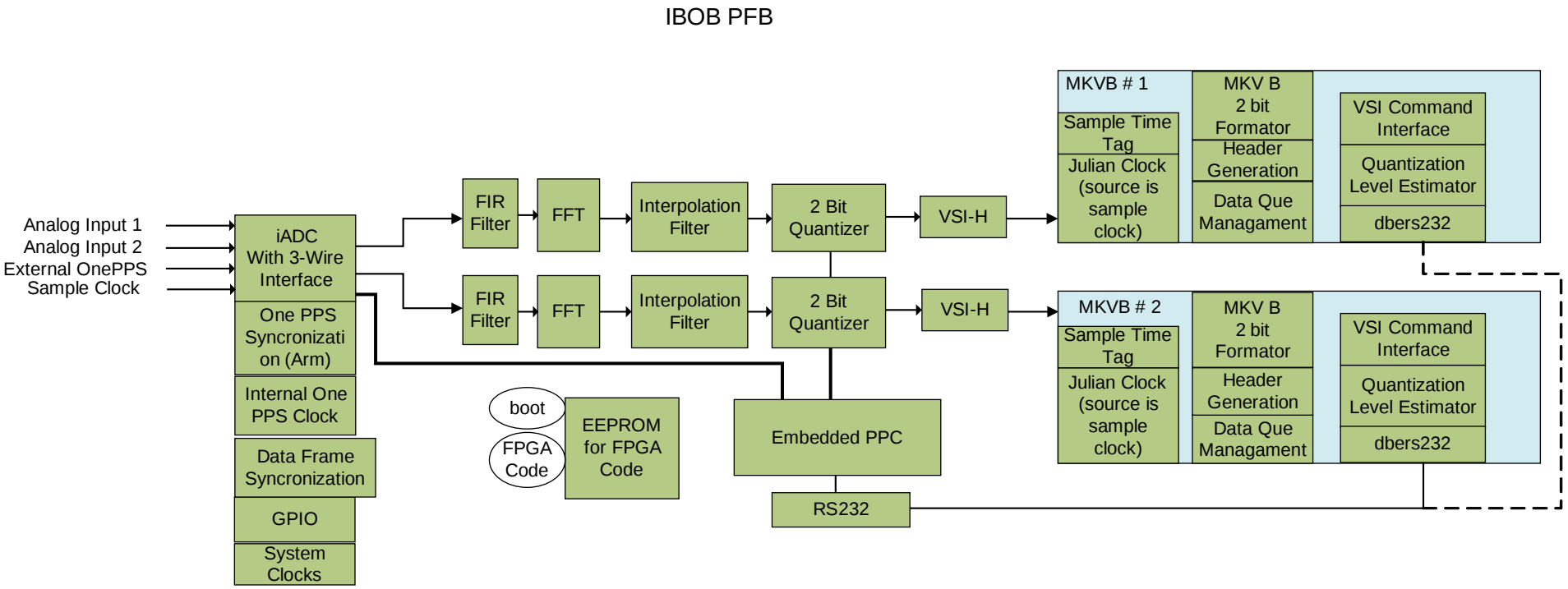
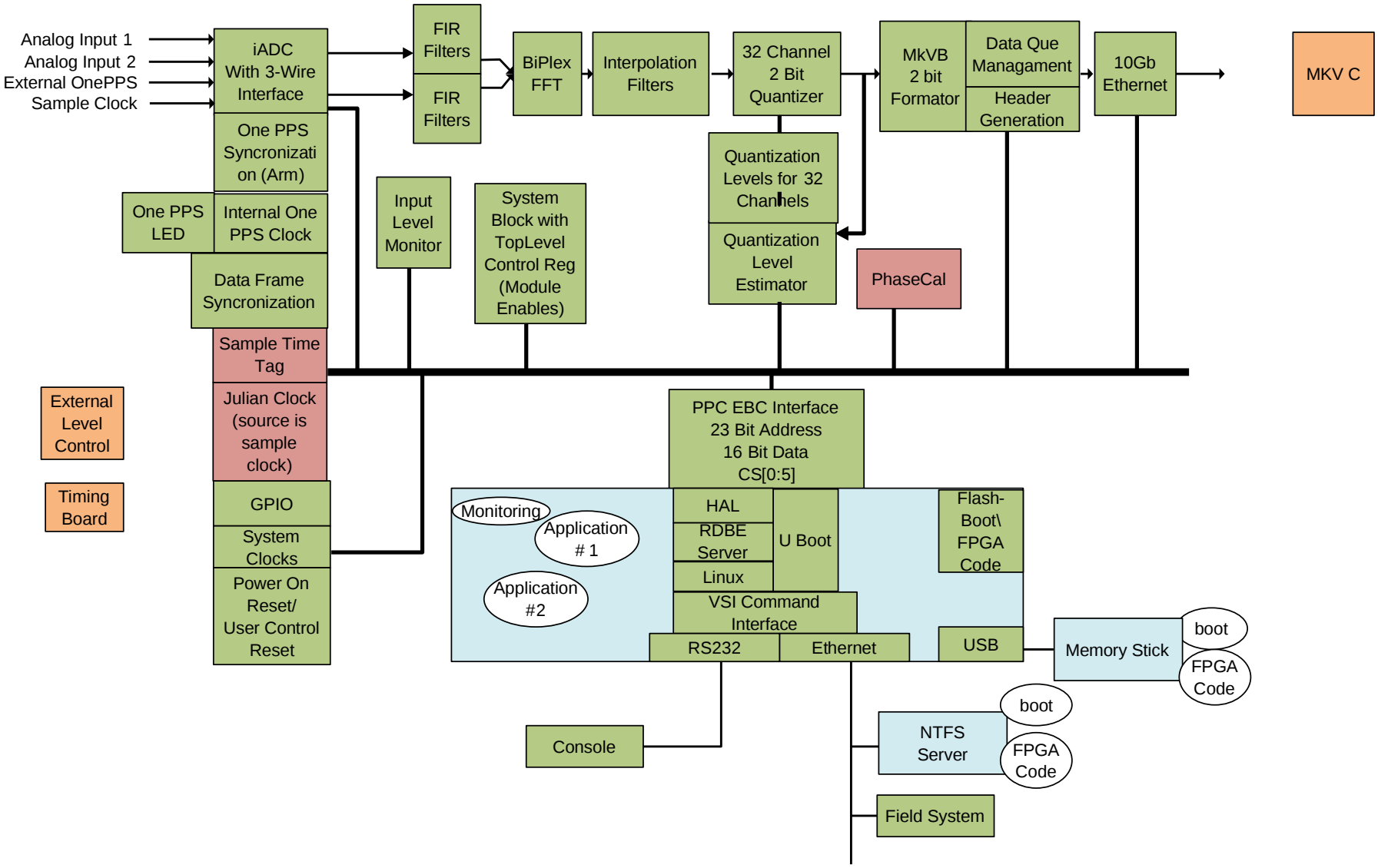
Demonstrations of RDBE-PFB Data Flow

Review the original PFB Data Flow



RDBE-PFB Data flow



Design Methodology

- Build a BSP to support the Roach Hardware.
- Haystack supportable design.
- Design using commercially availability tool flow.
- Code base needs to be maintained for life cycle of instrument.
- Design at the HDL level for code portability to future hardware architecture changes and device changes.
- Modular design for regression testing and verification of modules, especially when adding features to the design, upgrading tools and hardware.
- SVN repository for RDBE designs.
- Documentation.

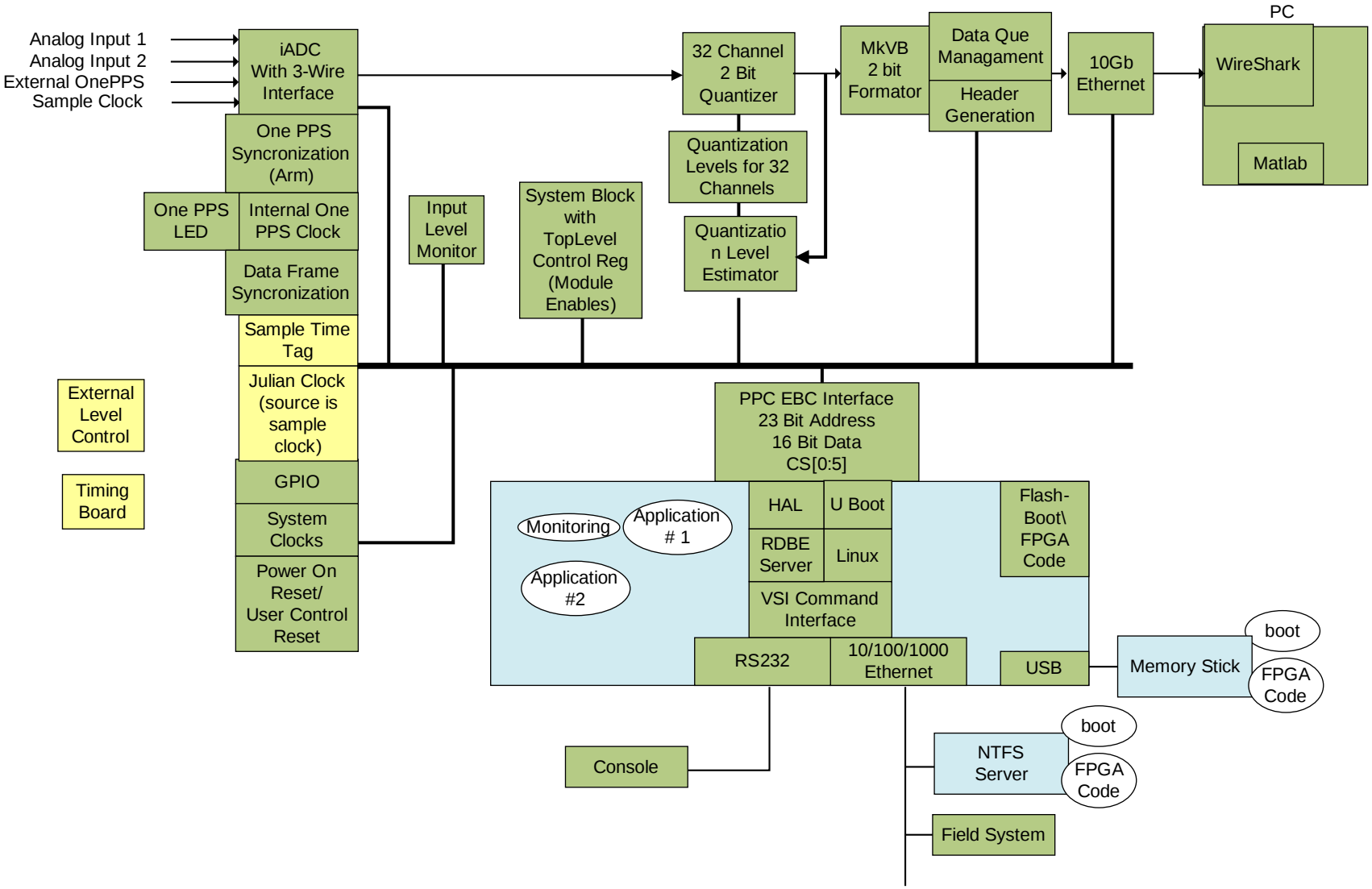
Demonstration/verification of RDBE-PFB Data Integrity

- U-boot/LinuxDevice Driver/RDBE Server Application.
- Processing of commands via console and LAN port.
- FPGA load from USB Stick, NTFS Server and Flash.
- System support modules tested/verified.
- Signal Processing Modules are independently verified in hardware.
- Mk5B (2bit) format for Geodesy has been tested/verified.
- Additional processing applications can be implemented.

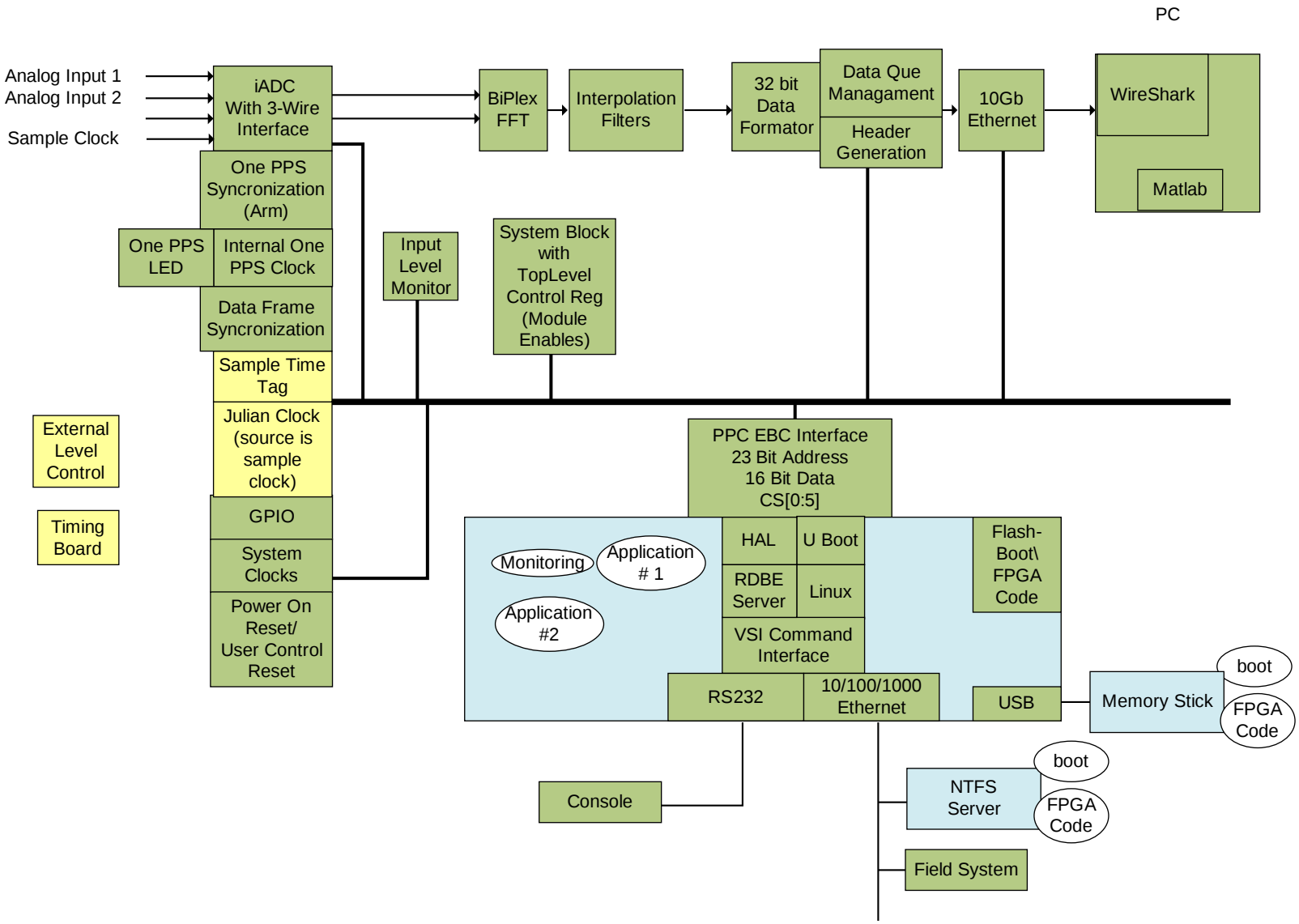
Data Flow Test setup

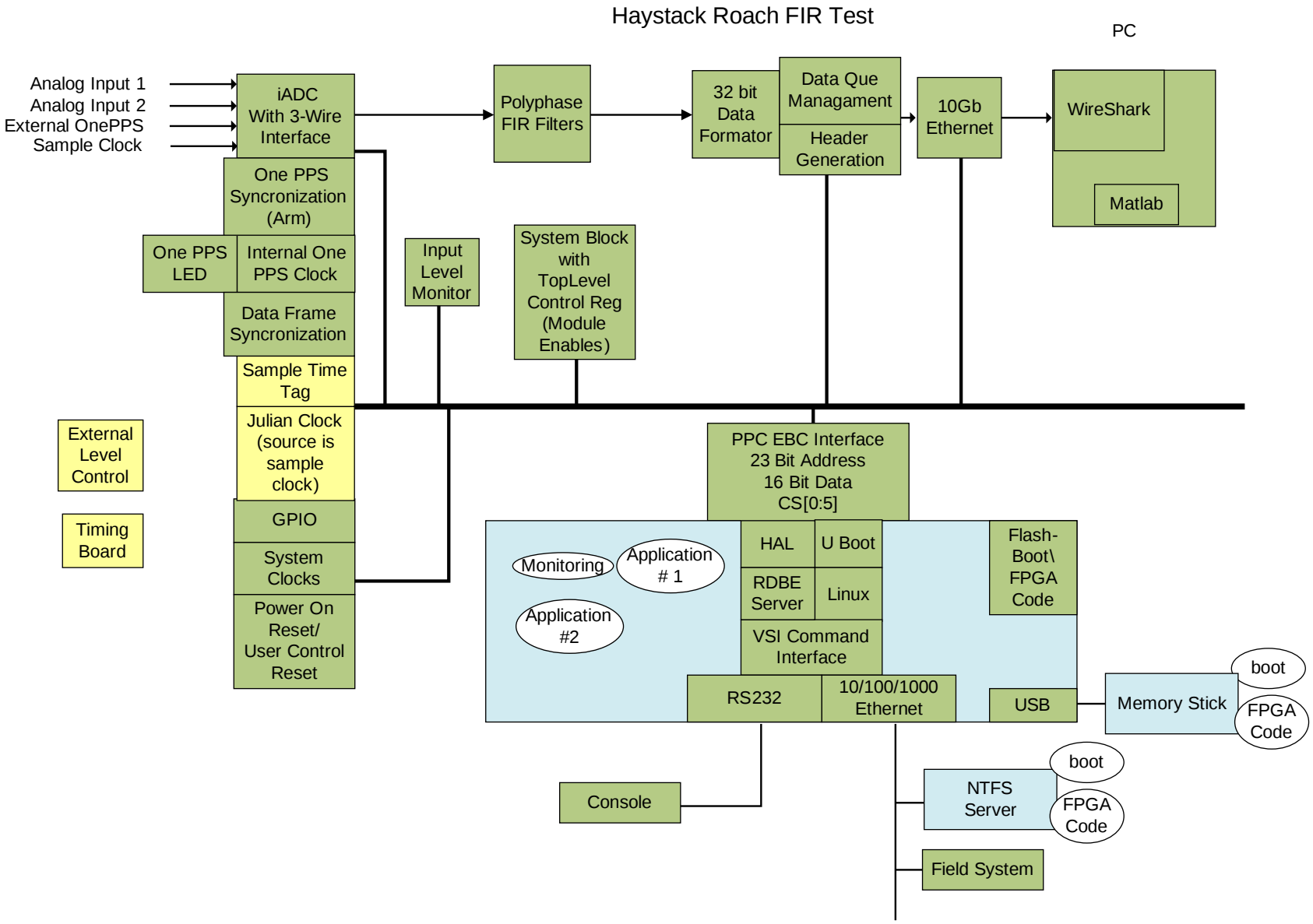
- Build top level module with DUT.
- Use signal/noise source as input to ADC.
- Run various (numerous) test on DUT.
- Collect 10Ge data packets using WireShark on PC.
- Analyze/verify data using Matlab, not only integrity but also timing.
- Conduct real time impulse response of filters to ensure correct coefficient ordering.

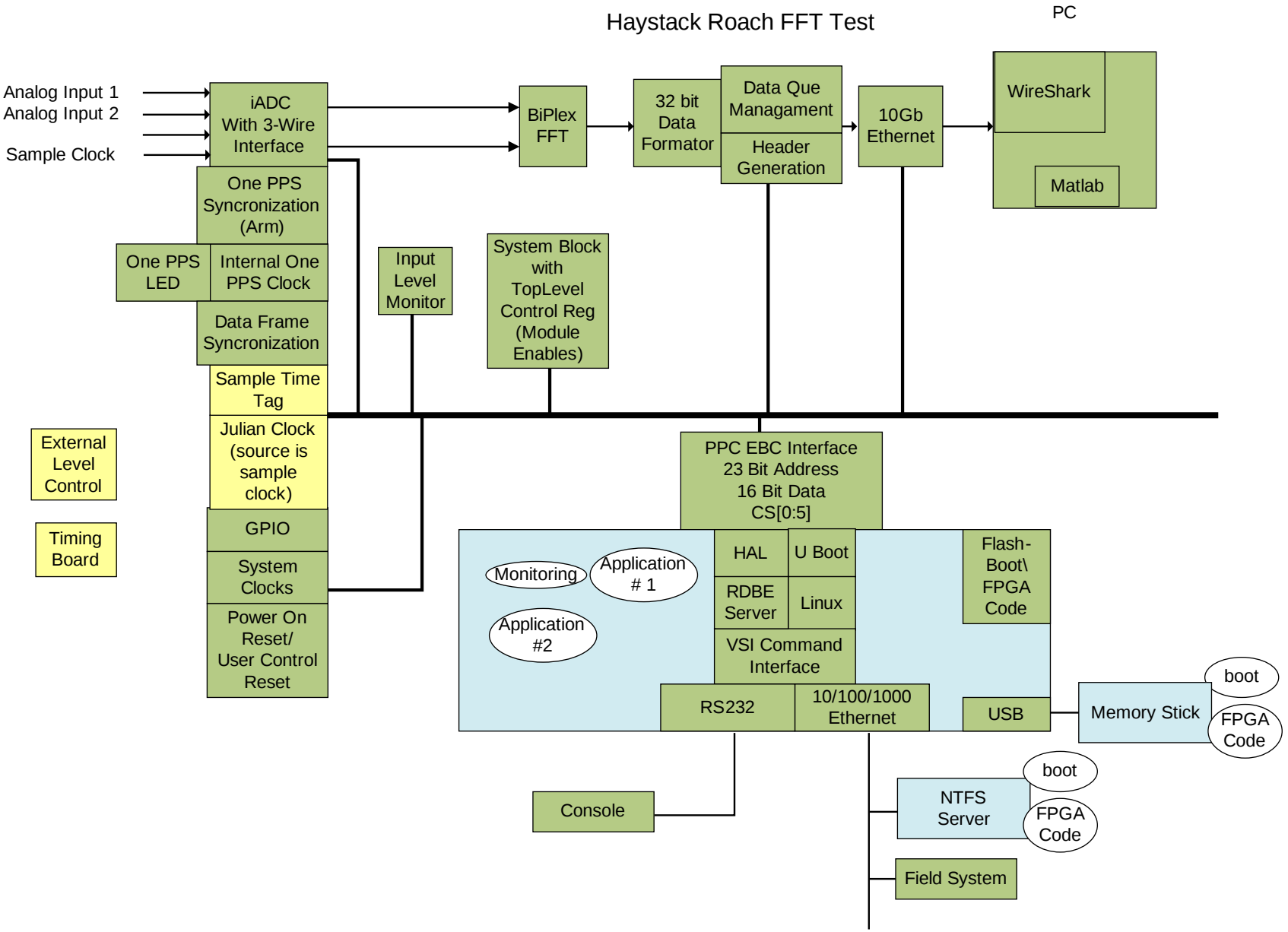
Haystack Roach Quantizer Top Level Test



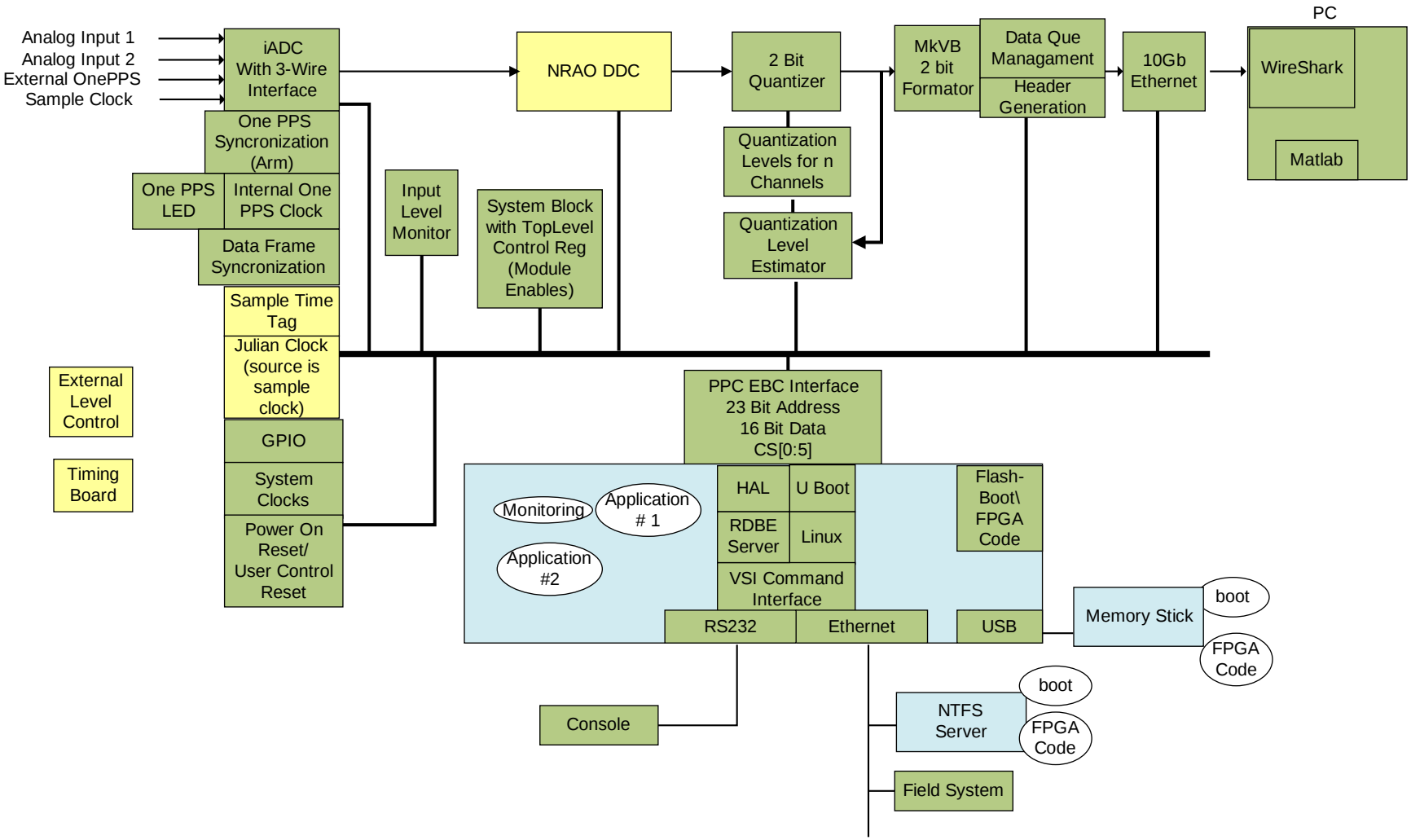
Haystack Roach Interpolator Test




Haystack Roach FIR Test
PC
Analog Input 1
Analog Input 2
External OnePPS
Sample Clock
iADC With 3-Wire Interface
Polyphase FIR Filters
32 bit Data Formator
Data Que Managament
Header Generation
10Gb Ethernet
WireShark
Matlab
One PPS Syncronization (Arm)
One PPS LED
Internal One PPS Clock
Data Frame Syncronization
Sample Time Tag
Julian Clock (source is sample clock)
GPIO
System Clocks
Power On Reset/ User Control Reset
Input Level Monitor
System Block with TopLevel Control Reg (Module Enables)
External Level Control
Timing Board
PPC EBC Interface
23 Bit Address
16 Bit Data
CS[0:5]
HAL
U Boot
Linux
RDBE Server
VSI Command Interface
Monitoring
Application # 1
Application #2
Flash-Boot\ FPGA Code
RS232
10/100/1000 Ethernet
USB
Memory Stick
boot
FPGA Code
Console
NTFS Server
boot
FPGA Code
Field System


Haystack Roach FFT Test
PC
Analog Input 1
Analog Input 2
Sample Clock
iADC With 3-Wire Interface
BiPlex FFT
32 bit Data Formator
Data Que Managament
Header Generation
10Gb Ethernet
WireShark
Matlab
One PPS Syncronization (Arm)
One PPS LED
Internal One PPS Clock
Data Frame Syncronization
Sample Time Tag
Julian Clock (source is sample clock)
GPIO
System Clocks
Power On Reset/ User Control Reset
Input Level Monitor
System Block with TopLevel Control Reg (Module Enables)
External Level Control
Timing Board
PPC EBC Interface
23 Bit Address
16 Bit Data
CS[0:5]
HAL
U Boot
RDBE Server
Linux
VSI Command Interface
Monitoring
Application # 1
Application #2
Flash-Boot\ FPGA Code
RS232
10/100/1000 Ethernet
USB
Memory Stick
boot
FPGA Code
Console
NTFS Server
boot
FPGA Code
Field System

NRAO Top Level



Task that still need to be completed

- Need to add Julian clock, level control and timing module.
- Phase Cal., 1 pps comparison module.
- Implement command interpreter with error checking.
- Implement VSI command set with error checking.
- RDBE Monitor application (Human Interface).
- Xport Monitor Application (External Monitoring, power reset).
- Documentation.
- Incrementally build PFB top level design (basically connect the SP modules together and test/verify at each stage).
- Testing with Mk5C when available.
- Operational testing. Comparison with iBOB.
- Implement 4Gbps version.
- NRAO DDC ?