Event System (EVG/EVR)

- Single Fiber can Transfer Clock, Delayed-Timings, Events (256), Data Buffers (2k-bytes)

Event system at Linac

- Fast and Synchronized global controls
- VME-based
- VxWorks/RTEMS
- EVG/EVR
- EPICS Driver/Device Support from community

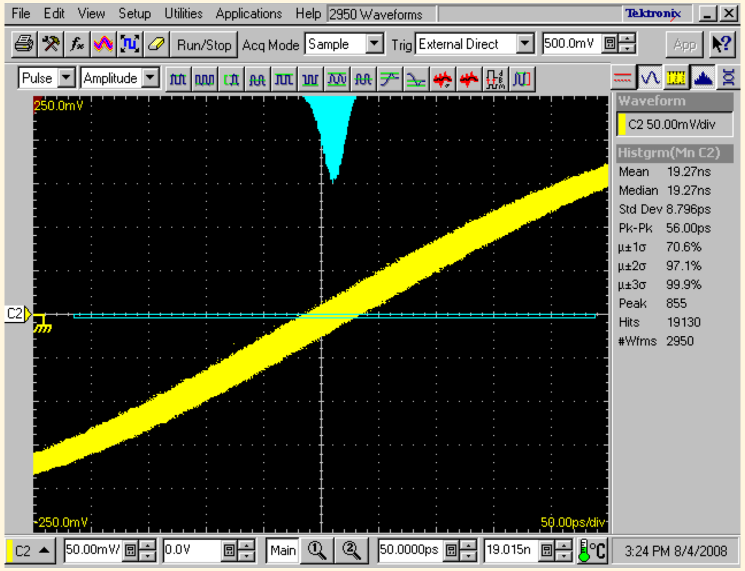

LLRF
Central Timing Station
50 Hz
571.2 MHz
Overlaid Timing Signal
Four Event Lines
Sub-Timing Stations at 15 Locations
Sub-Systems, rf, Beam Instrumentations, etc


LLRF
Event Generator at Central Timing Station
571.2 MHz and Event / 50 Hz
Single Fiber for each Station
Event Receivers at Sub-Timing Stations
Sub-Systems, rf, Beam Instrumentations, etc

Linac Event System

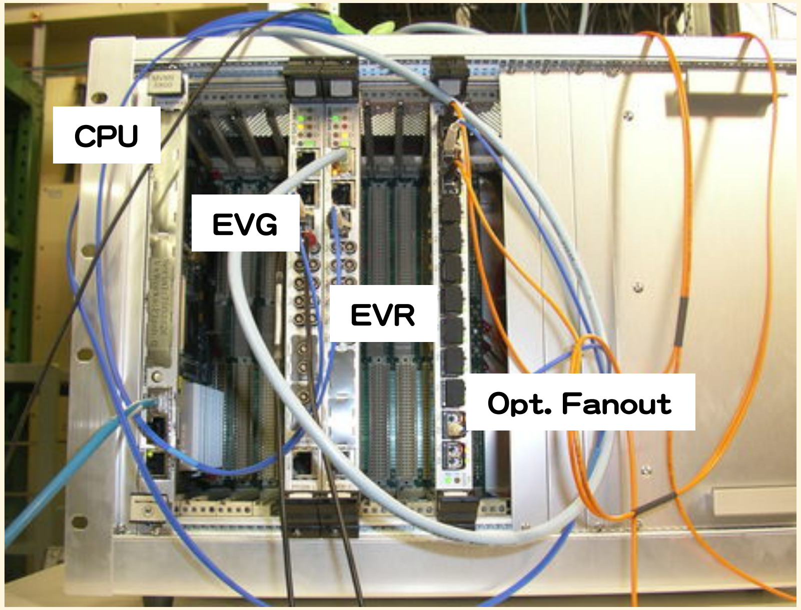
Basic parameters

- Event rate : 114.24MHz
- Fiducial rate : 50Hz
- Bit rate : 2284.8Mbps
- Timing jitter (Short term) : ~8ps
- No. of defined events : ~50
- No. of receiver stations : 18
- No. of Fast parameters : ~150
- Data rate : 2kbyte / ~36μsec



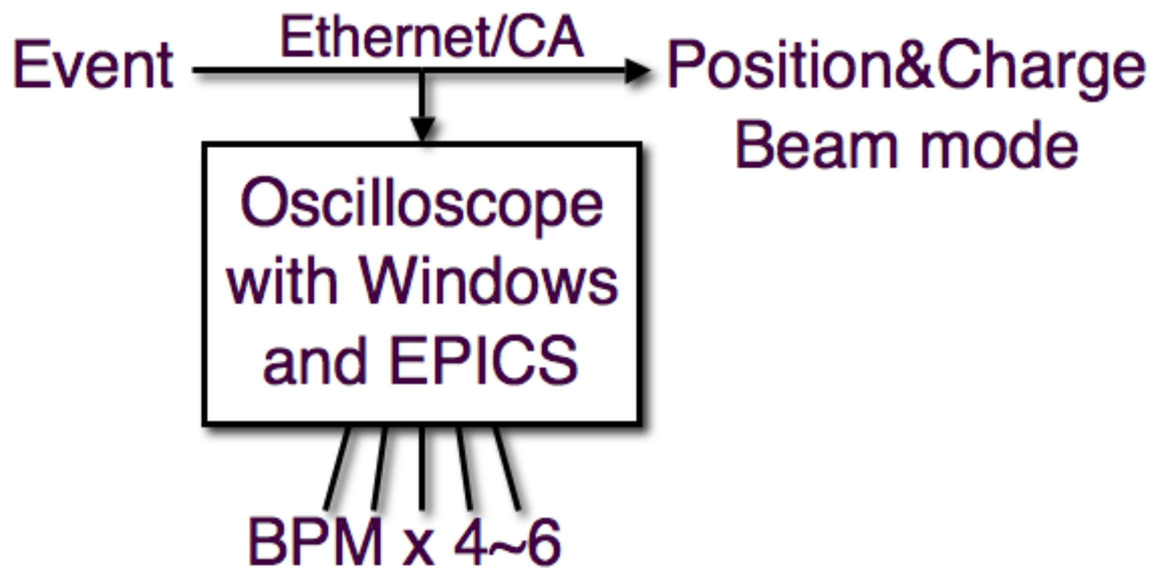
```
Waveform
C2 50.00mV/div
Histgrm(Mn C2)
Mean     19.27ns
Median   19.27ns
Std Dev  8.796ps
Pk-Pk    56.00ps
µ±1σ     70.6%
µ±2σ     97.1%
µ±3σ     99.9%
Peak     855
Hits     19130
#Wfms    2950
```

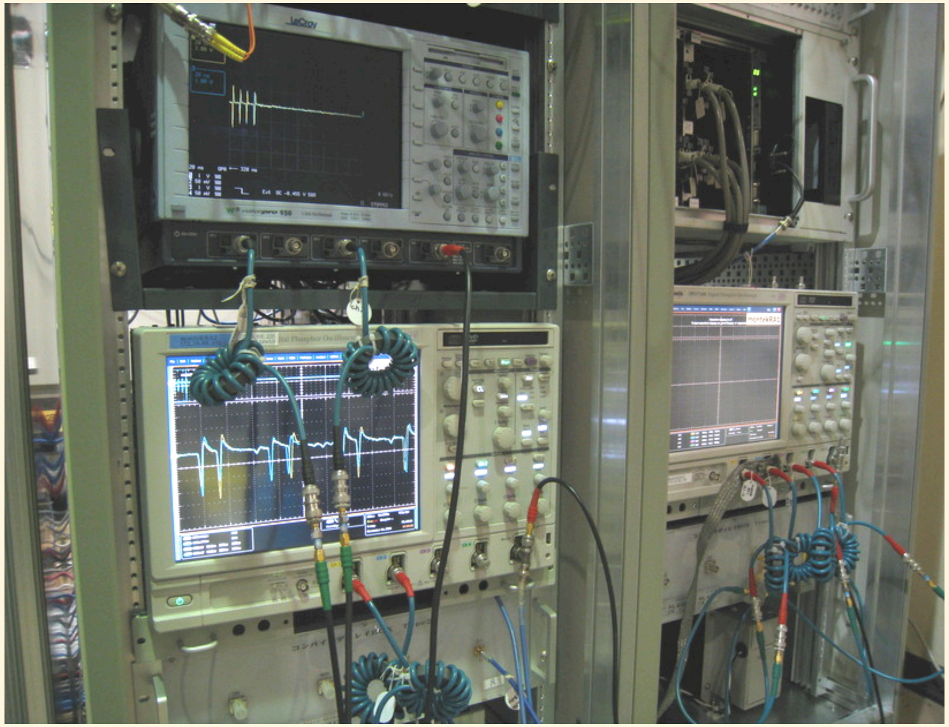
```
C2  50.00mV/  0.0V  Main  50.0000ps  19.015n  °C  3:24 PM 8/4/2008
```



BPM

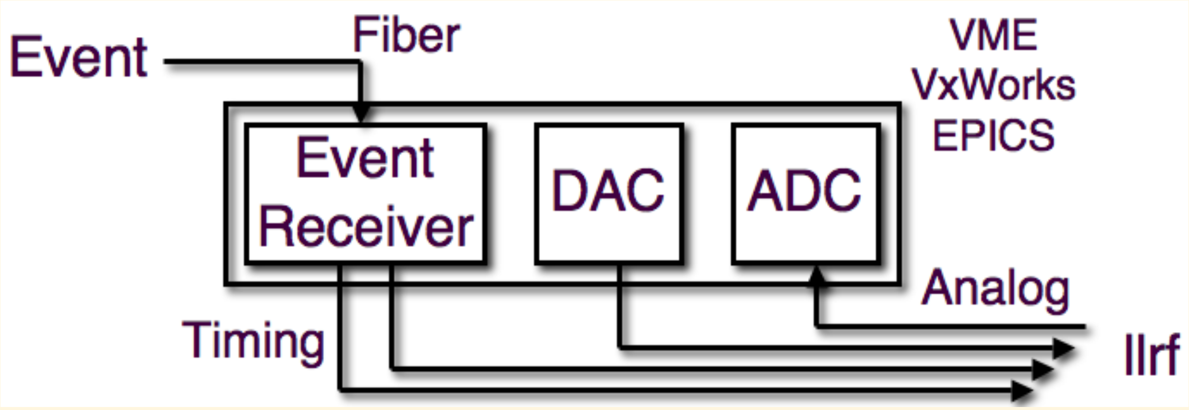
- DPO7104 can acquire data in 50Hz .
- Beam modes are recognized by events through network.
- Clients can monitor data of an interested beam mode.
- 100 BPMs are synchronized.
- For SuperKEKB we re-design completely for higher precision

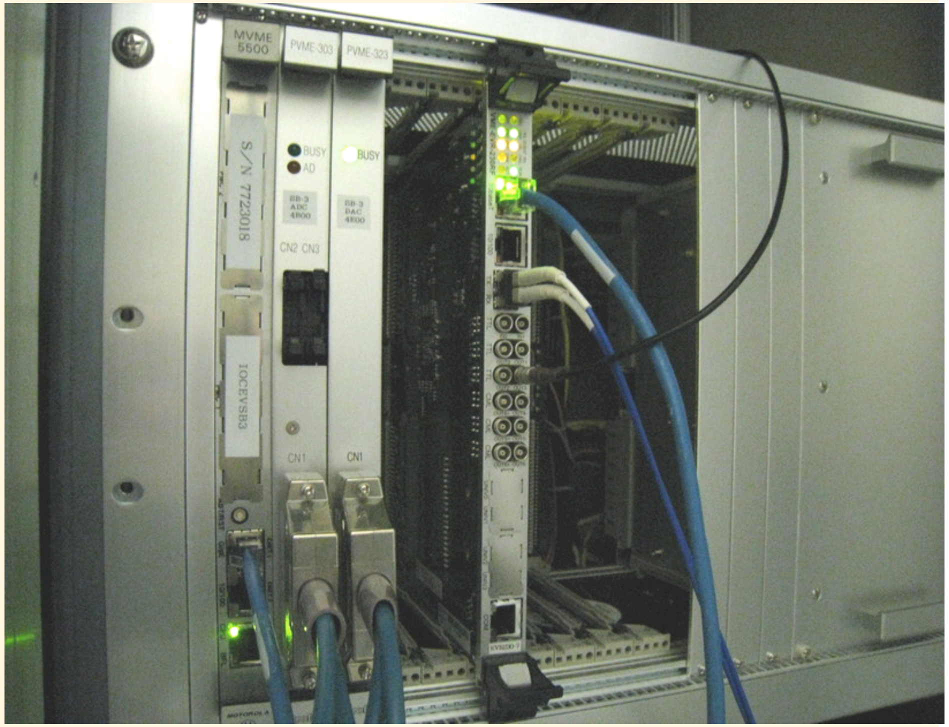




LLRF

- Timing and analog signals are essential for absolute energy, energy spread, and dual-bunch energy equalization.
- Signals can be switched pulse-by-pulse.
- Driver klystrons (SB), energy tuner klystron (KL), and sub-harmonic bunchers (SH) are managed by the event system.





RF Group

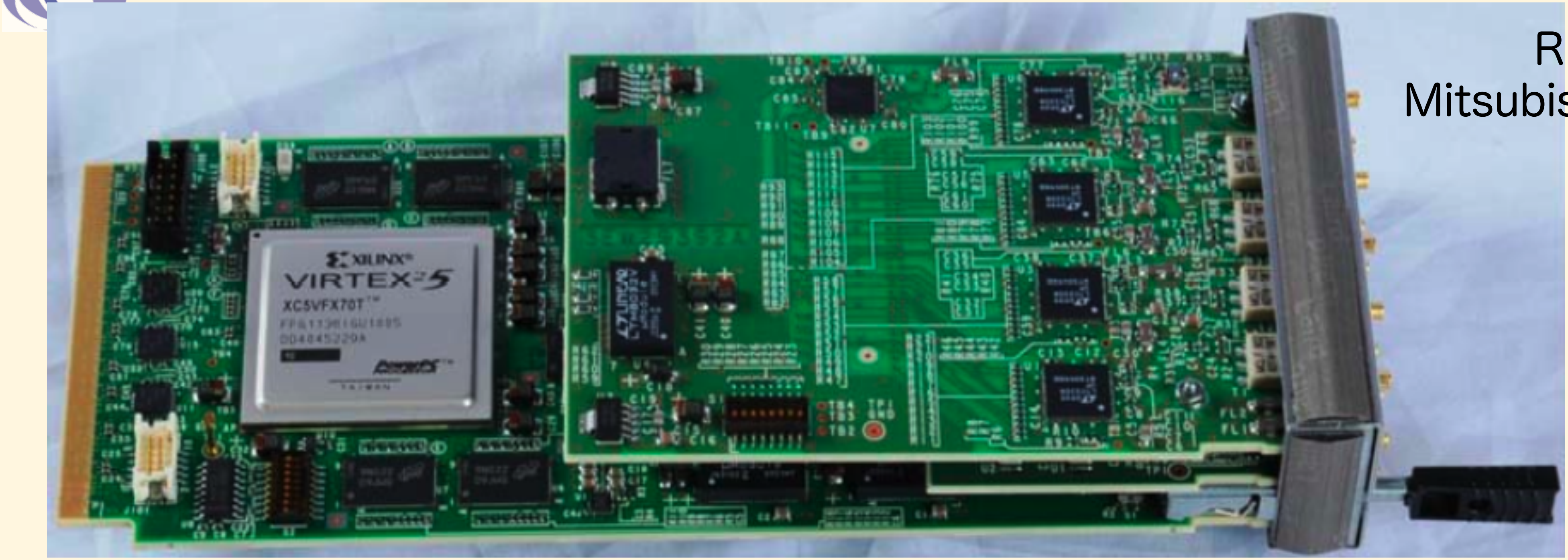
MicroTCA based LLRF Controller

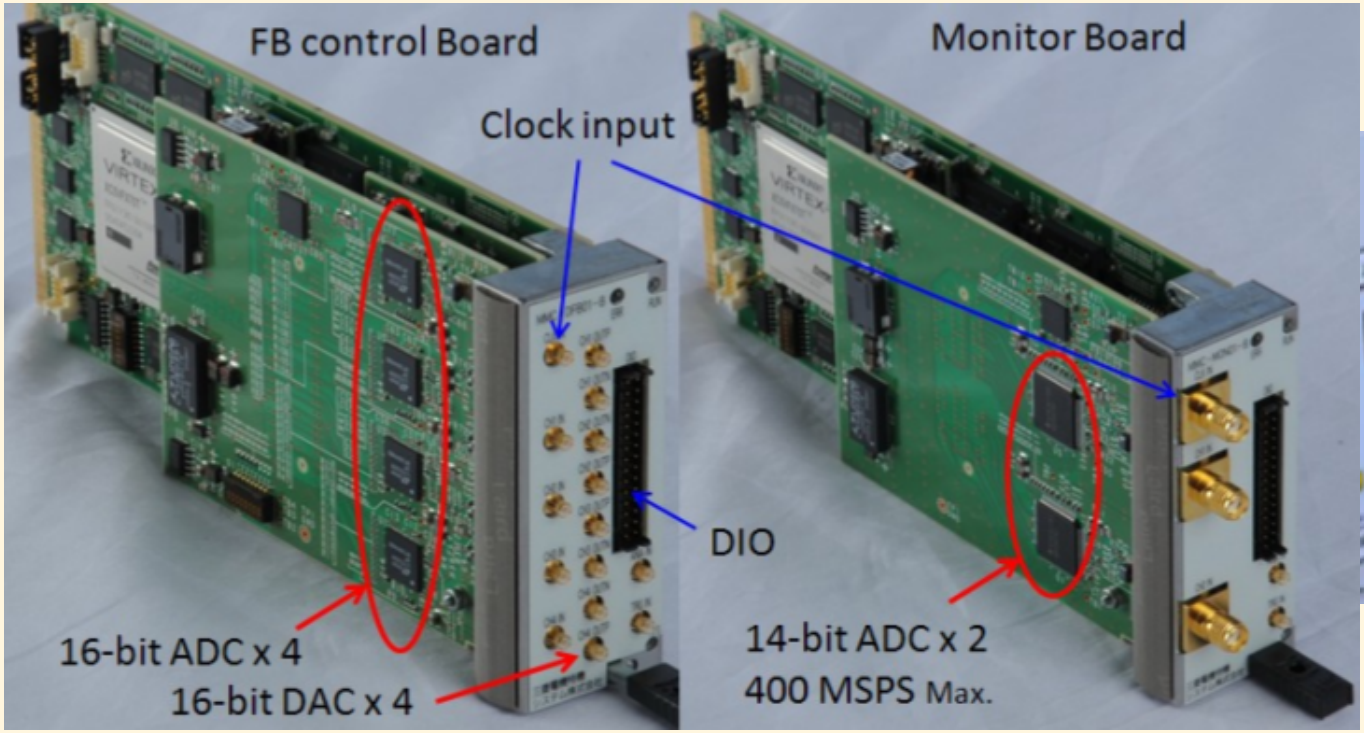
- Single-width full-height module
- Without physics experiment extension (MTCA.4)
  - We started earlier
  - Front-panel connectors only (rather busy)
- Digital part and Analog part are on isolated cards
  - ADC 16bit, 130Msps, x4
  - DAC 16bit, 500Msps, x4
  - Virtex5 with PPC440
  - RAM 640MB, Flash 64MB
  - Also monitor card employing the same digital part
    - ADC 14bit, 400Msps, 1.4GHz, x2
- Fabrication was carried at Mitsubishi Electric Tokki System

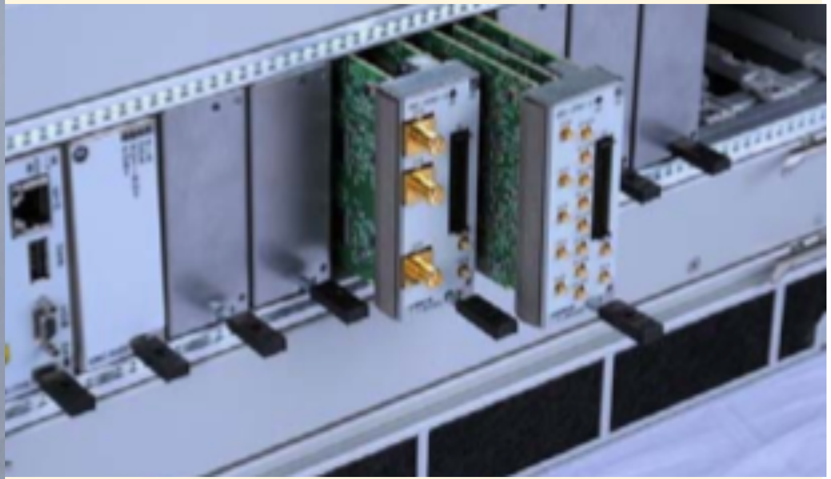
<http://www-linac.kek.jp/cont/epics/mtca/>

New LLRF Controller

RF Group
Mitsubishi Tokki







Reliability

Reliability

- **The end user expect rigid reliable operations**
- **Inner layers need flexibilities**
  - **Because of daily improvement**
- **Compromise between**
  - **Practical or ideal solutions**
  - **Aggressive and conservative**
  - **Under restrictions of**
    - **Time, safety, budget, man-power**
- **Here we think about adaptive reliability for 99.999% availability**

hardware
hardware Interface
equipment controls
beam controls
linac
ring
accelerator physics
beam delivery
detector
data acquisition
computing
physics, chemistry,
medical treatment

Reliability Increase without much Cost

- There should be “right way”
  - We hope to have it some day, but for now we need interims
- Surveillance for everything
  - Well-arranged system does not need this, but…
- Testing framework
  - Hardware/Middleware tests just before Beam
  - Software tests when installed
- Redundancy
  - In Many Hardware/Software components
  - Of course some of them are Expensive, but…

Possible Right ways for the Reliability

- Well-defined classes for accelerator controls
  - It surely reduces the coding errors
    - But one may insist on a complete class of accelerator
      - which we can never achieve
- Well-arranged naming conventions
  - It surely reduces human-operation errors
  - And enables more computer-aided tools
    - But real world was not so simple
- Well-specified deployment procedures
  - Enables more computer-aided tools
  - Even scripting languages were said to be bad
    - But we are lazy

Surveillance for everything

- We have written too many pieces of software
  - which assume certain circumstances unfortunately
    - which will fail some day
  - in scripting languages too rapidly and too easily
    - without documentations
- We manage too many computers
  - If only one, I’m almost sure I can make it stable
    - But in reality even hostname can be mis-labeled
- We installed too many network components
  - without good network database etc
    - which sometimes has bad routing information, etc
- We need surveillance

Surveillance for everything

- If certain installation of (software/hardware) was not ideal
  - Find out
    - What is the most important feature of the installation?
    - What is the easiest test for its healthiness?
  - Routine test is carried automatically
    - by cron or continuous scripts
    - If an anomaly found,
      - Alarm, e-Mail to the author, make error log
      - Restart related software, if not critical
      - Report to the human operator, if critical
  - Not ideal, but effective under limited human resources

Software Testing

- Moving operating environment
  - For better resource performance
    - We tend to do it because of the pressure from budget restrictions
  - May lead to malfunctions
    - We knew they may happen
- Automatic software (hardware) tests preferable
  - Under new environment (machine, compiler, network, etc)
    - Many kinds of important free software does them
    - Language systems, Linux Test Project
- We do some tests
  - But sometimes not enough
  - More thoroughly prepared tests needed

Testing Framework

- When we introduce new environment
  - Unit test
    - We don’t do it much yet
    - EPICS began to have it, “make runtests”
      - Collecting existent test cases
      - User can provide tests in Perl/Test framework
    - Hope to have for SAD and SADscripts
  - Regression tests
    - We have something, but not thorough, not exhaustive
    - Difficult to collect cases
  - Stress tests
    - We do it during operation (?)
    - We know computers rarely fail, but network/network-devices do
      - Find solution
      - Development of surveillances
      - Installation of failure-recovery or failover procedures

Testing Framework

- When we start new run
  - New software/hardware
    - We test unit by unit
    - But not through operational tools prepared
  - Maintenance works
    - We often forget to restore/initialize cables, switches, variables
    - Power-stop may bring another annoyance
- We need routine procedures which include
    - Hardware tests
    - Name/ID matching
    - Database tests
    - Software component tests
    - Software/Hardware simulation tests
  - Before beam operation
  - We do it mostly by operator observations based on written procedures
  - There were efforts to automate in the past

Redundancy

- Do we need redundancy?
  - Redundancy may be the last-resort measure
  - It may cost
  - Centralized facilities are easier to manage
    - If I have only one server, my life is much easier
  - But they become complicated monsters
    - Nobody understand everything
- Especially useful for maintenance
  - Not only for failure-recovery
    - Redundant systems of complicated system; $(\text{complicated})^2$
- Anyway we may have to prepare backups
  - Then automatic failover is just around the corner
    - And …

Redundant EPICS IOC

- **Redundant controllers are favorable**
    - as in PLCs
- **The project was started at DESY (M. Clausen)**
    - **Redundancy monitor task (RMT)**
        - Monitors healthiness of controllers
        - Manages primary redundancy resource (PRR)
    - **Continuous control executive (CCE)**
        - Synchronizes internal states
    - **Modifications for several others PRR's**
        - Scan tasks, Channel access server tasks, Sequencer, Drivers
        - Possibly user tasks
- **KEK joined in for wider applications**
    - **Linux (OSI) port**
    - **Gateway applications**
- **ATCA implementation possible**
    - **For ILC (?), microTCA (?)**

Software redundancy

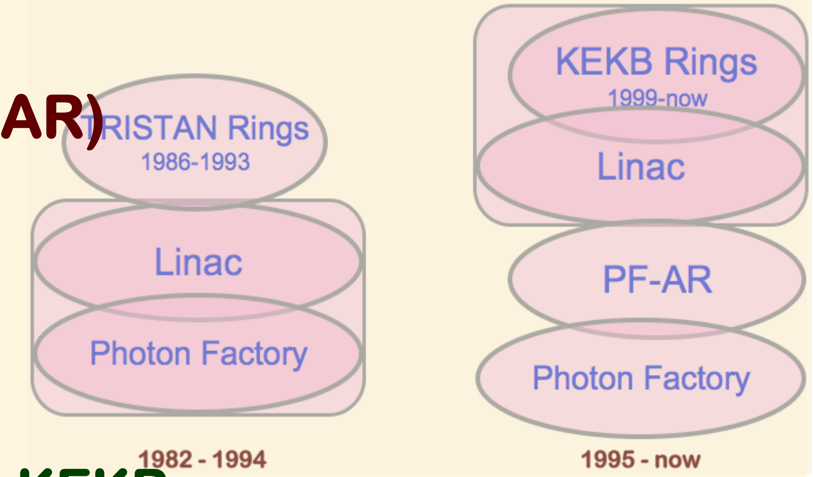
- EPICS IOC redundancy is slightly complicated
  - Since it has name resolution facility
  - More advanced
- Linac/KEK controls is simpler
  - Normally we run several middle-layer control servers
    - on separate machines
  - For EPICS gateway
    - We need redundant IOC technology
- Other existent servers
  - Recently more careful in redundancy
    - Like dchpd
    - Redundancy and replications

Past KEKB

KEKB and Linac Control Systems

Linac

- Controls Upgrade (1990~)1993
  - De-facto (and International) Standards, IP-only Networks
- No long Shutdown for KEKB upgrade
  - 3.5-times Energy increase, 10-times current increase
- Division changed at the end of Upgrade
- Three indirect User Facilities (KEKB, PF, PF-AR)
- Fewer resources



KEKB

- 5-year Shutdown after TRISTAN 1994-1998
  - Precision requirements were much different for KEKB
- Complete transition of Controls
  - from Nodal at TRISTAN to EPICS+SAD at KEKB
- Basically Single-user (Belle)

Equipment Controllers at Linac

- 1982~(1997) (1st generation)
  - 300 microprocessor-based controllers
    - Linked together with home-grown fiber-optic network
- 1993~now (upgrade of controls)
  - 150 PLCs (programmable logic controller)
    - Linked via only Fiber-optic Ethernet/IP
      - Control communication with servers and program development
- 1995~now (upgrade for KEKB)
  - 30 VXI for rf measurement
  - 5 VME / 10 CAMAC for Timing
  - 20 VME for Beam monitors
- 2006~ (upgrade of BPM readout)
  - 24 Oscilloscopes with WindowsXP IOC for 100 BPMs
    - 10Gs/s, 50Hz acquisition, local processing with 20 calibration parameter/BPM

Equipment Controllers at KEKB

- TRISTAN
  - Mostly CAMAC
    - Equipment group responsibility: CAMAC module and outside
- KEKB
  - 100 VME/IOC without Analog processing
  - 200 VXI/MXI mainframes for 900 BPMs
  - 50 CAMAC crates are kept for rf and vacuum
  - ARCNet boards for Magnet ps. settings, and others
  - GPIB for Magnet ps. readback, and others
  - PLCs for Magnet interlocks, and others

EPICS Transition at Linac

- **Home-grown RPC at Linac (1990~)**
  - Bad timing but no choice because of end of old mini-computer support
- **LynxOS Transition was developed (1994~1996)**
  - To cover both RPC and EPICS with pthread, posix
    - Mostly working, Failed to get funding for Hardware/Software upgrade
- **Gateways to EPICS in several ways**
  - Software-only IOC and Gateway (Clients to both RPC/CA)
  - Portable Channel Access Server of EPICS-3.12 (1995~)
  - Soft-IOC with device support to Linac RPC (2002~)
- **Real IOCs are increasing**
  - PLC(rf,vacuum,magnet) and Linux, Oscilloscope(bpm) with Windows, VME(llrf and timing)
  - RPC servers read EPICS IOCs, EPICS gateways read RPC servers

EPICS Transition at KEKB

- Some candidates discussed after Nodal at TRISTAN
  - RPC/CORBA based control design
  - Reflective memory (hardware shared memory) design
- No other choice than EPICS for KEKB
  - No man-power for control system software
  - The choice at SSC
  - International collaboration was attractive

Archiver/Logger

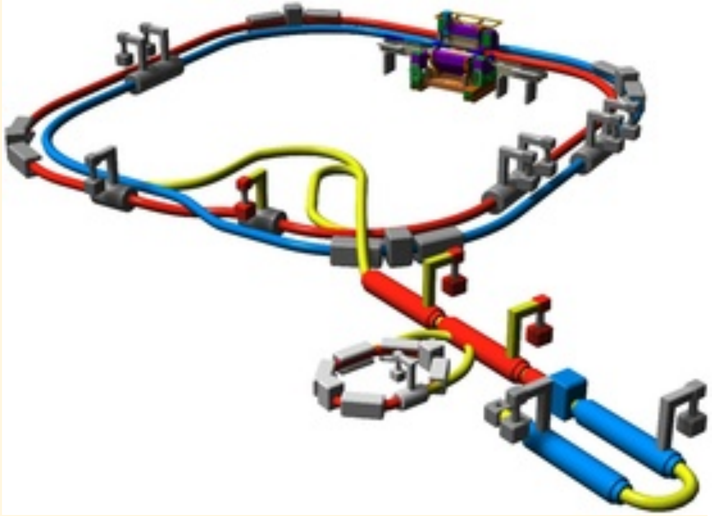
- **Linac**
  - Several archivers with different filters and stored in ascii
  - Replaced with two EPICS archivers (2002)
    - Channel archiver, with Java viewer, and Web-based viewer
    - Transition to CSS-based channel archiver
    - KEKBlog, SADscript-based viewer
      - Both ~400MB/day, Dynamic ADEL changes
- **KEKB**
  - KEKBlog, since 1998
    - Once there was a plan to replace it with Channel Archiver
      - Data conversion, no much performance difference
    - Only ADEL-based filter
      - 4GB/day
    - SADscript-based viewer is one of the most used applications
      - With Data analysis capability, easy manipulations

Scripting Languages

- Heavy use because of rapid prototyping
- Linac
  - (1992~) Tcl/Tk as Test tools on Unix
  - (1997~) Tcl/Tk as Main Operator Programming Tool
  - (Now) Mixture of SADscript/Tk, Python/Tk, Tcl/Tk
    - SADscript has most accelerator design capability
      - Covers many features like MATLAB, Mathematica, XAL, MAD
- KEKB
    - (Nodal interpreter and Fortran covered everything at TRISTAN)
  - Python covers many areas which is not covered by medm
  - SADscript is used by operators and physicists everyday
    - Realization of novel ideas in hours
      - Only some ideas are effective, so rapid prototyping is important

Towards SuperKEKB

SuperKEKB



- **Electron-positron asymmetric collider**
  - **Based on a decade of successful operation at KEKB**
- **Aims at 40-times higher luminosity**
  - **8 x 10^{35}cm^{-2}s^{-1} for further flavor physics studies**
  - **7GeV / 2.6A electron, 4GeV / 3.6A positron**
  - **β_y* ~ 0.3mm, $\varepsilon_x / \varepsilon_y$ ~ 4nm/9pm, σ_y ~ 50nm, σ_z ~ 6mm**
  - **Ante chamber, longer bend, damping ring, rf gun,etc**

KEKB Controls 1998 - 2010

- **EPICS as Main control Software Toolkit**
  - Became one of de-facto standard at 1995
  - Several fieldbuses were incorporated
    - VME, VXI, CAMAC, ArcNet, GPIB, etc
  - Reduced software design efforts much
- **Scripting Languages for Operational Software**
  - SADscript/Tk, Python/Tk, Tcl/Tk used much
    - Especially, SADscript as a bridge btw. Accelerator simulation, Numeric manipulation, Graphic interface and EPICS controls
  - Bright new idea in the morning meeting could make the operation much advanced in the evening
    - Great tool to optimize the operation

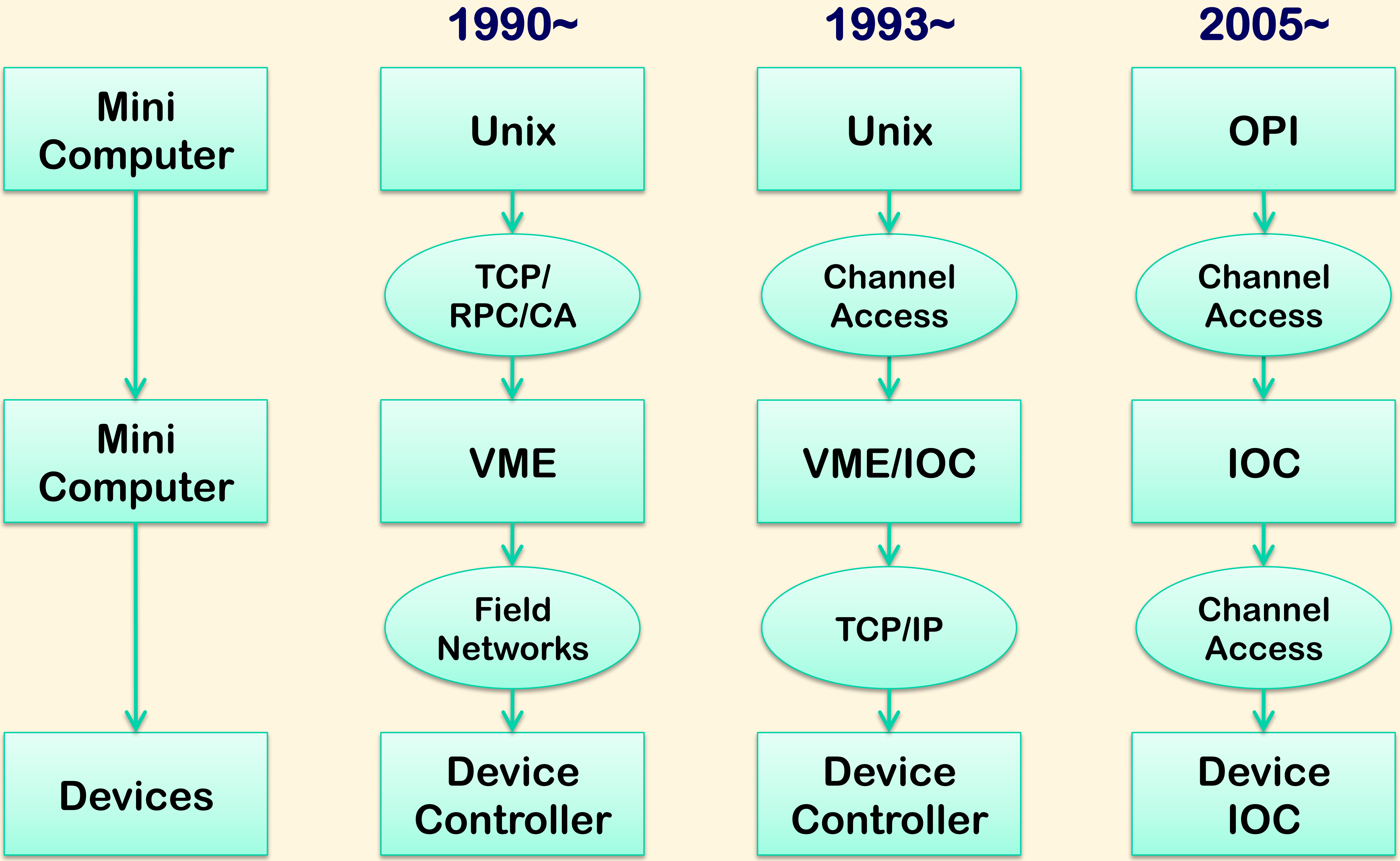
SuperKEKB Controls

- Inherit Good part of KEKB Controls
  - EPICS
  - Scripting languages
  - With simple rejuvenation of software/hardware
- Two Additional Concepts

1st: CA Everywhere

- EPICS Channel Access (CA) Everywhere
  - Or EPICS anywhere
- Embed EPICS control software (IOC) everywhere possible
- Reduce efforts on protocol design, testing, etc

Transition of Architecture



Overview of controls at KEK

- VME + Unix (1990~)
  - Standard model (later EPICS) configuration
    - With several fieldbuses

↓

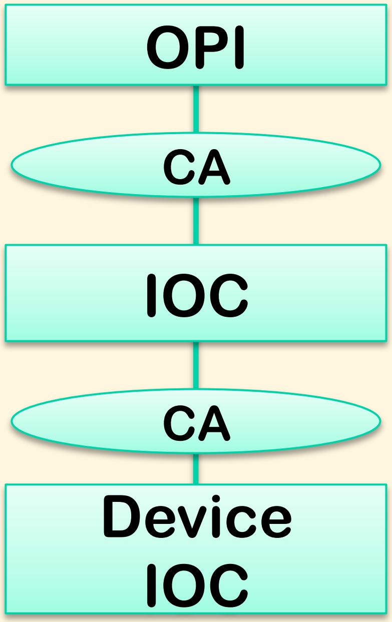
- Every controller on IP network (1993~)
  - 2-layer physical, 3-layer in logical (Linac, J-PARC)

↓

- Every controller with EPICS IOC (2005~)
  - Channel Access everywhere (CA Everywhere)
    - Good for rapid development and smooth maintenance
    - May need some consideration on network management

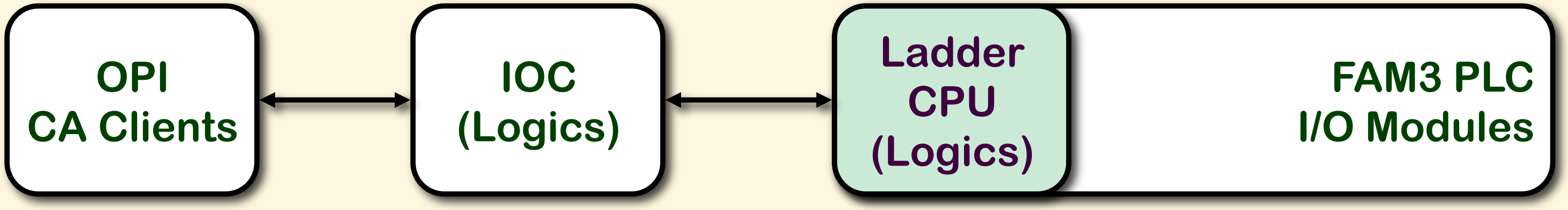
Embedded EPICS IOCs at (Super)KEKB

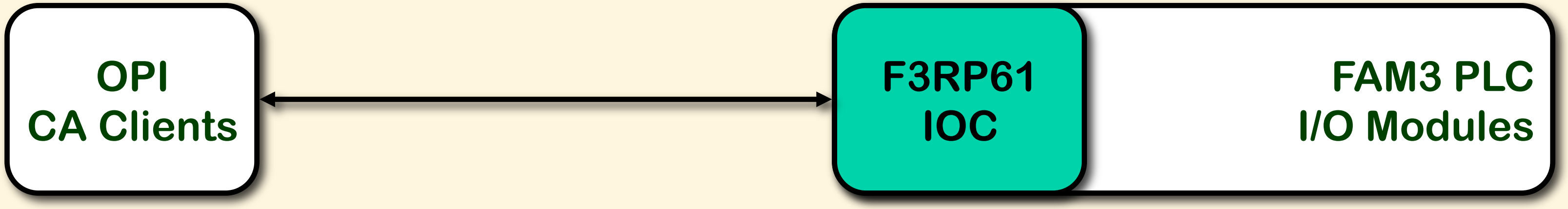
- Not only information server, but also the same software framework on every controller
  - Rapid development and smooth maintenance
- μTCA LLRF module: Linux/FPGA (Odagiri…)
- Yokogawa PLC: Linux CPU (Odagiri…)
- Oscillo. 50Hz measurement: Windows (Satoh…)
- MPS management :Linux/FPGA (Akiyama…)
- Timing TDC: Linux/Arm (Kusano…)
- Power modulator: Linux/FPGA (Kusano…)
- Libera BPM at 50Hz: Linux/FPGA (Satoh…)
- NI cRIO : CAS/FPGA (Odagiri…)
- Many more…



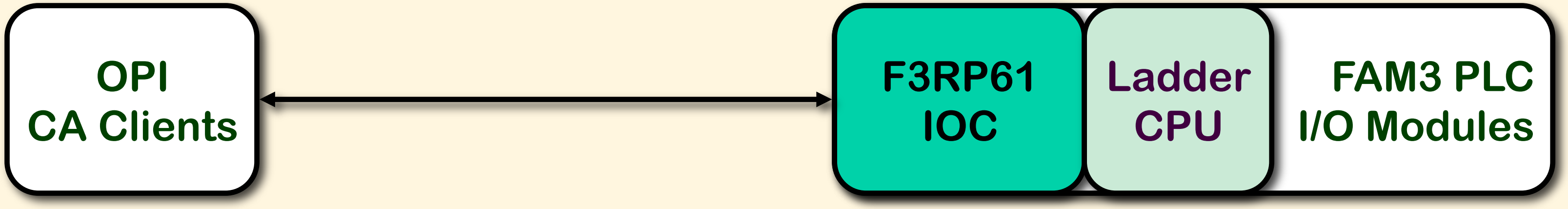
Simpler PLC Usage under EPICS

Conventional PLC usage with asynchronous access





If necessary, we can combine



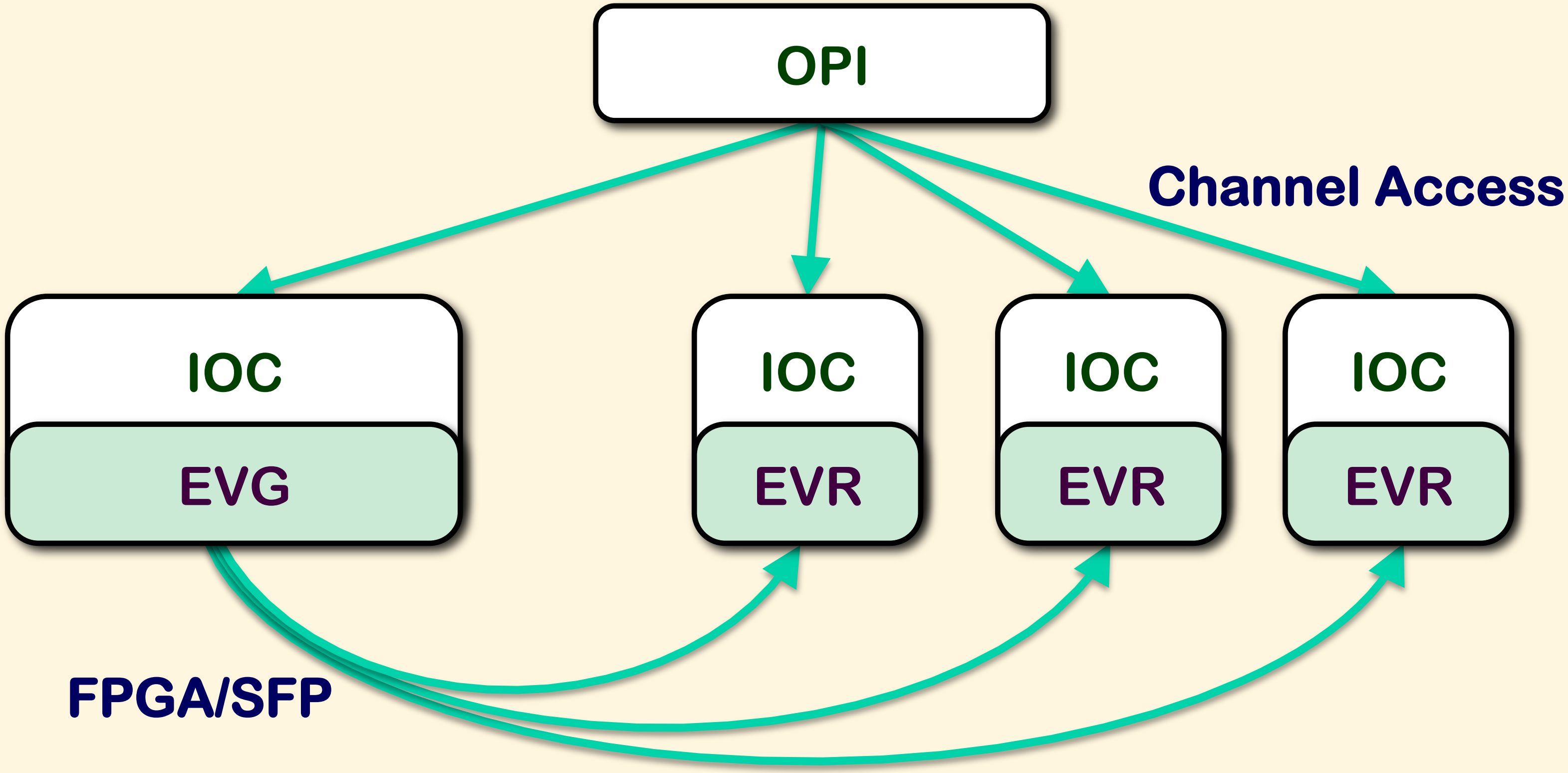
Logics are confined in PLC, and management is easier

2nd: Dual-layer Controls

- Another layer in addition to EPICS/CA
  - Event system helps EPICS with another channel
  - Additional functionality, synchronization and speed

Dual-layer Controls

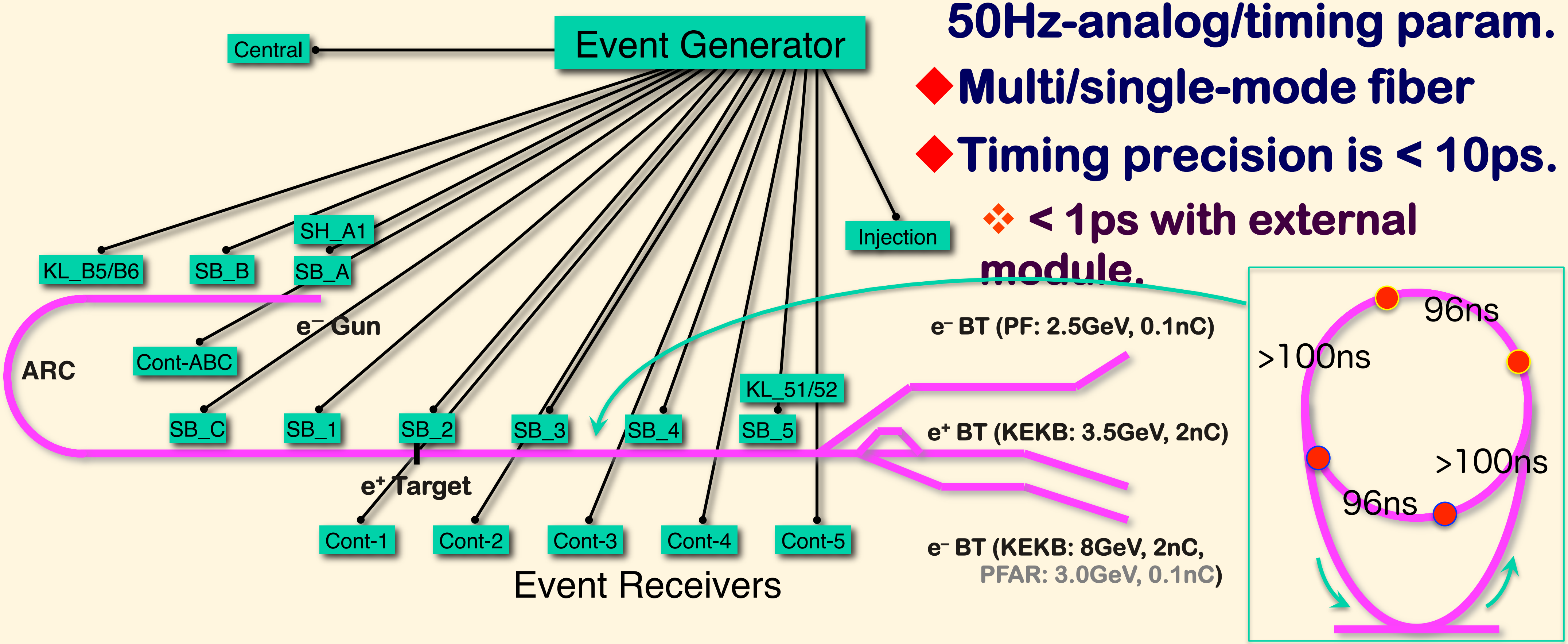
- **IOC controls via Conventional EPICS CA**
  - **Above 1ms, ordered controls**
- **Fast FPGA controls via SFP/Fiber**
  - **10ps ~ 100ms, 114MHz synchronous controls**



Fast Global Synchronous Controls

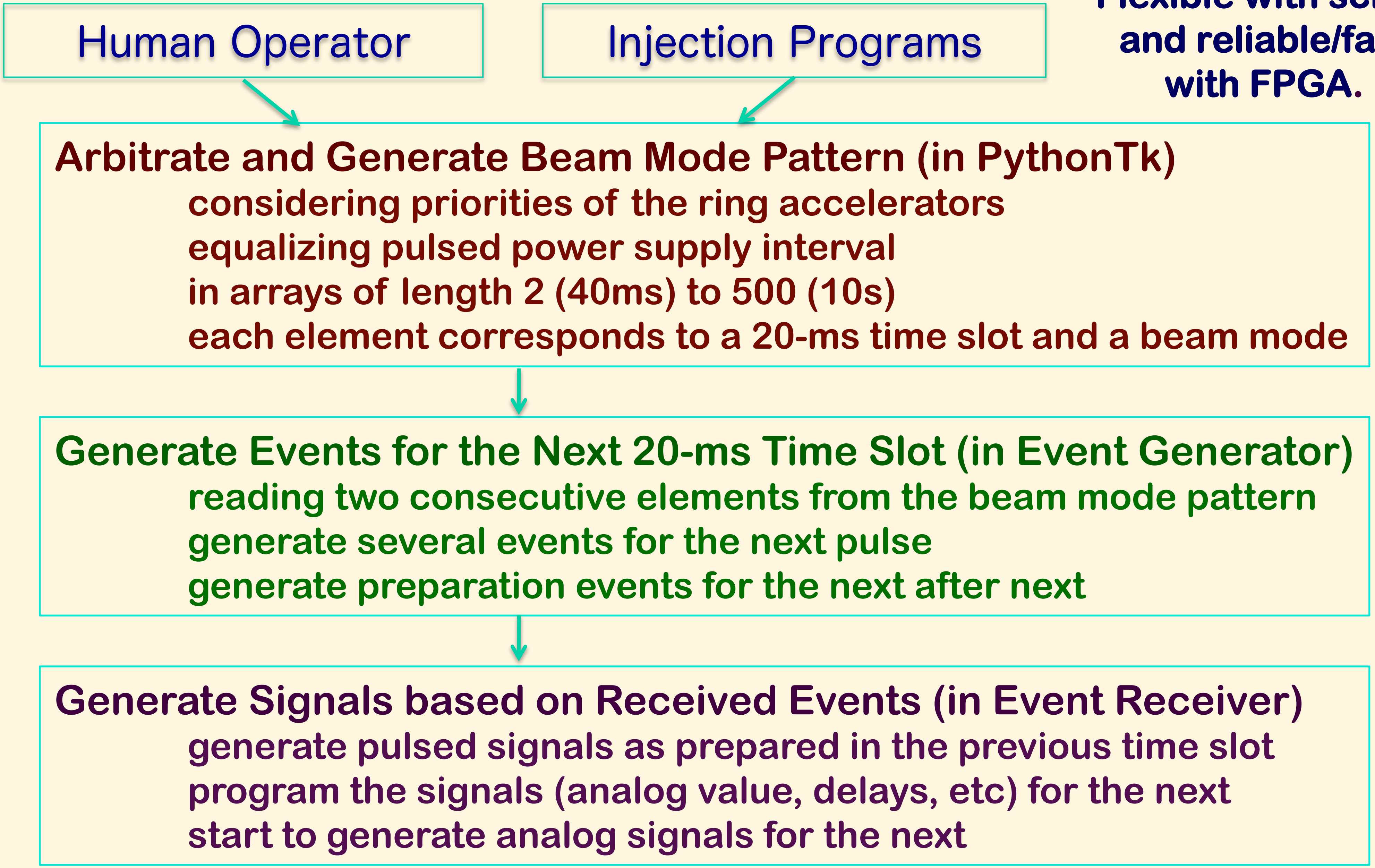
- MRF’s series-230 Event Generator / Receivers
- VME64x and VxWorks v5.5.1
- EPICS R3.14.9 with DevSup v2.4.1
- 17 event receivers up to now

- 114.24MHz event rate, 50Hz fiducials
- More than **hundred** 50Hz-analog/timing param.
- Multi/single-mode fiber
- Timing precision is < 10ps.
  - < 1ps with external module.



Event Manipulation

Flexible with script and reliable/fast with FPGA.



Human Operator

Injection Programs

Arbitrate and Generate Beam Mode Pattern (in PythonTk)

considering priorities of the ring accelerators
equalizing pulsed power supply interval
in arrays of length 2 (40ms) to 500 (10s)
each element corresponds to a 20-ms time slot and a beam mode

Generate Events for the Next 20-ms Time Slot (in Event Generator)

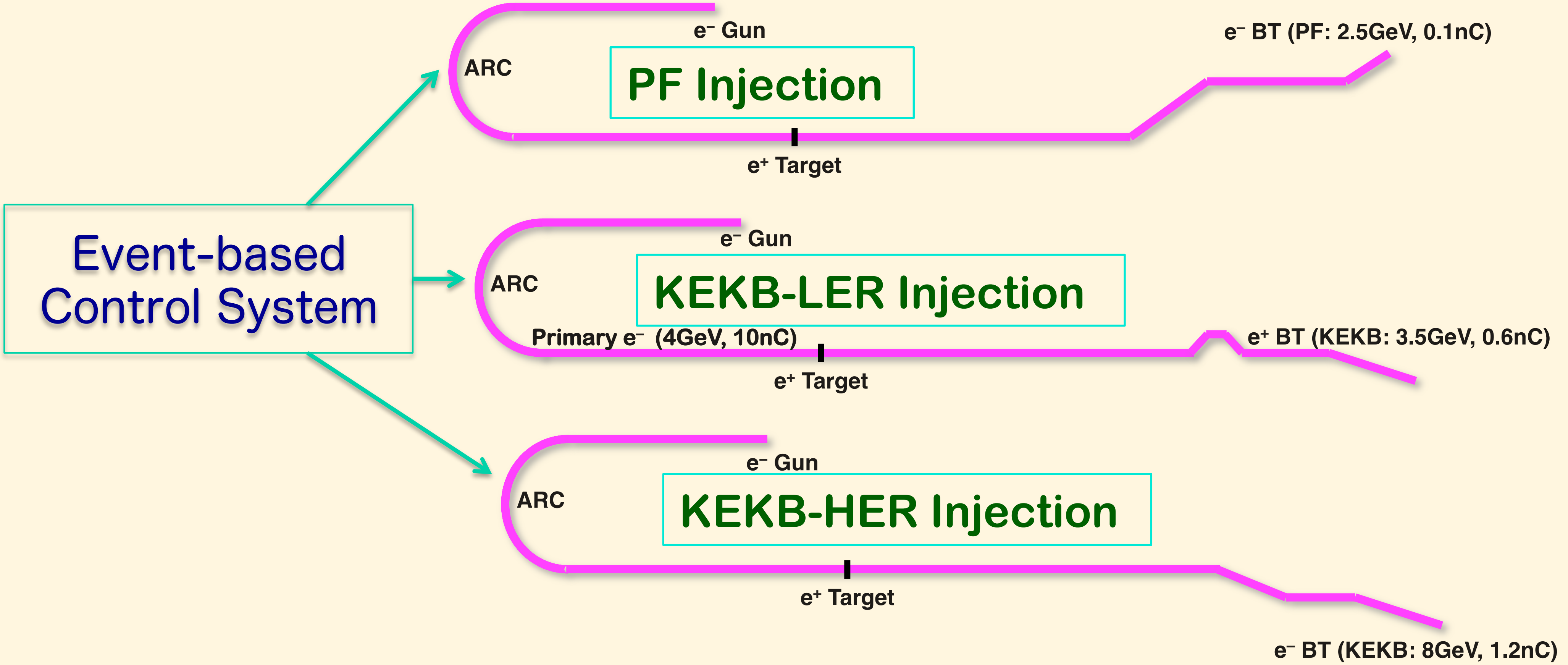
reading two consecutive elements from the beam mode pattern
generate several events for the next pulse
generate preparation events for the next after next

Generate Signals based on Received Events (in Event Receiver)

generate pulsed signals as prepared in the previous time slot
program the signals (analog value, delays, etc) for the next
start to generate analog signals for the next

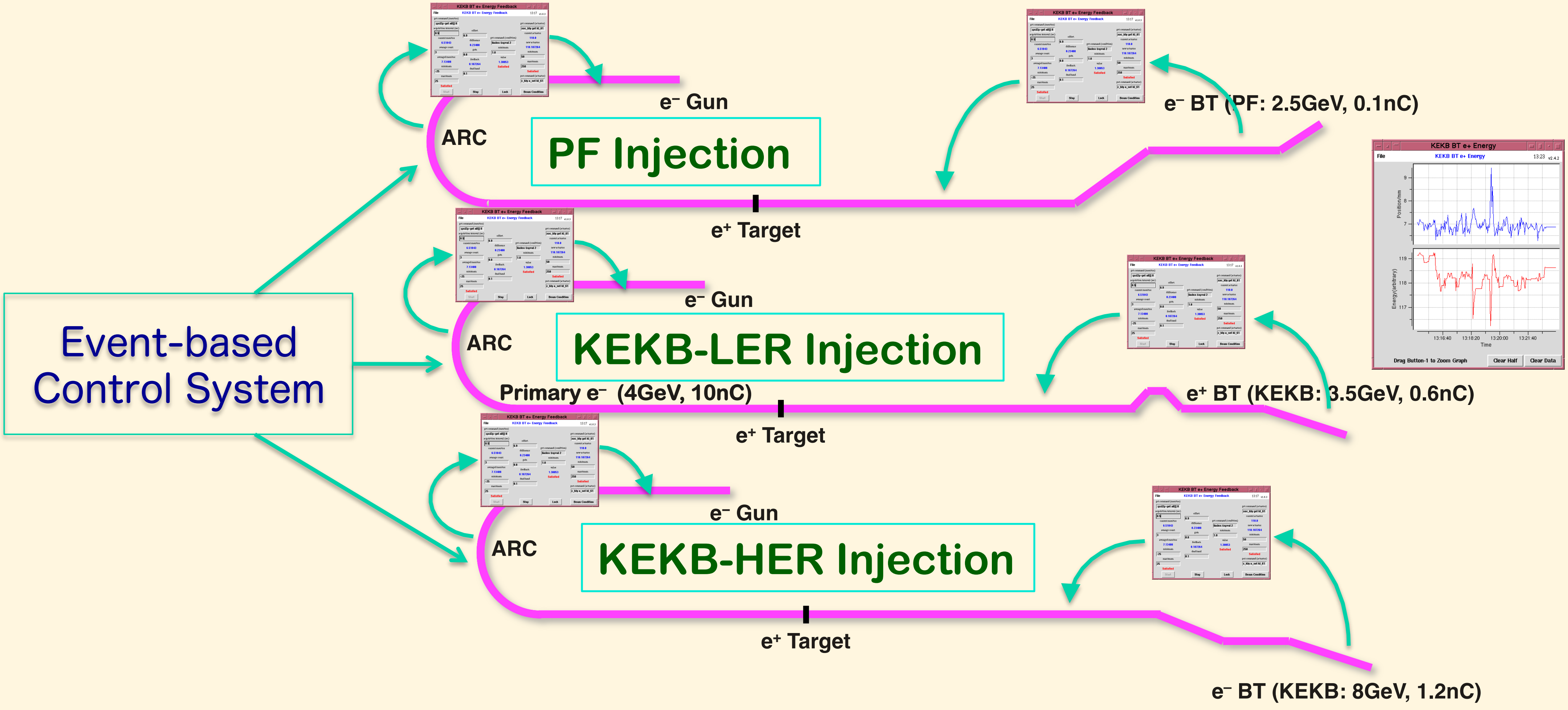
One Machine, Multiple Virtual Accelerators (VAs)

- **Control/Monitor are carried dependent on a VA**
  - **Mostly independent between VAs**
- **Independent parameter set for each VA, one of the VAs is controlled at a time**
  - **VAs for Injections (HER (e-), LER (e+), PF, PF-AR) and Linac-only in SuperKEKB project**



Multiple Closed Loop Controls Overlapped

- Closed loops can be installed on each VA independently
  - Tested at KEKB



Event-based Control System

PF Injection

e⁻ Gun

ARC

e⁺ Target

e⁻ BT (PF: 2.5GeV, 0.1nC)

KEKB-LER Injection

e⁻ Gun

ARC

Primary e⁻ (4GeV, 10nC)

e⁺ Target

e⁺ BT (KEKB: 3.5GeV, 0.6nC)

KEKB-HER Injection

e⁻ Gun

ARC

e⁺ Target

e⁻ BT (KEKB: 8GeV, 1.2nC)

Event-based Timing and Fast Global Synchronous Controls

- **Several Stable RF Frequencies (RF group)**
  - **114, 509, 571, 1298, 2856 MHz, Rubidium-based**
- **Independent Circumference Correction and Bucket Selection**
  - **KEKB: ~4x10^{-7}, PF, PF-AR: 4~20x10^{-6}**
- **SuperKEKB Injection Timing < 30ps**
  - **PF, PF-AR Injection with accidental sync. ~300ps**
- **4-ring Simultaneous Injection**
  - **Exchange energy, charge, sync. every 20ms**
  - **With damping ring (≧20ms) further complexity**
- **LLRF control and monitor are important**

Towards SuperKEKB

- **Upgrade of controllers for each type of device**
  - **Discussions with device groups, for aging controllers**
- **Base software components, OS, EPICS, CSS, (Scripting) Languages**
  - **Especially EPICS Collaboration-based software**
- **Operational software**
  - **Archiver, Archive viewer, Alarm, e-Log, etc**
- **Information sharing to offices**
  - **More Web based application software**
- **Seminar and training**
- **IP Networking, Wireless LAN, Console Desk, etc**

SuperKEKB Plan (1)

- For nano-beam scheme with 40-times higher luminosity
  - Many new facilities should be required
- Will start based on the existent environment
  - With additional concept of CA everywhere
- Help device groups to have better global controls
  - Replacement of old installations such as CAMAC
  - Solutions not only VME but also other types of controllers, embedded EPICS if possible
- Faster networks for the groups who can build controllers by themselves
- Better connection to operational environments
  - Keeping SAD environment, etc
  - Monitoring at offices

SuperKEKB Plan (2)

- Archiving scheme and viewer
  - Maybe existing KEKBlog and channel archivers
    - New viewer should be developed
- Alarm handler
  - CSS or Python (to simulate KEKBalarm)
    - Evaluating
- Operational Log
  - In house, two versions with different origins
    - Postgres + (Python/Zope and Flash/Flex)
- Scripts
  - SADscript/Tk, Python/Tk, (decreasing Tcl/Tk)
- Displays
  - CSS and MEDM/EDM

SuperKEKB Plan (3)

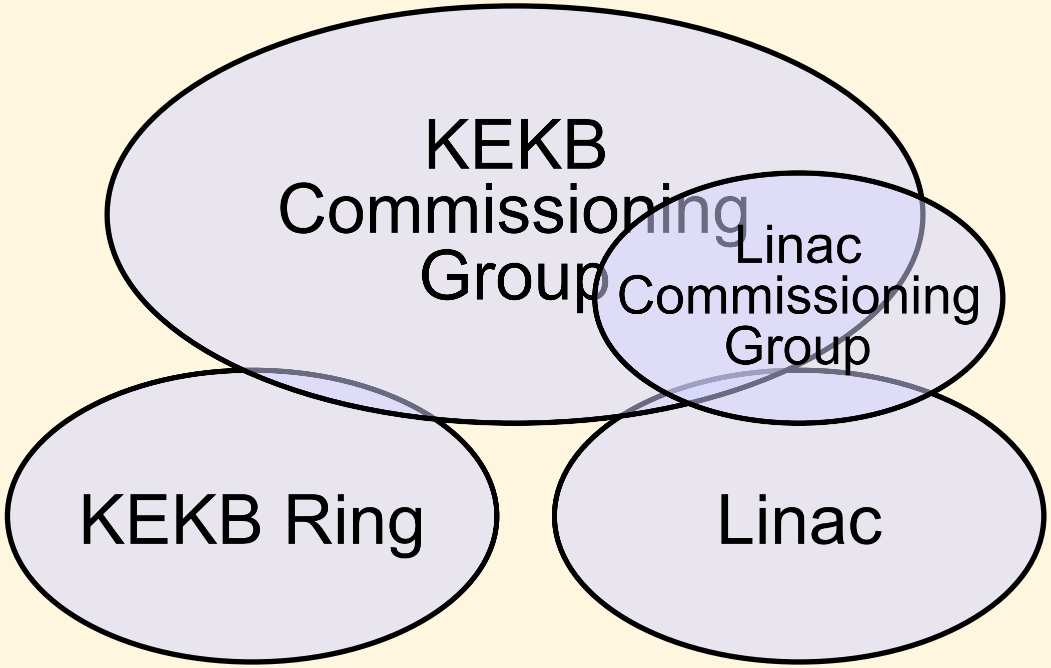
- Interviews to each device groups
  - Planning to have meetings and trainings
    - To collect user requirements, etc
  - Partially successful for old hardware replacements
    - Not yet effective for new functionalities
      - Whether both sides do not have experiences
- Additional controls
  - ex. Global orbit feedback
    - May provide EPICS interface, middle-speed feedback, etc

Operation

KEKB Commissioning Groups

- Formation of Commissioning Group (KCG)
  - Linac Commissioning (LCG)
    - 7 from Linac
    - ~10 from Ring
  - KEKB Ring Commissioning Group (KCG)
    - All LCG
    - ~20 from Ring
    - Several from Detector (BCG)
  - Commissioning software base was formed during Linac Commissioning (1997~)

  Tcl/Tk , Python/Tk, SADscript/Tk



Scripting Languages

- **Heavy use because of rapid prototyping**
- **Linac**
  - (1992~) Tcl/Tk as Test tools on Unix
  - (1997~) Tcl/Tk as Main Operator Programming Tool
  - (Now) Mixture of Tcl/Tk, SADscript/Tk, Python/Tk
    - SADscript has most accelerator design capability
      - Covers many features like MATLAB, Mathematica, XAL, MAD
- **KEKB**
    - (Nodal interpreter and Fortran covered everything at TRISTAN)
  - Python covers many areas which is not covered by medm
  - SADscript is used by operators and physicists everyday
    - Realization of novel ideas in hours
      - Only some ideas are effective, so rapid prototyping is important

SADScript

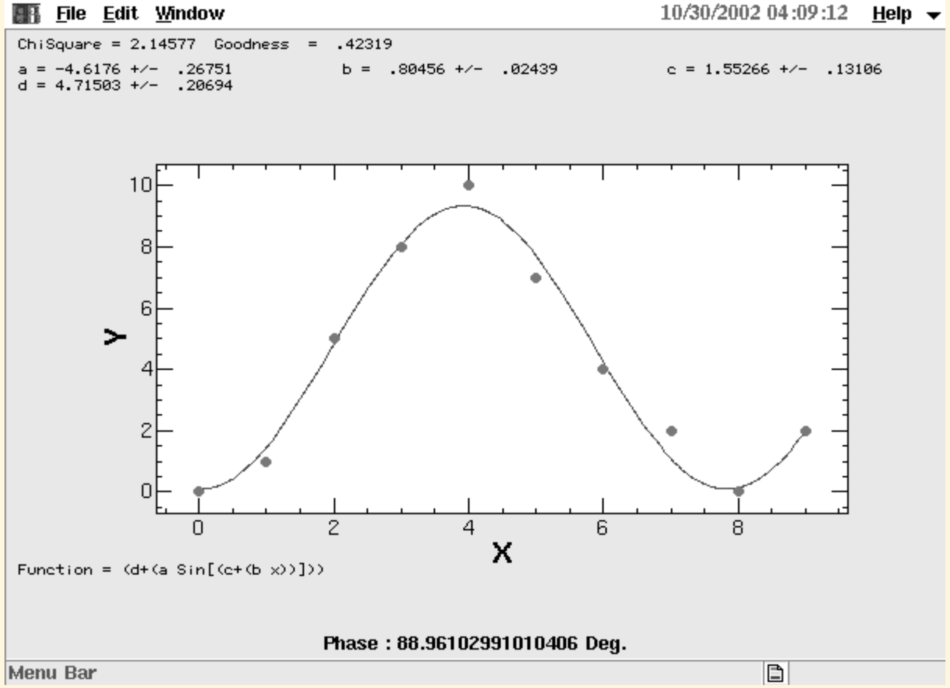
- **Mathematica-like Language**
  - Not Real Symbolic Manipulation (Fast)
  - EPICS CA (Synchronous and Asynchronous)

    CaRead/CaWrite[ ], CaMonitor[ ], etc.
  - (Oracle Database)
  - Tk Widget
  - Canvas Draw and Plot
  - KBFrame on top of Tk
  - Data Processing (Fit, FFT, …)
  - Inter-Process Communication (Exec, Pipe, etc)

    System[ ], OpenRead/Write[ ], BidirectionalPipe[ ], etc.
  - Greek Letter
  - Full Accelerator Modeling Capability
  - Also Used for non-Accelerator Applications
  - Comparable to XAL, but very different architecture

SADScript

◆ Example



```
ChiSquare = 2.14577  Goodness  =  .42319
a = -4.6176 +/-  .26751            b =  .80456 +/-  .02439            c = 1.55266 +/-  .13106
d = 4.71503 +/-  .20694
```

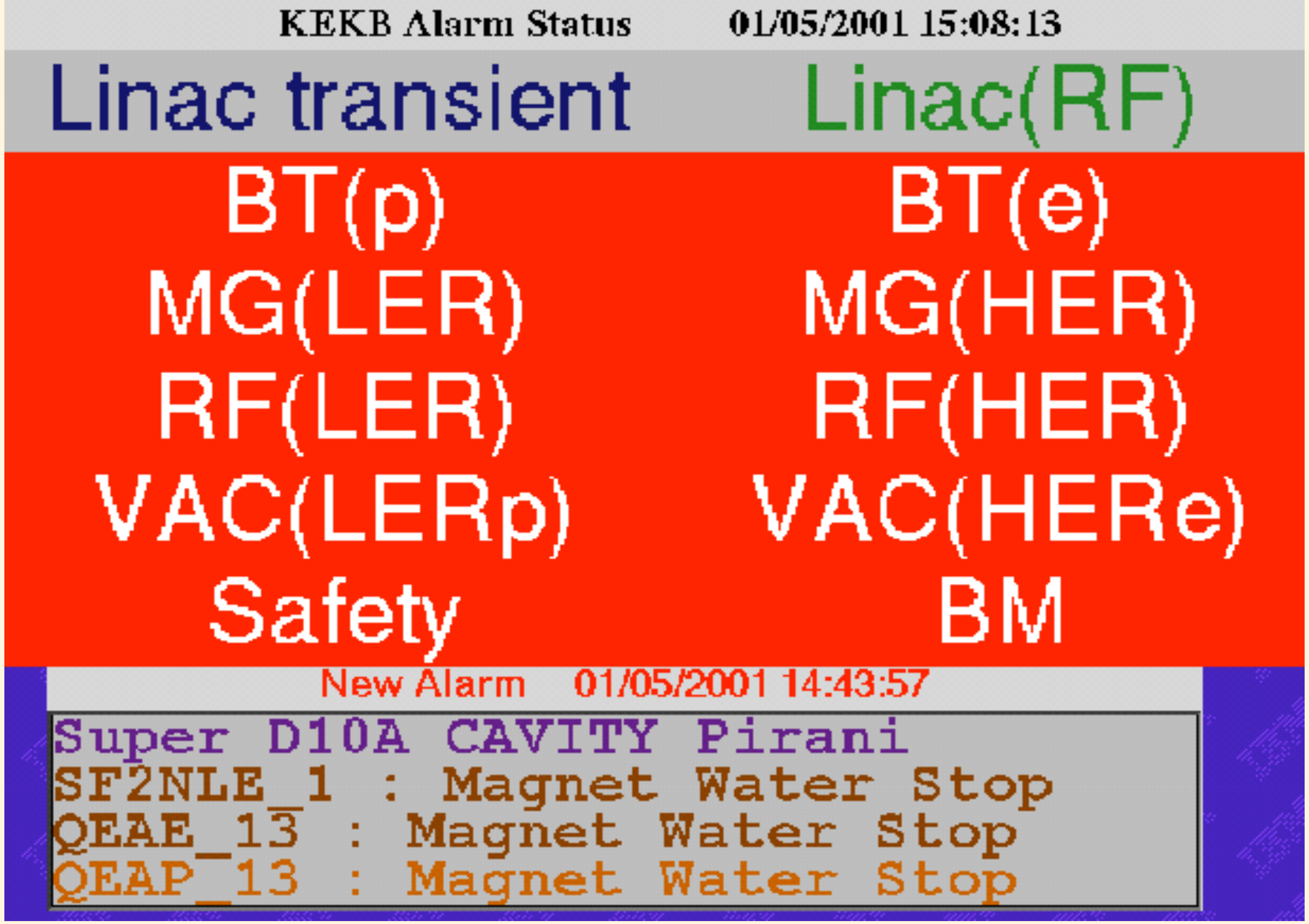
```
Function = (d+(a Sin[(c+(b x))]))
```

Phase : 88.96102991010406 Deg.

```
FFS;
w=KBMainFrame["w1",fm,Title->"t1"];
$DisplayFunction=CanvasDrawer;
W1=Frame[fm];
c1=Canvas[w1,Width->600,Height->400,
 Side->"top"];
Canvas$Widget=c1;
data = {{0,0}, {1,1}, {2,5}, {3,8}, {4,10}, {5,7}, {6,4}, {7,2}, {8,0}, {9,2}}
fit = FitPlot[data,a Sin[x b + c] + d, x, {a,5},{b,1},{c,1},{d,5}, FrameLabel->{"X","Y"}];
phase = StringJoin["Phase : ", (c/.fit[[1]]) 180/Pi, " Deg."];
f1=KBFComponentFrame[w1,Add->{KBFText[Text->phase]}];
TkWait[];
Exit[];
```

KEKB Alarm Panel

- **KEKB Alarm Main Panel covers Linac Alarms as well. ~10,000 Records are Monitored in One Panel. Detailed alarm information/history is available in a separate panel**



KEKB Alarm Status 01/05/2001 15:08:13

New Alarm 01/05/2001 14:43:57

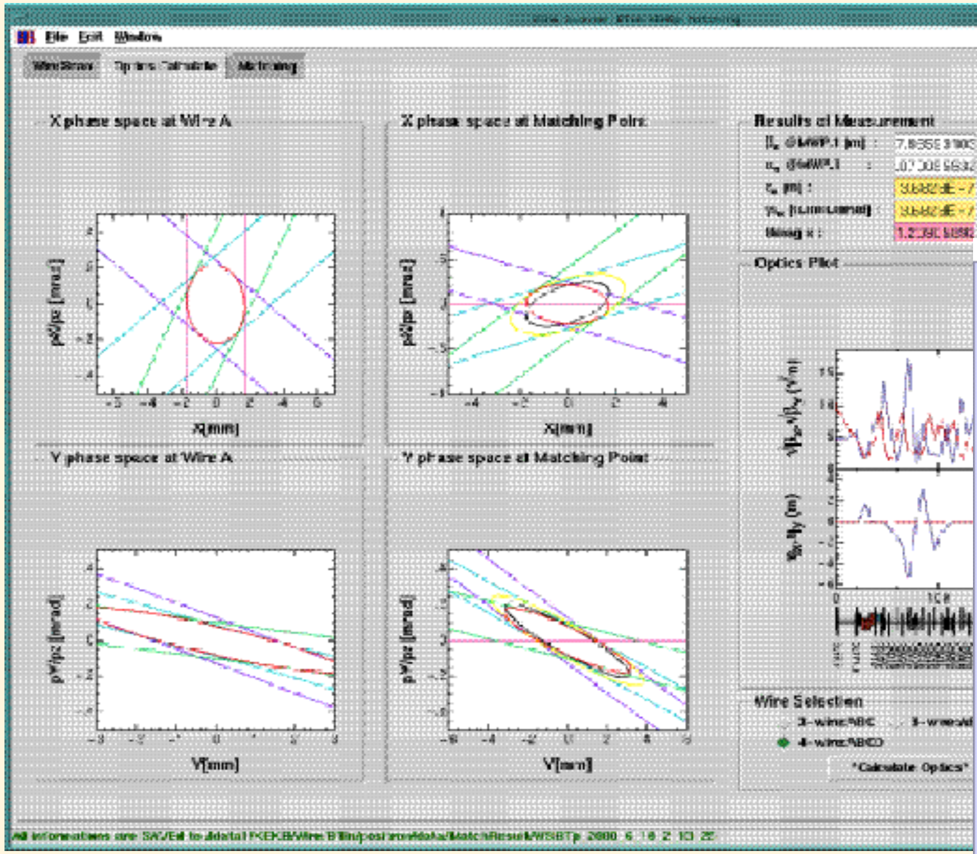
```
Super D10A CAVITY Pirani
SF2NLE_1 : Magnet Water Stop
QEAE_13 : Magnet Water Stop
QEAP_13 : Magnet Water Stop
```

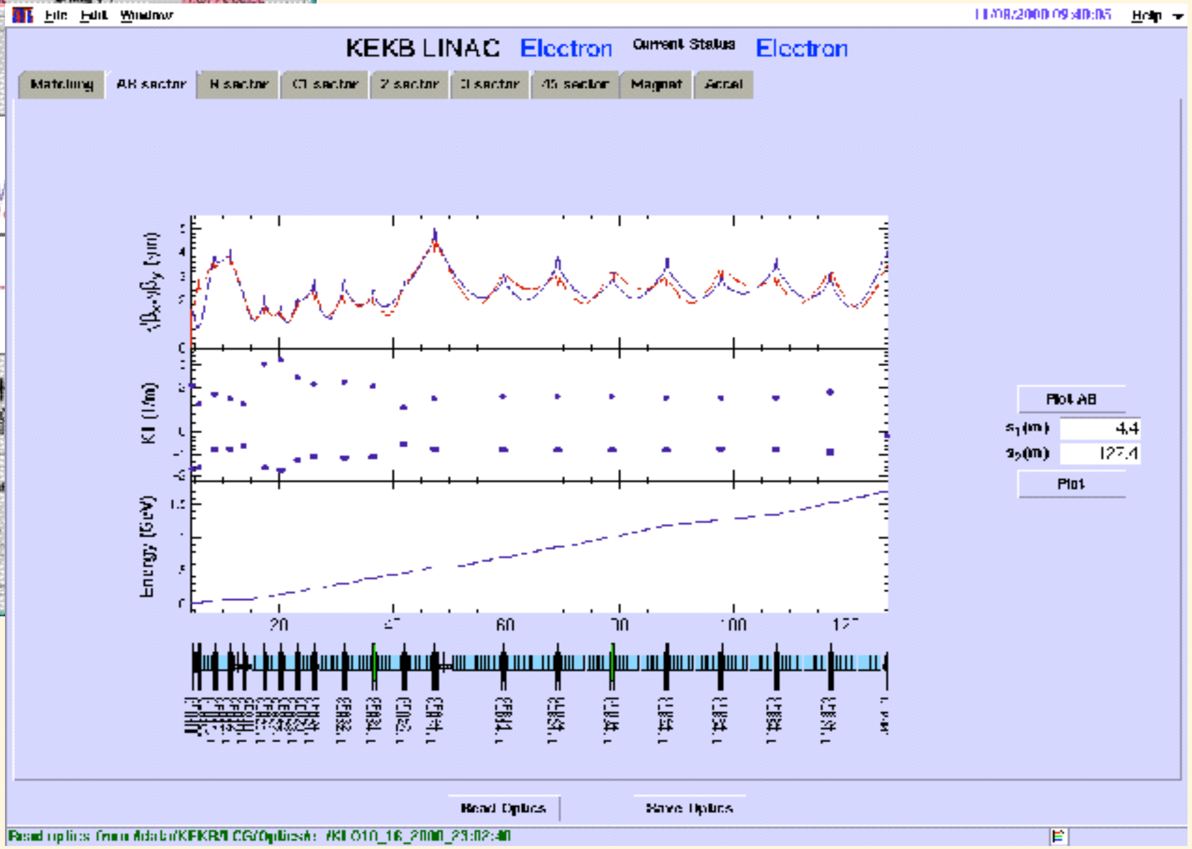
Linac

Ring

Beam Optics Panels in SAD

- **Beam Optics Matching and Optimization Panels in SADscript**

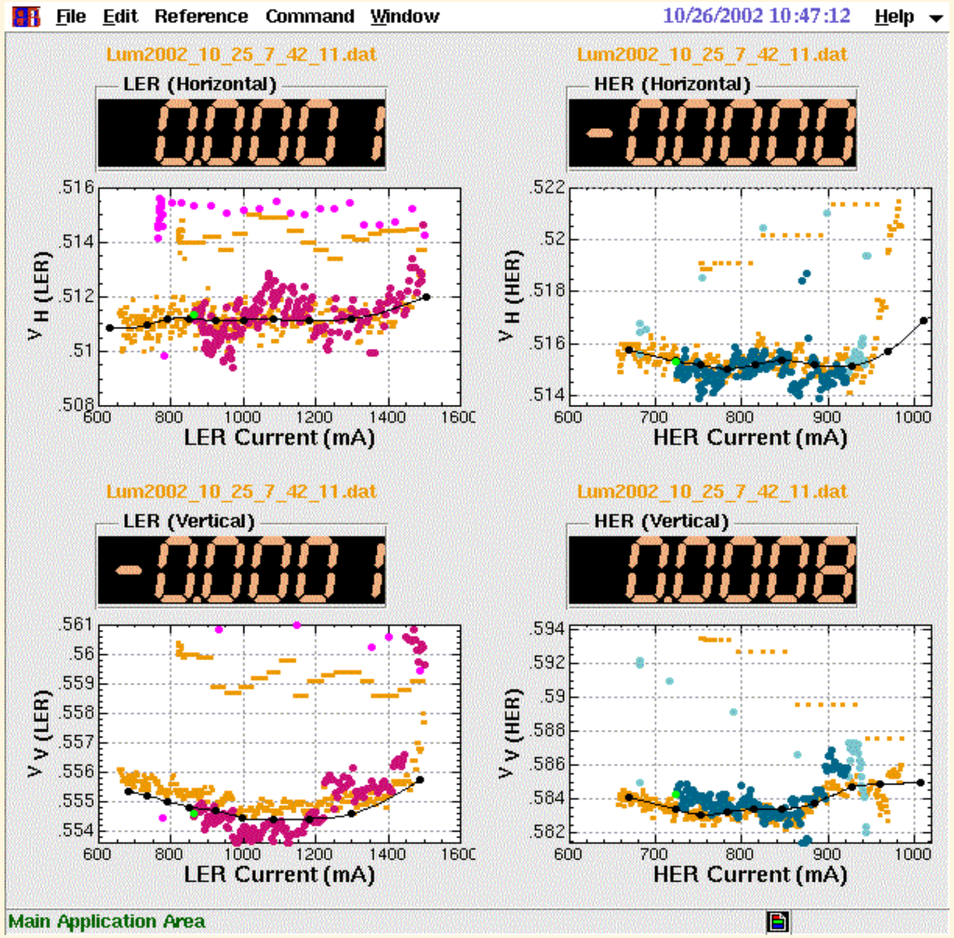


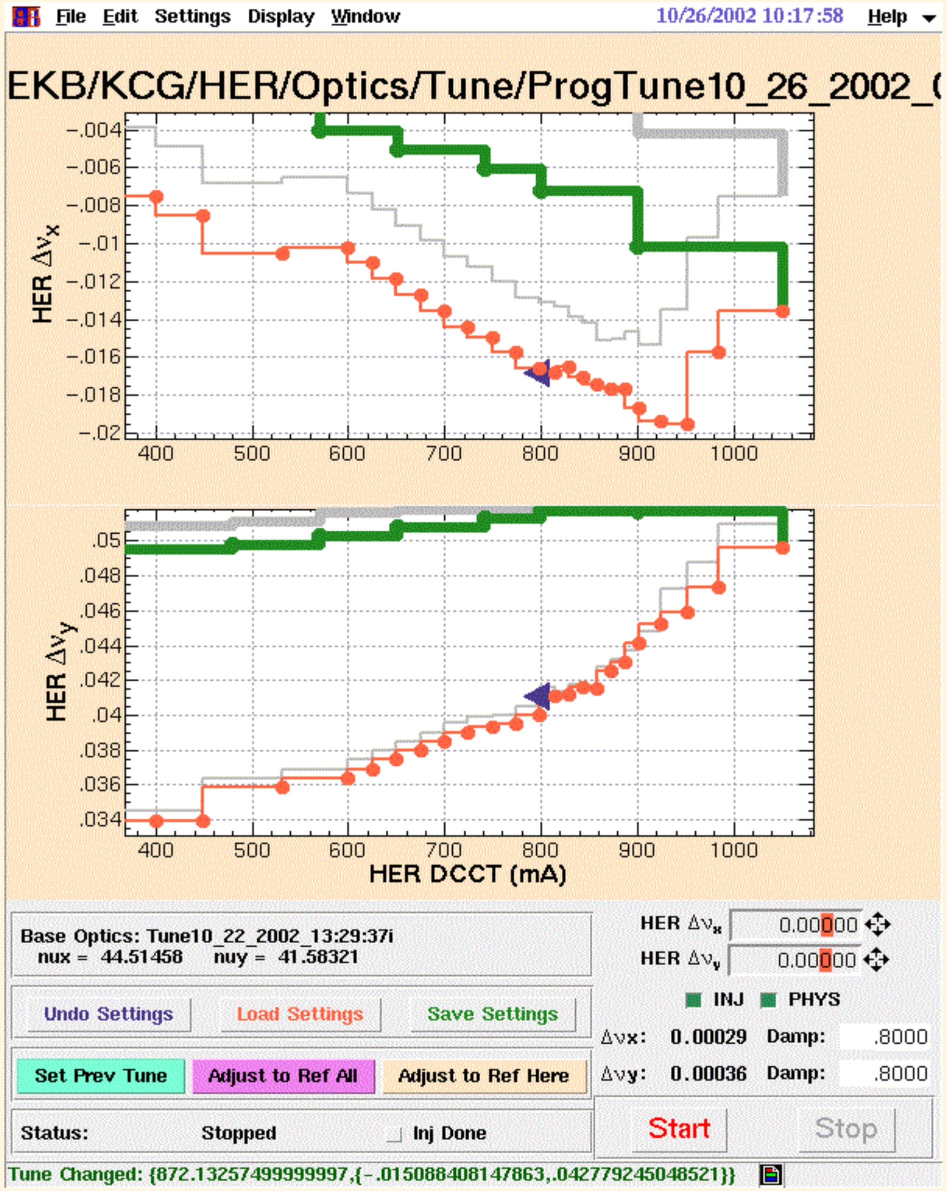


- **Some Parameters goes thru EPICS Gateways, others directly to Linac**

KEKB Operation Panel Examples

◆Tune Measurement and Tune Changer





Base Optics: Tune10_22_2002_13:29:37i
nux = 44.51458 nuy = 41.58321

Δνx: 0.00029 Damp: .8000

Δνy: 0.00036 Damp: .8000

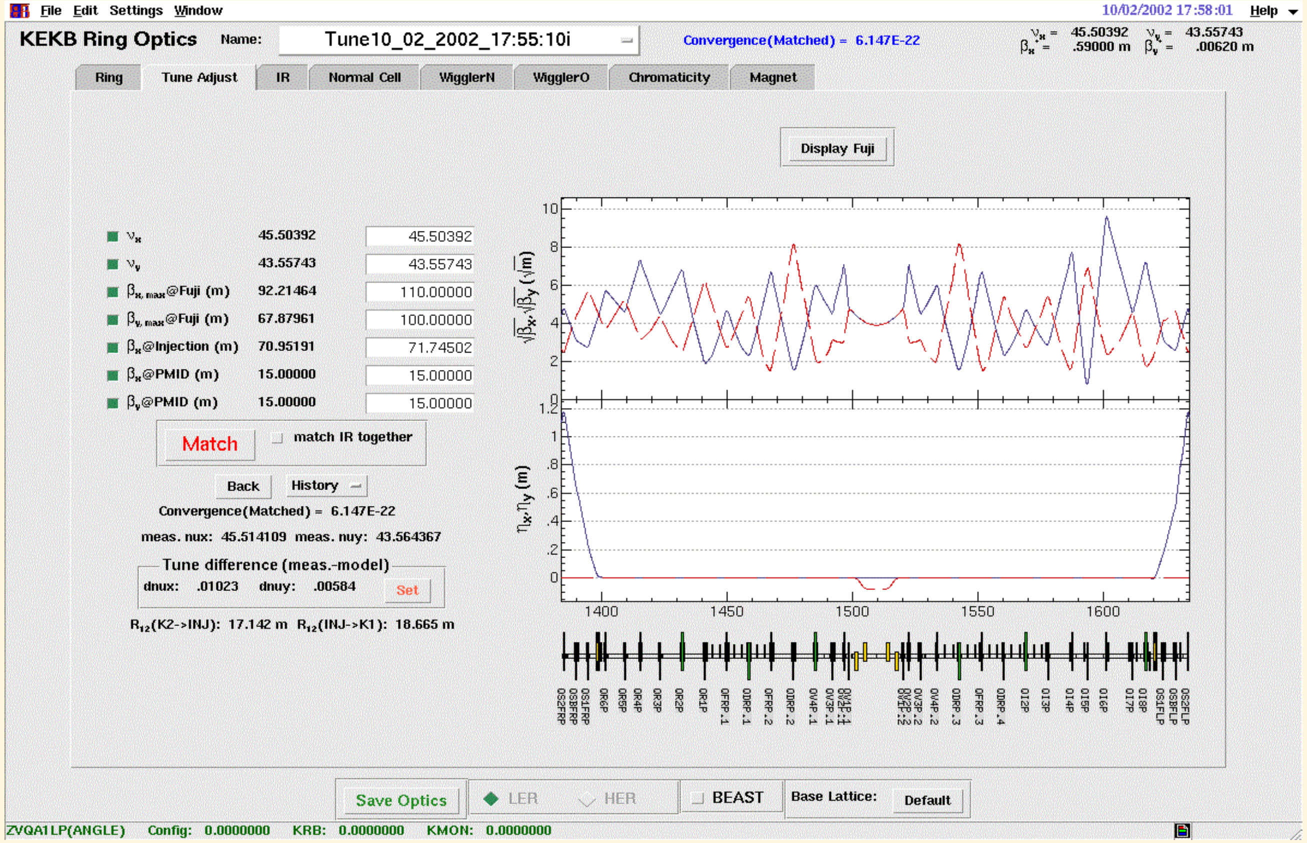
Status: Stopped □ Inj Done

Tune Changed: {872.13257499999997,{-.015088408147863,.042779245048521}}

Virtual Accelerator in KEKB

◆Tune/Optics Server

- **Maintain a Model of Real Accelerator**
- **Can Change Tune, Chromaticity, etc, on Request by Other Panels**
- **Act as a Virtual Accelerator**



File Edit Settings Window 10/02/2002 17:58:01 Help

KEKB Ring Optics Name: Tune10_02_2002_17:55:10i

Convergence(Matched) = 6.147E-22

ν_x = 45.50392 ν_y = 43.55743
β_x^* = .59000 m β_y^* = .00620 m

Ring | Tune Adjust | IR | Normal Cell | WigglerN | WigglerO | Chromaticity | Magnet

Display Fuji

ν_x	45.50392	45.50392
ν_y	43.55743	43.55743
$\beta_{x,\max}$@Fuji (m)	92.21464	110.00000
$\beta_{y,\max}$@Fuji (m)	67.87961	100.00000
β_x@Injection (m)	70.95191	71.74502
β_x@PMID (m)	15.00000	15.00000
β_y@PMID (m)	15.00000	15.00000

Match □ match IR together

Back History

Convergence(Matched) = 6.147E-22

meas. nux: 45.514109 meas. nuy: 43.564367

Tune difference (meas.-model)
dnux: .01023 dnuy: .00584 Set

R_{12}(K2->INJ): 17.142 m R_{12}(INJ->K1): 18.665 m

Save Optics ◆ LER ◇ HER □ BEAST Base Lattice: Default

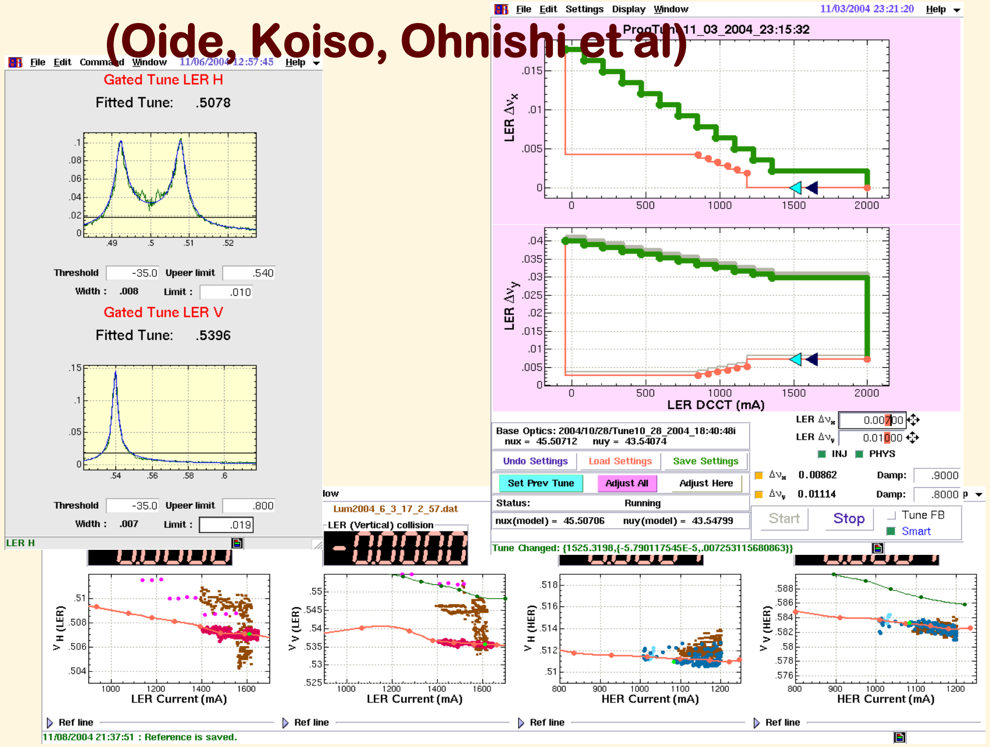
ZVQA1LP(ANGLE) Config: 0.0000000 KRB: 0.0000000 KMON: 0.0000000

Virtual Accelerator in KEKB

◆ For Example in KEKB

- ❖most Beam Optics Condition is maintained in the Optics Panel
- ❖Other Panels Manipulate Parameters Communicating with the Optics Panel

(Oide, Koiso, Ohnishi et al)



File Edit Command Window 11/06/2004 12:57:45 Help

Gated Tune LER H

Fitted Tune: .5078

Threshold -35.0 Upeer limit .540

Width : .008 Limit : .010

Gated Tune LER V

Fitted Tune: .5396

Threshold -35.0 Upeer limit .800

Width : .007 Limit : .019

LER H

File Edit Settings Display Window 11/03/2004 23:21:20 Help

ProgTune11_03_2004_23:15:32

Base Optics: 2004/10/28/Tune10_28_2004_18:40:48i
nux = 45.50712 nuy = 43.54074

LER Δν$_x$ 0.00700

LER Δν$_y$ 0.01000

Status: Running

nux(model) = 45.50706 nuy(model) = 43.54799

Δν$_x$ 0.00862 Damp: .9000

Δν$_y$ 0.01114 Damp: .8000

□ Tune FB
■ Smart

Tune Changed: {1525.3198,{-5.790117545E-5,.00725311568O863}}

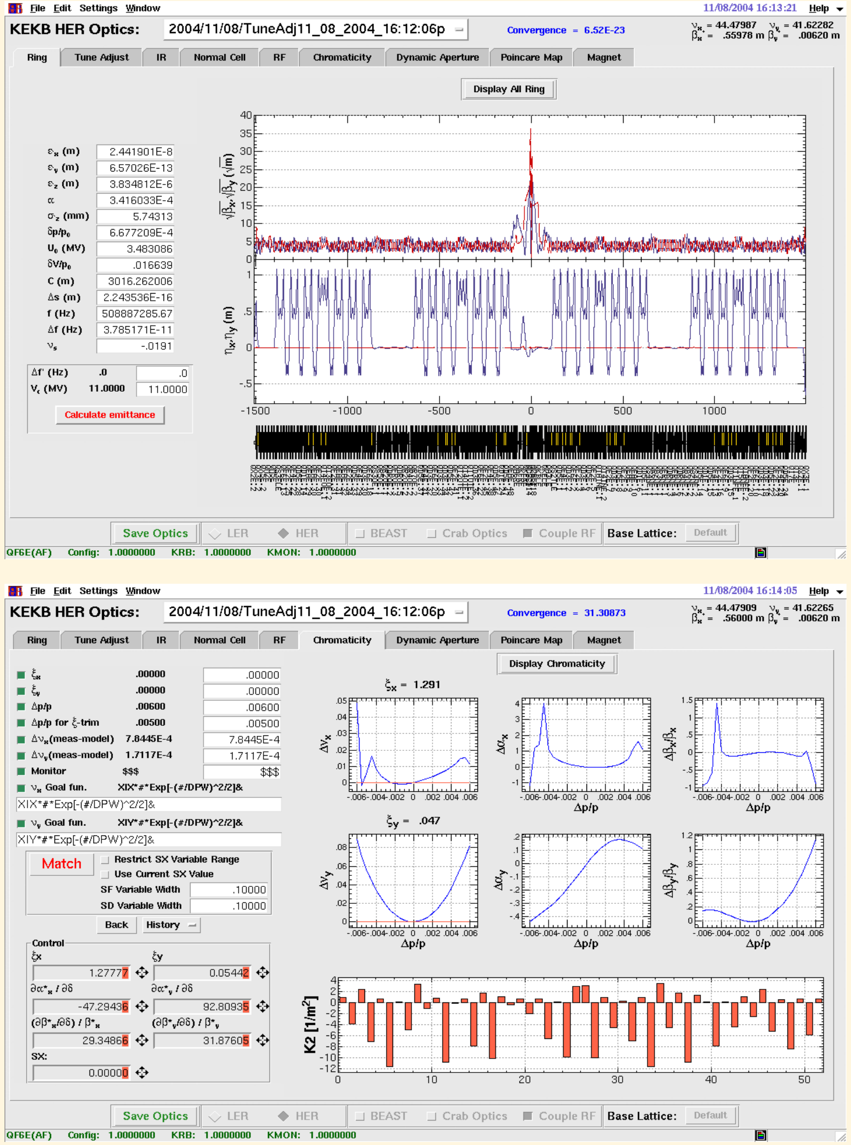
Lum2004_6_3_17_2_57.dat

LER (Vertical) collision

11/08/2004 21:37:51 : Reference is saved.

Tune Measurement/Changer

===>



File Edit Settings Window 11/08/2004 16:13:21 Help

KEKB HER Optics: 2004/11/08/TuneAdj11_08_2004_16:12:06p

Convergence = 6.52E-23

ν_x = 44.47987 ν_y = 41.62282
β_x^* = .55978 m β_y^* = .00620 m

ε_x (m)	2.441901E-8
ε_y (m)	6.57026E-13
ε_z (m)	3.834812E-6
α	3.416033E-4
σ_z (mm)	5.74313
$\delta p/p_0$	6.677209E-4
U_0 (MV)	3.483086
$\delta V/p_0$	.016639
C (m)	3016.262006
Δs (m)	2.243536E-16
f (Hz)	508887285.67
Δf (Hz)	3.785171E-11
ν_s	-.0191

Δf' (Hz) .0 .0

V_c (MV) 11.0000 11.0000

QF6E(AF) Config: 1.0000000 KRB: 1.0000000 KMON: 1.0000000

File Edit Settings Window 11/08/2004 16:14:05 Help

KEKB HER Optics: 2004/11/08/TuneAdj11_08_2004_16:12:06p

Convergence = 31.30873

ν_x = 44.47909 ν_y = 41.62265
β_x^* = .56000 m β_y^* = .00620 m

■ ξ_x	.00000	.00000
■ ξ_y	.00000	.00000
■ Δp/p	.00600	.00600
■ Δp/p for ξ-trim	.00500	.00500
■ Δν$_x$(meas-model)	7.8445E-4	7.8445E-4
■ Δν$_y$(meas-model)	1.7117E-4	1.7117E-4
■ Monitor	$$$	$$$

■ ν_x Goal fun. XIX*#*Exp[-(#/DPW)^2/2]&

XIX*#*Exp[-(#/DPW)^2/2]&

■ ν_y Goal fun. XIY*#*Exp[-(#/DPW)^2/2]&

XIY*#*Exp[-(#/DPW)^2/2]&

□ Restrict SX Variable Range

□ Use Current SX Value

SF Variable Width .10000

SD Variable Width .10000

Control

ξx 1.27777 ξy 0.05442

∂α*$_x$ / ∂δ -47.29436 ∂α*$_y$ / ∂δ 92.80935

(∂β*$_x$/∂δ) / β*$_x$ 29.34866 (∂β*$_y$/∂δ) / β*$_y$ 31.87605

SX: 0.00000

ξ_x = 1.291

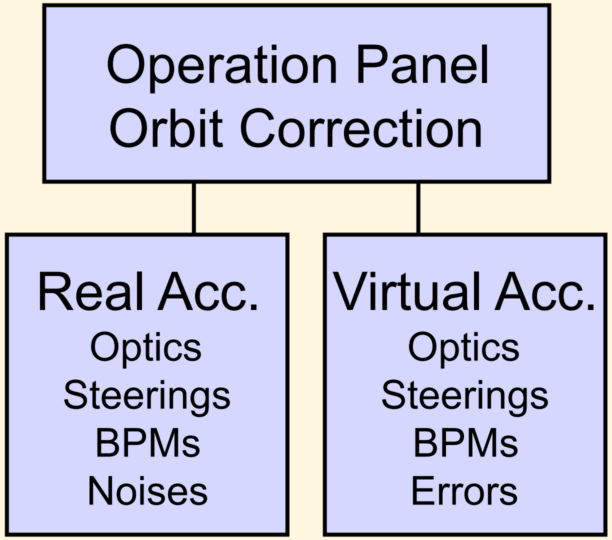
ξ_y = .047

QF6E(AF) Config: 1.0000000 KRB: 1.0000000 KMON: 1.0000000

Optics Panel

Example Virtual Accelerator

- ◆Virtual Accelerator may Provide the Both Fake Steerings and Fake BPMs Maybe with Simulated Errors/Noises
- ◆Orbit Correction Application may Work On Those Fake Information



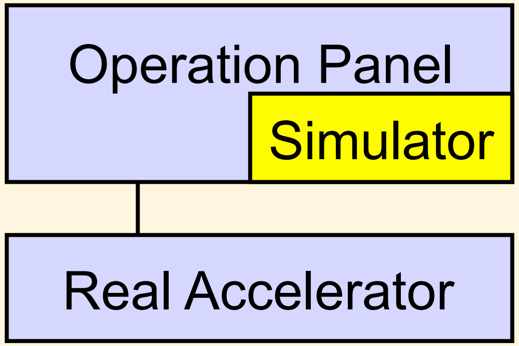
Operation Panel
Orbit Correction

Real Acc.
Optics
Steerings
BPMs
Noises

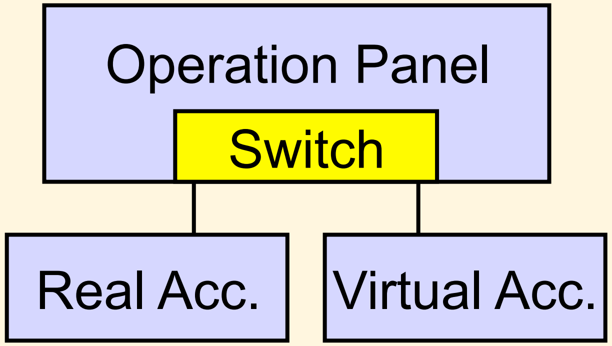
Virtual Acc.
Optics
Steerings
BPMs
Errors

Virtual Accelerator with EPICS

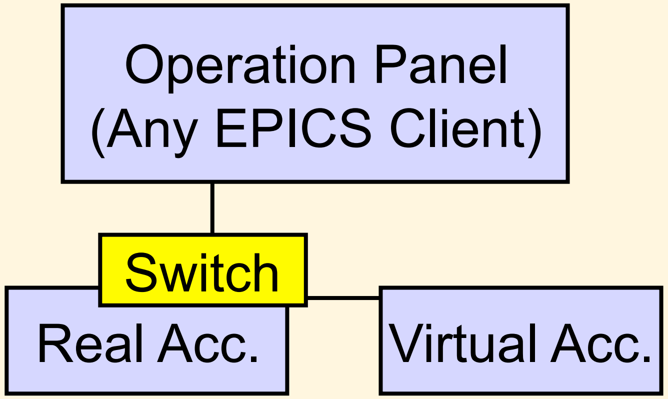
- **Fake Accelerator Implementation**
  - **With EPICS Channel Access**
- **In A Single SAD Application**
  - **Built-in Simulator in Operation Panel**
  - **Only SAD Applications**



- **Separate Simulator (Virtual Accelerator)**
  - **Needs Some Switching Mechanism**



- **Separate Simulator (Virtual Accelerator)**
  - **In EPICS Semantics (EPICS Simulation Server)**
  - **Any Operation Panel (not only SAD)**
  - **SAD Simulation Server should Act as EPICS Channel Access Server**



EPICS Simulation Mode

◆EPICS Database - Simulation Mode

A set of fields to support simulation are supplied on all hardware input records.

SIMM = YES makes this record a simulation record.

A link to a database value to put the record into simulation mode is specified

in the SIML. A non-zero number puts the record into simulation mode.

SVAL is the engineering units value used when the record is in simulation mode.

SIOL is a database location from which to fetch SVAL when the record is in simulation mode.

SIMS is the alarm severity of the record if it is in simulation mode.

❖That is, EPICS Records may have Proxy Records

EPICS Simulation Mode

SIMM - Simulation Mode

» This field has either the value YES or NO. By setting this field to YES, the record can be switched into simulation mode of operation. While in simulation mode, input will be obtained from SIOL instead of INP.

SIML - Simulation Mode Location

» This field can be a constant, a database link, or a channel access link. If SIML is a database or channel access link, then SIMM is read from SIML. If SIML is a constant link then SIMM is initialized with the constant value but can be changed via dbPuts.

SVAL - Simulation Value

» This is the record's input value, in engineering units, when the record is switched into simulation mode, i.e. when SIMM is set to YES.

SIOL - Simulation Value Location

» This field can be a constant, a database link, or a channel access link. If SIOL is a database or channel access link, then SVAL is read from SIOL. If SIOL is a constant link then SVAL is initialized with the constant value but can be changed via dbPuts.

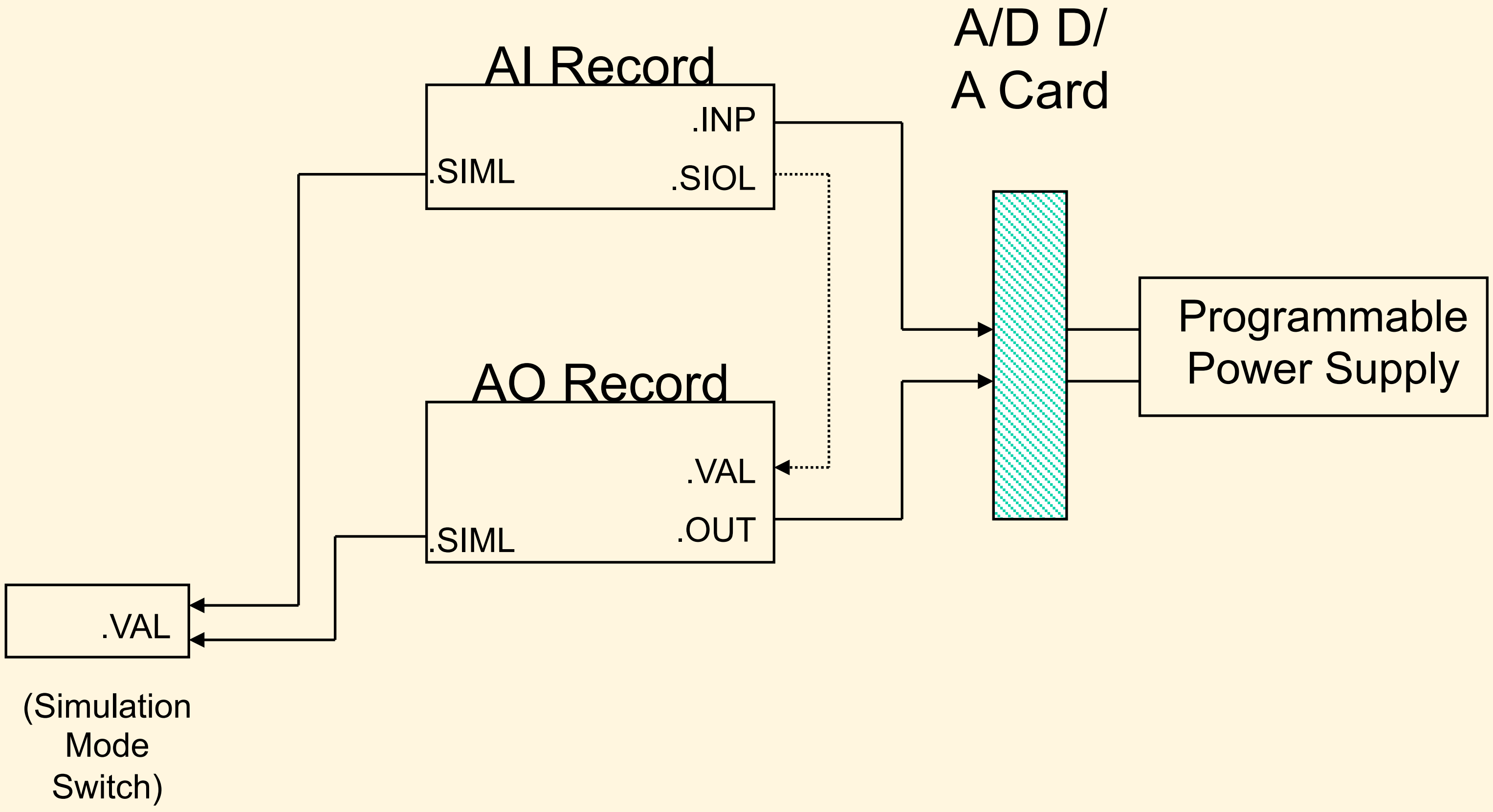
SIMS - Simulation Mode Alarm Severity

» When this record is in simulation mode, it will be put into alarm with this severity and a status of SIMM.

Simulation Mode

◆EPICS Simulation Mode Simple Example

❖Tests Logic without Hardware

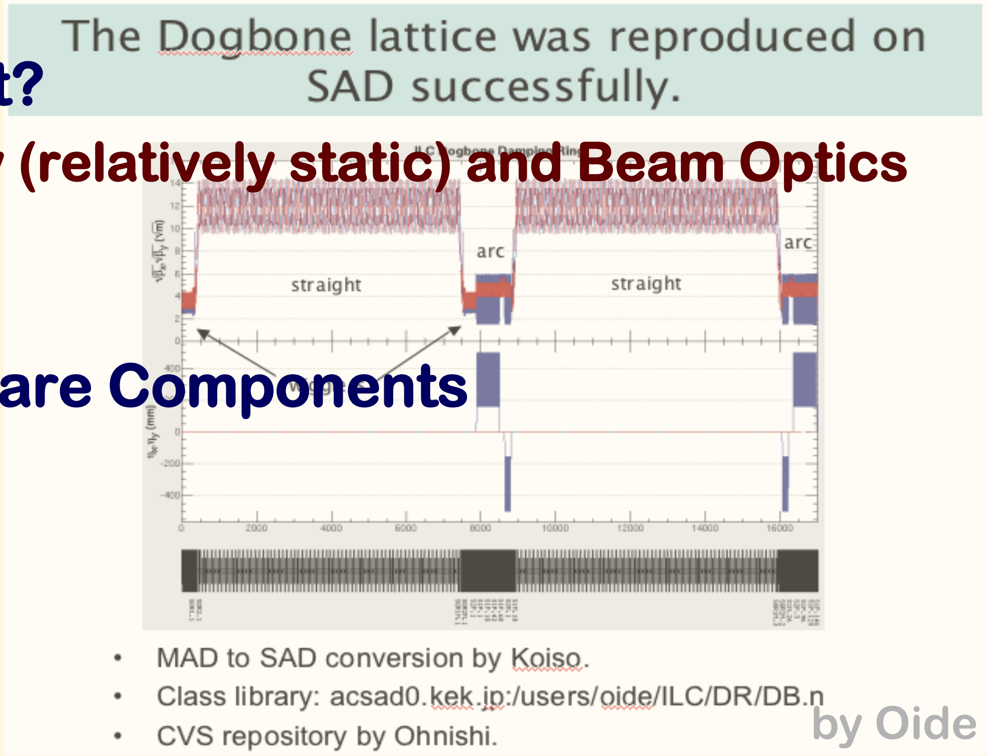


SAD as EPICS Simulator

- **Implementing a Virtual Accelerator**
- **SAD Simulator in Channel Access Server**
  - **Serves Channel Values Requested by Channels (Records) in Simulation Mode (SIOL),**
    Acting as a Channel Access Server
  - **Slightly more Difficult to Implement (at the First Stage)**
- **SAD Simulator in Channel Access Client**
  - **Provides Channel Values Needed by Channels (Records) in Simulation Mode (SVAL)**
  - **Easier to Implement (?)**
    - **Needs Some Studies**

Beam Optics Database

- **Repository of Inputs to Simulation Codes?**
- **XSIF Extended Standard Input Format**
  - **Many Simulation Codes utilize it**
  - **SAD does not**
  - **Currently a Conversion Tool is Used to for These Input Formats**
  - **XSIF (LIBXSIF) inclusion in SAD?**
- **Yet another Generalized Input Format?**
  - **Separation between Beamline Geometry (relatively static) and Beam Optics (more varying)**
  - **Could be structured into XML**
- **Relational information to each Hardware Components**
  - **We do not prefer complicated relations**



The Dogbone lattice was reproduced on SAD successfully.

- MAD to SAD conversion by Koiso.
- Class library: acsad0.kek.jp:/users/oide/ILC/DR/DB.n
- CVS repository by Ohnishi.

by Oide

Summary

Accelerator Controls

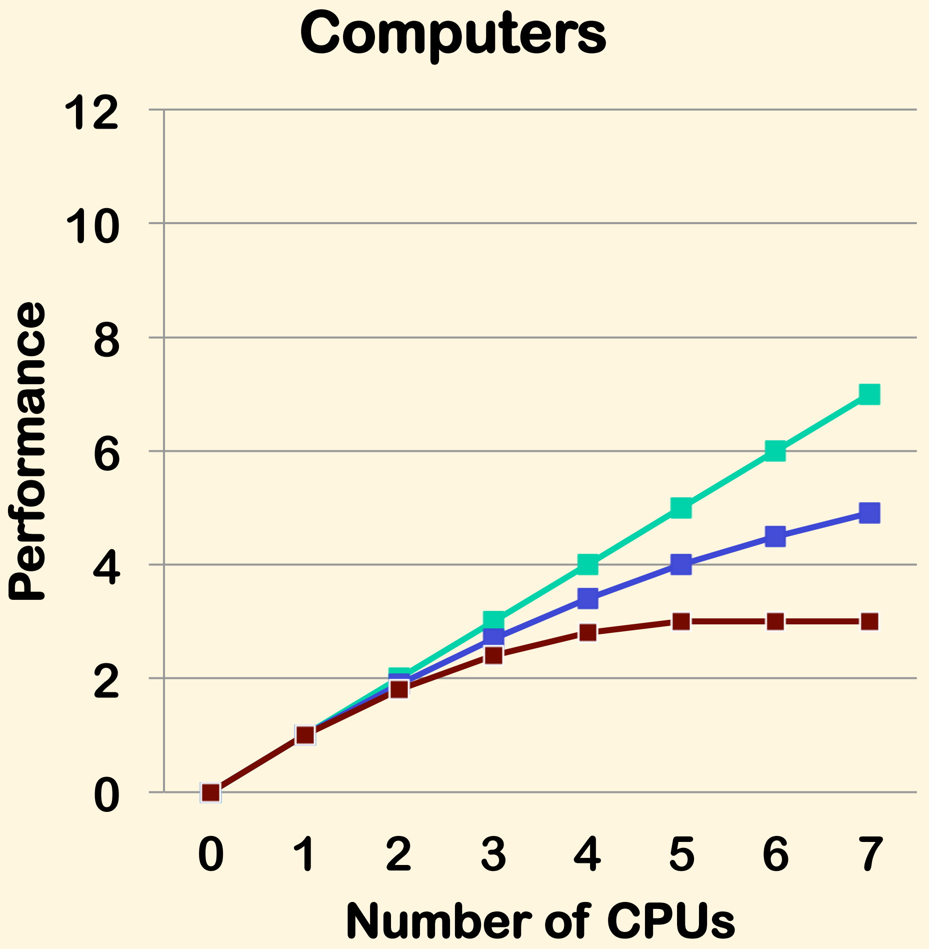
- **It's a fan to interact with all the components of the accelerator through control hardware and software**
- **It's a fan to interact with all the staff members of the project in order to design and improve controls**
- **We can contribute to the machine performance and the results even without realizing it**

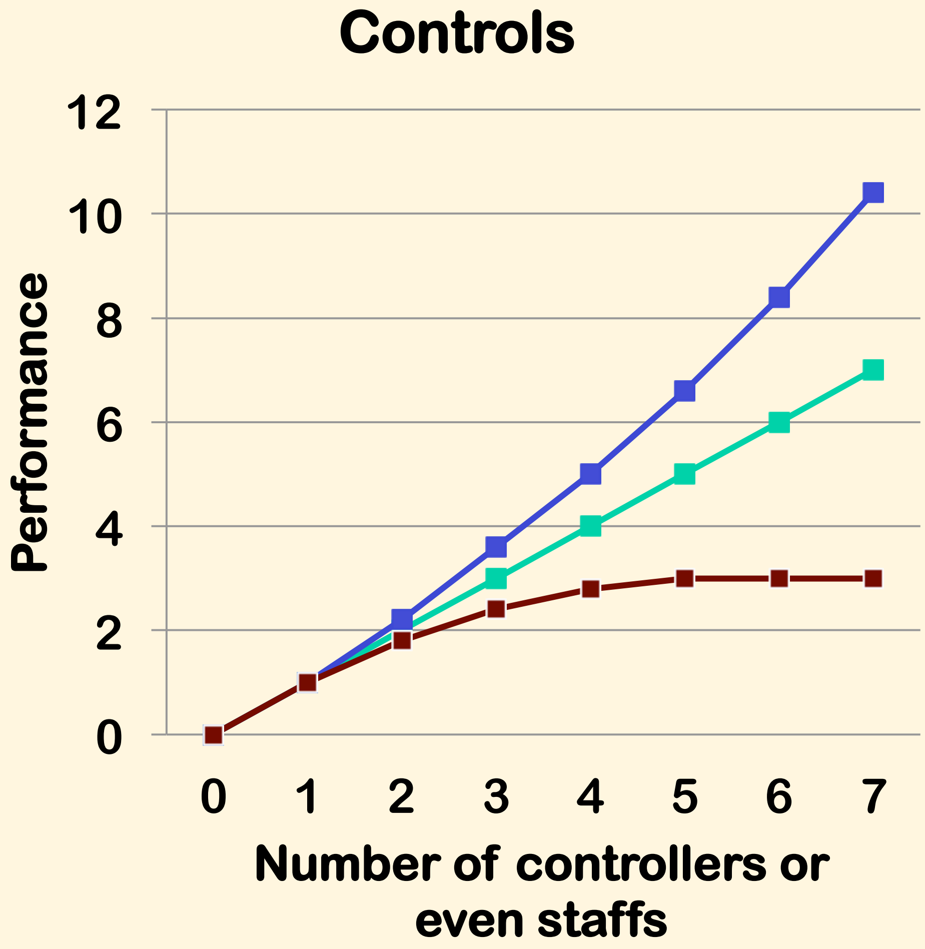
Conferences

- Exchange collaboration ideas internationally
  - Collaboration is important because we don't want to reinvent the wheel
  - Asia, Europe and North America regions
- ICALEPCS, San Francisco/NIF, Oct.2013
- PCaPAC, Kolkata/VECC, Dec.2012
- EPICS collab. meeting, Daejeon/KSTAR, Oct.2012
- WAO, San Francisco/SLAC, Aug.2012
- IBIC, Tsukuba, Oct.2012
- IPAC, New Orleans, May.2012

Performance achievement

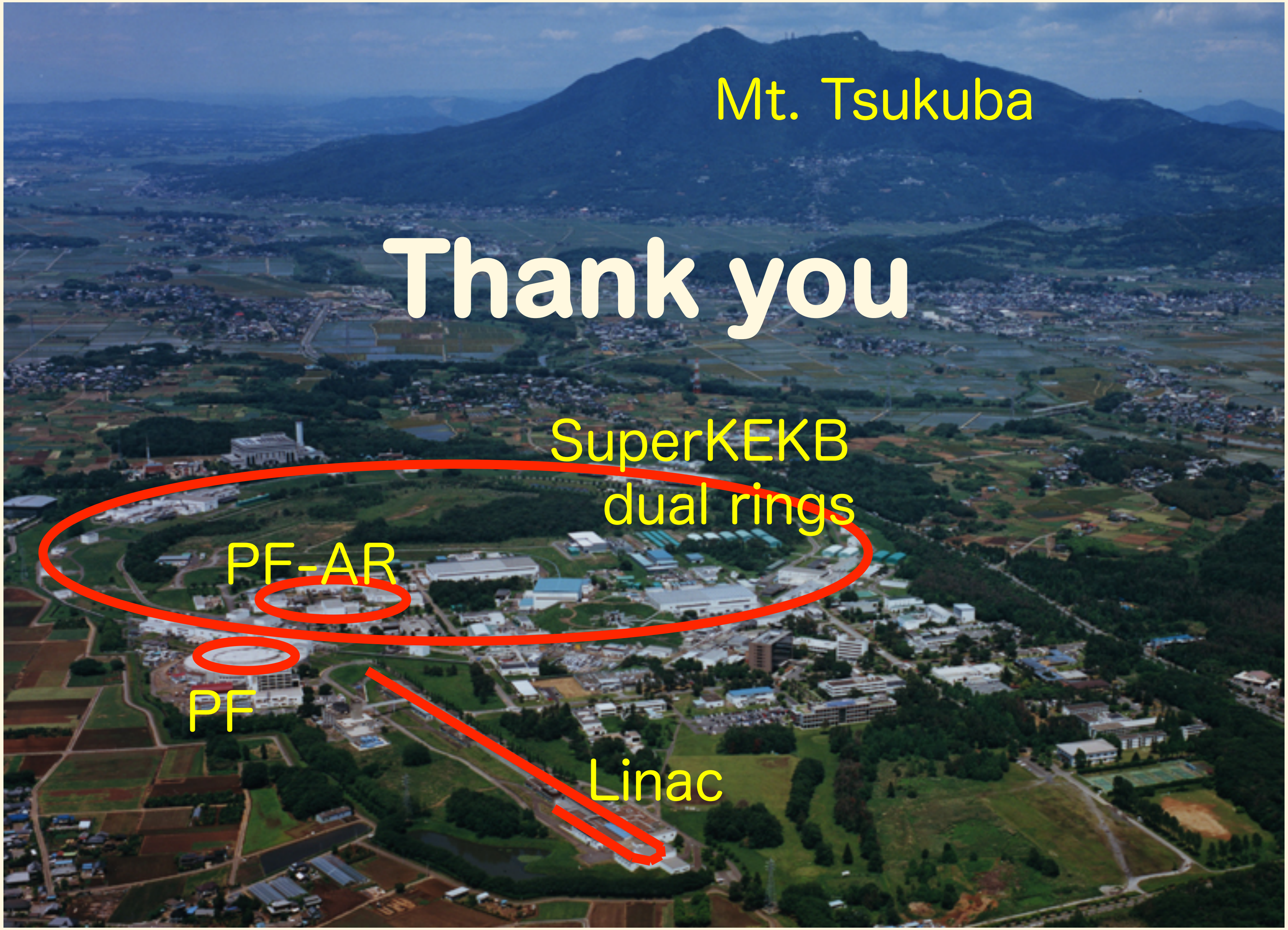
If the system is small single controller or single person can cover it. However, … We need global optimization instead of local optimization





Conclusion

- ◆Two kinds of virtual accelerators appeared in this slides. That may mean something.
- ◆Control efforts have contacts with all activities in the particle accelerator. Controls are at the privileged position to enjoy it.
- ◆We should provide maximum performance out of each component and achieve ultimate goal.
- ◆With some Phronesis (Greek: practical wisdom, ability to understand the universal truth), we believe we can achieve the target.


Mt. Tsukuba
Thank you
SuperKEKB
dual rings
PF-AR
PF
Linac

Thank you

Phronesis

- アリストテレスの知恵の考え方
  - Sophia とは逆に普遍的な真実を理解しようとする能力(らしい)、全体的な解決方法を探す能力
- 加速器制御
  - 実現する方法はたくさんあるが、各装置の能力を引き出し、性能向上のための新しいアイデアを実現し、物理実験の成果を最大限に高める方法はそれほど多くない
  - 同じ基盤の上で各グループがアイデアを交換できることが重要なのではないか

Thank you