

Paradigms for high-availability LLRF systems

Larry Doolittle, LBNL

WG4 of LLRF07, Knoxville, October 2007

Availability defined as

$$\frac{\langle \text{Uptime} \rangle}{\langle \text{Uptime} \rangle + \langle \text{Downtime} \rangle}$$

traditionally also represented as

$$\frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

in our context it's worth subdividing MTTR (mean time to repair) into MTT detect + MTT replace + MTT recover.

Recovering from failure

Minimize labor, e.g., build self-configuring boards

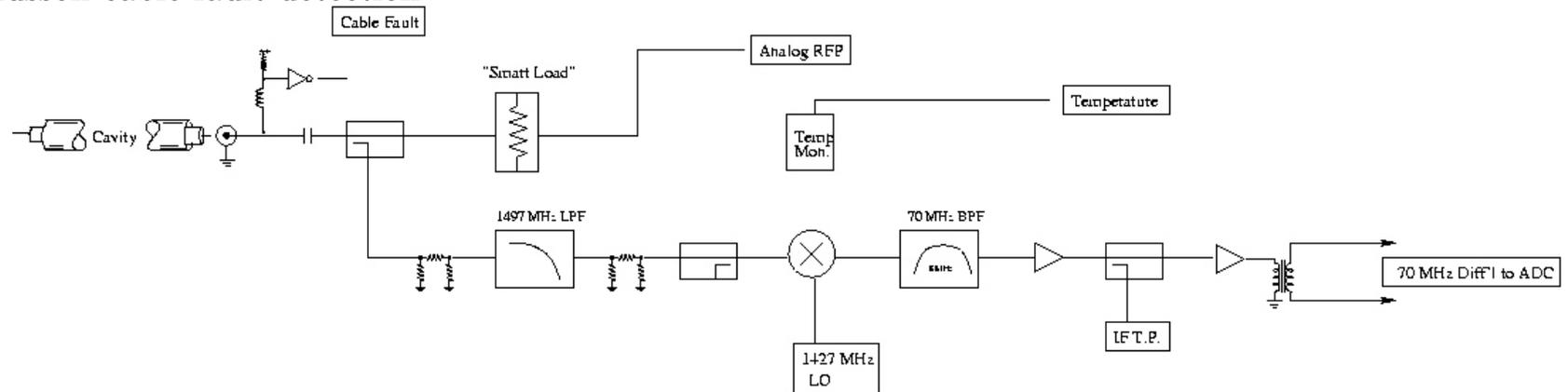
Testing

detecting failure (especially POST)

LO power monitoring

Clock glitch detection

Musson cable fault detection



Bathtub curve - find infant mortality before trying to run beam

Global redundancy

Support “limp-along mode” of system after a limited number of failures (e.g., one cavity, or one forward power cable)

Local redundancy

Triple-redundancy/voting

Software Failures

Software failures can be characterized by keeping track of software defect density in the system. This number can be obtained by keeping track of historical software defect history. Defect density will depend on the following factors:

- Software process used to develop the design and code
(use of peer level design/code reviews, unit testing)
- Complexity of the software
- Size of the software
- Experience of the team developing the software
- Percentage of code reused from a previous stable project
- Rigor and depth of testing before product is shipped.

Defect density is typically measured in number of defects per thousand lines of code (defects/KLOC).

[credit to eventhelix.com]

Failure aversion gives extra motivation to write simple software, and more importantly, to structure systems so that simple software will meet the needs of the accelerator.

Ideal maintenance cycle

1. *In-situ* diagnostics to determine which module has degraded/failed
2. Information presented to operators in control room
3. Workarounds automatically engaged to keep accelerator running as well as possible in the face of all but the worst failure modes
4. Robotic exchange of module in tunnel during daily maintenance minute
5. New module is tested *in-situ*, presumably by firing an RF pulse without beam
6. New module works without operator intervention, using calibration information combined from new module bench measurements, old module plant characterization tables, and one calibration beam pulse

[Doolittle and Simrock, ILC-Americas Workshop SLAC October 14-16, 2004]