# Controls and Machine Protection

***Joint International Accelerator School***
***on***
***Beam Loss and Accelerator Protection***

Enzo Carrone

Newport Beach, CA – Nov. 10th, 2014

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

# Introduction

SLAC

## Off to a good start

**“Tous les paramètres sont normaux et la trajectoire est normale”**

# During 39 seconds…

The software generated a number too big for the system to handle: the computer shut down and passed control to its redundant twin which, **being identical to the first**, came to the same conclusion and shut down a few milliseconds later.

The rocket, now without guidance, changed direction to compensate for an imagined error and collapsed in its own turbulence.

# Some causes



1. Software reused from the Ariane 4 series, which has special requirements and lower horizontal velocities.

2. An error while converting a 64 bit floating point number to a 16 bit integer caused an overflow: a custom floating point format for which the processor could have generated an exception error.

3. Some conversions (in Ada code) on the computers are protected from bad conversions, but this one was disabled.

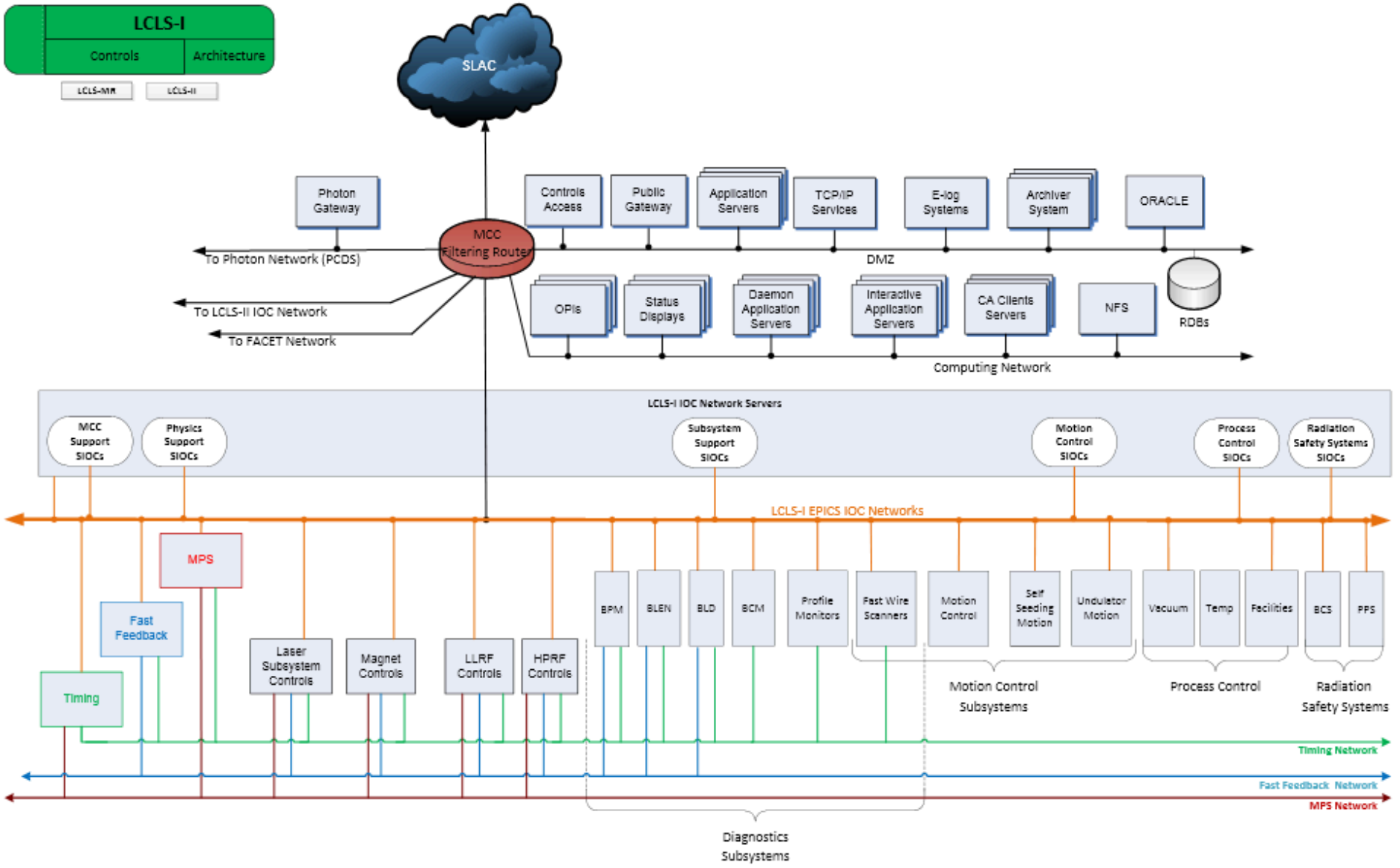
# Some causes – Cont'd

SLAC

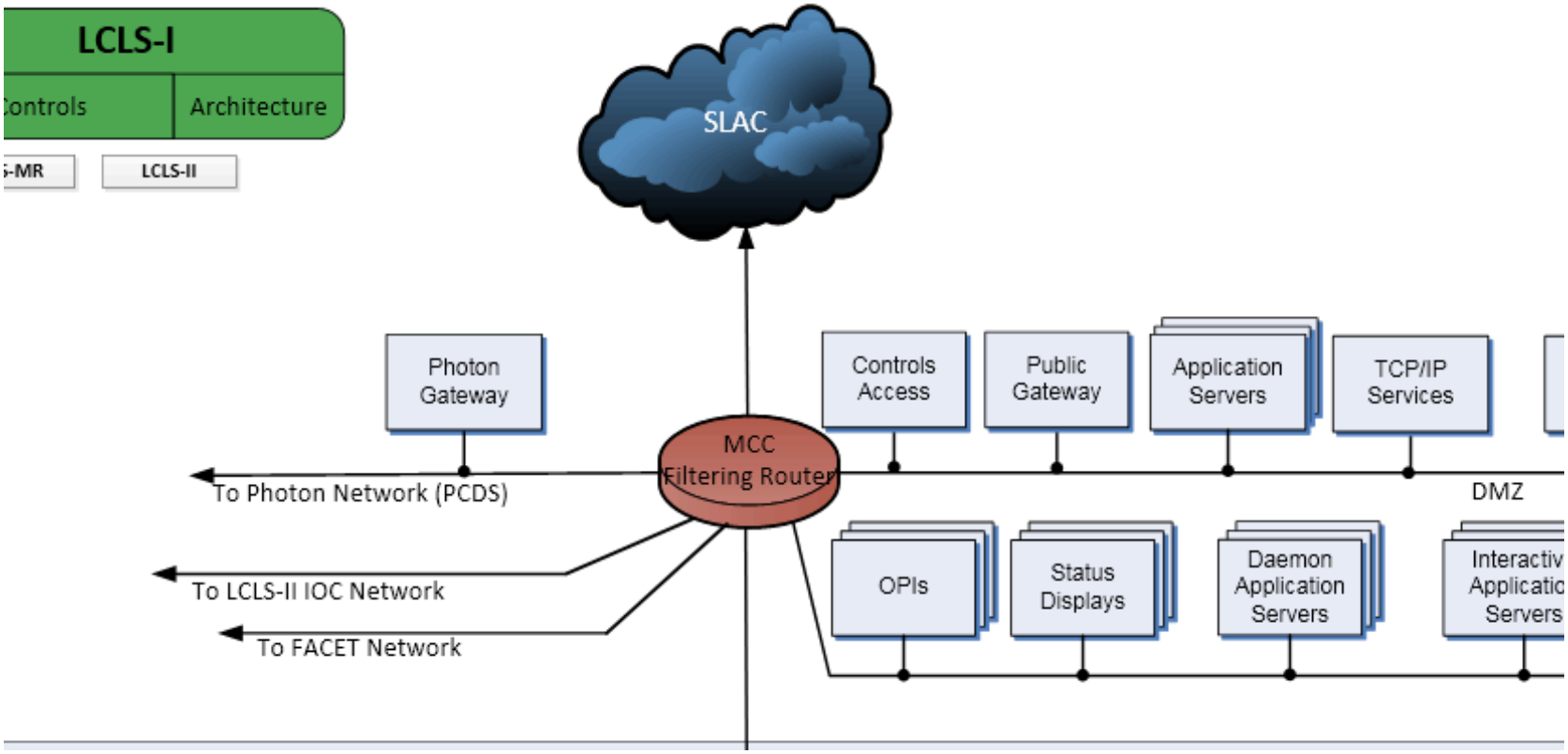
4. The primary inertial sub-computer and its backup **both** shut down because of this, and the primary started a memory dump.

5. The main computer looked at the data dump and interpreted it as flight data. The nozzles swiveled to their extreme position to try to "right" the rocket, causing it to break apart. When the boosters broke off, the whole launcher automatically self-destructed.

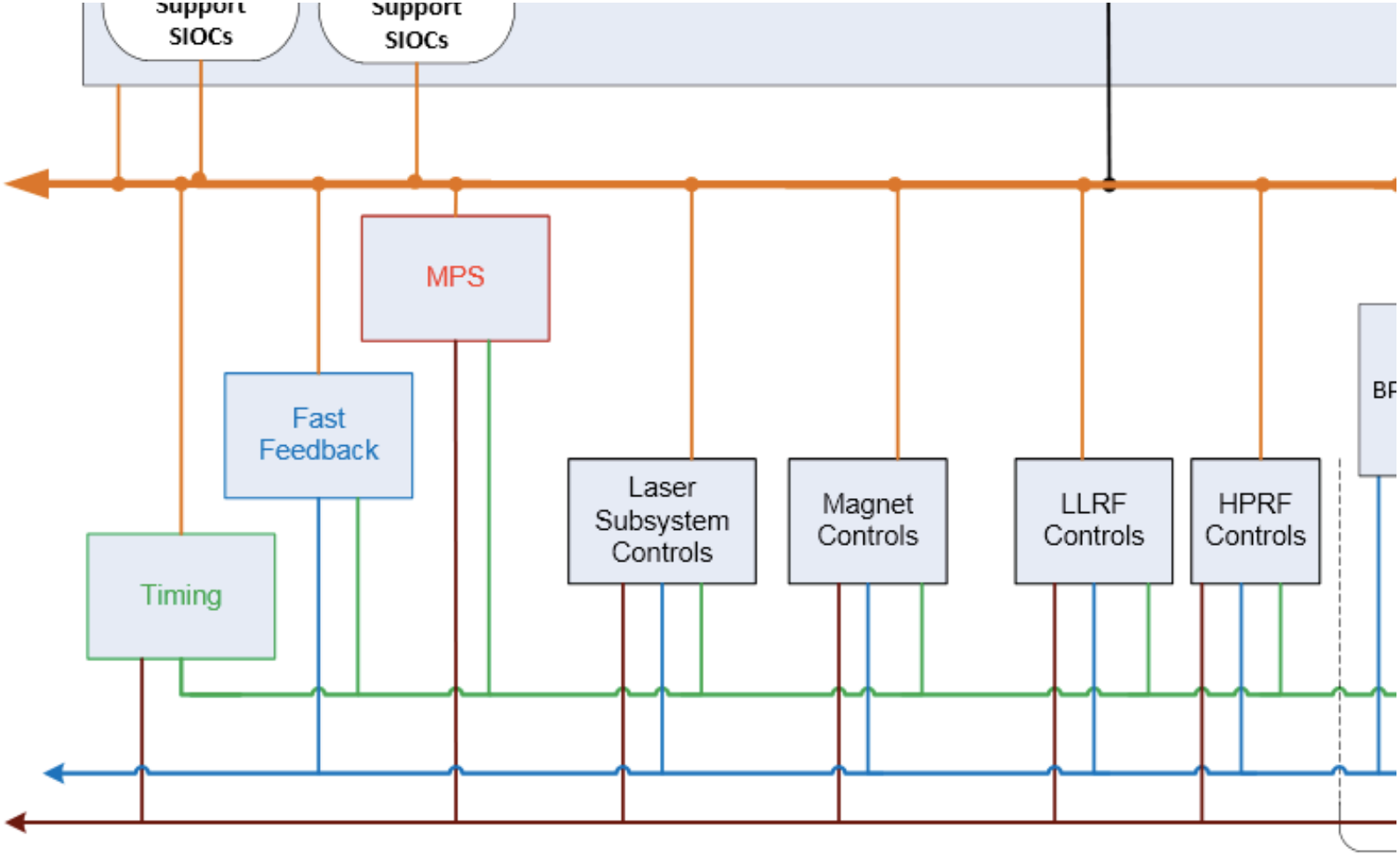
# Controls are complex systems



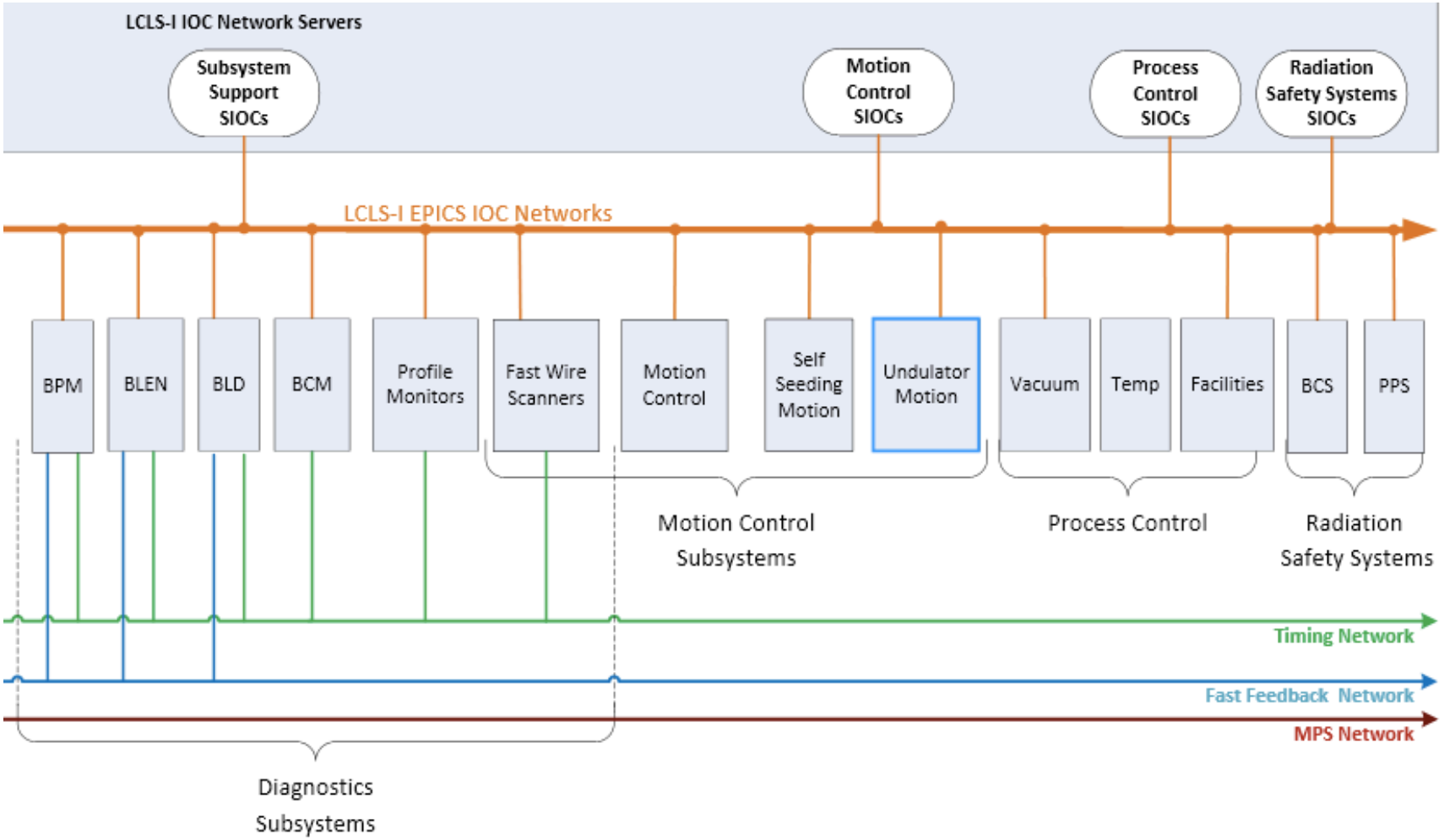
# Controls Architecture



# Controls Architecture



# Controls Architecture



# What can go wrong?

SLAC

## Anything.

1. System fails unsafe;
2. System fails when it shouldn't have failed;
3. You are trying to make changes to a program, only to discover you were working on the wrong version;
4. Wrong data received from sensors (but interpreted as true);
5. The system was changed and cannot be brought back to a previous state;
6. The systems needs to be upgraded/changed, but there is not enough documentation to do it;
7. System compromised by a malicious piece of code, gone unnoticed for a long time.

# How to mitigate some risks (today's agenda)

SLAC

Not all risks are created equal.

1. Redundancy;
2. Lifecycle management;
3. System architecture;
4. Configuration control;
5. Quality Assurance;
6. Standards;
7. Tests;
8. Documentation;
9. Cyber Safety.

# 1. Redundancy

SLAC

Sometimes it is two capacitors on a circuit-board in case one fails; other times it is the duplication of a whole system, such as in the US missile programs of the 1950s, where the Air Force built redundancy into each and every component of the entire missile.

It allows to design for and quantitatively demonstrate reliability.

Risk of common-mode failures b/w diverse software is hard to eliminate: any shared specification can lead to common-mode failures.

# 1. Redundancy – Cont'd

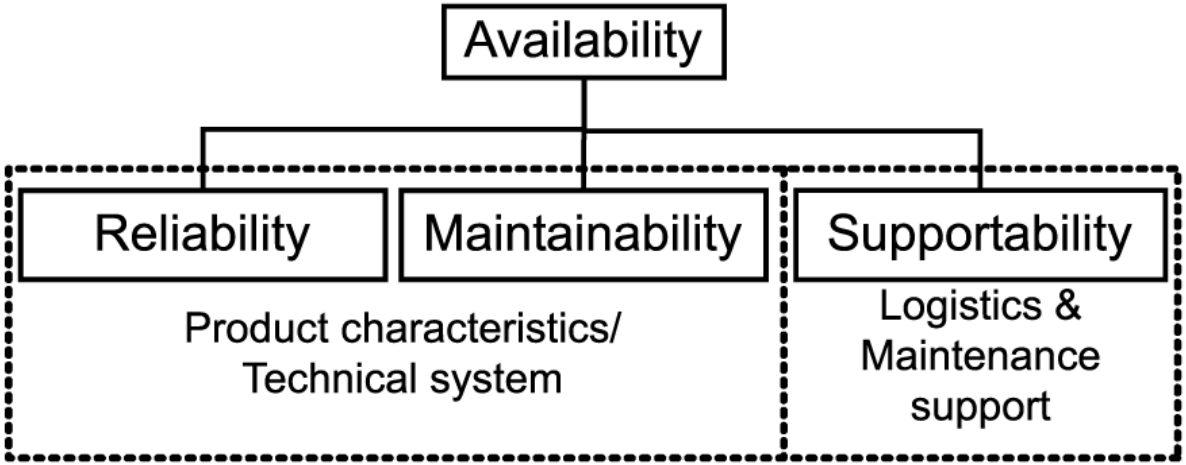
SLAC

Space Shuttle's extensive redundancy makes its workings "mysterious and unpredictable even to its designers".

Redundant elements also require further "managerial" systems to determine, indicate, and/or mediate failures (e.g. four engines on aircrafts).

Apollo 11 Capsule: "Whilst the primary control is automatic, for vehicle operation, man has been added to the system as a redundant component who can assume a number of functions at his discretion dependent upon his diagnosis of the state of the system. Thus, manual control is secondary" –i.e., Neil Armstrong was THE redundant element!

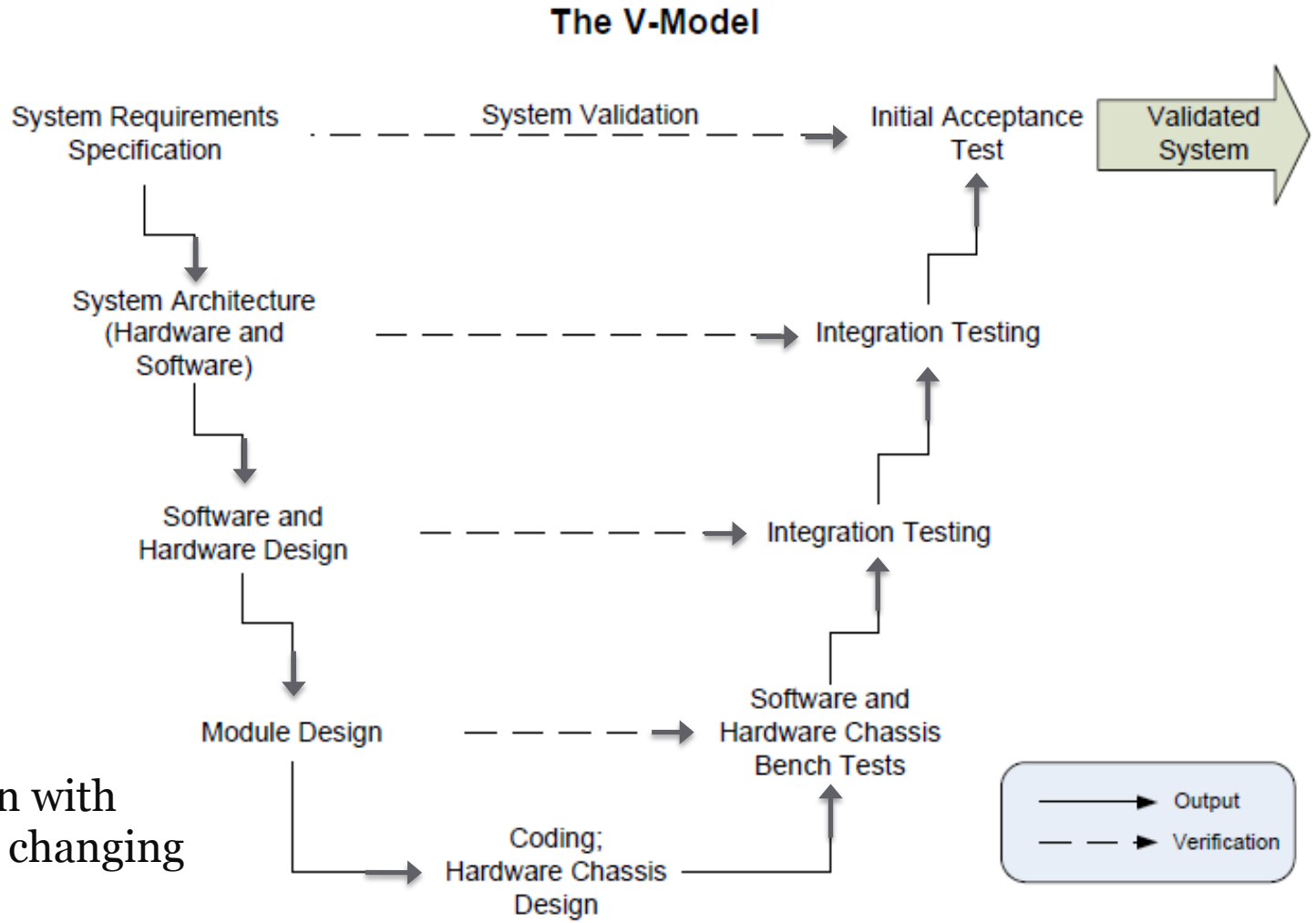
# Some constraints




Challenges:

- Diagnostics
- Maintenance
- Test Equipment
- Spare parts procurement
- Knowledge management
- Training

# 2. Lifecycle Management (Systems Engineering)




Interaction with PPS when changing modes

# Instrument Development - SW FW Environment

SLAC

Modern high performance instruments

- High speed, high resolution ADCs on Front End take in transducer data
- High speed DACs on Back End drive actuators
- FPGA between ADCs and DACs digitally process signals and implement measurement/control algorithm

A tool for FPGA algorithm development:

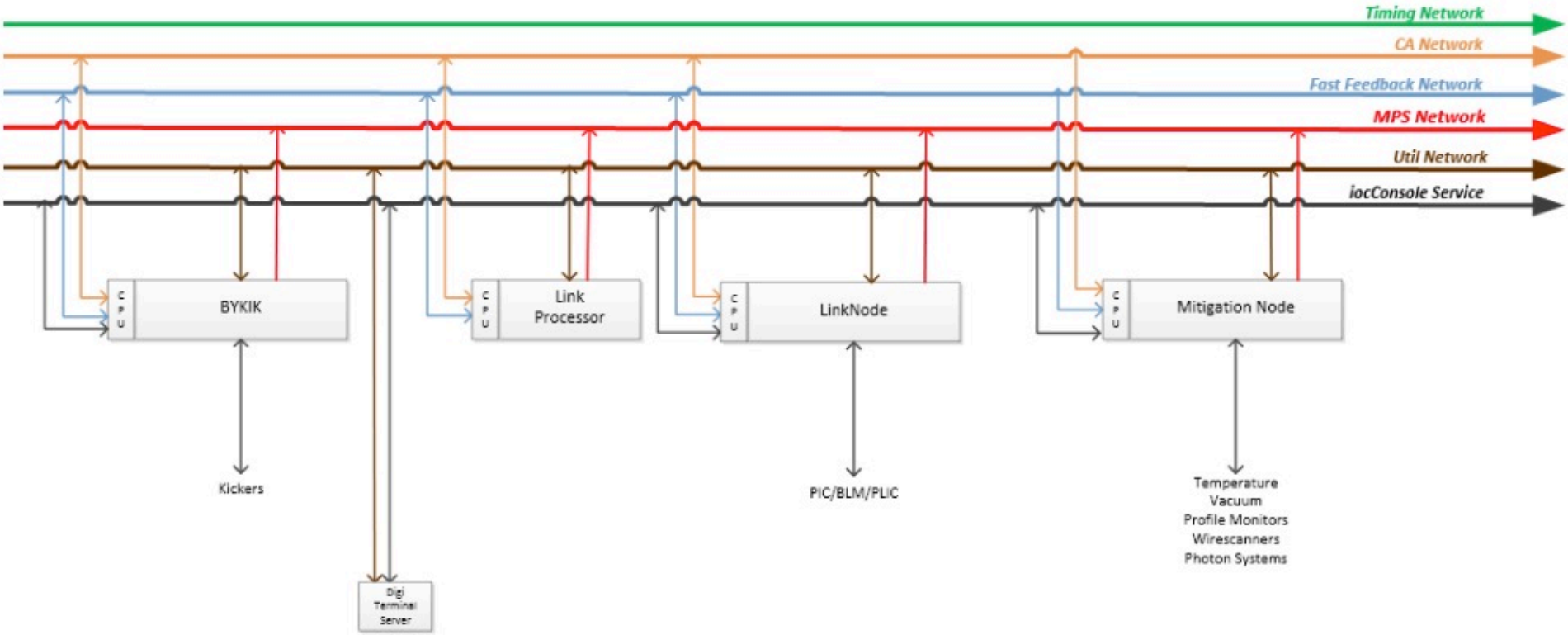
MatLab Simulink +Xilinx System Generator (Altera DSP Builder) implements Simulink algorithms with vendor IP blocks.

(VHDL Libraries à la UNICOS)

FPGAs act as digital HW, but should be treated as SW

# 3. Architectures

# MPS


Timing Network
CA Network
Fast Feedback Network
MPS Network
Util Network
iocConsole Service
CPU
BYKIK
Kickers
Digi Terminal Server
CPU
Link Processor
CPU
LinkNode
PIC/BLM/PLIC
CPU
Mitigation Node
Temperature
Vacuum
Profile Monitors
Wirescanners
Photon Systems

# BCS: A hybrid architecture

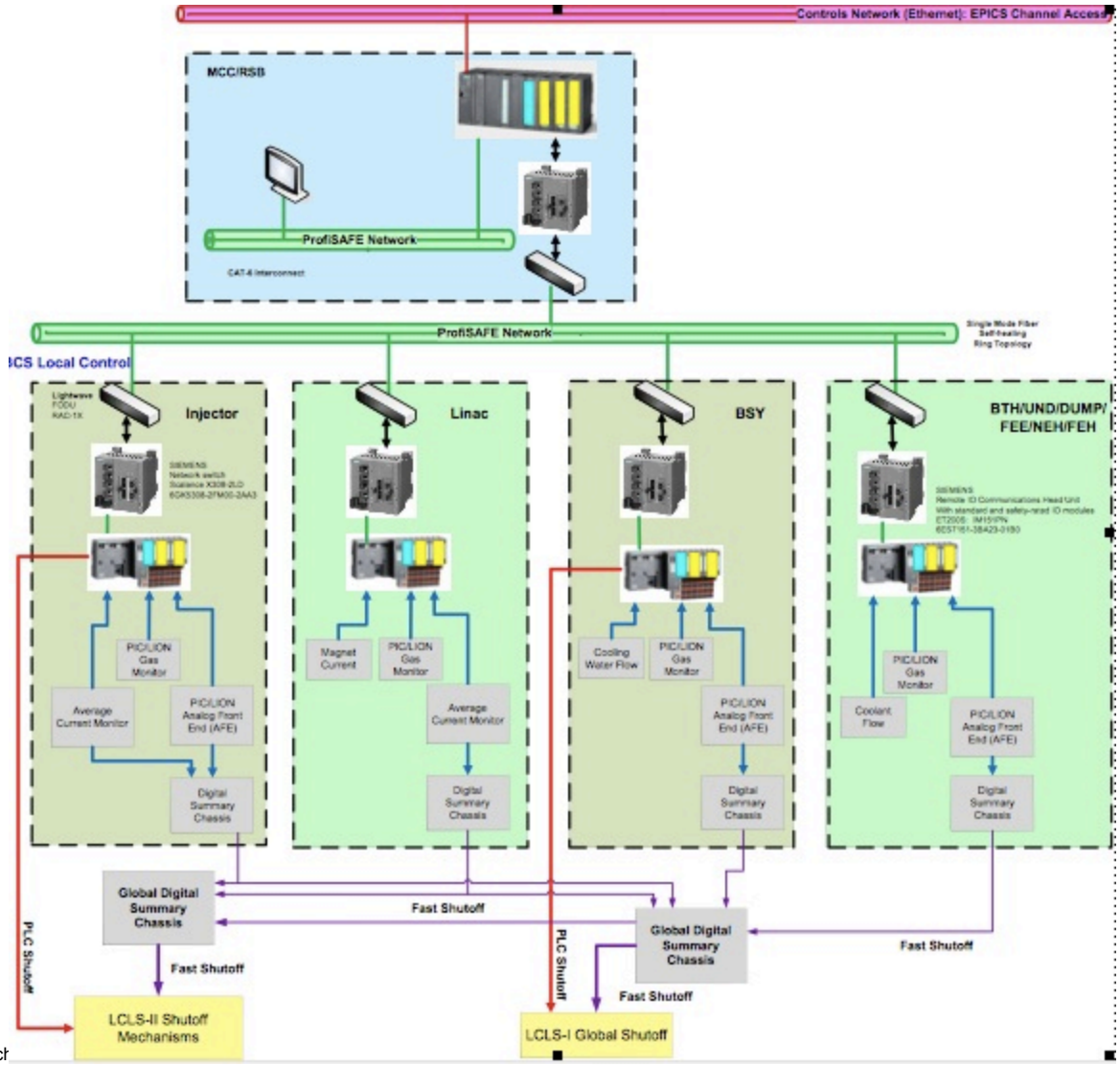
SLAC

Beam Containment System (BCS): keeps beam within prescribed channels and limits, preventing generation of excessive level of radiation in occupied areas.

It also protects the integrity of safety-related beam line components.

BCS performs this function by monitoring beam power and beam loss, and shutting beams down if a limit is exceeded.

# BCS: A hybrid architecture – Cont'd



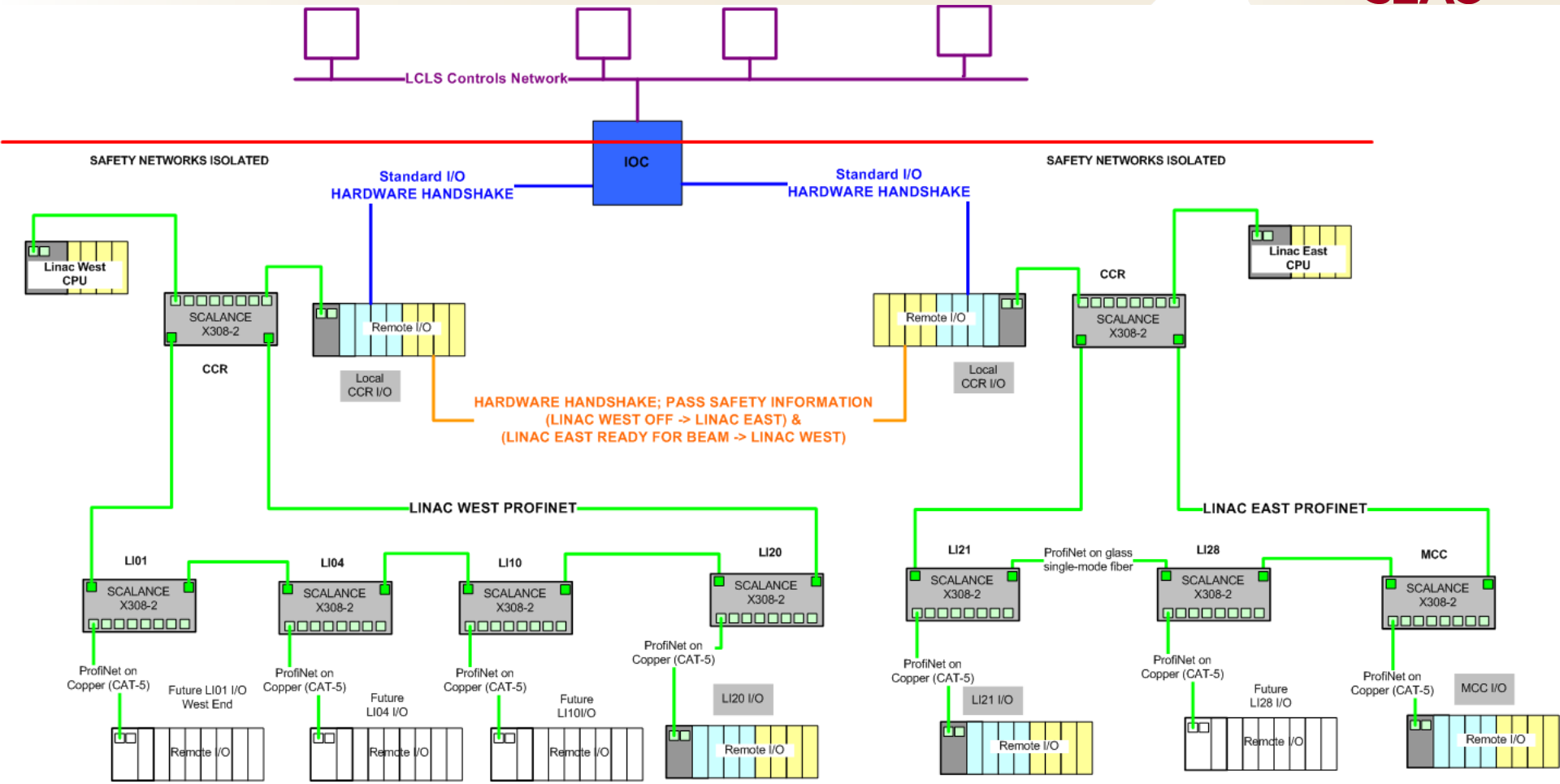
# BCS: A hybrid architecture – Cont'd

SLAC

Remote I/O drop acts as a data hub:

- Slow sensors directly connect to PLC
- Custom chassis for fast sensors
- Chassis provide "fault" information to local Digital Summary chassis
- Global Digital Summary chassis for fast shut off of output devices
- PLC provides diagnostics and configuration control

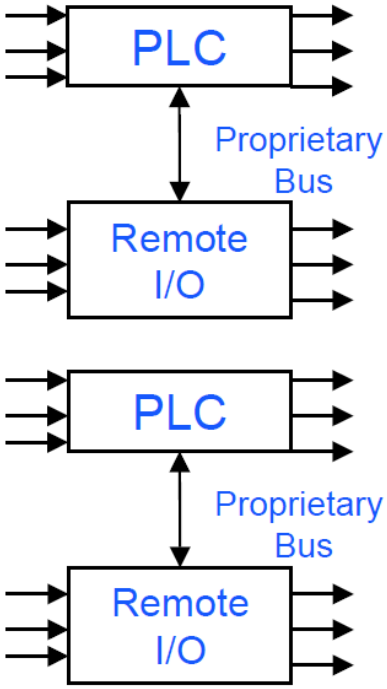
# PPS

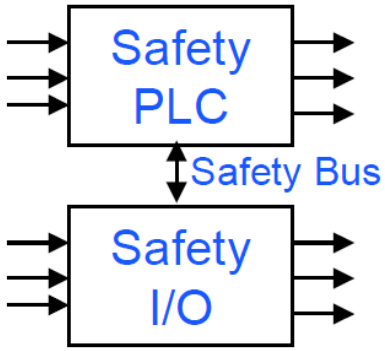

LCLS Controls Network
IOC
SAFETY NETWORKS ISOLATED
Standard I/O
HARDWARE HANDSHAKE
Standard I/O
HARDWARE HANDSHAKE
SAFETY NETWORKS ISOLATED
Linac West CPU
SCALANCE X308-2
CCR
Remote I/O
Local CCR I/O
HARDWARE HANDSHAKE; PASS SAFETY INFORMATION
(LINAC WEST OFF -> LINAC EAST) &
(LINAC EAST READY FOR BEAM -> LINAC WEST)
Remote I/O
Local CCR I/O
CCR
SCALANCE X308-2
Linac East CPU
LINAC WEST PROFINET
LINAC EAST PROFINET
LI01
LI04
LI10
LI20
LI21
LI28
MCC
SCALANCE X308-2
ProfiNet on Copper (CAT-5)
ProfiNet on glass single-mode fiber
Future LI01 I/O West End
Future LI04 I/O
Future LI10I/O
LI20 I/O
LI21 I/O
Future LI28 I/O
MCC I/O
Remote I/O

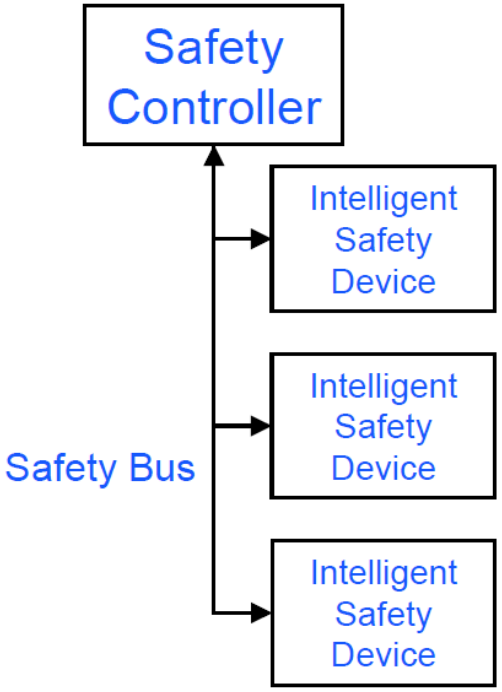
# Evolution of Safety Systems at SLAC

SLAC

| Accelerator Custom Design | Process Safety Design | Machine Safety Design |
|---|---|---|



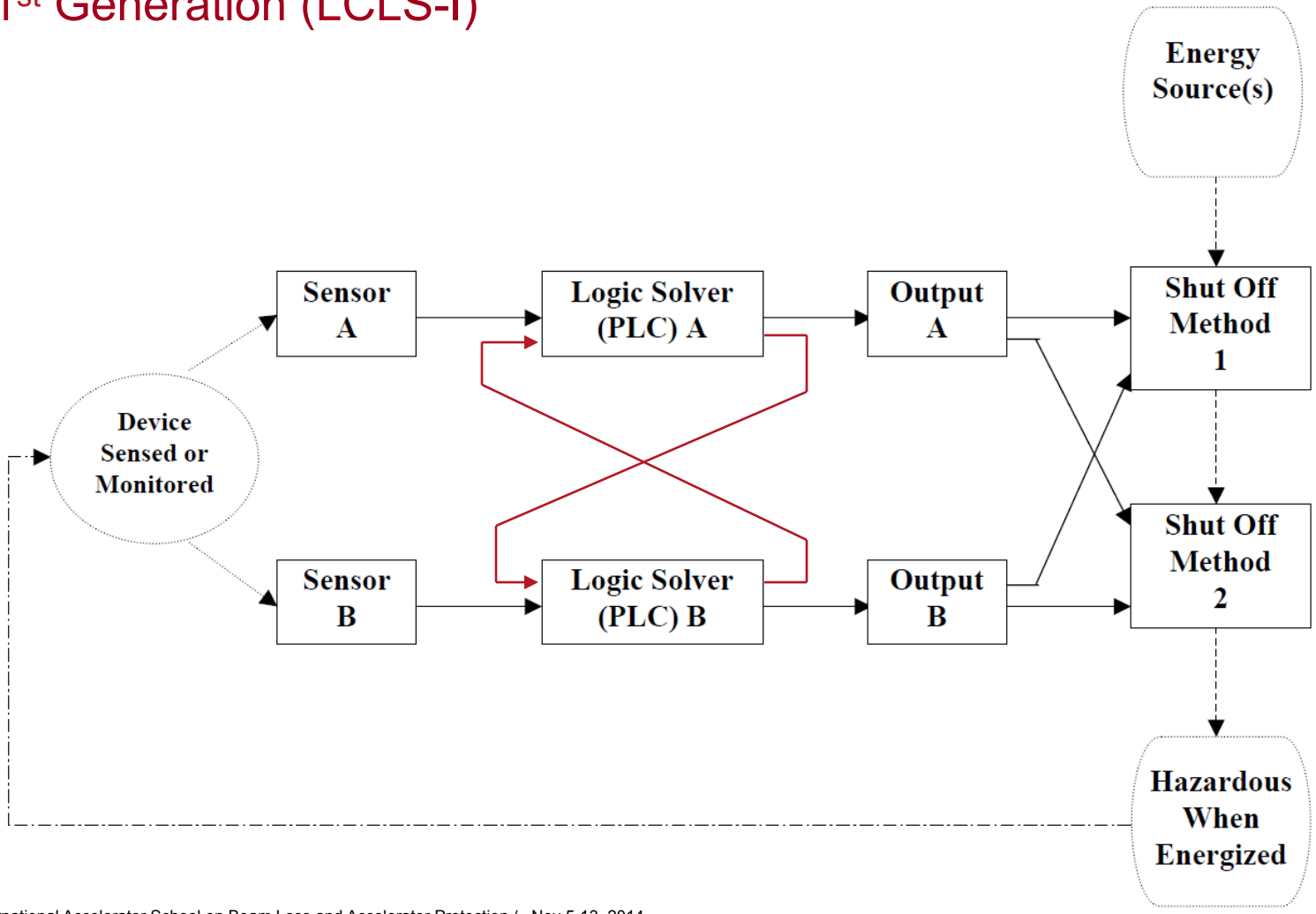




**2008 (LCLS-I)** → **2011 (CCR Upgrade)** → **2014 (LCLS-II)**

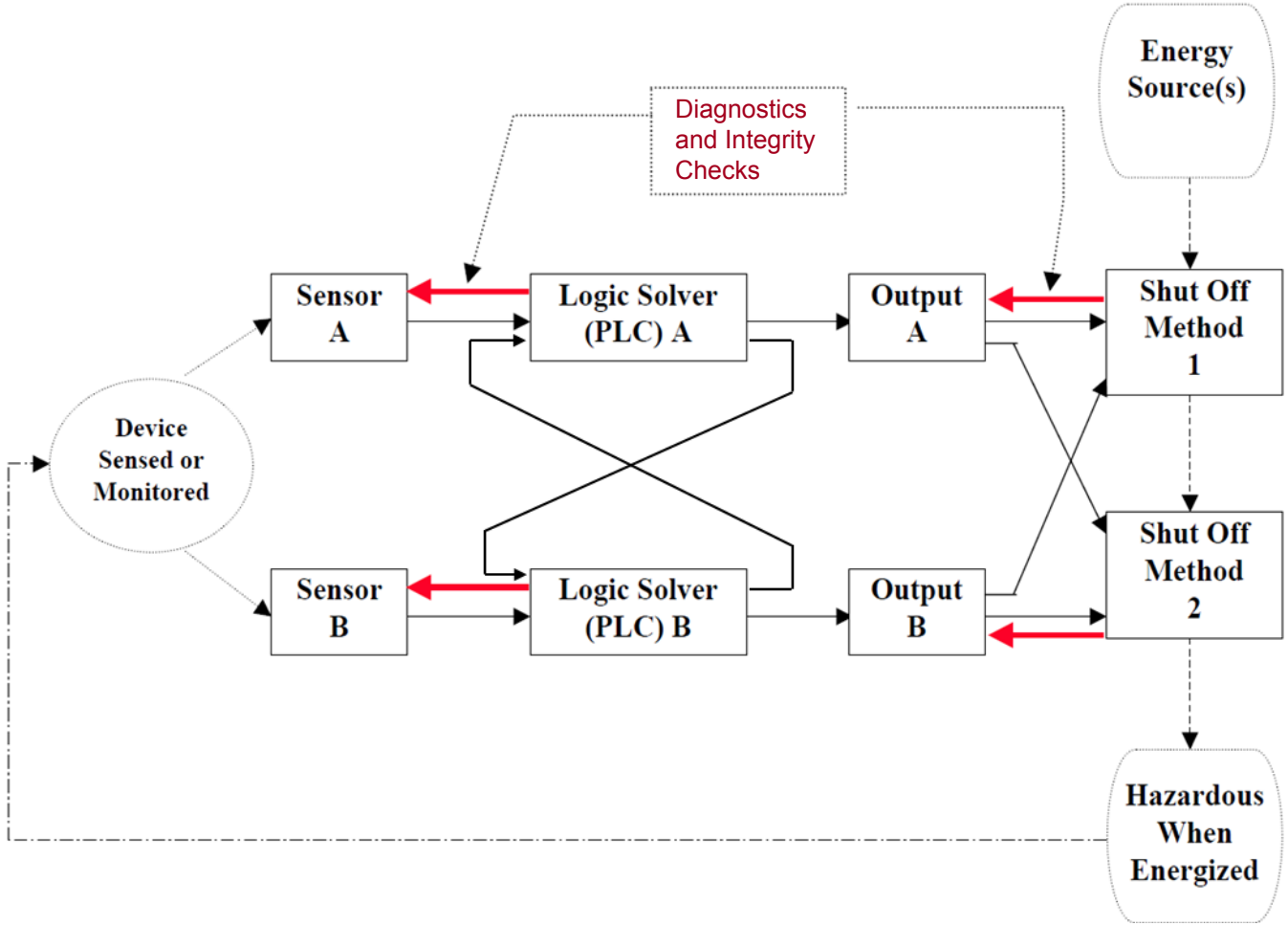
# Traditional Architecture

## 1st Generation (LCLS-I)



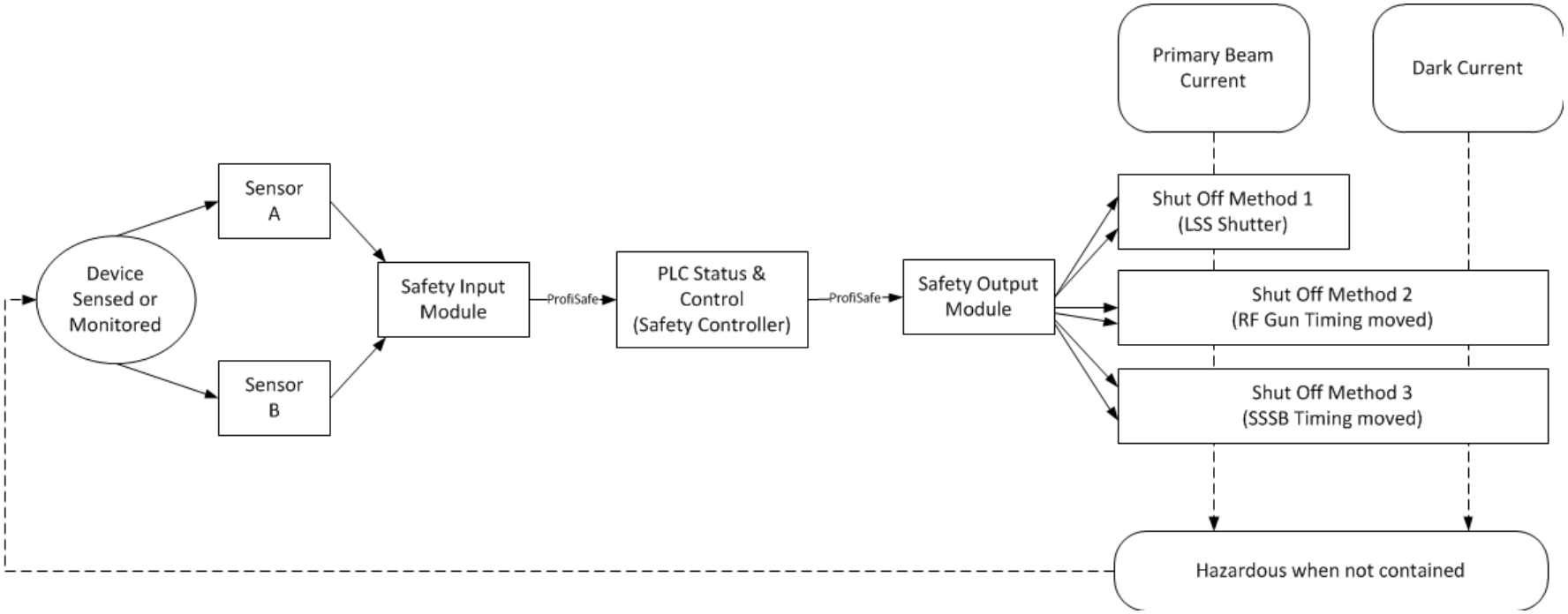
# Modified Traditional Architecture

## 2nd Generation (CCR Upgrade)



# Safety PLC

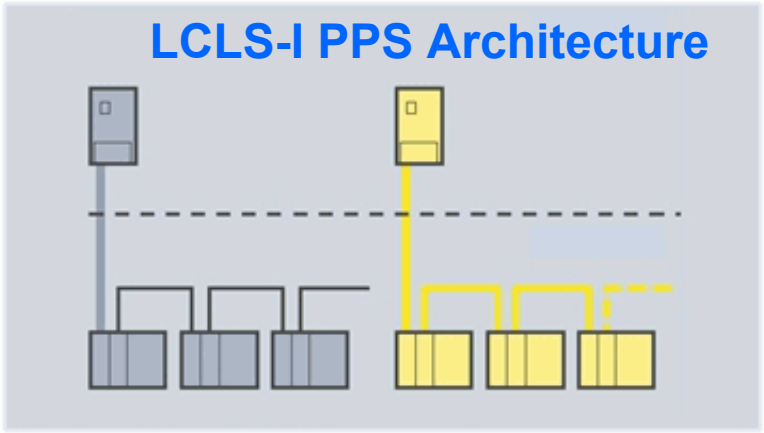
## 3rd Generation (BCS Upgrade)



# Option: A Single-PLC System

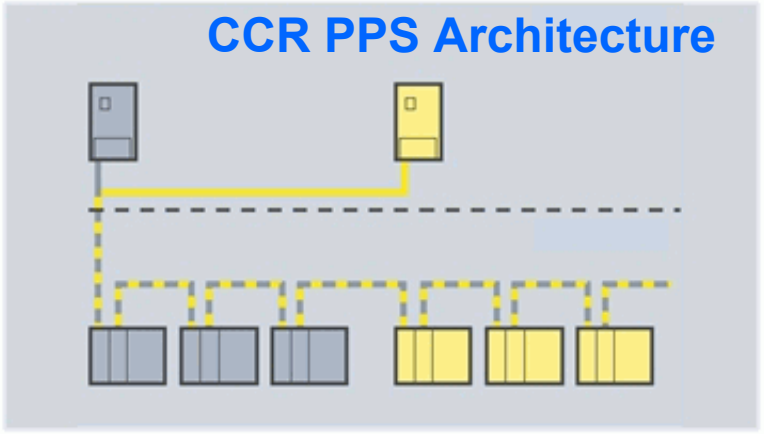
SLAC

## Identical functionality based on a different system architectures

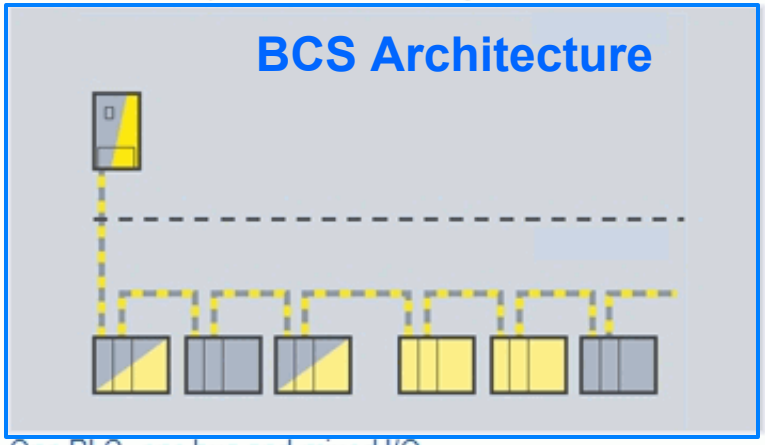



Separate PLCs, I/O and bus systems

One PLC but separate I/O and bus systems




Separate PLCs, one bus system, separate I/O




One PLC, one bus and mixed I/O

# Reliability of a Single-PLC Based System

SLAC

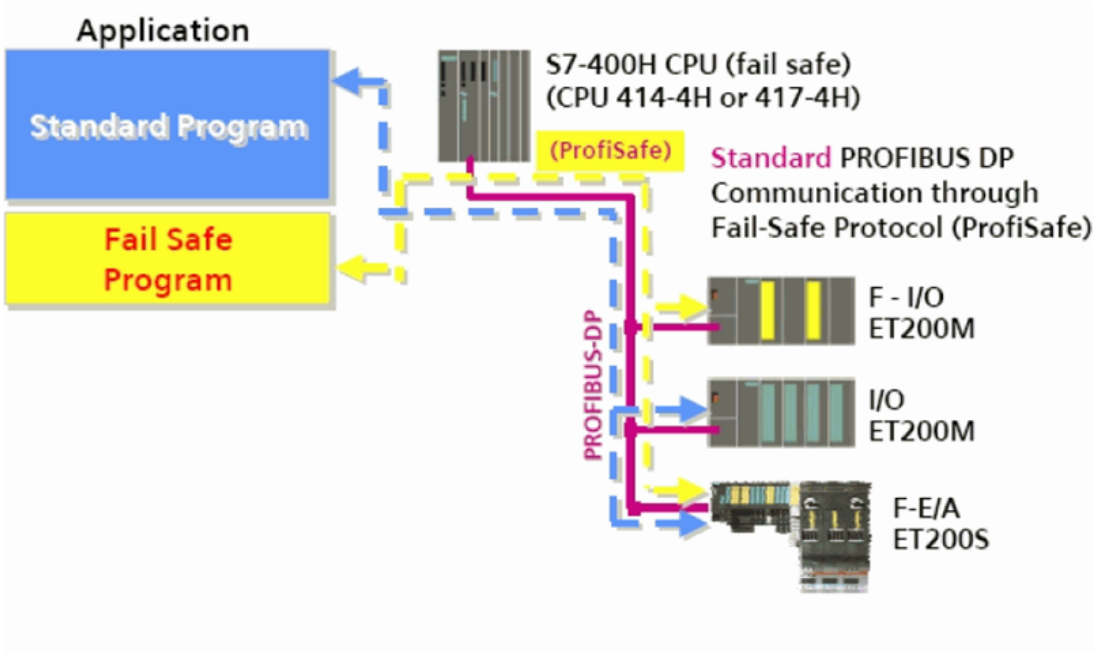
**A single CPU operating in a 1001D structure is certified by TUV for SIL 3.**

Four layers of protection:

1. **Failsafe input module** (monitored signal wire; local protection with lockout & reset; discrepancy analysis and time out for faulty inputs; communication watchdog);
2. **Safe communication network** (fault detection, fault reaction, recovery);
3. **Safety rated controller** (redundant evaluation of inputs);
4. **Failsafe output module** (tests itself with internal test pulses; safe state failure mode for communication failure; no reliance on PLC for errors handling).

# Coexistence of standard and safety program in a controller

- Separation of standard and failsafe program;
- Changes in the standard program do not affect the integrity of the safety program;
- One engineering platform for both applications;
- Protection of the Safety program from unauthorized access.

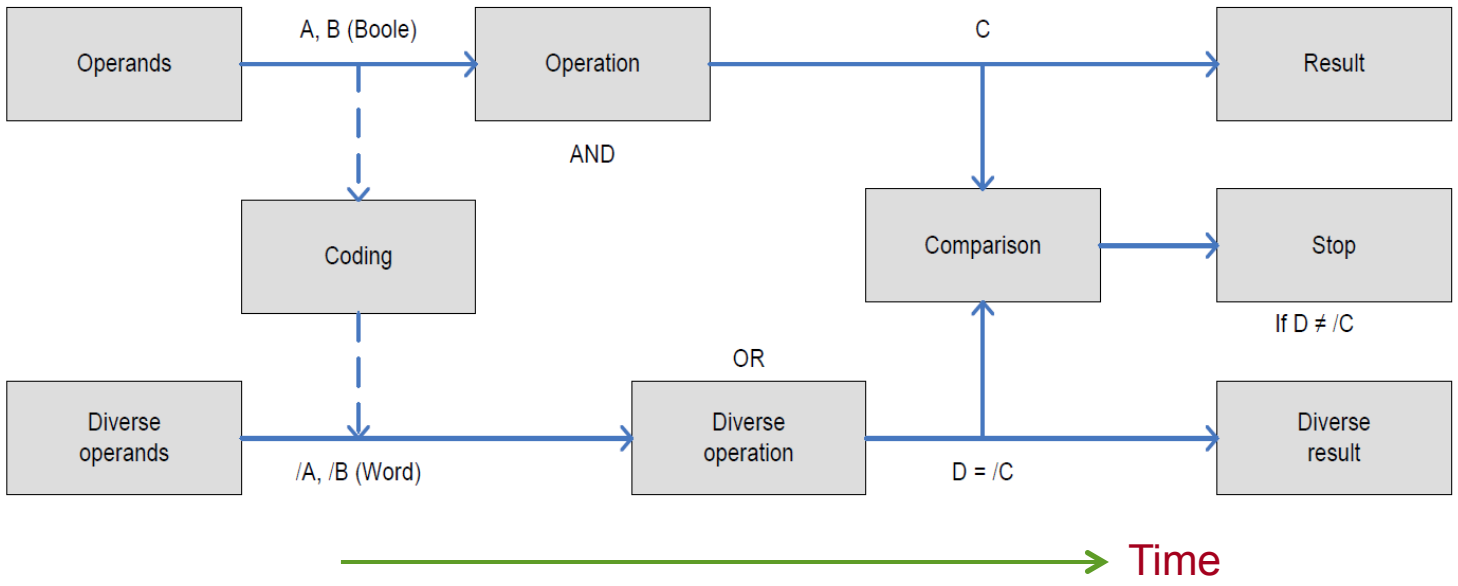



**Field Hardware (PLC inputs) is still redundant.**

# Logic Solver Internal Redundancy

SLAC

## Time Redundancy/Diversity instead of Structural redundancy




- Two copies executed in parallel: one using 16-bit word mode, the second using single-bit binary instruction; the compiler changes one into another (diversity);
- When D ≠ C, the safety program is executed. Twice (redundancy);
- Bool and Word Operations processed in different parts of the CPU;
- Two independent hardware timers.

# 4. Configuration Control

A DOE National Laboratory, 2007:

There was a failure of a PLC;

The PLC was repaired using the incorrect version of software.

# Day two

SLAC

There were two simultaneous core dumps.

The recovered system reveals problems; the status reported on EPICS was not all correct.

Development Software Locations:

Not in CVS
A:\Floppy Disks
C:\Desk Top Drive
D:\CD Burner
F:\Flash Drives
V:\Group Drive
Z:\Employee Drive

# Day three



The recovered system required certification, but the certification document was not consistent with the previously executed version.

A hardcopy was not available, so a document was printed from Microsoft Word.

Track Changes function in Microsoft Word created two versions.

An investigation ensued.

# Software Control

| Requirements | Personnel Protection | Machine Protection |
|---|---|---|
| **Use of Configuration Versioning System (CVS) for Software** | YES | YES |
| **Manage check-in/out of CVS with procedures** | YES | YES |
| **Track & Check Checksum** | Yes; Additional "Safety Signature" available for Safety-Rated PLCs | YES |
| **Software Download is password protected** | YES | YES |
| **Download over network?** | No; not allowed. Local ProfiBus connection only allowed. | YES |
| **Download to wrong CPU across network?** | No; isolated networks and different CPU names/IP addresses even if on same network | No; isolated networks and different CPU names/IP addresses even if on same network |
| **Protection against wrong safety program load** | Hardware configuration is loaded; safety modules have hardware DIP switches. Hardware configuration error causes fail-safe shutdown | NO |

CVS: Single location to retrieve approved SW

# Software Control – Cont'd

| Requirements | Personnel Protection | Machine Protection |
|---|---|---|
| **Physical Isolation from Controls Network** | NO | NO |
| **Possible accidental download of Safety-Critical program from controls network** | NO; local download only | YES |
| **Possible intentional sabotage of Safety-Critical program from controls network** | NO; local download only | YES |
| **Possible accidental download of Access Control (non-safety critical) progam from controls network** | YES | YES |
| **Possible intentional sabotage of Access Control (non-safety critical) program from controls network** | YES | YES |
| **Possible accidental changes to EPICS HMI panels from controls network** | YES | YES |
| **Possible intentional change to EPICS HMI panel from controls network** | YES | YES |

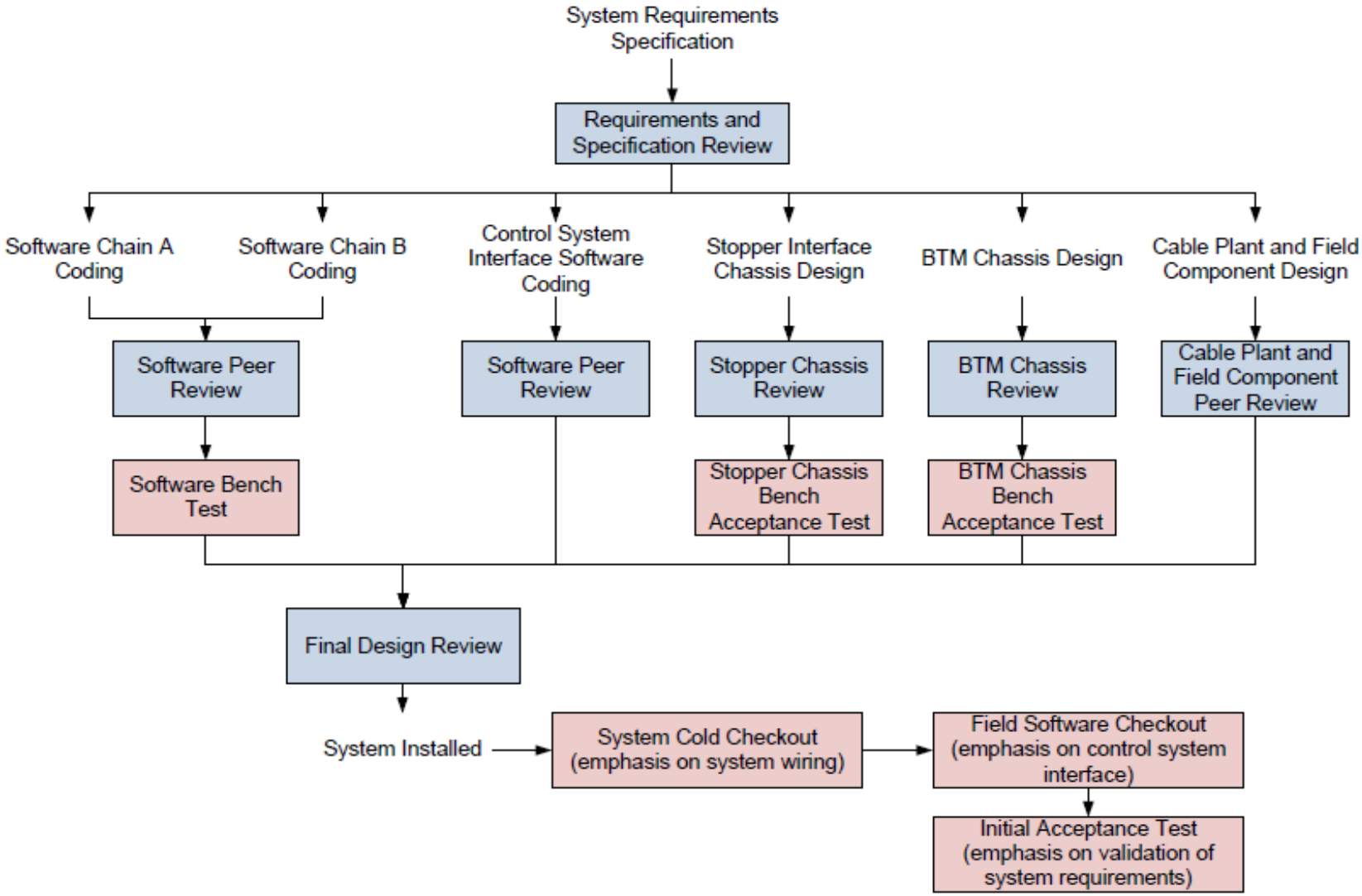
# Firmware updates

SLAC

When do you patch/update your Server/Industrial PC/PLC OS/firmware?

- Is it a critical fix?
- Can it wait?
- Will the code/IP still work?
- Database to track the fw update?
- Point of contact for vendors?
- Maintenance contract?
- When committed to the new version, all systems have to be changed.
- Once done the entire system has to be re-tested.

# 5. Quality Assurance



# Review Process

SLAC

## Minimum Recommended Reviewer Complement

| | |
|---|---|
| Minor Modification, familiar methods | 1 external reviewer |
| Minor Modification, new methods | 2 external reviewers |
| Medium Change, familiar methods | 2 external reviewers |
| Medium Change, new methods | 2 or more external reviewers,<br>1 external to Control Dept |
| Large Change, familiar methods | 3 external reviewers,<br>1 external to Controls Dept |
| Large Change, new methods | 3 or more external reviewers,<br>1 or more external to Controls Dept,<br>1 external to the laboratory |

# Review Process

SLAC

- Minor Modifications: adding or moving an emergency off button, BSOIC, or Ion Chamber, equivalent device substitutions such as upgraded annunciator panels, or minor logic changes that improve performance but are not changes in the logic specification;

- Medium Changes: redesigns of stopper, BTM, BSOIC, PIC Chassis, or power supply interface chassis, or minor changes in PPS logic specification;

- Large Changes: new PPS zones, new BCS regions, complete PPS rebuilds or significant logic modification.

# Programming Flavors

SLAC

**Fixed Programming Language (FPL):** ASIC, embedded Code… - only operating parameters may be changed. E.g., smart sensor.

**Limited Variability Language (LVL):** Program runs in restricted memory space (restricted by hardware). IEC 61131-3. E.g., PLC.

**Full Variability Language (FVL):** C, JAVA, etc. Allowed to write any value in any memory space.

# VHDL



VHSIC Hardware Description Language (VHDL) –where VHSIC: Very High Speed Integrated Circuit.

Elegantly represents parallel processing.

Modular language, similar in visual style to Ada.

Features: inclusion of time/clock processing and the facilities for describing low-level circuits.

It can be simulated, rather than executed, and these simulations may give different results to actual implementations if VHDL is poorly constructed. This means that **simulation cannot, in general, be considered equivalent to software testing**.

# IEC 61508, Part 2 and 7

SLAC

- Use of structured and modular design
- Restricted use of asynchronous constructs
- Design for testability
- Restrictive use of ambiguous constructs
- Transparent and easy to use code
- Defensive code and range checking (o pick up faults or anomalies and respond in a pre-determined way)
- Use of comments and annotations
- Limits on module sizes and number of ports to increase readability
- Avoidance of multi-dimensional arrays, goto type commands
- Avoidance of redundant logic and feedback loops
- Avoidance of latches, asynchronous reset

# 6. Standards (Hardware)

# CAMAC



**Computer Automated Measurement And Control:**

- Standard bus and modular crate electronics standard for DAQ and Controls, defined in 1972;
- Solved the low channel density problem of NIM;
- Up to 24 modules in a crate, interfaced to PC;
- Not hot-swappable because of backplane design;

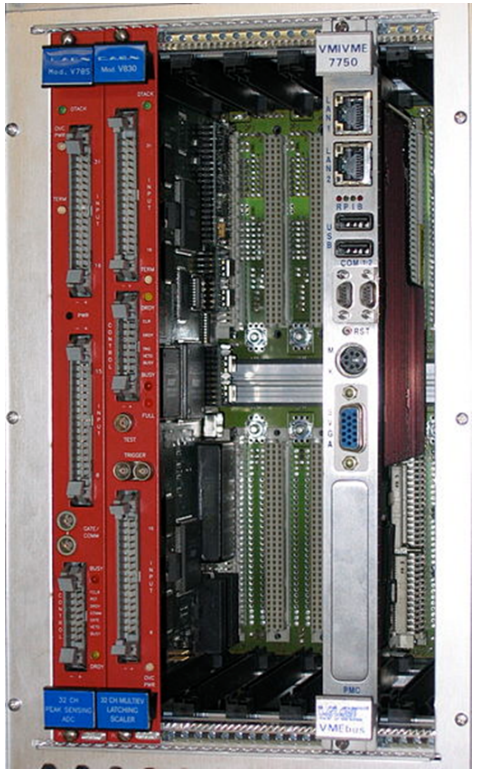
- Dataway manages: Module power, address bus, controls bus and data bus;
- 24-bit communication b/w controller and selected module.

# VME



## Versa Module Europa

- Standard backplane bus, defined in 1981;
- Architecture not scalable for high speed (single ended parallel bus not for Gbps);
- EMI shielding not specified;
- Developed for Motorola 68000 line of CPUs (the bus is equivalent to the pin of 68000 run out onto a backplane);
- Faster bus (from 16 to 64 bit), up to 40 MHz (VME64).



# They can get quite busy…



# µTCA



- First modular computer architecture with completely serial multi-Gbps backplane
- Serial ports are bidirectional pairs in star or mesh topology
- Intelligent Platform Manager Interface IPMI for cooling and thermal management, control and monitor

Modules:

- Cooling Units (CU);
- Power Modules (PM);
- Advanced Mezzanine Card (AMC) for electronics, CPU, hard drives;
- Rear Transition Module (RTM);
- µTCA Central Hub (MCH)

# Reliability



Built-in hot-swap.

Designed for high-reliability:

- Intelligent Platform Manager Interface IPMI for cooling and thermal management, control and monitor;
- Built-in crate and component status monitoring and remote management/diagnostics;
- Independent monitoring channel within the crate.

# Network Attached Devices (NAD)

SLAC

**AKA "Pizza Box".**

Guidelines for NAD design have to be provided on a case-by-case basis, i.e., ***there is not a standard to design a NAD-based system***.

Every board is unique, as it is the creation of the design engineer.

A NAD-based board can be equipped with plenty of cards from many vendors.

# 7. Tests

SLAC

System and Integrity checks built into the lifecycle.

Tests have to be reviewed and documented.

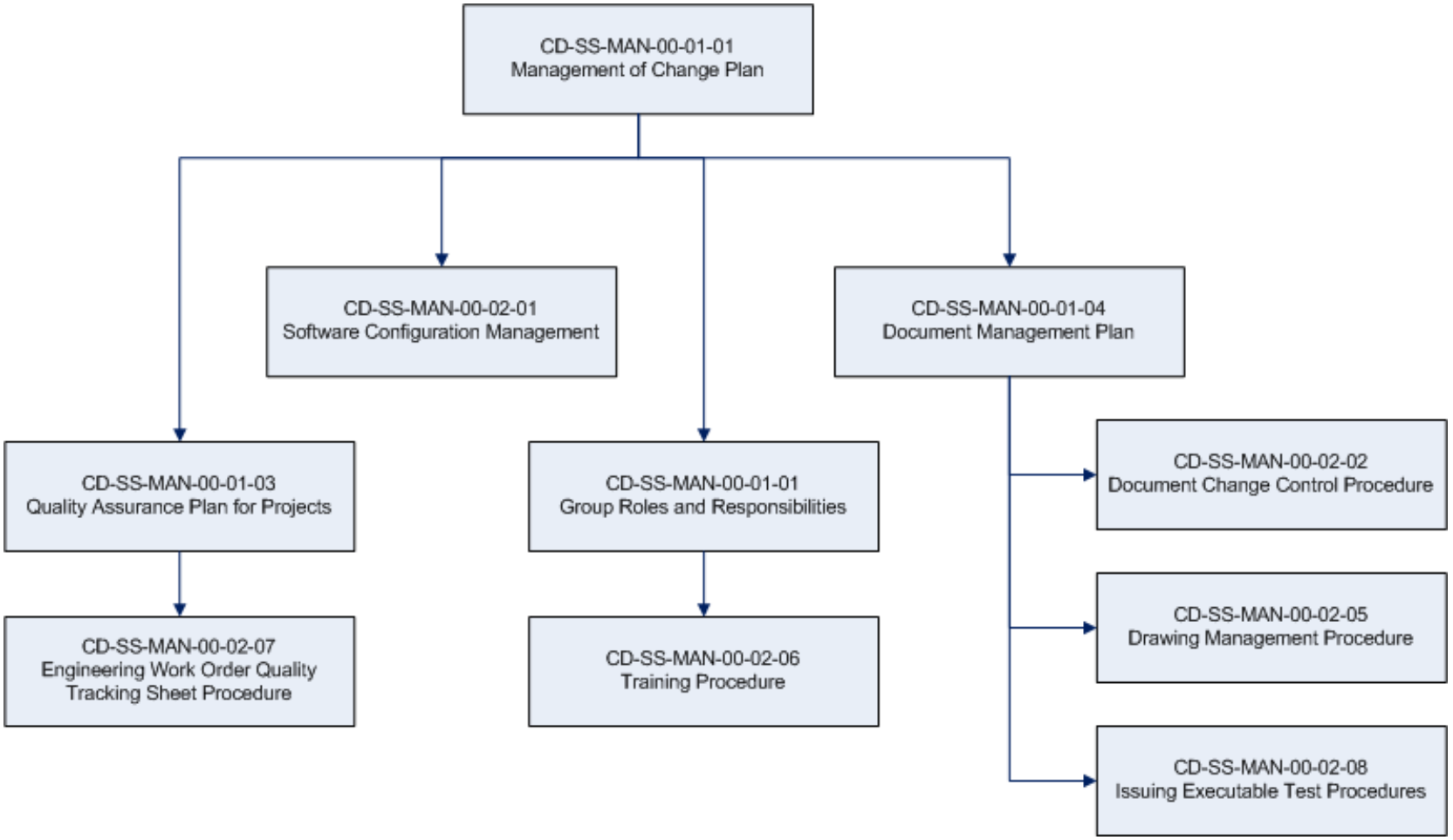
System and integrity check should be performed regularly.

Test procedures can have zero-error tolerance (cold- and hot-checkout, test bench, test validation, test execution).

Executed Test procedures have to go under configuration management.

Test procedures should go through a closeout process.

# 9. Documentation



# 10. Cyber Safety

# Evolution of regulatory landscape: EO

SLAC

**Obama's Cybersecurity Executive Order (Feb 2013):**

- Agencies to share information with companies
- NIST to develop a framework with industries
- Review CS regulation
- Voluntary compliance

Deadline: Feb 2014

Leadership: NIST

# Evolution of regulatory landscape

SLAC

**Obama's Cybersecurity Executive Order (Feb 2013).**

**Framework for Improving Critical Infrastructure Cybersecurity (NIST, Feb 2014):** a system of regulations and the means used to enforce them.

1. Core functions (activities and references);
2. Implementation tiers (guidance);
3. Framework profile (how to integrate CS functions within a CS plan).

# Framework's core

SLAC


Asset Management," "Access Control," "Detection Processes".

"External information systems are catalogued," "Data-at-rest is protected," "Notifications from detection systems are investigated".

# NIST SP 800-53 – IT CS Land

NIST Special Publication (SP) 800-53 (Computer Security Guide) – Rev. 4 published in April 2013

**Information Security Program**:

- Risk assessment
- Policies and procedures
- Subordinate plans
- Training
- Periodic testing
- ATS
- Incident response
- Continuity of Operations

# DOE Order 205.1B

DOE O205.1B **Cyber Security Program**:

- Integrated, Enterprise-Wide
- Risk Managed Approach (vs. Controls Compliance)
- Configuration Management
- Consistent with NIST
- Line Management Accountability
- Integrated into GOCO model
- Protection of DOE information and information systems
- Mission focused
- Governance

# DOE Order 205.1B – Cont'd

DOE oversight is conducted through **Contractor Assurance Systems**: this provides the *Authorization Function.*

**Risk Management Approach** (RMA): Framing; Assessing; Responding; Monitoring.

**What do you know, you are not alone anymore!**

# IEC 17799 and ISA-TR99

SLAC

**IEC 17799:** Information Technology – **Code of practice for Information Security Management** :

High level, broad in scope, conceptual in nature and a basis to develop your own security standard and security management practices.

**ISA-TR99: Integrating Electronic Security into the Manufacturing and Control System Environment:**

Guidance to user and manufacturers, analyzing technologies and determining applicability to securing Manufacturing and controls.

# IEC 15408 and IEC 27001

SLAC

## IEC 15408 (3.1) – Information Security Management Systems (ISMS):

Framework to specify security **functional and assurance requirements** through the use of Protection Profiles. **Vendors** can implement security attributes, **testing labs** can evaluate the products.

## IEC 27001:2005 - Common Criteria for Information Technology Security Evaluation (aka CC):

System to bring information security under **explicit management control**:

- Policies and governance; Asset management; HR security;
- Access control; Incident management; Business continuity; etc.

# Your interface with IT

SLAC

**Key to success is to engage in a proactive, collaborative effort between management, controls engineers and IT Department.**

**NIST 800-53 is king.**

**Along came NIST 800-82.**

- Many times a CS team (an "enclave") exists already.
- "Ah, we're not sure we can share such information with you"…
- They might even tell you that it is impossible to gain access.
- You are the bridge between 800-53 and 800-82.
- You will have to provide the expertise to implement it.

# NIST SP 800-82

SLAC

- Latest release: Rev.2, **May 2014**
- "***Defense-in-depth***" strategy: layering security mechanism so that impact to one mechanism as a result of failure is minimized.
- Includes:
  - ICS policies based on DHS Threat Level;
  - Implementing multi-layer network topology;
  - Provide logical separation b/w corporate and ICS networks;
  - Use DMZ (i.e., no direct communication b/w ICS and corporate);
  - Fault-tolerant design;
  - Redundancy for critical components;
  - Privilege management;
  - Encryption.

# NIST 800-82 Do we speak the same language?

SLAC

| Requirement | ICS | IT |
|---|---|---|
| **Performance** | Time critical, deterministic. | High throughput. Reliable. Jitters ok. |
| **Availability** | Continuous processes. No start/stop. Planned outages, no rebooting. Redundancy. | Tolerable. |
| **Risk Mgmt** | Human safety paramount. Compliance, losses, damages. Fault intolerant. | Data integrity paramount. Priority: CI(A). Recover by reboot. |
| **Architecture Security** | Protect edge systems (PLC, DCS). | Protect assets and info (often centralized). |

# NIST 800-82 - Do we speak the same language? – Cont'd

SLAC

| Requirement | ICS | IT |
|---|---|---|
| **Physical Interaction** | Yes, certification. | Coming up (internet of things) |
| **Time-Critical Response** | Yes (e.g. HMI password shouldn't compromise emergency actions) | No |
| **System Operations** | Control Engineers are not IT. Legacy, proprietary systems (support?) | More tools available. |
| **Resources** | Limitations on CPU, third part security solutions | Plenty |

# NIST 800-82 - Do we speak the same language? – Cont'd

SLAC

| Requirement | ICS | IT |
|---|---|---|
| **Communications** | Often proprietary | Plenty |
| **Change Mgmt** | Requires tests by vendors and re-certifications. Old OSs no longer supported. | Same |
| **Managed Suppt** | Often single vendor. | Often same challenge. |
| **Component Lifetime** | 15-20 years | 4 years |
| **Access to Components** | Remote, hazardous locations | Easy |

# Towards a two-tiers approach

SLAC

For Laboratories, risk tolerance for “generic” ICS is different than for Personnel Protection or Medical Technology (protect lives, information, asset, etc.).

Somebody will have to identify boundaries and interfaces.

In highly regulated environments, once a standard is chosen and committed, the organization is auditable against it –it’s a big deal.

You might not always have the luxury of a Cyber Security team (**also knowledgeable about ICS**) at your service.

# FPGA Cyber Attack Modes

SLAC

1. **Black box attack**: feeding all possible input combinations to the FPGA chip and registering the corresponding output states. Provides the potential to reverse-engineering the FPGA electronic design integrated into the chip.

2. **Read-back attack**: Reading the chip configuration via the JTAG interface used in most FPGAs for debugging and maintenance. Recently, FPGA vendors have improved the protection measures against unauthorized access to chip configuration.

3. **Cloning attack**: in SRAM FPGA chips, a configuration file is stored in a non-volatile memory external to the FPGA chip. This may allow the retrieval of bit-streams while loading the configuration in the FPGA, and later to clone the stolen FPGA electronic design. The protection against this threat is encrypting bit-streams during their transmission from a non-volatile memory to the FPGA chip. Measures have been already implemented in most modern FPGAs to prevent this possibility.

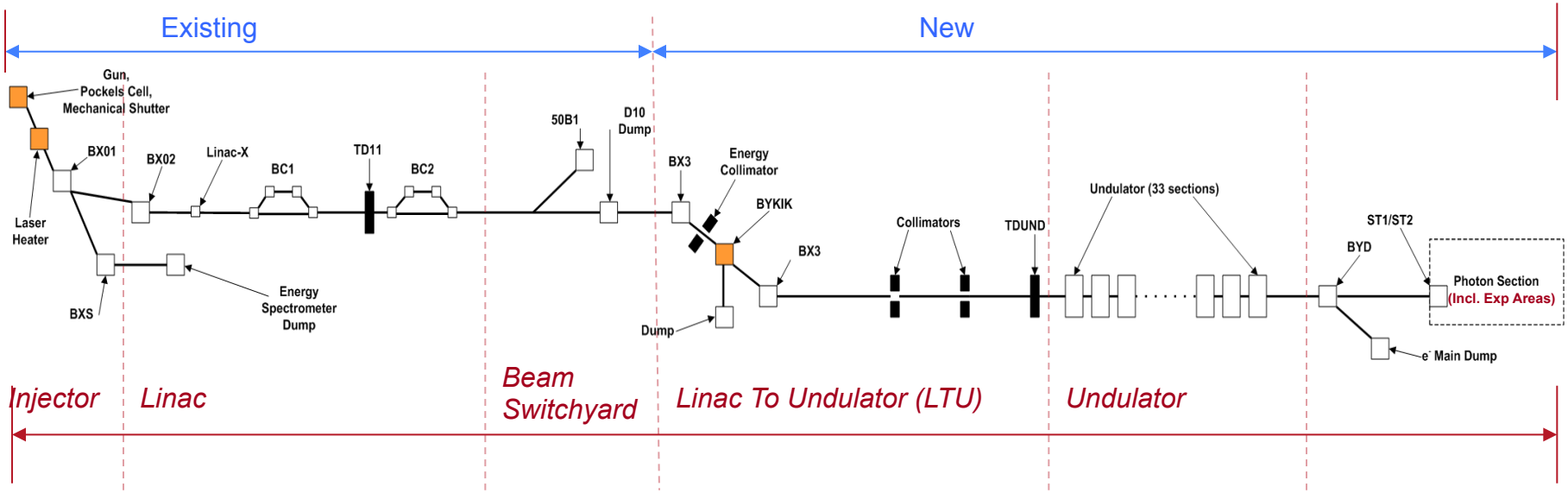
# FPGA Advantages

SLAC

The FPGA technology has certain **beneficial properties** for assuring cyber security:

1. Does not rely on a complex OS and therefore does not have dormant, unused functionalities that can be attacked. The FPGA chip just works deterministically through the calculations that it was programmed for in the application development process.
2. No known viruses and malware for VHDL code;
3. FPGA-based devices have a simple and structured design: their V&V processes will more likely detect the presence of potential malicious designs;
4. Physical access to the FPGA chips is strictly controlled: code is located in a flash memory (on a separate chip) without offering any physical access for modification while in on-line operating mode.
5. Programming and re-programming can only be done through a specific interface.

# An example: LCLS



# Photon Beam Parameters

SLAC

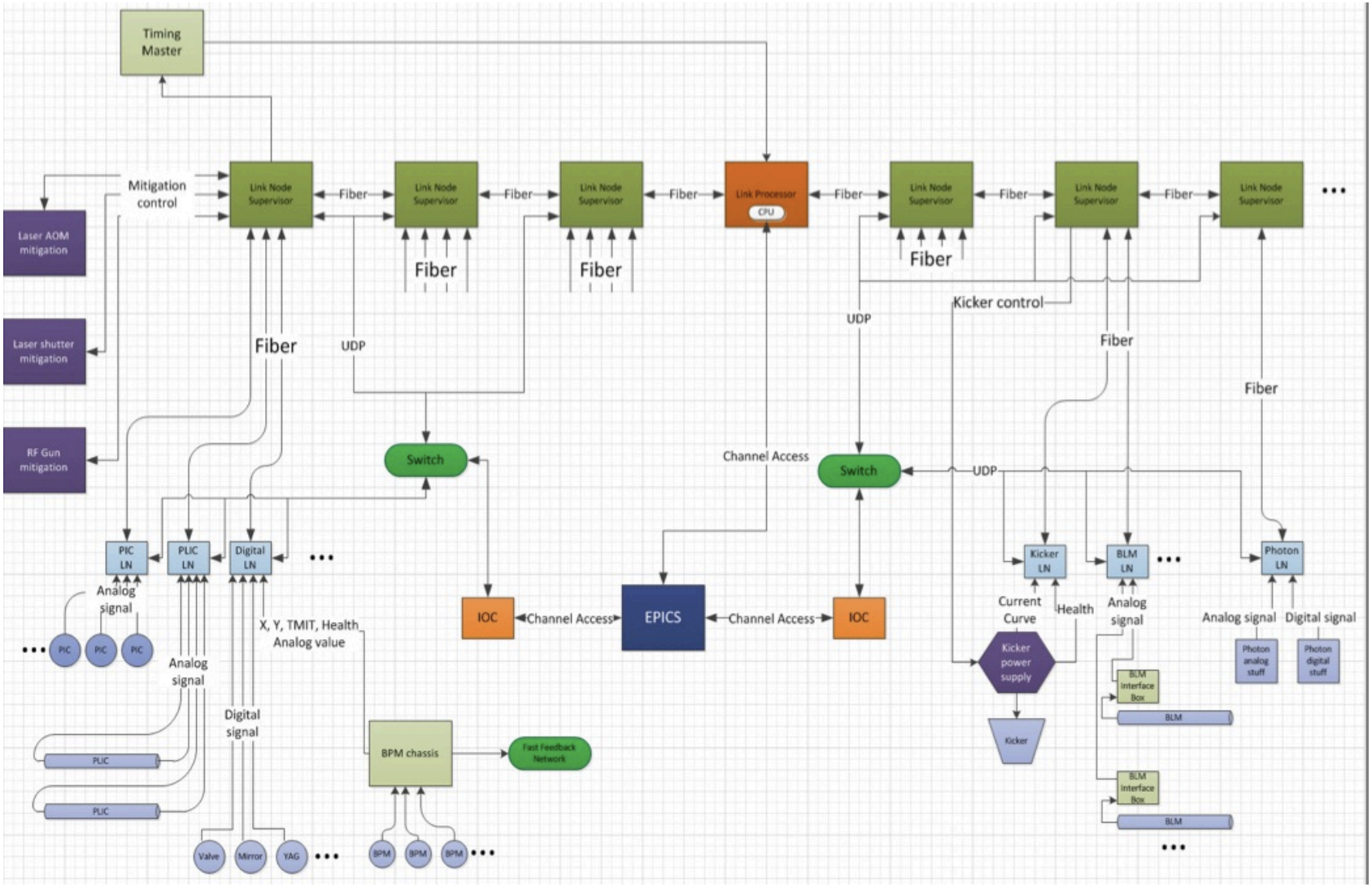
| Photon Beam Parameters | Symbol | hard x-rays | soft x-rays | unit |
|---|---|---|---|---|
| Fundamental wavelength | $\lambda_r$ | 6.2-1.3 | 43.7-6.2 | Å |
| Photon energy | $\hbar\omega$ | 2000-9600 | 285-2000 | eV |
| Final linac $e^-$ energy | $\gamma mc^2$ | 6.7-14.7 | 2.5-6.7 | GeV |
| FEL 3-D gain length | $L_G$ | 3.3 | 1.5 | m |
| Photons per pulse | $N_\gamma$ | 2 | 20 | $10^{12}$ |
| Peak brightness | $B_{pk}$ | 20 | 0.3 | $10^{32}$ § |
| Average brightness (120 Hz) | $\langle B \rangle$ | 160 | 8 | $10^{20}$ § |
| SASE bandwidth (fwhm) | $\Delta\omega/\omega$ | ~0.2-0.5 | ~0.2-1.0 | % |
| Final pulse duration (fwhm) | $\Delta\tau_f$ | 50-250 | 70-400 | fs |

# Electron Beam Parameters

SLAC

| Electron Beam Parameters | | | | |
|---|---|---|---|---|
| Bunch charge | $Q$ | 0.15 | 0.25 | nC |
| Init. bunch length (rms) | $\sigma_{z0}$ | 0.65 | 0.65 | mm |
| Final bunch length (rms) | $\sigma_{zf}$ | 7 | 20 | $\mu$m |
| Final peak current | $I_{pk}$ | 3.0 | 1.0 | kA |
| *Proj.* emittance (injector) | $\gamma\varepsilon_{x,y}$ | 0.4-0.6 | 0.4-0.6 | $\mu$m |
| *Slice* emittance (injector) | $\gamma\varepsilon^{s}_{x,y}$ | 0.4 | 0.4 | $\mu$m |
| *Proj.* emittance (undulator) | $\gamma\varepsilon^{U}_{x,y}$ | 0.5-1.6 | 0.5-1.6 | $\mu$m |
| Single bunch rep. rate | $F$ | 120 | 120 | Hz |
| UV laser energy on cath. | $u_l$ | 25 | 25 | $\mu$J |
| UV laser diam. on cath. | $2R$ | 1.2 | 1.2 | mm |
| $e^-$ energy stability (rms) | $\Delta E/E$ | 0.04 | 0.07 | % |
| $e^-$ $x,y$ stability (rms) | $x/\sigma_x$ | 15, 10 | 25, 20 | % |
| $e^-$ timing stability (rms) | $\Delta t$ | 50 | ? | fs |
| Peak current stab. (rms) | $\Delta I/I$ | 10 | 6 | % |
| Charge stability (rms) | $\Delta Q/Q$ | 2.5 | 2.5 | % |
| FEL pulse energy stability | $\Delta N/N$ | <10 | <10 | % |

# Schematic View



# MPS Requirements

SLAC

- Turn off or limit rate of electron beam when faults are detected to prevent damage to sensitive machine components;
- Protect Undulator permanent magnets from electron beam;
- Shut off beam (detect and mitigate) within one pulse at 120 Hz (i.e., 8.33 ms for LCLS-I, 90 μs for LCLS-II).

# Additional Requirements

SLAC

- Protection of laser heater system from injector laser;
- Protection of photon section from FEL X-rays;
- Allow fault conditions to set different maximum rates for each mitigation device;
- Automatic beam rate recovery (after a fault is corrected, beam rate raised to before fault value);
- Securely bypass faults.

# MPS Inputs

- Inputs from obstructions
  - Vacuum valves
  - Screens (OTRs, YAGs)
  - Beam stoppers
- And beam loss monitors
  - Toroids
  - Protection Ion Chambers (PICs)
  - Beam Loss Monitors (BLMs)

They generate the primary MPS beam loss signals

- Watchdog
- Currently monitoring ~2100 inputs

# Sensors

- Vacuum Valve Position
- Waterflow Status
- Magnet Power Supply Status
- Temperatures
- In-beam Diagnostics Status
- Beam Position
- Beam Charge
- RF System Status
- Beam Containment Status
- Beam Loss Monitors

# Mitigation Devices

SLAC

- Laser Heater Mechanical Shutter
- Photocathode Laser Mechanical Shutter
- Gun Trigger Permit (was Pockels cells)
- Pre-Undulator Fast Kicker (BYKIK)

# What it is and how it works

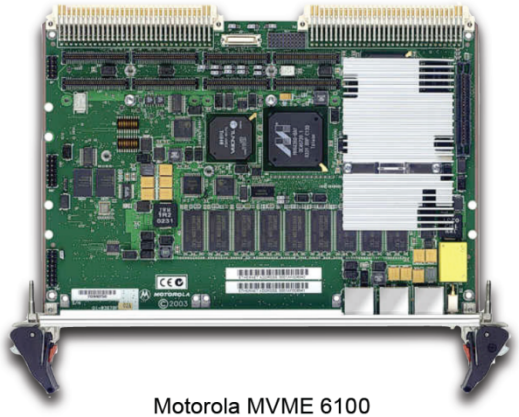
SLAC

A star network consisting of two entities: Link Processor and Link-Nodes (interconnected over private GigbE network).

The MPS determines the maximum allowed beam rate by processing device fault input signals (from **Link Nodes**, input multiplexers) with a rate limiting algorithm (executed on **the Link Processor**).

# Components

- Link Processor:
  - Runs MPS algorithm
  - Makes decisions based on sensor states
  - Interfaces to timing system

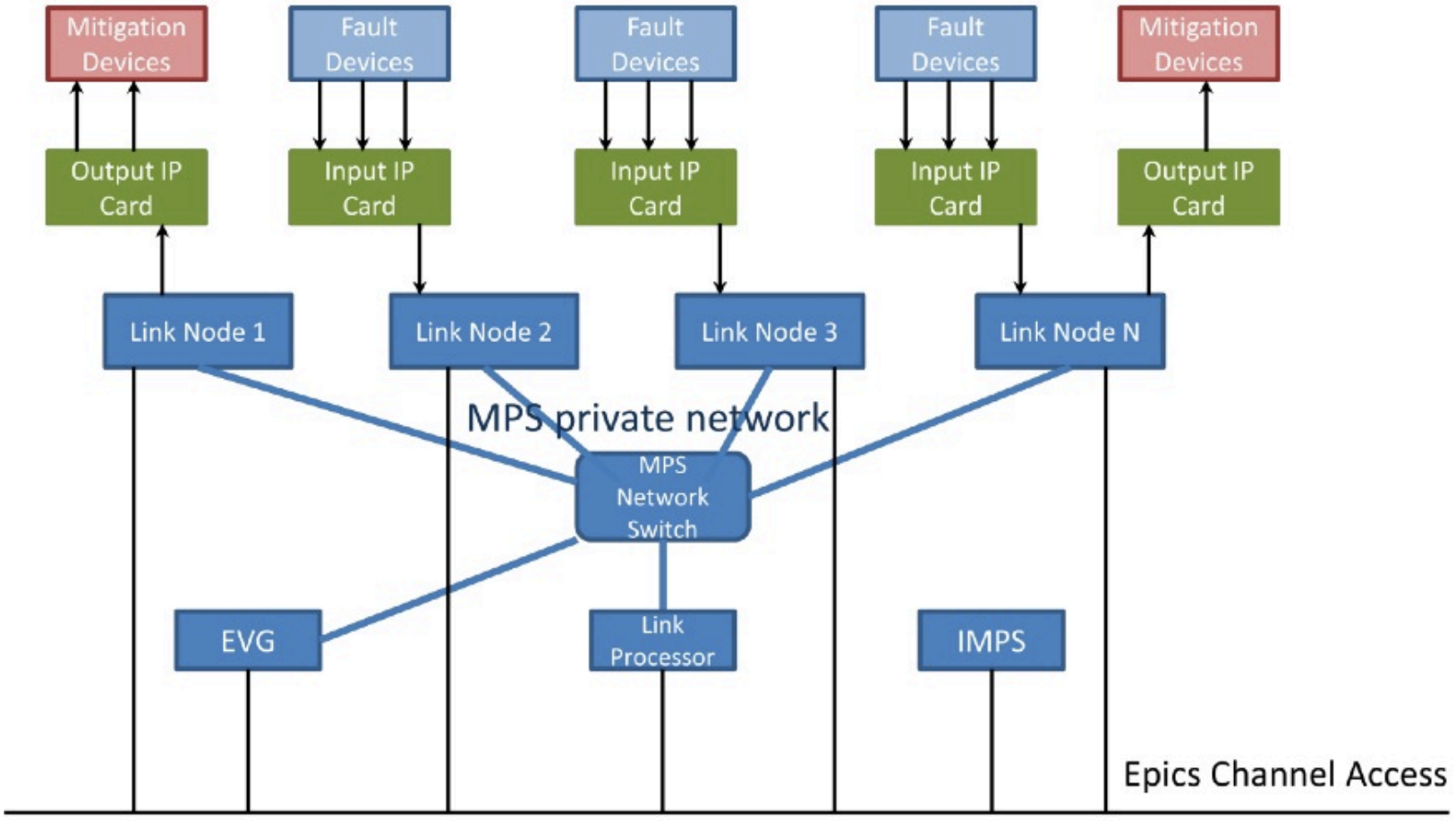


Motorola MVME 6100

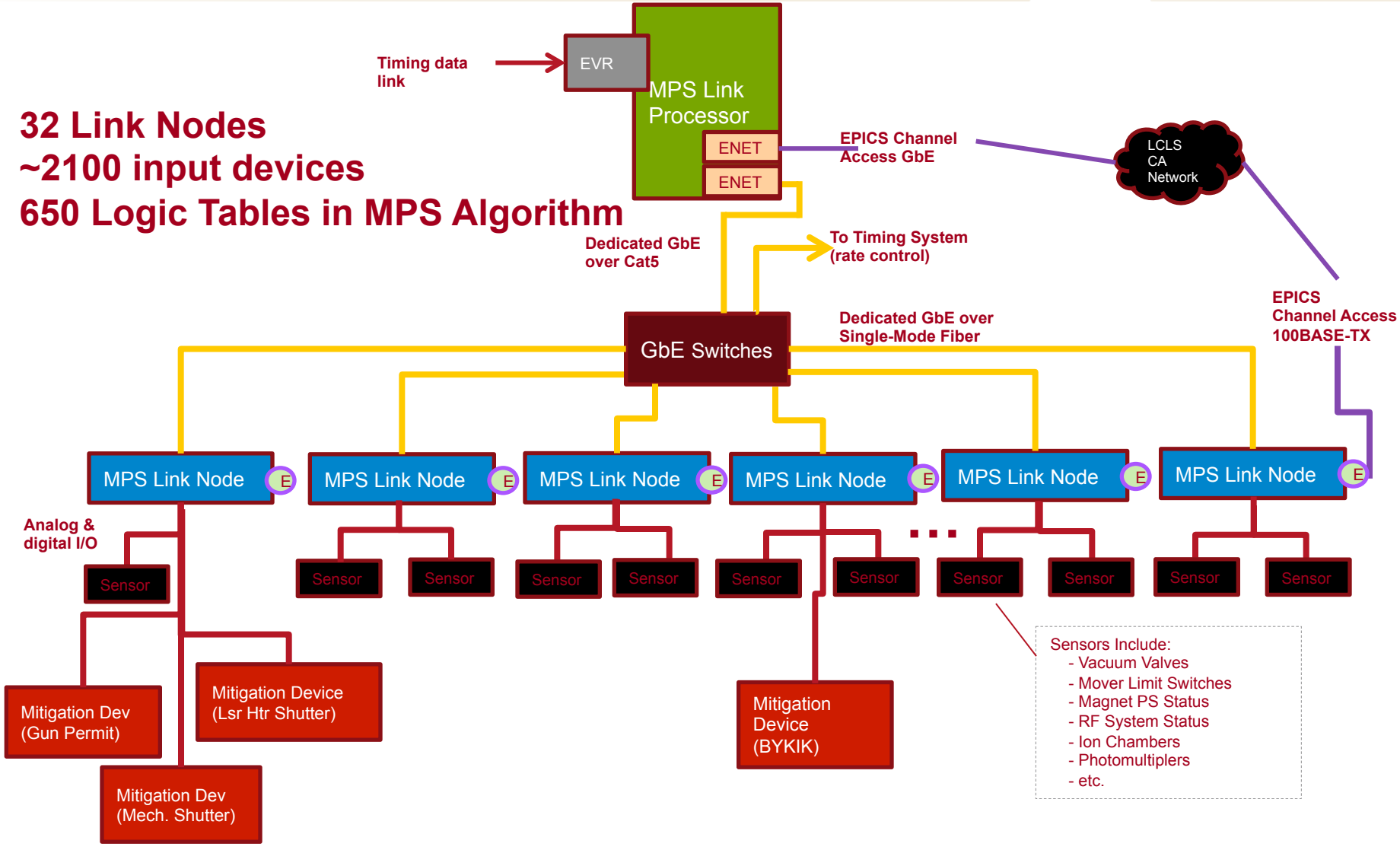
PMC-EVR-200

- Link-Node:
  - Sensor signal collection point
  - Drives mitigation devices
  - Integrates sensor subsystems

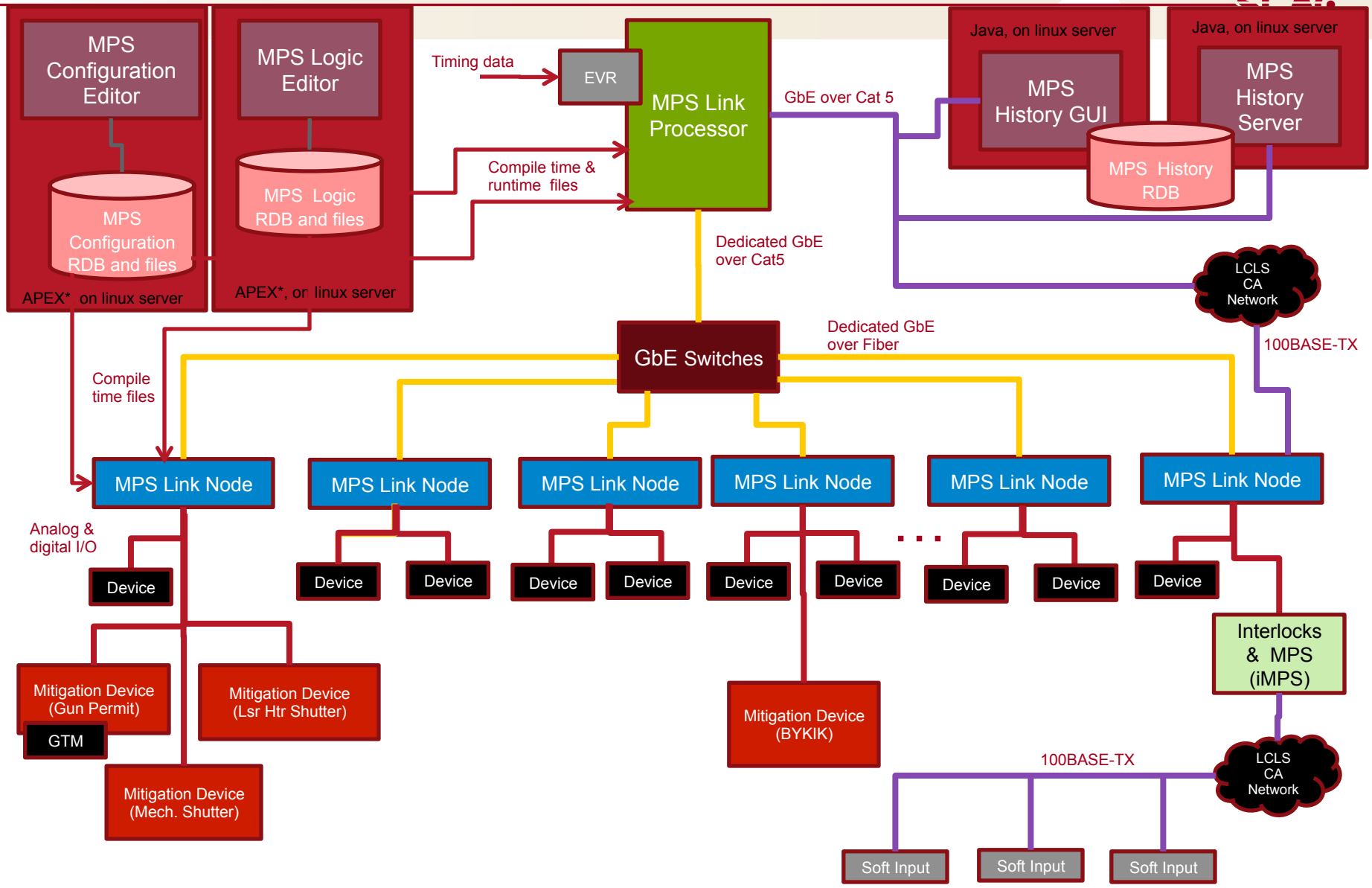
# Overview


Mitigation Devices
Fault Devices
Fault Devices
Fault Devices
Mitigation Devices
Output IP Card
Input IP Card
Input IP Card
Input IP Card
Output IP Card
Link Node 1
Link Node 2
Link Node 3
Link Node N
MPS private network
MPS Network Switch
EVG
Link Processor
IMPS
Epics Channel Access

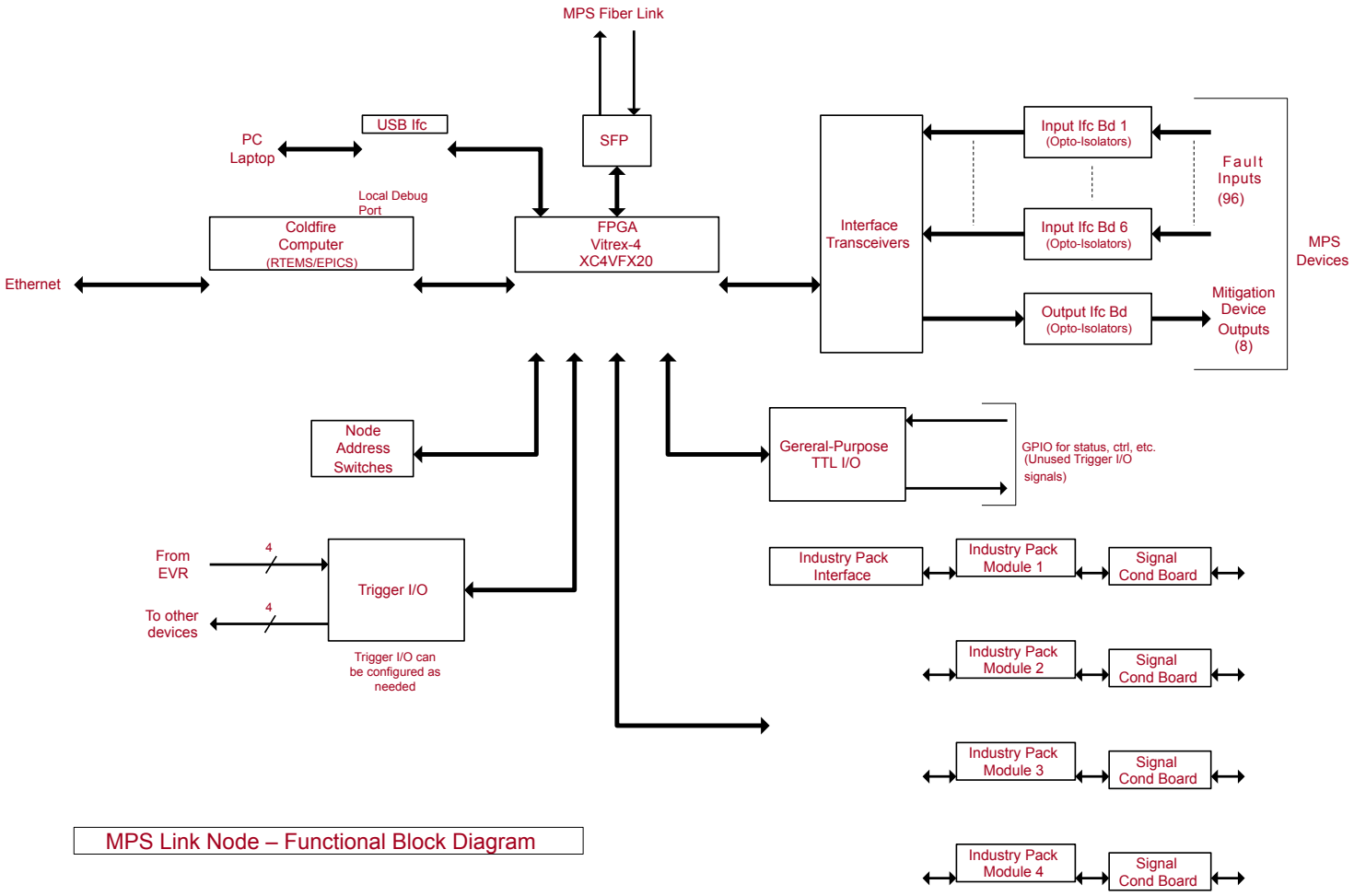
# Architecture



# Full Architecture



# Link Node Functional Block Diagram




MPS Link Node – Functional Block Diagram

# Link Node Architecture

SLAC

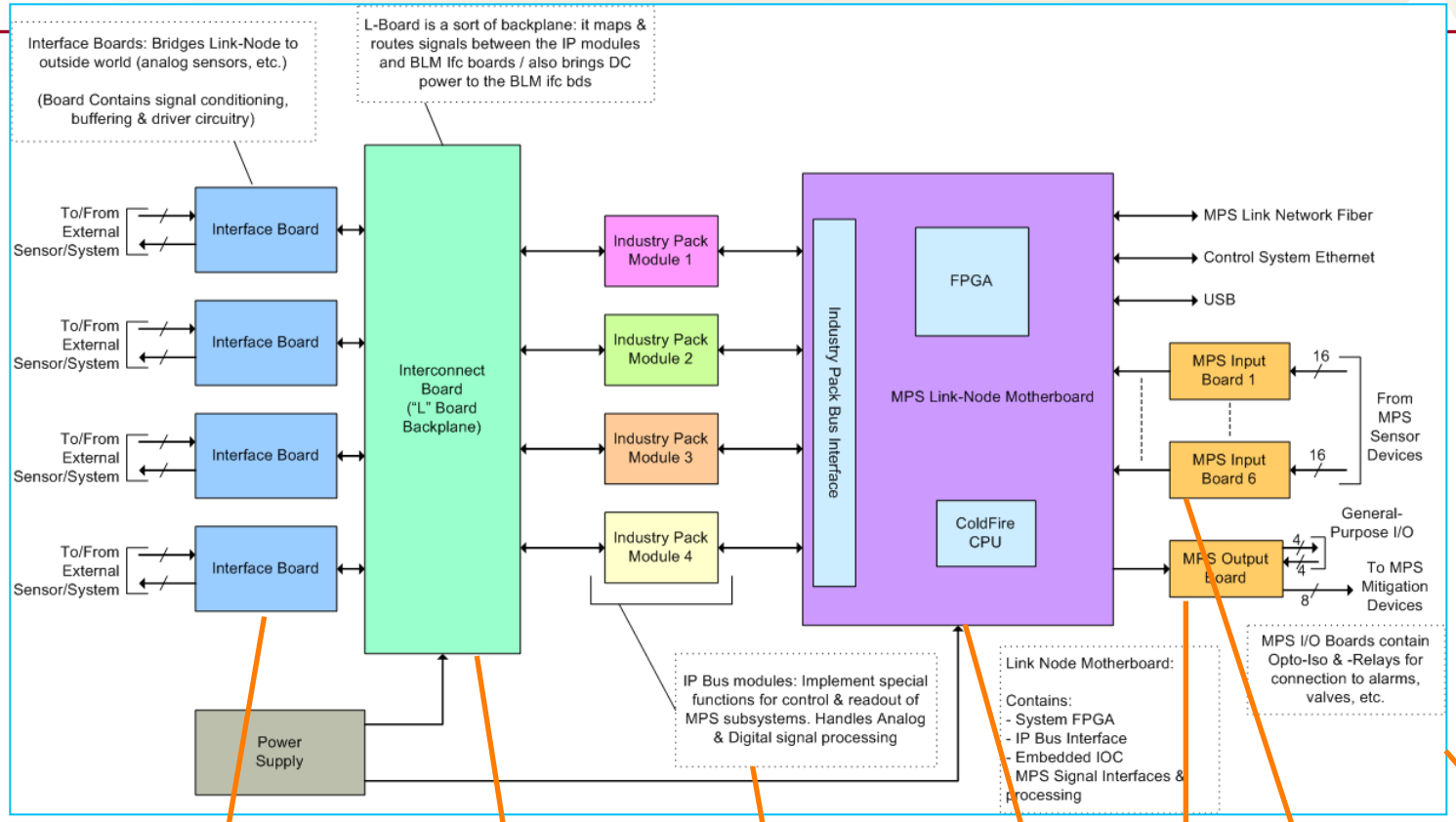
"Pizza Box" (NAD) with configurable board arrangement for configuration and control of loss monitor devices.

It contains:
- MPS "Engine" in Virtex-4 FPGA
- MPS Digital I/O
- Embedded Coldfire CPU
- Industry Pack (IP) bus interface
- GigE Interface
- USB Interface (dev & maintenance)

It is configured in different "flavors":
- Standard (MPS Digital I/O Only)
- BLM (Undulator Beam Loss Monitor)
- PIC (Beam Loss Ion Chamber)
- BYKIK (Fast Kicker Magnet)

- Interfaces to on Board FPGAs
- Configuration, Control of MPS loss monitor devices

# Link Node - A NAD

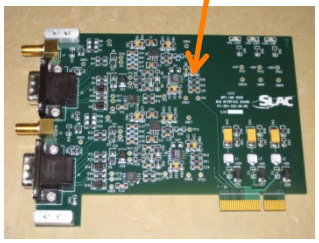


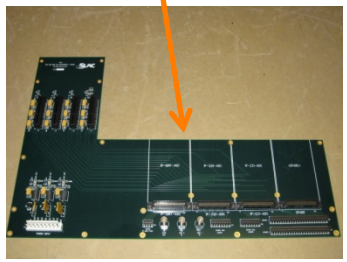

## Link-Node:

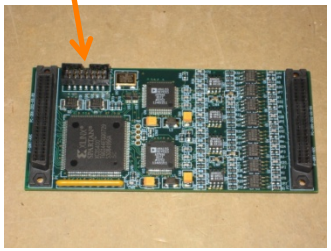
Config w/ different boards:

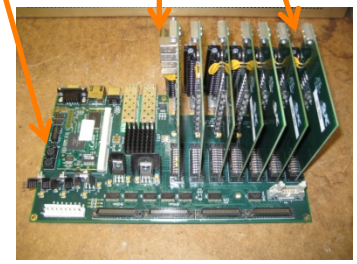
Standard:
- Motherboard
- MPS I/O Boards

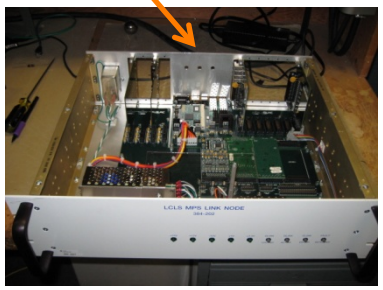
App Specific:
- IP Boards
- L-Board
- Interface Boards



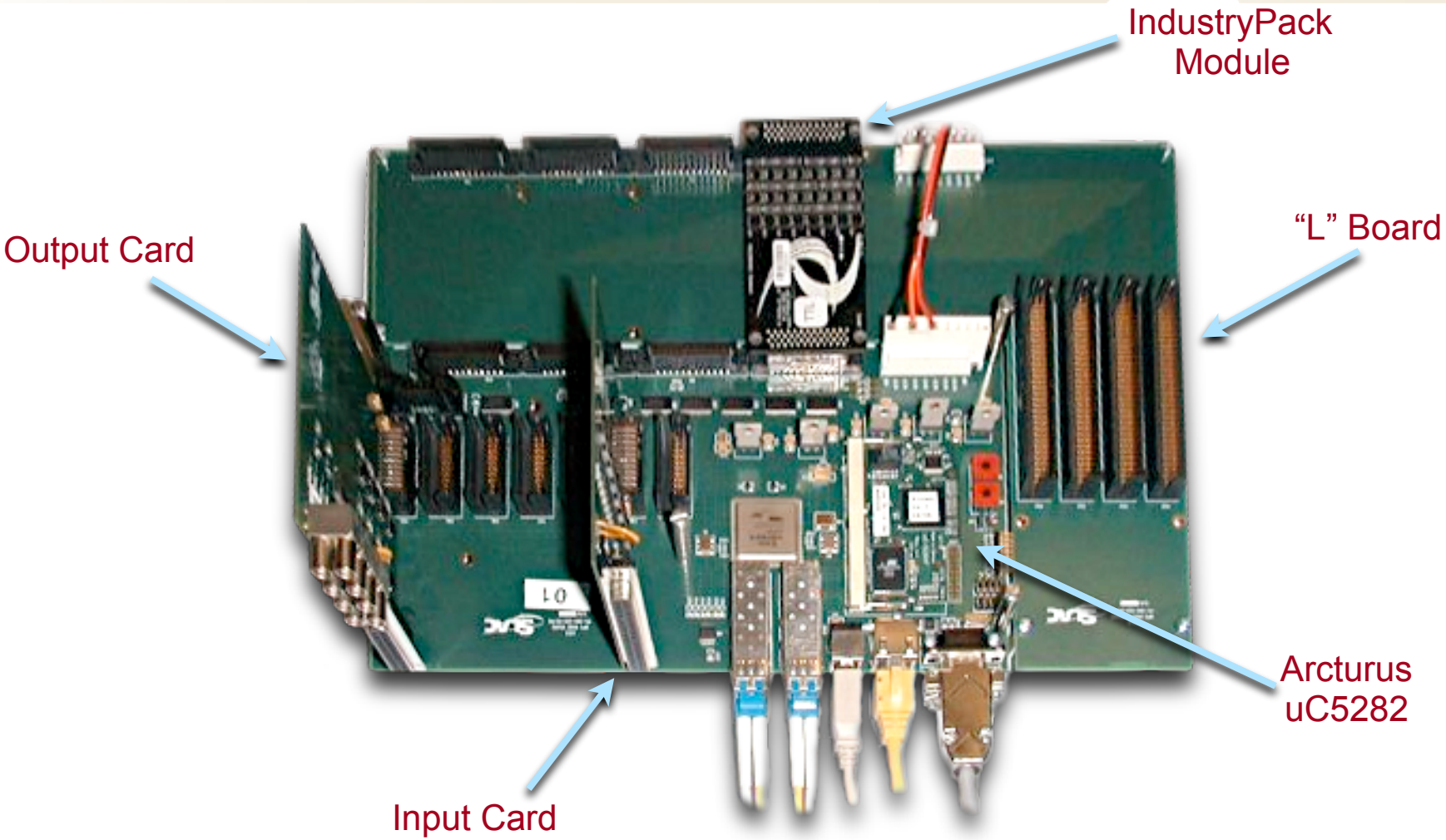








# Link Node – Cont'd



# MPS Link Processor

SLAC

- Gathers device status from Link Nodes.;
- Determines maximum allowed beam rate at each mitigation device;
- Broadcasts “Permit” message to Link Nodes;
- Handles fault bypassing, latching;
- Logs machine state changes with MPS history servers;
- Can load/unload MPS logic at run-time.

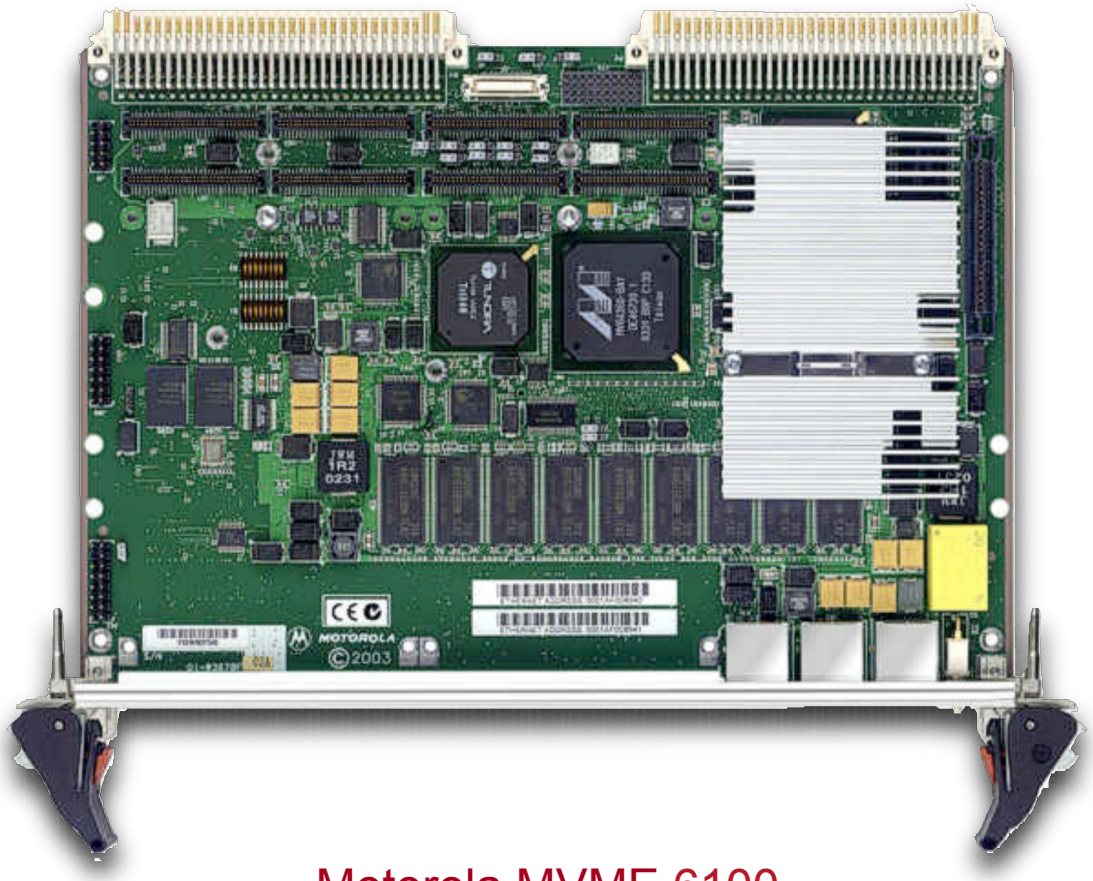
# MPS Link Processor – Cont'd

SLAC

- Motorola MVME 6100
  - VME64 single board computer
  - 1.267 GHz MPC7457 PowerPC
  - 2 GB DDR266 RAM
  - Two Gigabit Ethernet interfaces
  - Serial port for IOC console

- Micro-Research Finland PMC-EVR-200

# Link Processor Modules




Motorola MVME 6100

PMC-EVR-200

# Link Node Flavors

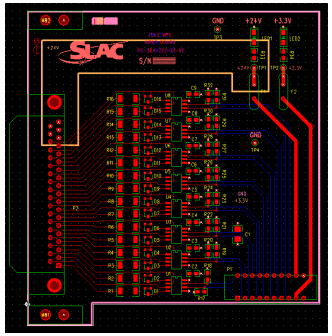
1. Mitigation Control (standard);
2. BYKIK;
3. BLM;
4. PIC.

# 1. Standard

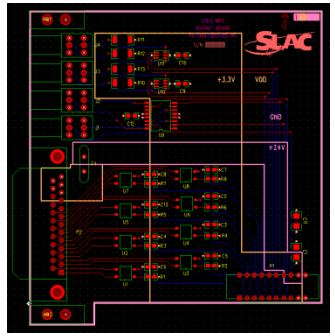
Standard (Mitigation)

- Digital I/O Only
- Up to 96 Inputs
- 8 Outputs

Input Board

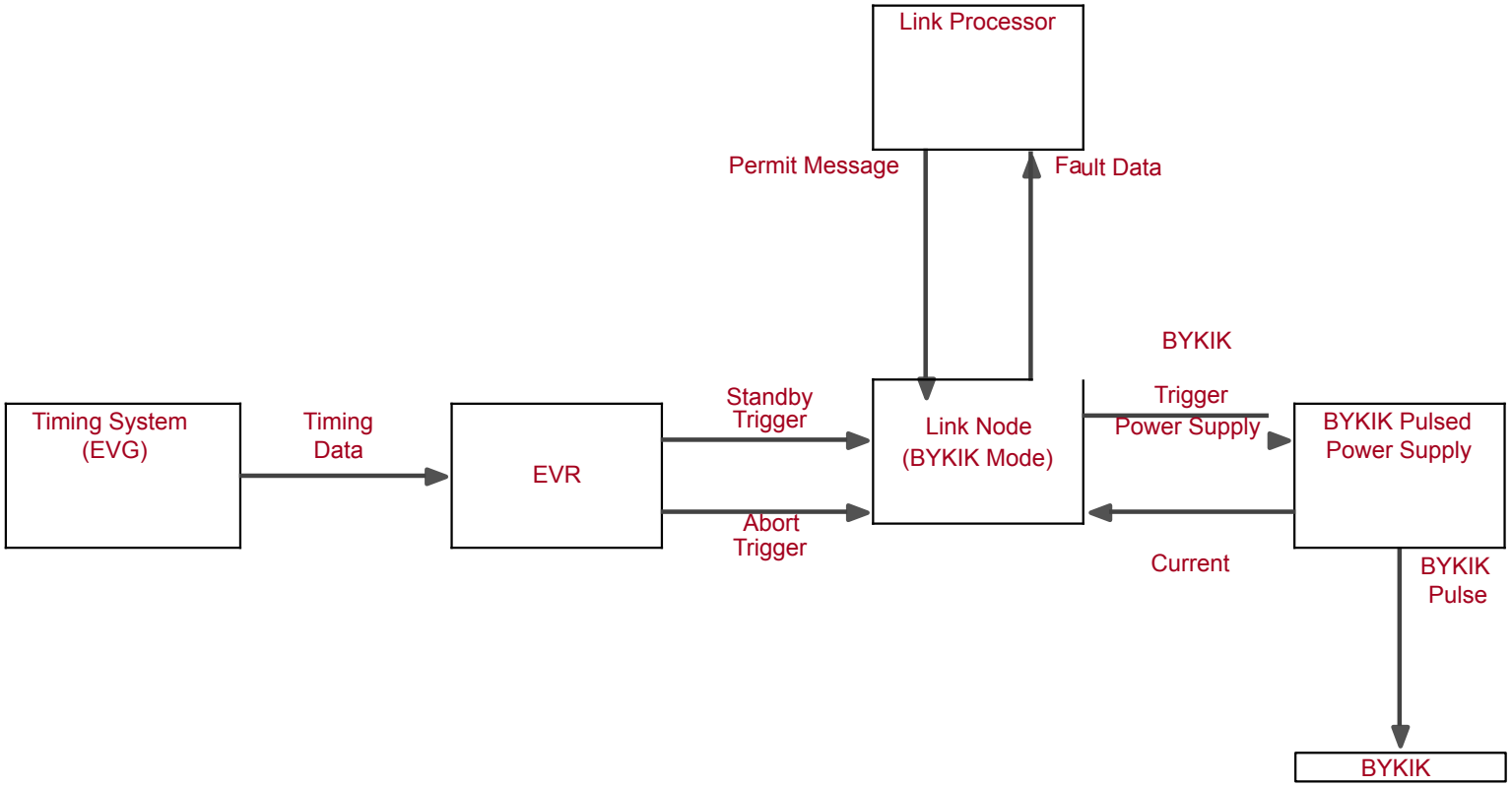


Output Board

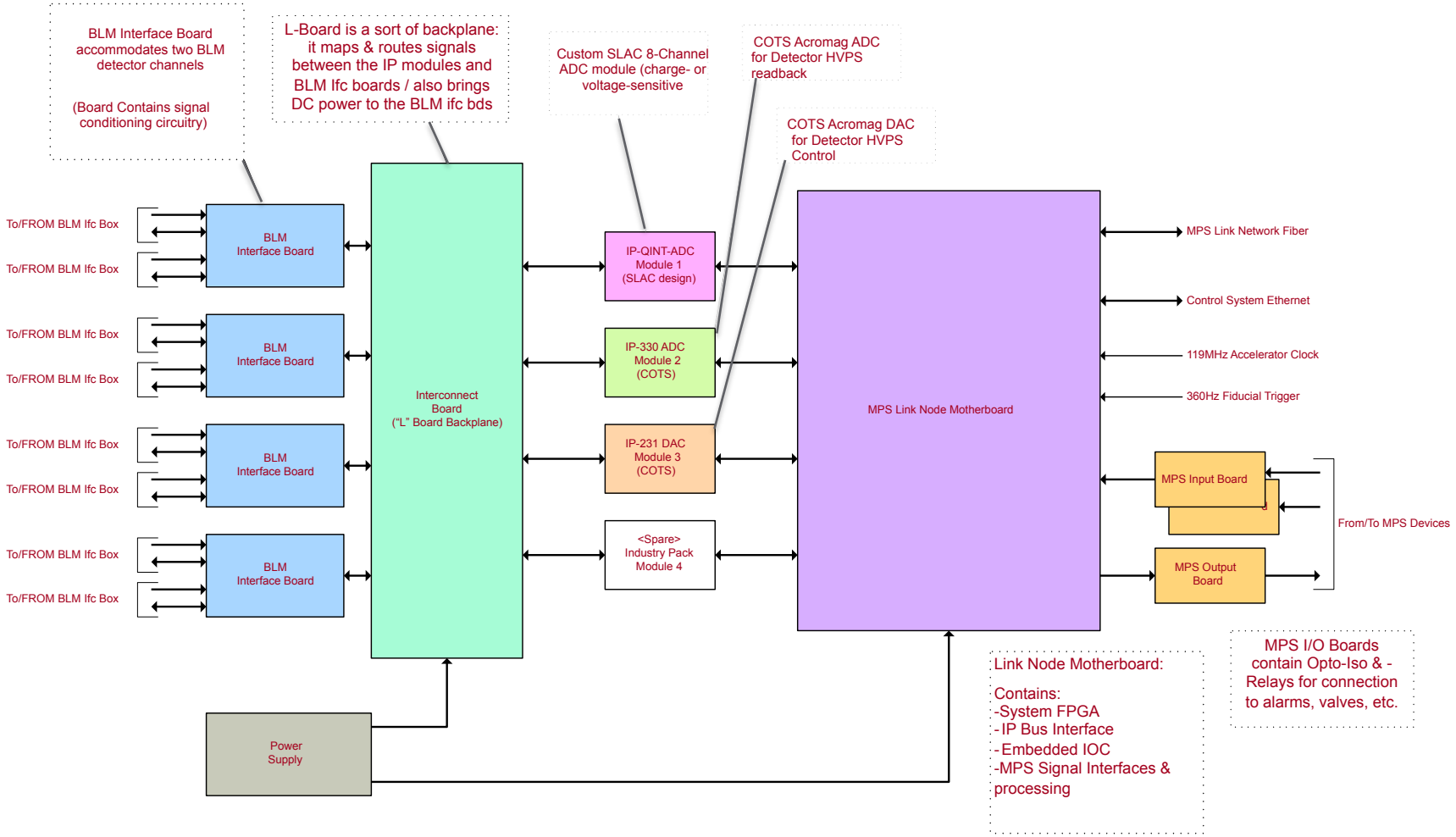


Temperature
Vacuum
Profile Monitors
Wire Scanners
Photon System

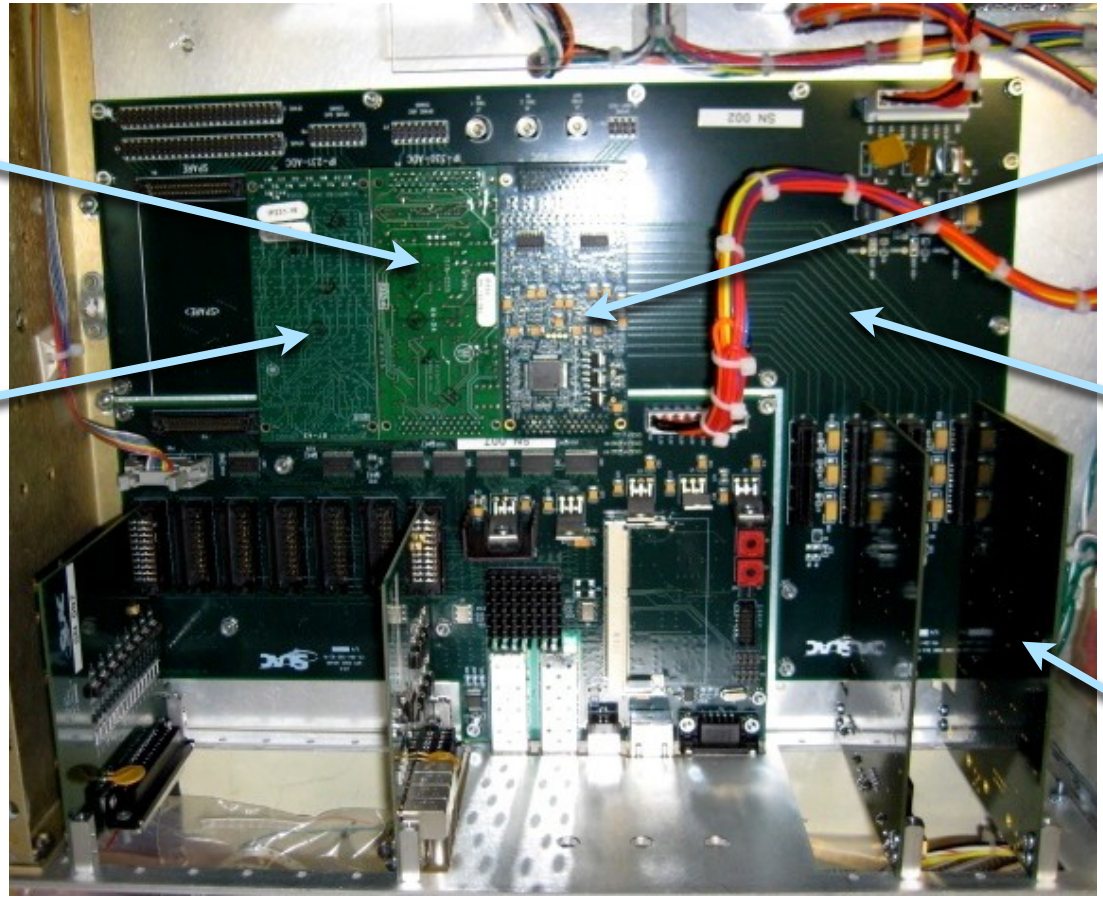
# 2. BYKIK


Link Processor
Permit Message
Fault Data
BYKIK
Timing System
(EVG)
Timing
Data
EVR
Standby
Trigger
Abort
Trigger
Link Node
(BYKIK Mode)
Trigger
Power Supply
BYKIK Pulsed
Power Supply
Current
BYKIK
Pulse
BYKIK

# 3. BLM


BLM Interface Board accommodates two BLM detector channels
(Board Contains signal conditioning circuitry)
L-Board is a sort of backplane: it maps & routes signals between the IP modules and BLM Ifc boards / also brings DC power to the BLM ifc bds
Custom SLAC 8-Channel ADC module (charge- or voltage-sensitive
COTS Acromag ADC for Detector HVPS readback
COTS Acromag DAC for Detector HVPS Control
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
To/FROM BLM Ifc Box
BLM Interface Board
BLM Interface Board
BLM Interface Board
BLM Interface Board
Interconnect Board ("L" Board Backplane)
IP-QINT-ADC Module 1 (SLAC design)
IP-330 ADC Module 2 (COTS)
IP-231 DAC Module 3 (COTS)
<Spare> Industry Pack Module 4
MPS Link Node Motherboard
MPS Link Network Fiber
Control System Ethernet
119MHz Accelerator Clock
360Hz Fiducial Trigger
MPS Input Board
MPS Output Board
From/To MPS Devices
Power Supply
Link Node Motherboard:
Contains:
-System FPGA
- IP Bus Interface
- Embedded IOC
-MPS Signal Interfaces & processing
MPS I/O Boards contain Opto-Iso & - Relays for connection to alarms, valves, etc.

# 3. BLM – Cont'd



# Thresholds

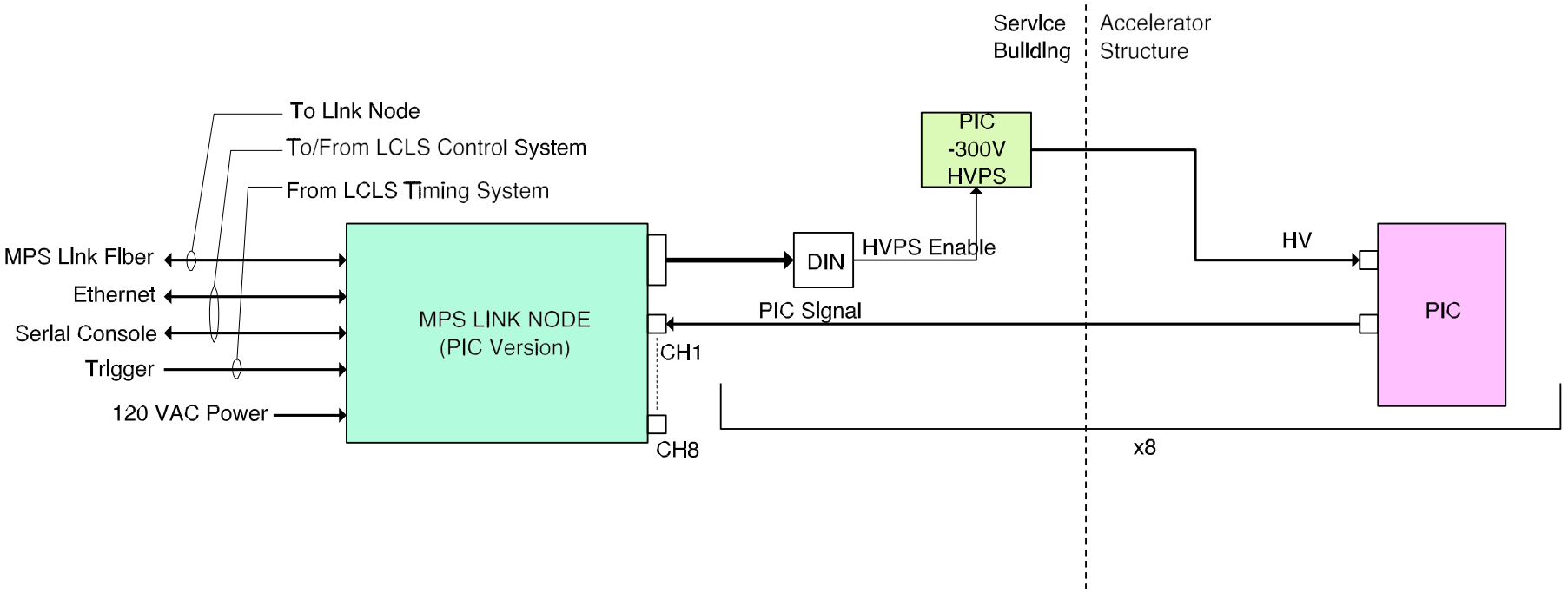
SLAC

Loss monitors with different types of sensitivity are deployed in different locations.
The signal is gated to coincide with the beam arrival time and compared to a programmable threshold and will indicate a fault if exceeded.

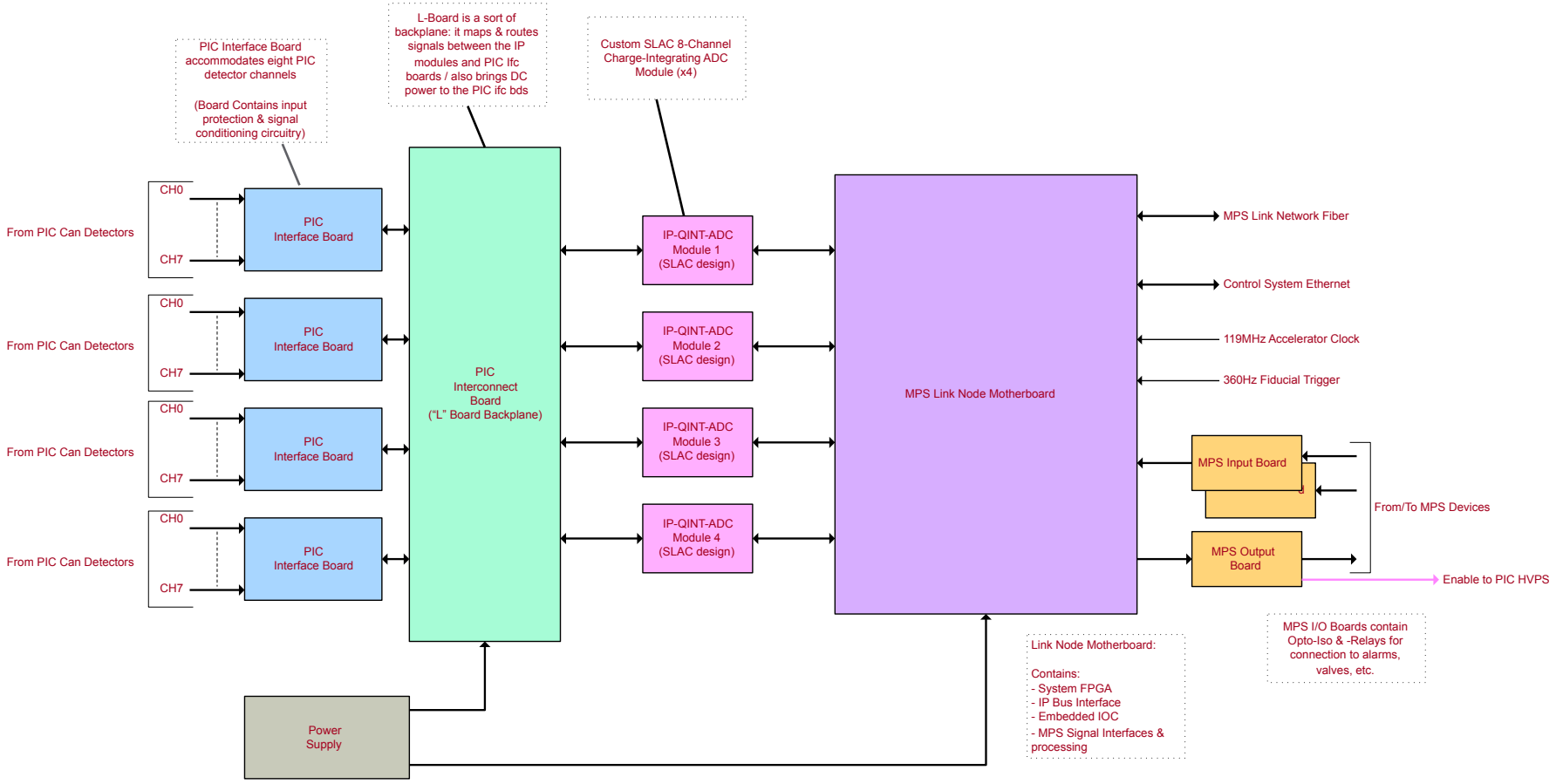
Two programmable threshold settings are required:

1. Exceeding the first threshold can allow the beam rate to be lowered by the MPS;
2. Exceeding the higher threshold will cause the beam to be shut off and require the MPS to be manually reset.

# 4. PIC


Service Building
Accelerator Structure
To Link Node
To/From LCLS Control System
From LCLS Timing System
MPS Link Fiber
Ethernet
Serial Console
Trigger
120 VAC Power
MPS LINK NODE
(PIC Version)
CH1
CH8
DIN
HVPS Enable
PIC
-300V
HVPS
HV
PIC Signal
PIC
x8

# 4. PIC – Cont'd

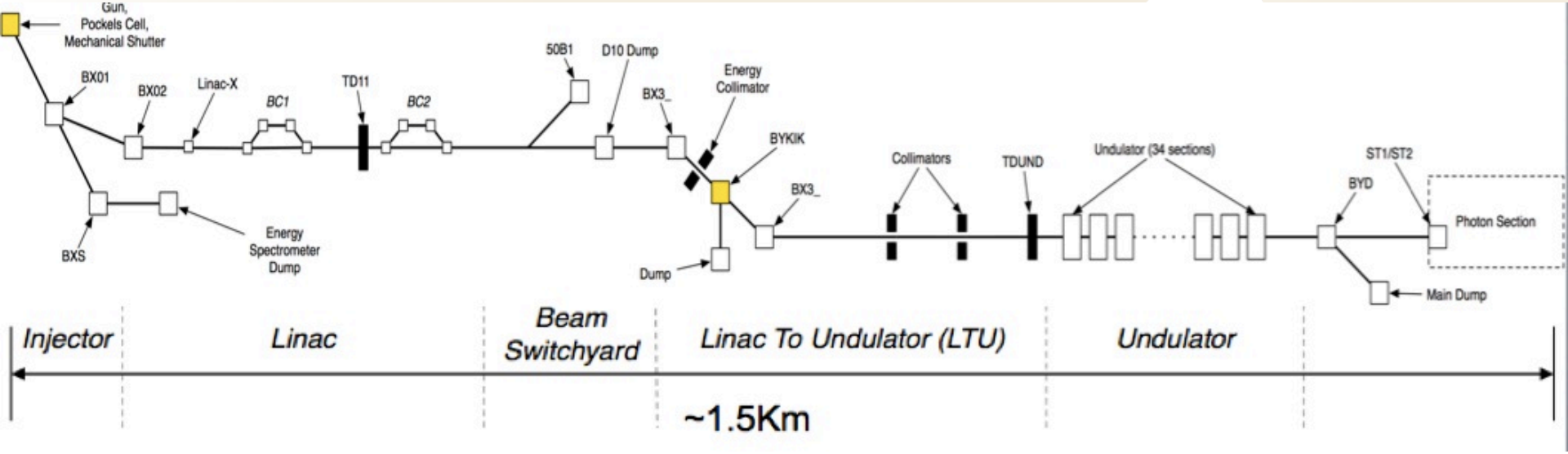

PIC Interface Board accommodates eight PIC detector channels
(Board Contains input protection & signal conditioning circuitry)
L-Board is a sort of backplane: it maps & routes signals between the IP modules and PIC Ifc boards / also brings DC power to the PIC ifc bds
Custom SLAC 8-Channel Charge-Integrating ADC Module (x4)
From PIC Can Detectors
CH0
CH7
PIC Interface Board
PIC Interconnect Board ("L" Board Backplane)
IP-QINT-ADC Module 1 (SLAC design)
IP-QINT-ADC Module 2 (SLAC design)
IP-QINT-ADC Module 3 (SLAC design)
IP-QINT-ADC Module 4 (SLAC design)
MPS Link Node Motherboard
MPS Link Network Fiber
Control System Ethernet
119MHz Accelerator Clock
360Hz Fiducial Trigger
MPS Input Board
MPS Output Board
From/To MPS Devices
Enable to PIC HVPS
Power Supply
Link Node Motherboard:
Contains:
- System FPGA
- IP Bus Interface
- Embedded IOC
- MPS Signal Interfaces & processing
MPS I/O Boards contain Opto-Iso & -Relays for connection to alarms, valves, etc.

# OUTPUTS – Mitigation Devices

SLAC

1. BYKIK Pulsed Magnet
2. Injector Mechanical Shutter
3. Laser Heater Shutter
4. Gun LLRF PAC

# 1. BYKIK




- AKA Single Bunch Beam Dumper (SBBD), can kick beam into the dump pulse-by-pulse;
- Positioned half way down machine;
- Can limit beam rate to any rate from 0 to 120 Hz while the Linac is kept at a constant rate;
- Allows the MPS to protect the Undulator region while beam tuning and configuring is performed upstream.

# 1. BYKIK – Cont'd

SLAC

If the amplitude does not fall within a specified range at trigger time, the BYKIK has failed and the MPS uses the Gun RF and injector mechanical shutter to continue rate limiting at the correct rate.

The MPS continues to trigger the BYKIK after a BYKIK fault. If the BYKIK begins to operate correctly, the MPS returns to using the BYKIK as the mitigation device.

# 2 – Injector Mechanical Shutter

SLAC

Pockels Cell (old): Beam rate changes would change laser profile, due to thermal effects. Replaced by:

- Injector Mechanical Shutter
  - Slower mechanical device;
  - It inhibits the Injector UV laser before it hits the gun's cathode;
  - Not a pulsed device: either full rate light or 0 Hz;
  - Control verified with optical position sensors.

# 3 – Laser Heater Shutter



It inhibits the IR Laser Heater Laser Light.

The Laser Heater Optical Transition Radiation (OTR) screens, which are used to align the IR laser and electron beams, can be damaged by normal operational intensities of the IR laser and require an attenuator to be inserted when observing the IR laser on the OTR screen.

# 4 – Gun LLRF PAC

SLAC

The RF Gun PAC is triggered at 120 Hz: MPS will inhibit it to abort the beam, triggering at standby time and allowing the RF system to stay in thermal equilibrium.

MPS will use the RF Gun to limit beam rate from 120 Hz to 10 Hz.

During a 0 Hz MPS rate limit both the Gun LLRF PAC and the injector mechanical shutter will be used to inhibit the electron beam.

# Communication – Link Processor

SLAC

- EPICS communication uses MVME 6100's GbE interface;
- MPS/Link Node communication uses second GbE interface and a custom real-time protocol stack (RTS).
- RTS provides deterministic communication.
- UDP/IP protocols allow familiar development, easy testing, standard hardware and software tools.

# Communication – Link Node

SLAC

- Arcturus uC5282's 100BASE-TX interface used for EPICS communication
  - Configuration of Link Node at startup
  - Device input thresholds and debounce times

- High-speed GbE MPS communication using FPGA's RocketIO/fiber SFP transceiver

# Communication Methods

SLAC

- Link Processor sends Synchronization and Permit messages at 360 Hz;
- Status messages returned from Link Nodes upon receipt of sync messages;
- Link Processor faults all Link Node inputs of Link Nodes that do not provide response within 8.3 ms;
- Link Nodes stop beam at mitigation devices after 3 ms without a permit message.

# Output to History Server



The Link Processor logs all fault and status messages to an MPS History server application, which stores the messages in an Oracle database in real-time.

MPS messages are stored separately from the normal logging system so that no messages are lost; they are also forwarded to the normal message logging system so that they can be correlated to other logged events.

An MPS history viewer is available to the operators via the MPS GUI.

# History Server

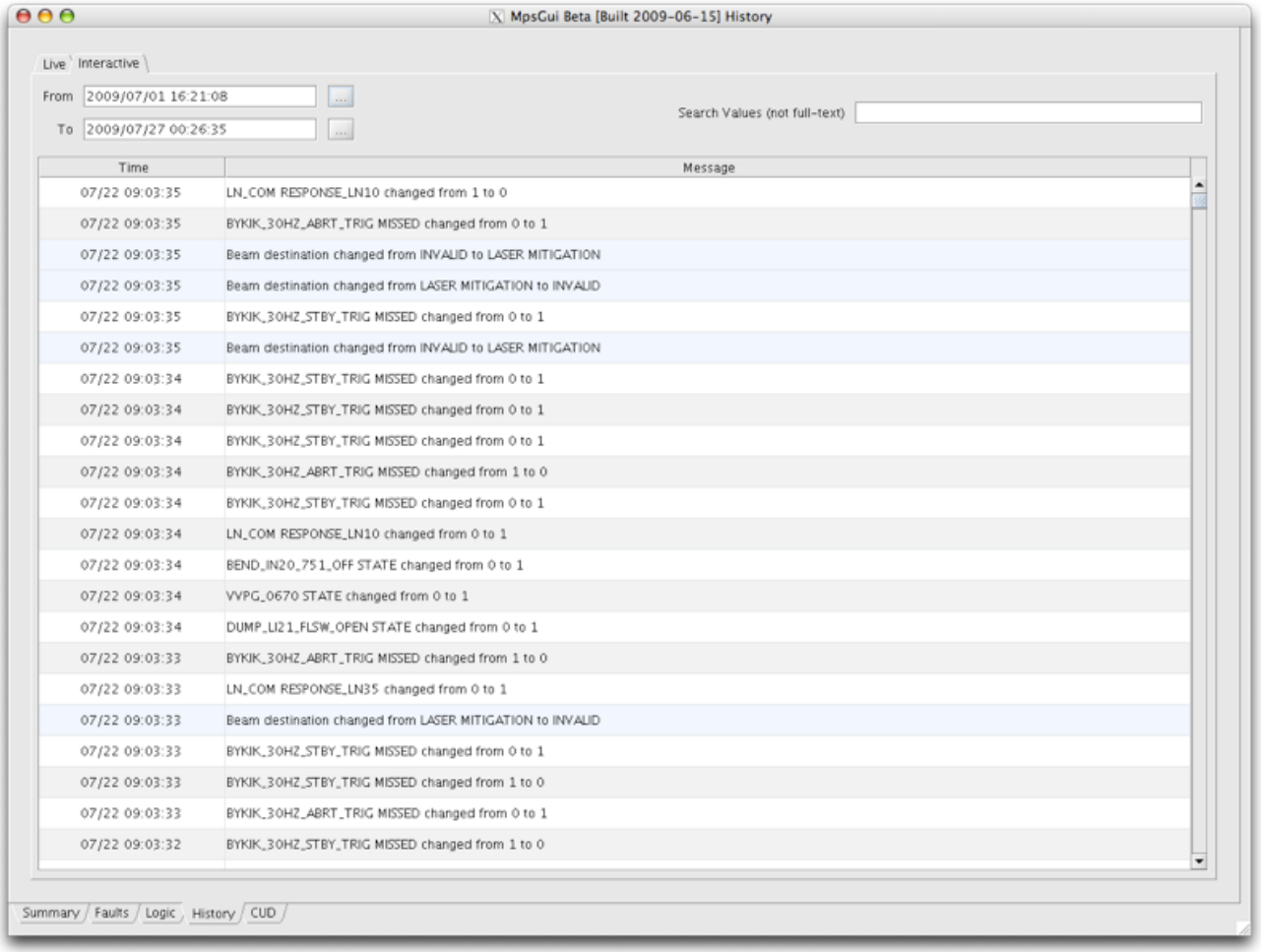
SLAC

- Events logged:
  - Device state changed;
  - Beam rate changed;
  - Destination changed.

- History Servers notify Link Processor the their existence;
- Link Processor connects to the first available History Server;
- Link Processor automatically connects to next server available server if current server fails to receive message.

# History GUI

*Date range*



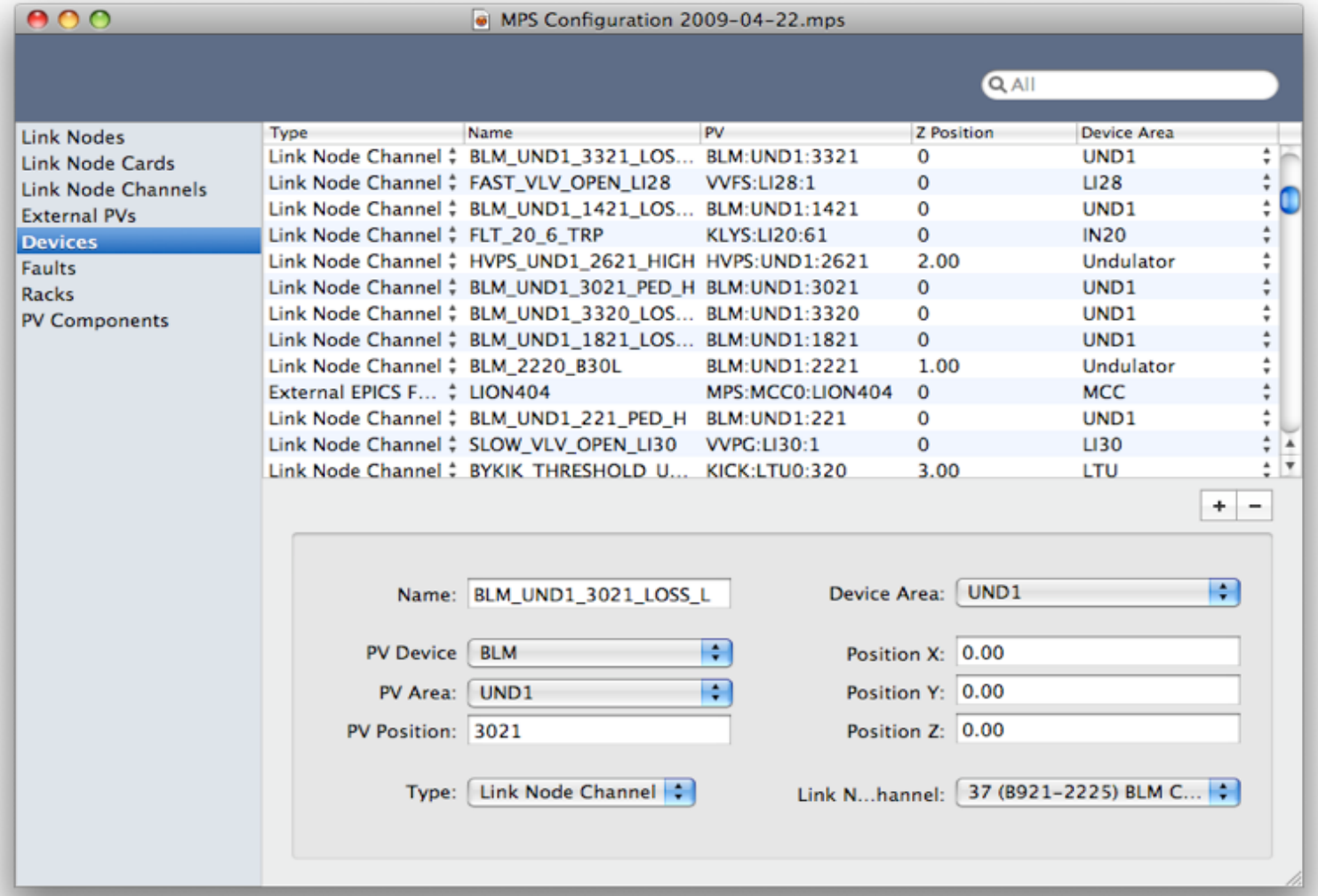

*Text filter*

*Filtered history messages*

# System Configuration

- MPS inputs configured using “MPS Database Editor.app”;
- MPS logic created with “MPS Logic.app”;
- Both applications store data as SQLite3 files;
- Data is portable and easy to access;
- Python modules export data in a variety of formats.

# Configuration Editor



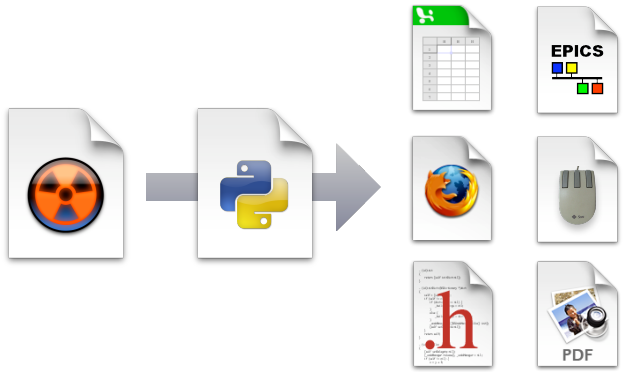
# Configuration Editor

SLAC

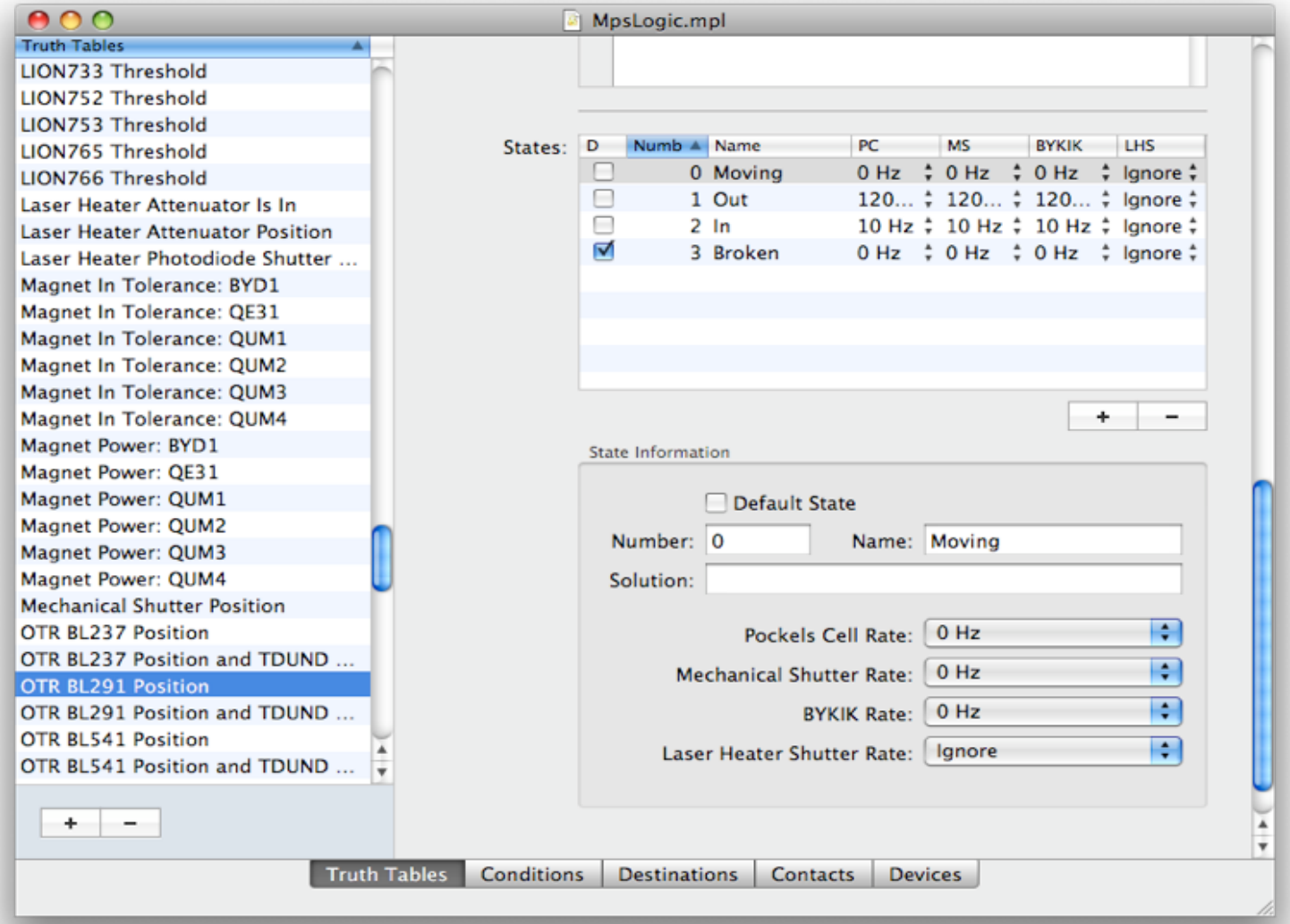
## Configuration file exported to .db, .edl, .stt, .h, .tex, and .csv using Python module

- **.csv**
  - LN and LP boot time configuration
- **.db**
  - LP EPICS records (five records per input)
- **mpsEpicsFaultInputs.stt**
  - LP state notation used at compile time – monitors EPICS fault inputs
- **MPSFaultNumbers.h**
  - LP and MPS Logic compile time configuration
- **.edl**
  - EDM displays for users
- **.tex, .pdf**
  - LaTeX documentation converted to PDF

- **mpsdb.sqlite3**
  - Copy of configuration file for MPS GUI
- **MPSDatabase.sql, pvlist.txt**
  - Dumped configuration file
  - List of records in .db files



# Logic Editor

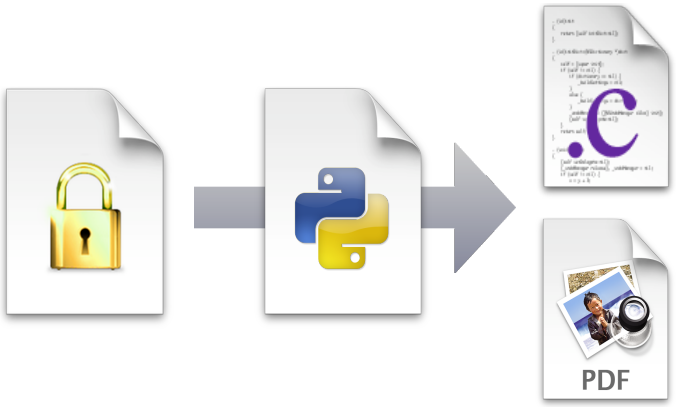


# Logic Editor – Cont'd

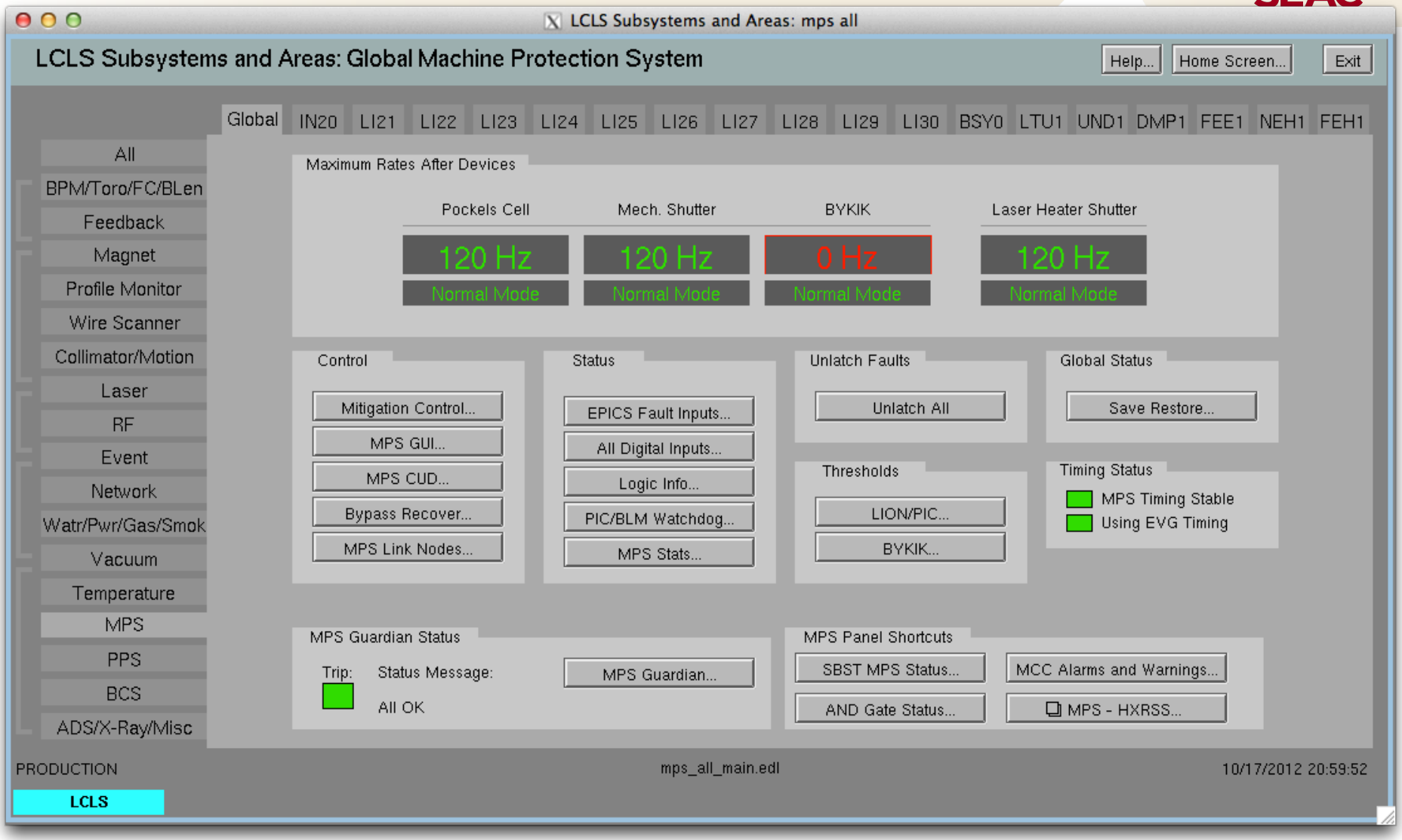
SLAC

***Logic file exported to .h, .tex, .txt, .pdf and .csv using Python module***

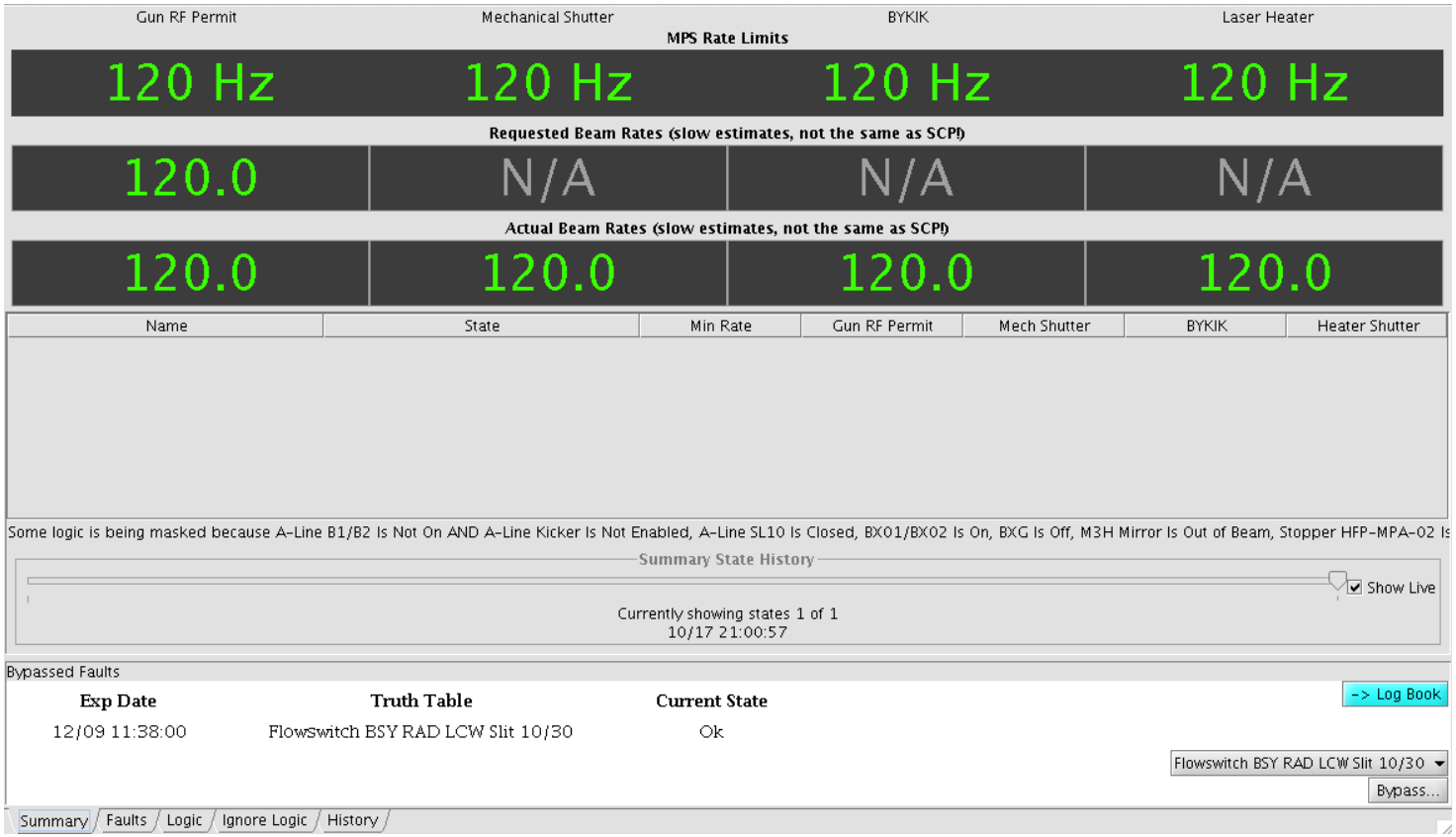
- **Algorithm.h**
  - Loadable and unloaded from LP at run time – Built against a specific MPS Input configuration
  - Logic Table Definitions
  - Can contain C specific logic for more complex logic
- ***.pdf, *.txt**
  - Logic documentation



# MPS Global Panel



# Main Fault UI




*Current rates*

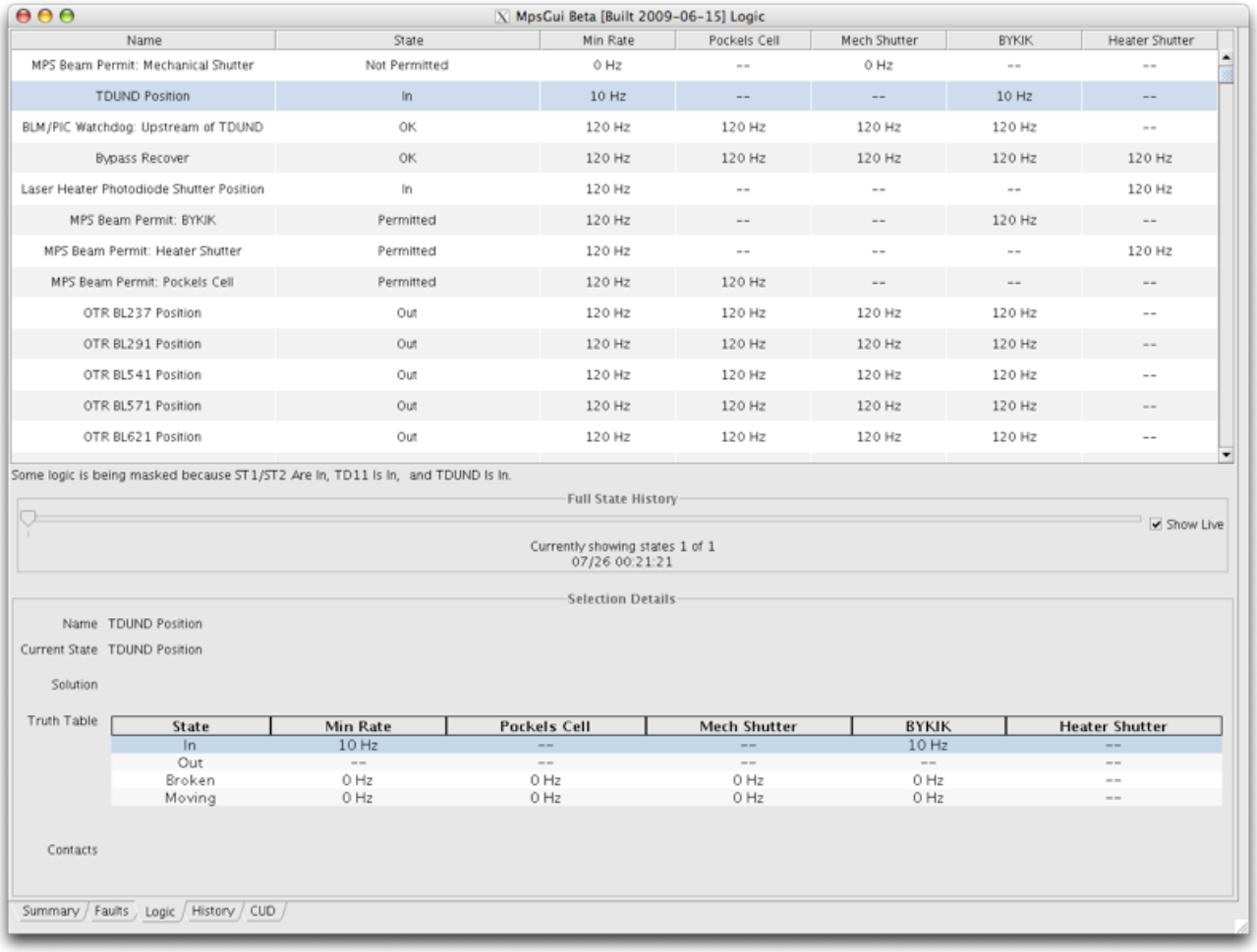
*Current rate limiting truth tables*

*History slider*

*Active bypasses*

- Java-based
- Main user Ifc in control room

# User Interface - Logic

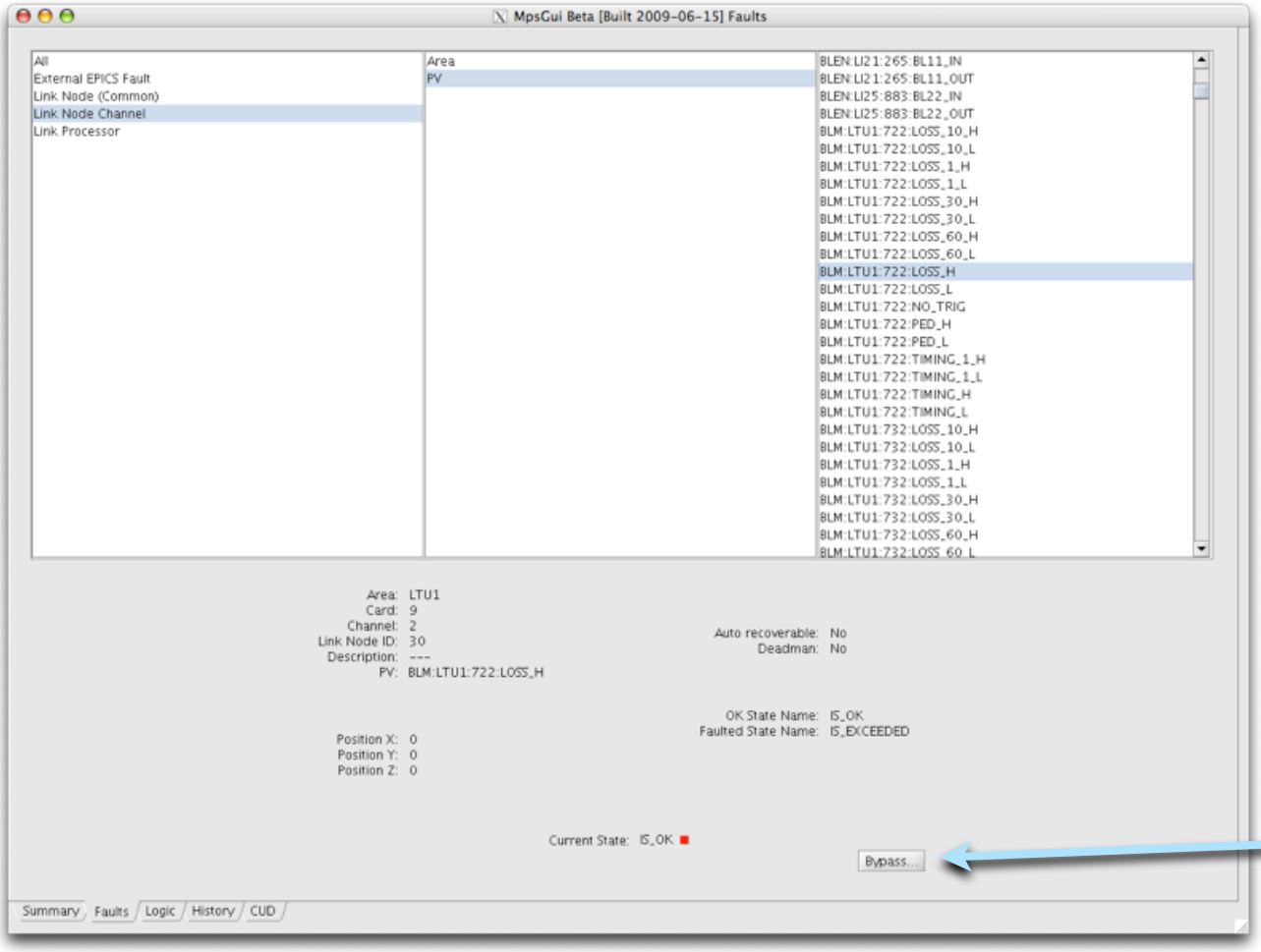



*All truth tables and states*

*History slider*

*Selected truth table details*

# User Interface - Inputs




*Input selection*

*Input details*

*Bypass button*

# Faults Bypass

SLAC

Device faults can be bypassed via an EPICS display by selecting a fault, choosing its bypass state, and supplying a bypass duration.

For example, an operator can choose to bypass a flow switch for one day by selecting the flow switch input, selecting its OK state, and giving a bypass duration of 24 hours.

All bypasses are logged and automatically timed by the MPS system. The operator is alerted when the bypass time is reached, forcing the operator to re-evaluate bypasses.

# Just for fun…

SLAC

Power Supply Failure

- Commercial module (internal flaw)

Ion Chamber noise pickup

- New electroncs 100x more sensitive than old
- Fixed by shielding

Radiated EMI from Link-Node

- Getting into Cell Phone repeater in Ctrl Room Bldg

Pockels Cell rate limit caused laser beam profile changes → affected emittance

# Timing challenges

Timing system is very flexible, and easily allows trigger timing to be moved away from beam time.

BLMs and BYKIK rely on beam time triggers to detect beam loss and kick beam.

Lots of misfiring.

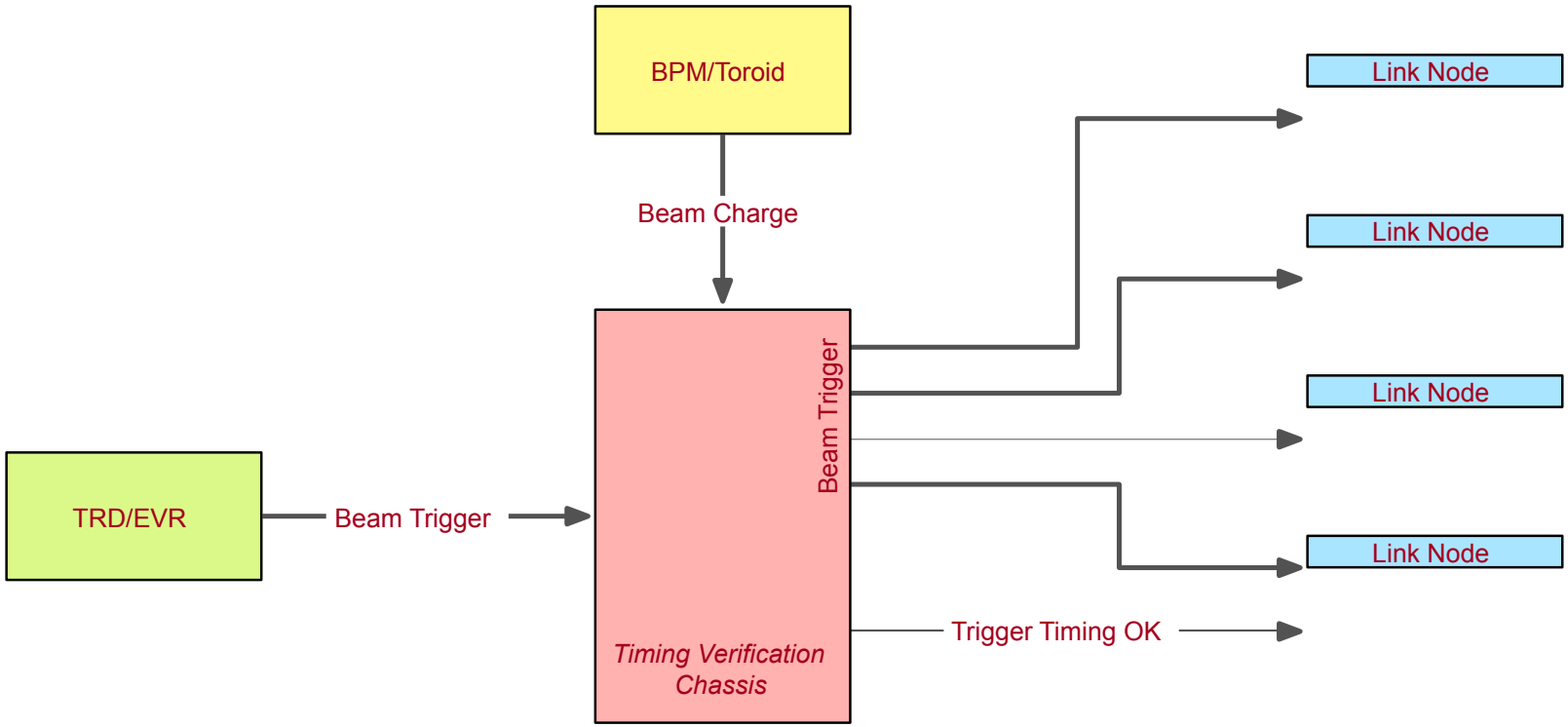
# Timing verification

SLAC

- Solution 1:
  - EVR trigger is injected to QADC analog input via directional coupler;
  - If EVR is not seen by QADC, faults LN virtual input.

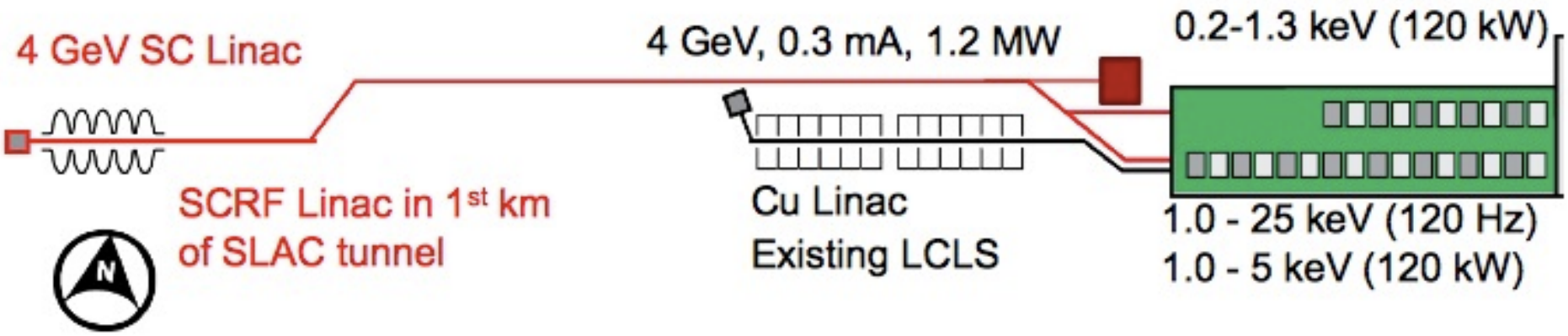
- Solution 2:
  - Use BPMs and toroids to detect beam time;
  - Compare BPM and toroid beam time with timing system's beam time triggers;
  - Output OK status to Link Node digital input while trigger timing is correct.

# Timing Verification Chassis



# Coming soon: LCLS-II





# LCLS-II MPS Requirements

SLAC

Similar to LCLS-I, except:

- LCLS-II beam loss must not interfere with LCLS-I MPS (and vice-versa);
- LCLS-II MPS must be capable of independently rate-limiting the beam in either of its two undulator beam lines;
- LCLS-II MPS is a stand-alone system separate from LCLS-I MPS, such that outages in one system do not effect the other;
- There will be two TDUND stoppers in LCLS-II. MPS must suppress faults in the two undulator beam lines according to which TDUND stopper is in;
- MPS must monitor an additional pulsed magnet in LCLS-II that switches beam between two undulator beam lines.

# Acknowledgments



Special thanks to

**Matt Boyes**

Of SLAC, for his invaluable contribution and the stimulating conversations.

# The End

SLAC

Thank you!

I hope you enjoyed it!