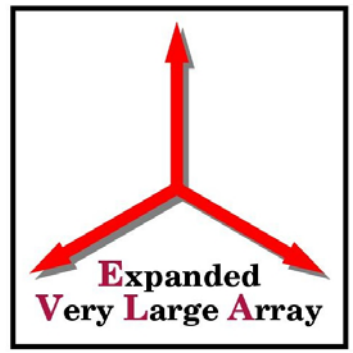


CMIB Software

Bruce Rowen

Topics

- Design Concepts
- Communication Methods
- XML
- CMIB Software Structure
- CMIB Module Drivers

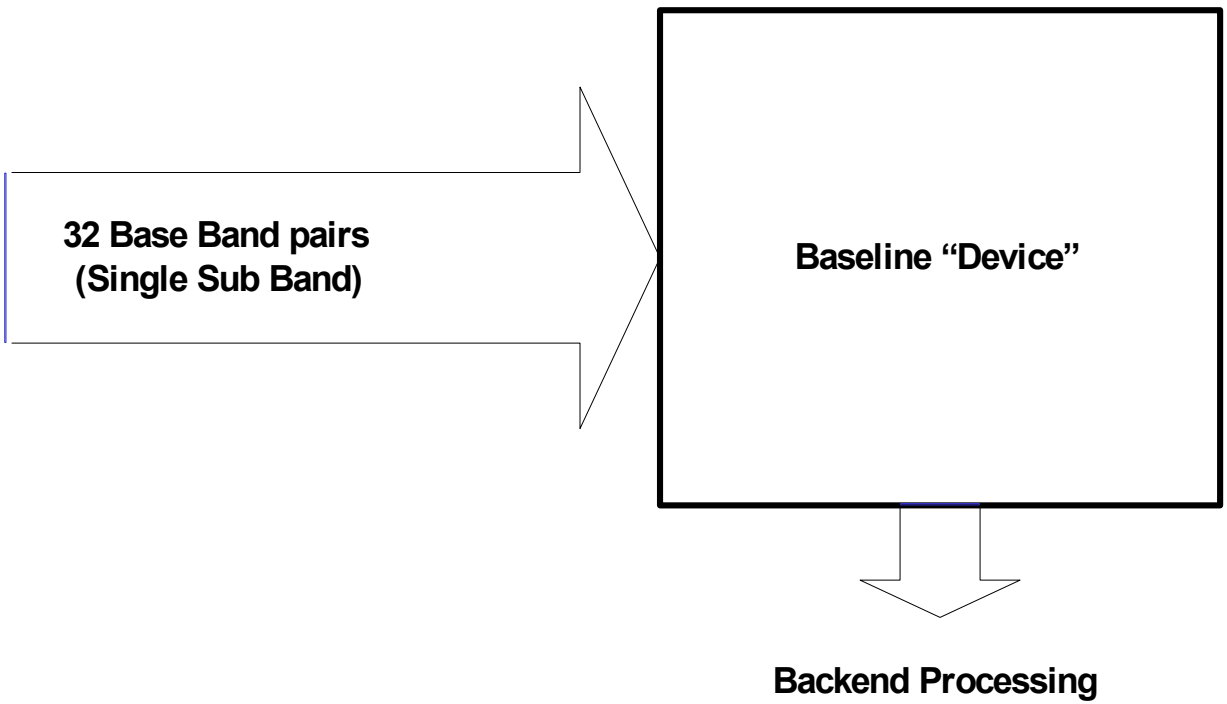
Topics

- Design Concepts
- Communication Methods
- XML
- CMIB Software Structure
- CMIB Module Drivers

Design Concepts

- Each CMIB operates as a virtual correlator “device”
- Correlator device can have different levels of abstraction
- Can operate stand alone or integrated into system
- Device generally appears stateless externally
- Device is self contained for diagnostics and healing

Design Concepts
(Baseline Board)



Design Concepts
(Baseline Board)

- Define actions and properties of a single Base Band Pair
  - Identified by StationID (0-255), SubBandID (0-17), BaseBandID (0-7)
  - Define Properties:
    - Sample Size & Rate
    - Lag Block Size, Start #, End #, & Extent (Recirculation)
    - Products (RR, LL, RL, LR)
    - Activation Time (Implies action queues)
    - Other details!

Design Concepts

- Access device directly or via VCI
- Subscribe to various levels of system “chatter”
- Query device for current state
- Control hardware at various sub-function levels (recirculation, correlation, accumulation, etc.)
- Control at near-register level if truly paranoid

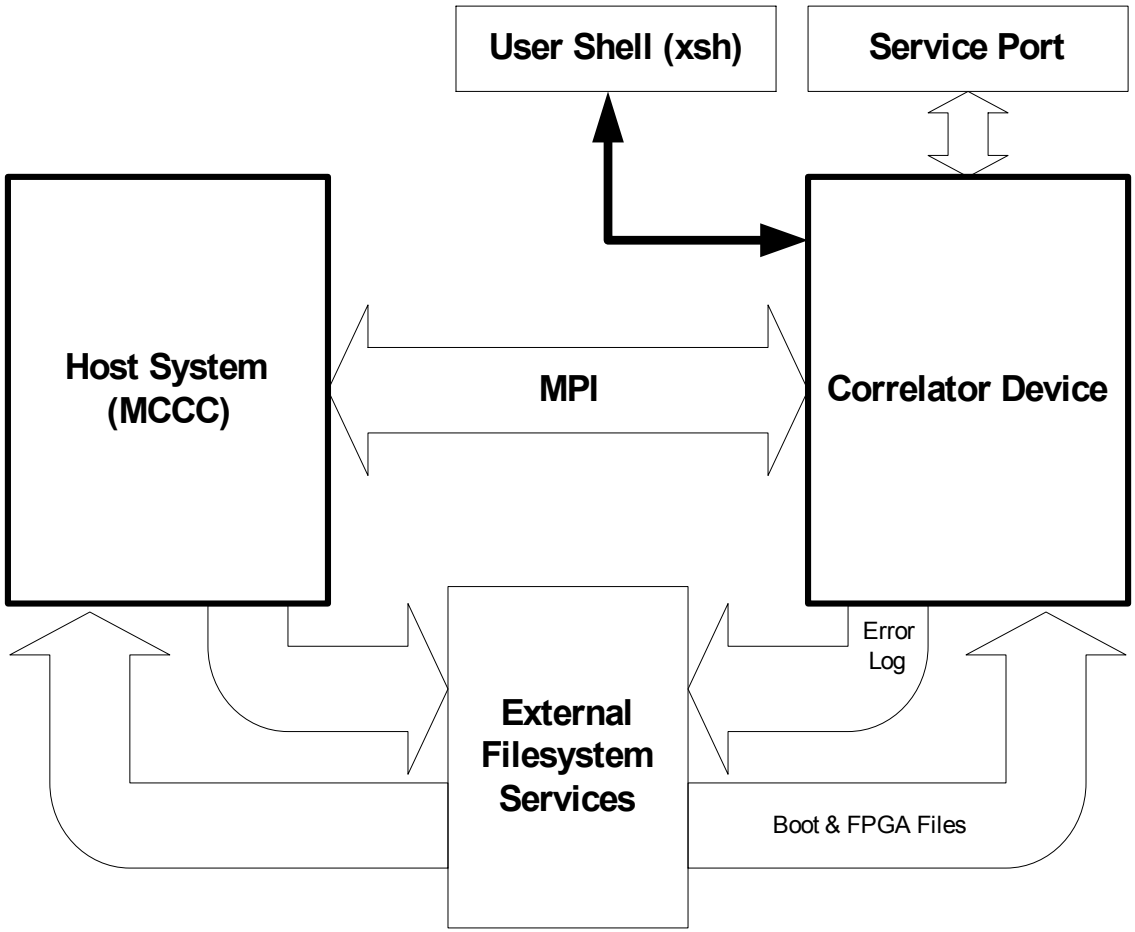
Topics

- Design Concepts
- Communication Methods
- XML
- CMIB Software Structure
- CMIB Module Drivers

Communication Methods

- Network supports both M&C plus OS-file services needs
- Ethernet is primary access medium. Serial port is secondary
- Two primary access ports, “on-line” and “service”
- Network services (NFS, etc.) are “hidden”

Communication Methods



Communication Methods

(on-line system)

MPI – Message Passing Interface

Convenient framework for message passing among heterogeneous hosts. Underlying protocol is transparent and flexible (SSH, RSH etc.)

Software methods are simple read/write operations with options for blocking, polling, and status checking.

Targets are identified by ID tag groups. Messages can contain separate tags for expedited dispatching in multithreaded systems.

Communication Methods

(Service Port)

- Socket based
- Can use telnet for interactive session
- Can use browser for GUI session
- Same functionality as On-Line port
- This is the “back door” for service people, tweakers, and hacks
- Single correlator device only unless redirection through MCCC (broker services)

Communication Methods
(User Shell)

- Generic password challenge system login
  - Network “rlogin” or serial port dumb terminal login
  - Generic Linux user environment plus tools to assist with hardware communication
  - Most likely use is for operating system work or network isolated testing
- Use of xsh (XML Shell) for service port emulation

Topics

- Design Concepts
- Communication Methods
- XML
- CMIB Software Structure
- CMIB Module Drivers

XML

- XML eXtensible Markup Language
- XML describes how data is organized, HTML describes how data is displayed
- Provides a very flexible method of passing data among different systems
- Parsers, verifiers, various tools, etc. are prolific
- Isolates changes in data handlers from affecting other systems

XML Schemata

- Provides a template for the XML
- Defines as much (or as little) of the data format as you want
- Used by verifiers and DOM (Document Object Model) tools
- Allows specification of element attributes such as ranges, types, counts, etc.
- More versatile than DTDs and use XML syntax

- These will define all correlator interfaces

XML

Do a mid-level configuration of a correlator chip

```
<?xml version = “1.0”?>
<correlator>
        <correlatorID = “3.4”>
                <baseBandID = “0”>
                        <productSourceID = “1”/>
                        <products = “RL”/>
                        <products = “RR”/>
                        <products = “LL”/>
                        <products = “LR”/>
                        <lagCount = “128”/>
                </baseBandID>
                <baseBandID = “2”>
                        <productSourceID = “3”/>
                        <products = “RR”/>
                        <products = “LL”/>
                        <lagCount = “512”/>
                </baseBandID>
                <masterDump = “x”/>
        </correlatorID>
</correlator>
```

Topics

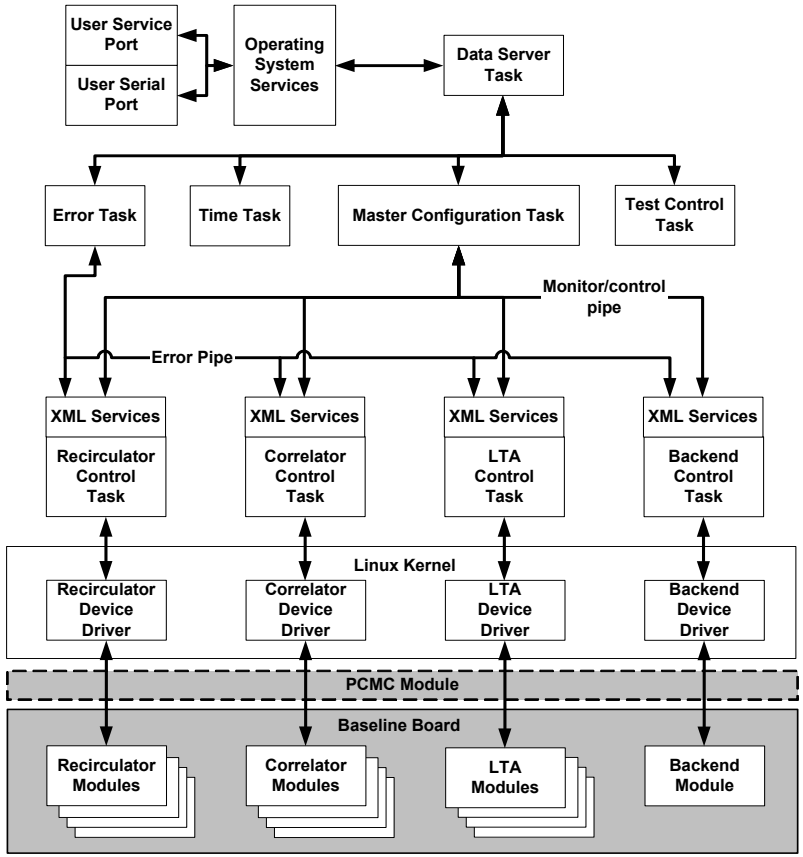
- Design Concepts
- Communication Methods
- XML
- CMIB Software Structure
- CMIB Module Drivers

CMIB Software Structure

- Modular!
- Use standard Linux device drivers to integrate hardware modules into the operating system
- Make use of standard communication channels
- Make each layer above the hardware as isolated as practical
- Module control task for each functional hardware module

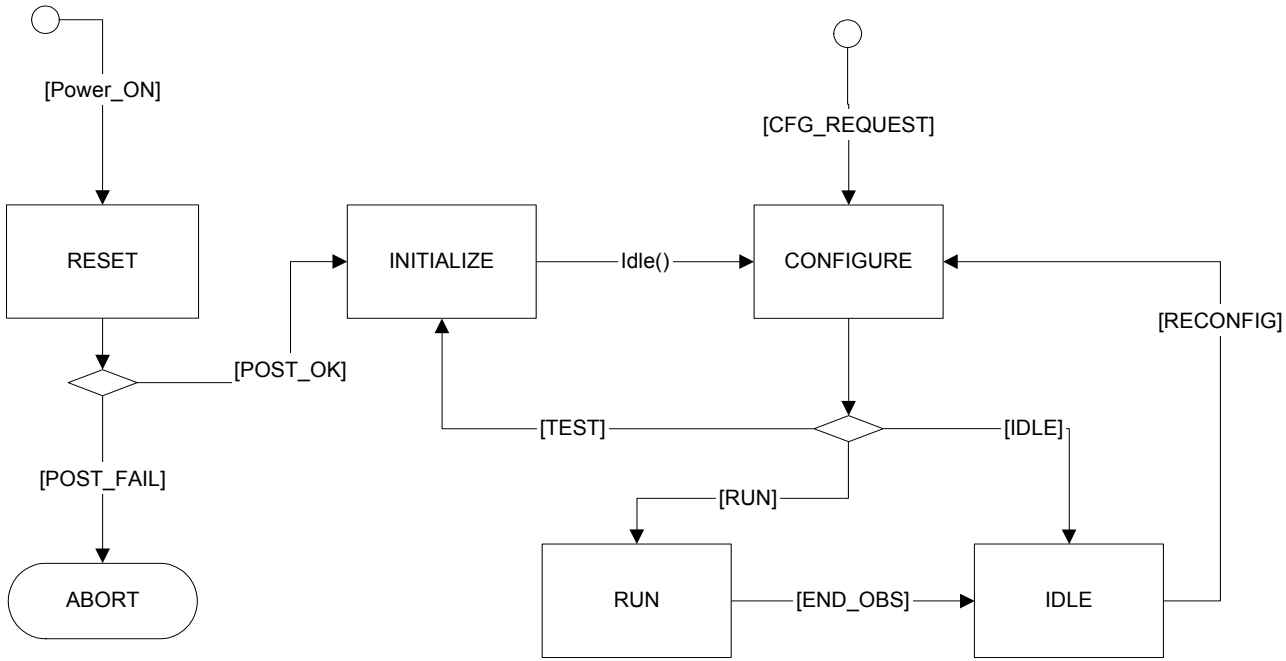
CMIB Software Structure

(Baseline Board)



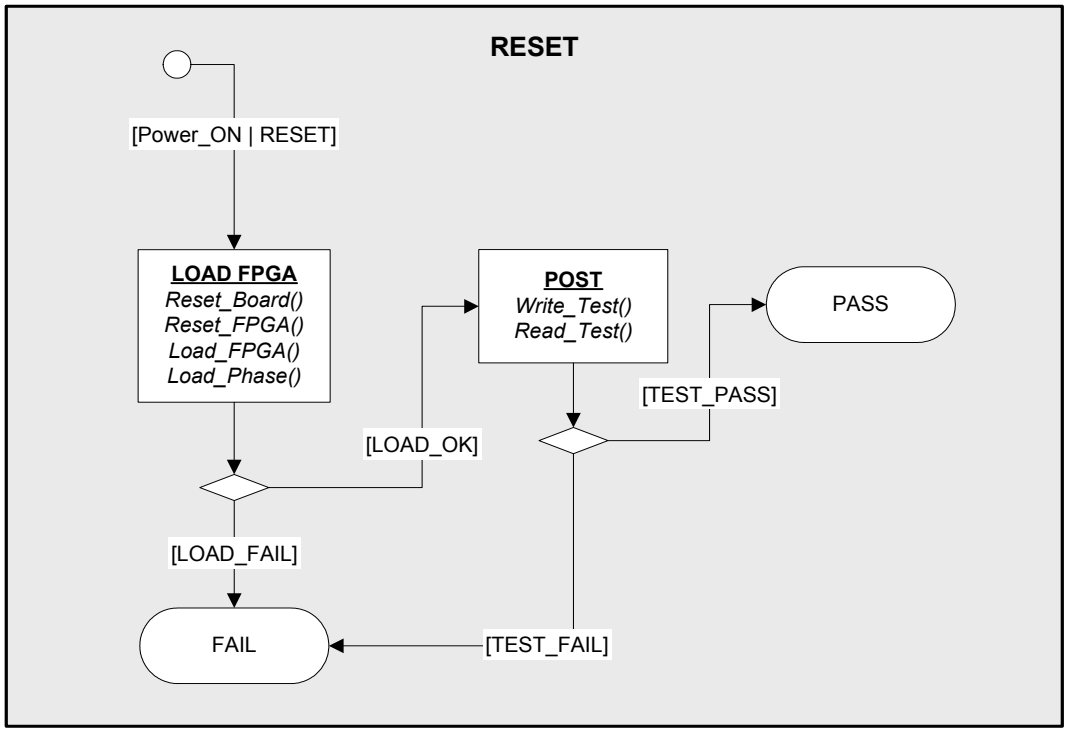
CMIB Software Structure

(Recirculator Control Task)



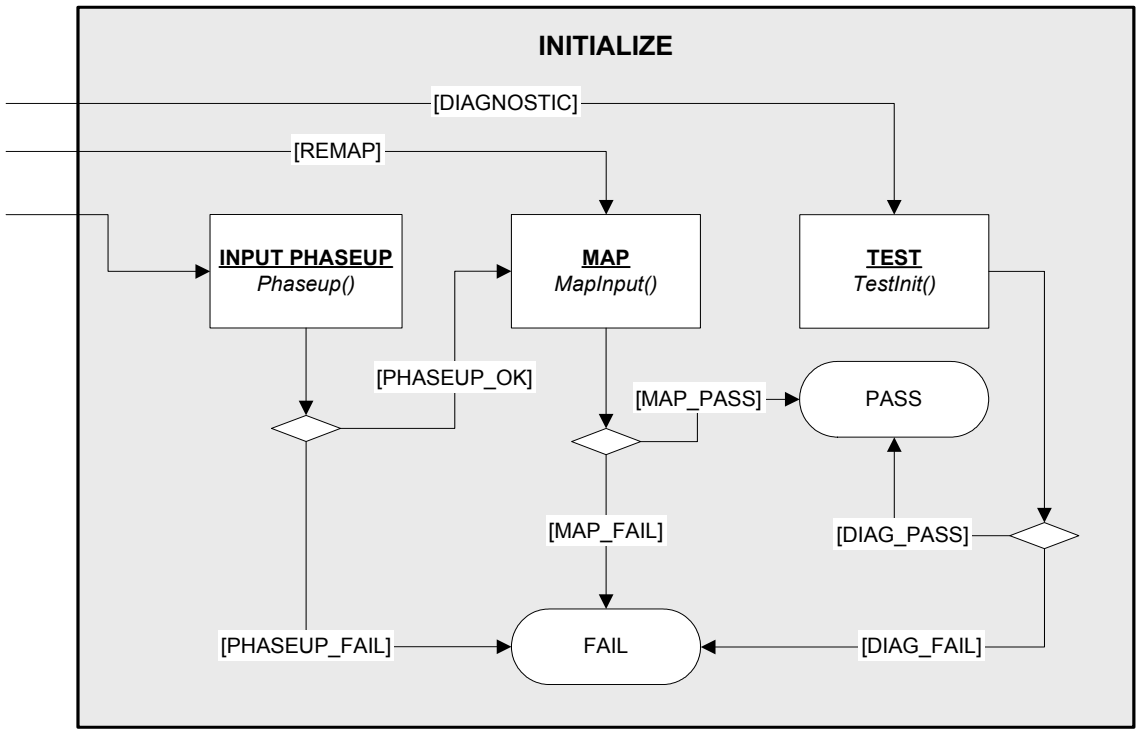
CMIB Software Structure

(Recirculator Control Task)



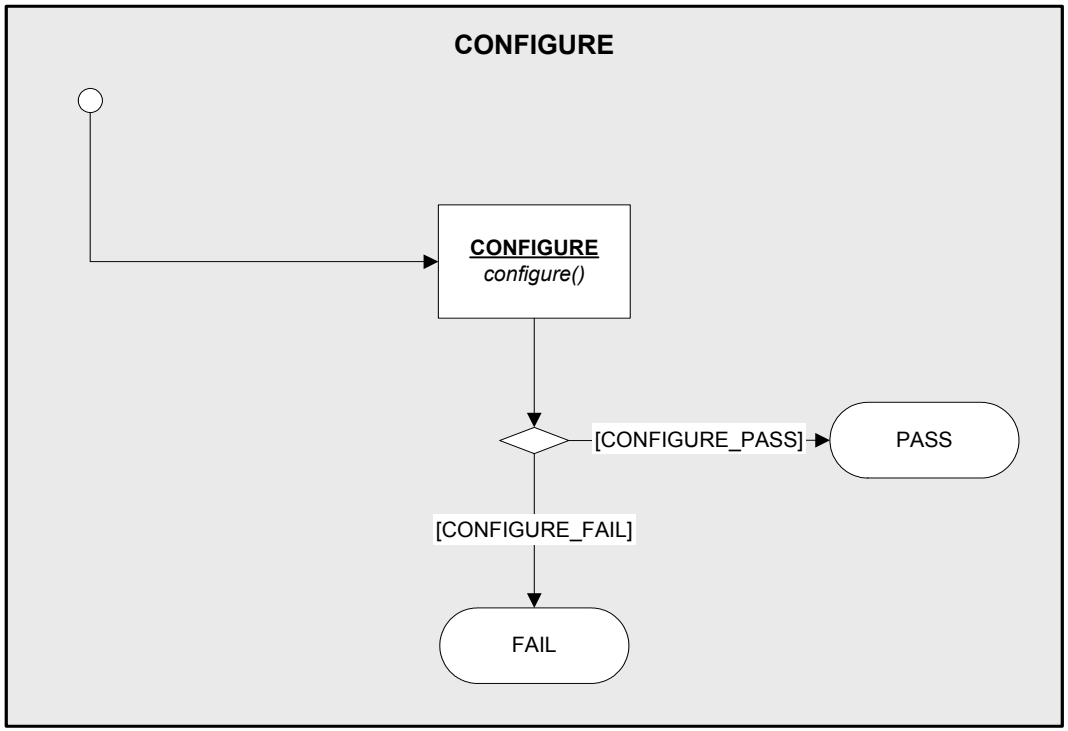
CMIB Software Structure

(Recirculator Control Task)



CMIB Software Structure

(Recirculator Control Task)



Topics

- Design Concepts
- Communication Methods
- XML
- CMIB Software Structure
- CMIB Module Drivers

CMIB Module Drivers

- Build to operate as standard Linux device drivers
- Attached to the kernel during boot
- Self configuration and POST
- Multitask ready
- Support “open”, “close”, “read”, “write”, and “ioctl”
- Full reflection of module register sets through ioctl enumerations

CMIB Module Drivers

(Recirculator Driver)