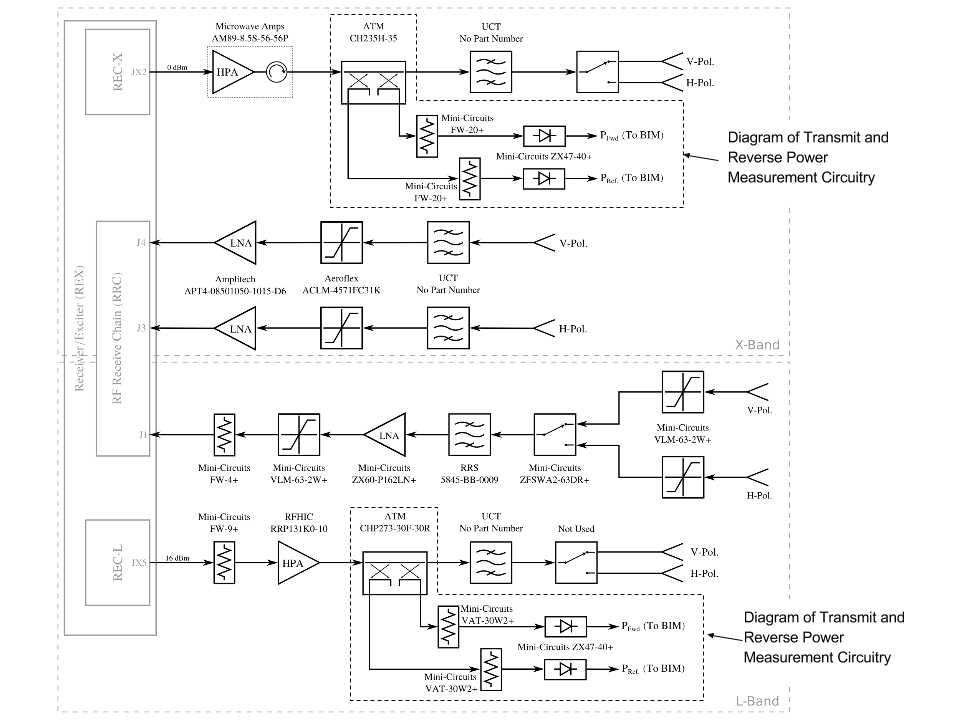
Requirement Review

Some of the figures and table from the user requirement that are relevant will be referred to in this document

Figure 1: System Diagram of NeXtRAD RF Frontend With Part Numbers

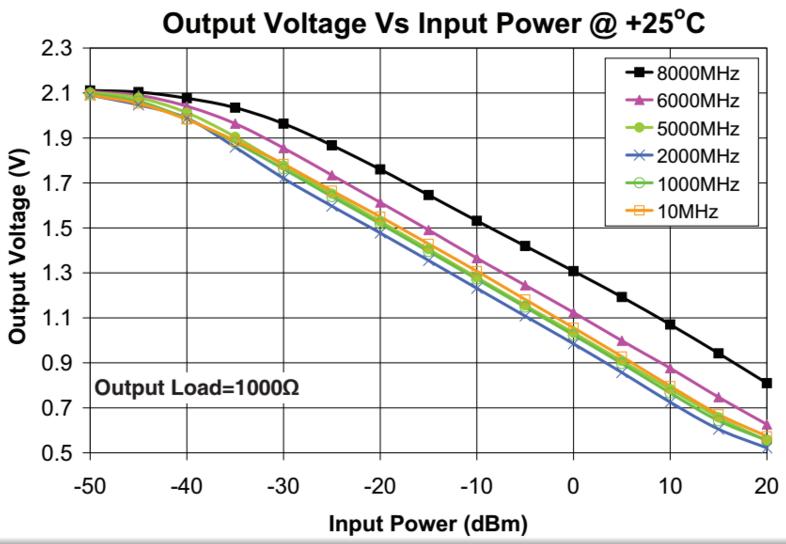


Figure 2: Output Voltage vs Input Power @+25OC for the Mini-Circuits ZX47-40+

|  |  |  |
| --- | --- | --- |
| Frequency Band | Component | Coupling(dB) |
| L Band | ATM CHP273-30F-30R | 30±1 |
| X Band | ATM CH235H-35 | 35±1.50 |

Table 1: The coupling in dB for the couplers

|  |  |  |
| --- | --- | --- |
| Frequency Band | Component | Attenuation(dB) |
| L Band | Mini-Circuits VAT-30W2+ | 30±0.3 |
| X Band | Mini-Circuits FW-20+ | 20 |

Table 2: The attenuation of the attenuators in dB

|  |  |
| --- | --- |
| Frequency Band | Net Attenuation due to coupling and attenuator (dB) |
| L | 60 |
| X | 55 |

Table 3: Net Attenuation due to coupling and attenuator

Keys points to note:

1. Sample and hold circuitry will be needed because the pulse length are between 1𝞵s and 20𝞵s which is very short.
2. The forward and reverse power (peak values) must be sampled during the transmission, and these digital values from the measurements will need to be transfer via ethernet to the Node Controller.

The Desired Solution

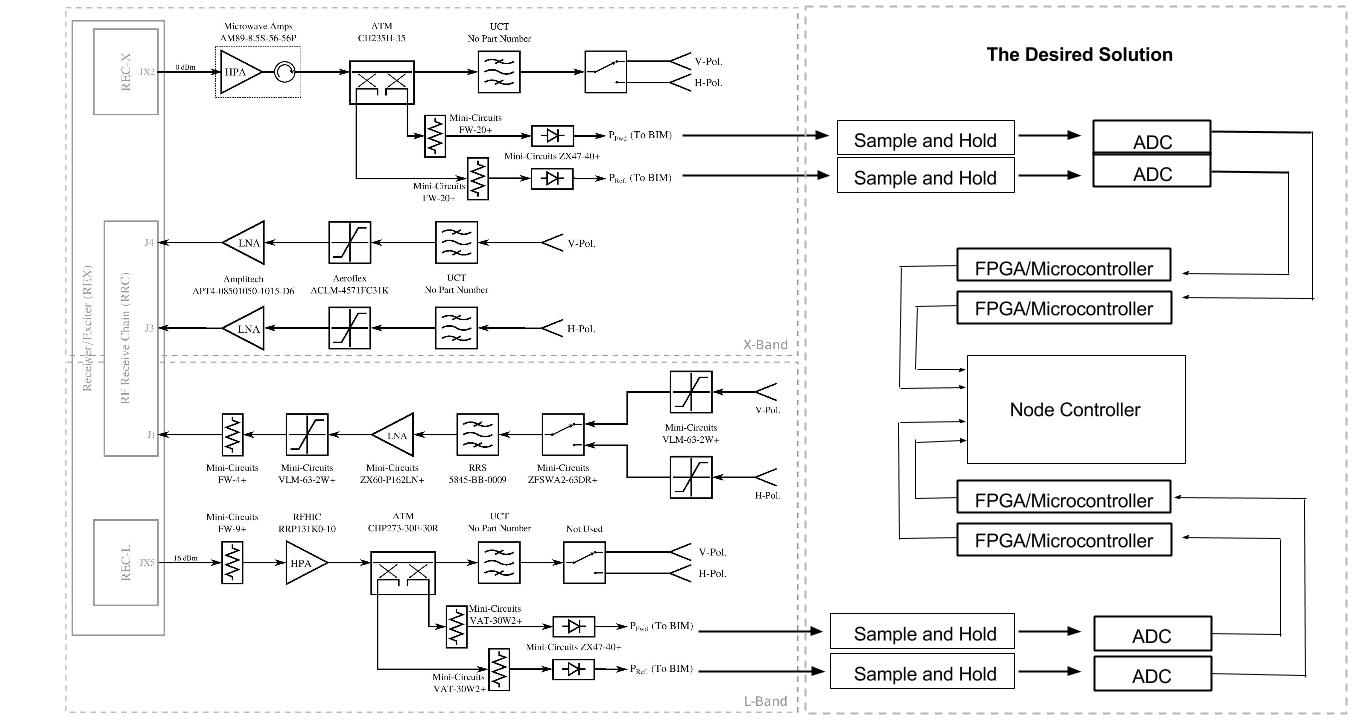


Figure 3: System Diagram of NeXtRAD RF Frontend With Part Numbers along the Sampling Circuitry

The solution that will be produced will have the basic the structure shown in Figure 3.

**The Possible Solutions for the Sample and Hold**:

*Why sample and hold circuitry*

In point 1 a reason for why sample and hold circuitry is needed was given but it is a vague so we will give a more detailed reason. The sampling and digitization of the Pref. and Pfwd for both bands will most likely be done using a microcontroller or an FPGA that has an ADC. If we put a sample and hold between the output of the forward and reverse power measurement circuitry and the ADC then we can relax the timing requirements of the ADC, that is to say rather than having the device sample when a pulse is being transmitted we can now allow it to sample the value held by the sample and hold circuitry sometime between pulses. There is about 1ms between pulses. This will mean that the device and its ADC need not be working at a very fast clock rate.

*Solutions:*

1. Build A Sample and Hold Circuit from Scratch

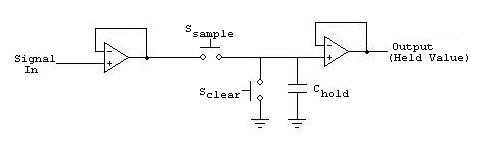


Figure 4: Sample and Hold Circuit

Source: <http://www.eecs.tufts.edu/~dsculley/tutorial/opamps/opamps7.html>

In Figure 4 there are two buffers. The first one is the input buffer and it ensures that the input signal will not be affected by the capacitor when it is charging. The second one is the output buffer and it ensures that the capacitor does not discharge over time. When Ssample is pressed Chold is charged. When Sclear is pressed Chold is discharged.[[1]](#footnote-0)

In this case the only major things to be chosen are the amps and Chold. The capacitor needs be charged within the 1𝞵s because 1𝞵s is the shortest possible pulse length. During charging Ssample needs switched on and Sclear needs to be switched off. Before charging occurs Chold needs to be discharged first to clear the previous voltage level. In order to have the Chold charge up fast enough the delays in the amps must be taken into consideration.

When the prepulse occurs Chold must be cleared by switching on Sclear while Ssample is off. After it has been cleared Sclear must be switched off and Ssample must be switched on so then when the pulses occurs the sample and hold will be ready to sample it. This means that within a 30𝞵s interval both switches need to be switched on and off at least once for they need to react within a few microseconds therefore the switches can not be push buttons. Perhaps transistors can be used in this case.

After the charging occurs the Ssample must be switched off to prevent the capacitor from discharging when the pulse is not there.

The output of the sample and hold would then be fed into the input of the ADC.

2. Buy A Sample and Hold Circuit:

Designing the Sample and Hold Circuit is time consuming so another alternative is to just by a Sample and Hold Circuit

Sample and hold specs:

\*Charge to 5% of the final input value within 0.5 𝛍s.

\*It must be able to handle up 2v at the input.

A possible sample and hold device:

Here is a list of possible Sample and Hold Devices that can be used:

**HA5351**

***Its Features:***

Fast Acquisition to 0.01%. . . . . . . . . . . . . . . . . 70ns (Max)

Low Offset Error . . . . . . . . . . . . . . . . . . . . . . . ±2mV (Max)

Low Pedestal Error. . . . . . . . . . . . . . . . . . . . ±10mV (Max)

Low Droop Rate . . . . . . . . . . . . . . . . . . . . . . 2µV/µs (Max)

Wide Unity Gain Bandwidth . . . . . . . . . . . . . . . . . . 40MHz

Low Power Dissipation . . . . . . . . . . . . . . . .220mW (Max)

Total Harmonic Distortion (Hold Mode) . . . . . . . . . -72dBc - (VIN = 5VP-P at 1MHz)

Fully Differential Inputs

On Chip Hold Capacitor

Pb-Free (RoHS Compliant)

With the HA5351 the voltage level within it would only drop by 2mV within a 1ms. Which is not much considering that the output range for the Mini-Circuits ZX47-40+ is .5V to 2.1V. Meaning that there is a range of 1.6V.

**ADC Specifications:**

1. There is a time lapse of 1ms between each pulse meaning that the sample and hold circuit will hold the value of a pulse for at least 0.97ms since the a prepulse will occur 30𝛍s before a pulse and the sample and hold will need to be cleared within that 30𝛍s. To allow to uncertainties the adc must be able do it’s conversion within 0.7ms. So that about 2 samples per millisecond.

2. Must be able to take inputs of up to 2.1V.

3. There is no strict requirement for quantization error. A reasonable maximum quantization error is ±2.5mV for this case.

**The Possible Solutions for the FPGA/Microcontroller**:

**Solution 1:**

It was proposed that a microcontroller be used to process signals from the ADC to the node controller. This was suggested on the grounds that a microcontroller is cheap and relatively easy to programme. However, a microcontroller with ethernet compatibility would have to be used. Raspberry Pi was chosen as one of the best candidate microcontroller for this operation. A Raspberry Pi needs an external ADC.

**Solution 2:**

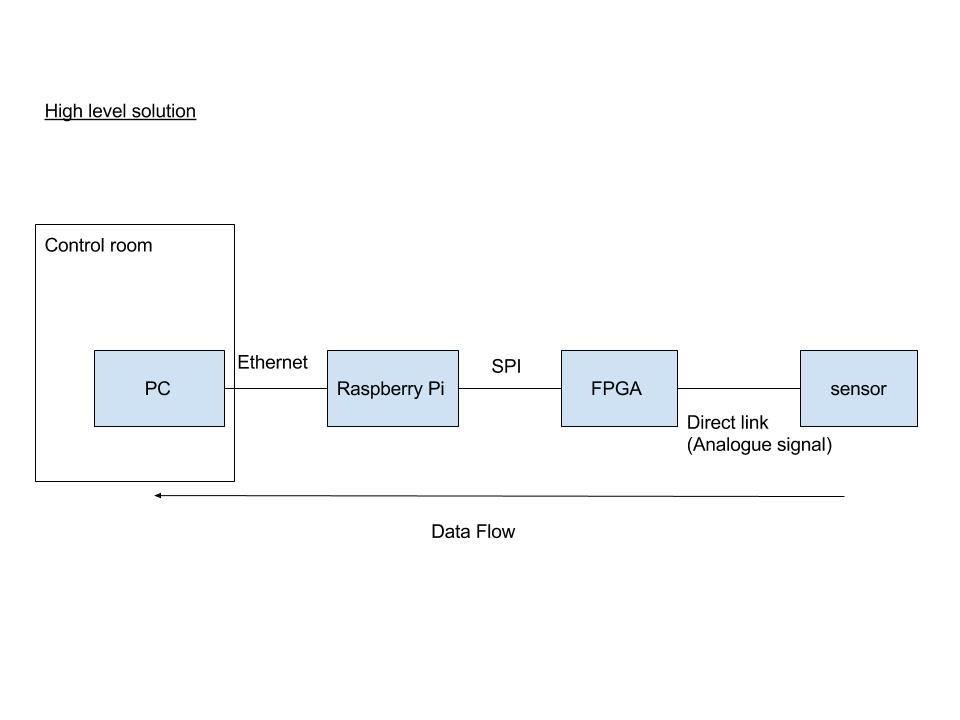
An FPGA was proposed as an alternative to process the signals from the ADC to the Node Controller. An FPGA has advanced timing capabilities. It also uses External ADC, the Pmods.

**The Chosen Solutions for the FPGA/Microcontroller**:

On analysis of the solutions above on the divergent step of the design process, the best solution for implementation was chosen based on the design specifications. This is where we found out that use of a microcontroller to process signals directly from the ADC side is not a best solution due to limitation on timing requirement of a Raspberry Pi. Communicating with the node computer using the fpga via ethernet directly would prove problematic because implementing udp/tcp on FPGA is demanding. To overcome this data will be sent via spi to a Raspberry Pi and then the Raspberry Pi will send the data to the node computer via ethernet.

The Pmods were used for ADC. Nexys 3 and Raspberry Pi 2 were used for implementation of the solution.

**The implemented solution**



*The codes are attached. The solution gave errors that were linked to possible licence issues of the software.*

**Keep Getting Error:**

ERROR:NgdBuild:604 - logical block 'XLXI\_2' with type 'pmodad1' could not be

resolved. A pin name misspelling can cause this, a missing edif or ngc file,

case mismatch between the block name and the edif or ngc file name, or the

misspelling of a type name. Symbol 'pmodad1' is not supported in target

'spartan6'.

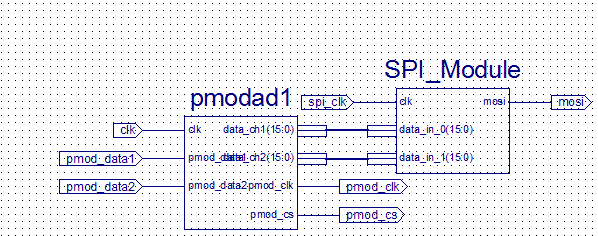
ERROR:NgdBuild:604 - logical block 'XLXI\_1' with type 'SPI\_Module' could not be

resolved. A pin name misspelling can cause this, a missing edif or ngc file,

case mismatch between the block name and the edif or ngc file name, or the

misspelling of a type name. Symbol 'SPI\_Module' is not supported in target

'Spartan6'.



1. Eecs.tufts.edu. (2016). [online] Available at: http://www.eecs.tufts.edu/~dsculley/tutorial/opamps/opamps7.html [Accessed 24 Jun. 2016]. [↑](#footnote-ref-0)