CASPER Memo 017
Packetized FX Correlator Architectures

Peter McMahon, Alan Langman*, Dan Werthimer, Don Backer, Terry Filiba, Jason Manley, Aaron Parsons and Andrew Siemion

September 2007

Abstract

We outline several possible architectures for packetized FX correlators that use Ethernet switches for the interconnect between F engines and X engines. We highlight advantages and disadvantages of each.

Introduction

For large N dish arrays, correlators are constructed by first channelizing input from each antenna, and then performing the cross-multiplications on each frequency channel produced. We refer to each channelizer as an "F engine", and each multiplication engine as an "X engine". CASPER Memo 003 [1] outlines Aaron Parsons' development plans for an 8-station FX correlator, built using IBOB and BEE2 boards. For a broader overview of CASPER's signal processing plans, including correlators, see ref. [2].

Although the architectures presented in this memo are intended to be reasonably general, we have made several assumptions. Most importantly, we assume that F engines will produce data at a rate less than 10Gbits/sec, and that X engines will consume data at a rate less than 10Gbits/sec. Unless otherwise specified, we assume that the number N of F engines is equal to the number of X engines, although we do this primarily for convenience, and there is no reason why the architectures presented cannot be extended to use a number of X engines different from the number of F engines. We also do not discuss correlator output, which has quite varied requirements depending

*Karoo Array Telescope

on the application, and regardless doesn't affect the interconnect architectures we describe. In general, output can be facilitated by using 10GbE or 1GbE connections directly from the X engines to a data storage machine, or by using 10GbE or 1GbE connections from the switches already used in the interconnect architectures, to a data storage machine.

Another important point to note is that all architectures in this memo are discussed under the assumption that signals from dishes/antennas are single polarization only. However, extending the architectures to the dual polarization case is simply a matter of treating the two polarization signals from each dish as single "logical" signals, and using full-Stokes X engines.

Our architectures have been considered with the Berkeley Emulation Engine 2, the Internet Break-Out Board (IBOB) and its successor, the ROACH board, as the target hardware in mind. When we discuss Architecture 3, we give a specific example of how the architecture can be mapped to ROACH hardware. It is likely that future correlators built by CASPER will use this architecture with ROACH boards in a similar way to that which is presented.

Architecture 1: 10Gbit Ethernet

Aaron Parsons's correlator, described in ref. [1], is based on an architecture that uses a single 10GbE switch as its interconnect mechanism between F engines and X engines. We now describe the most basic form of this architecture. A single 10GbE port on each F and X engine is connected to a single port on a 10GbE switch. Figure 1 illustrates the architecture. Each F engine sends frequencies $f_0, f_N, f_{2N}, \ldots$ to X engine 0, frequencies f_1, f_{N+1}, f_{2N+1} to X engine 1, through frequencies $f_{N-1}, f_{2N-1}, \ldots$ to X engine $N-1$. This can be accomplished by having each F engine create N 10GbE packets, one for each X engine, and address them appropriately so that the switch will deliver the packet containing frequencies $f_0, f_N, f_{2N}, \ldots$ to X engine 0, and so on.

This architecture is simple, and requires only one switch and a small amount of wiring. The switch needs $2N$ 10GbE ports.

Architecture 1.1: 10Gbit Ethernet, with better efficiency

Architecture 1 is conceptually sound, but it is somewhat wasteful. We assume that 10GbE ports are expensive, and thus wish to maximize the utility of

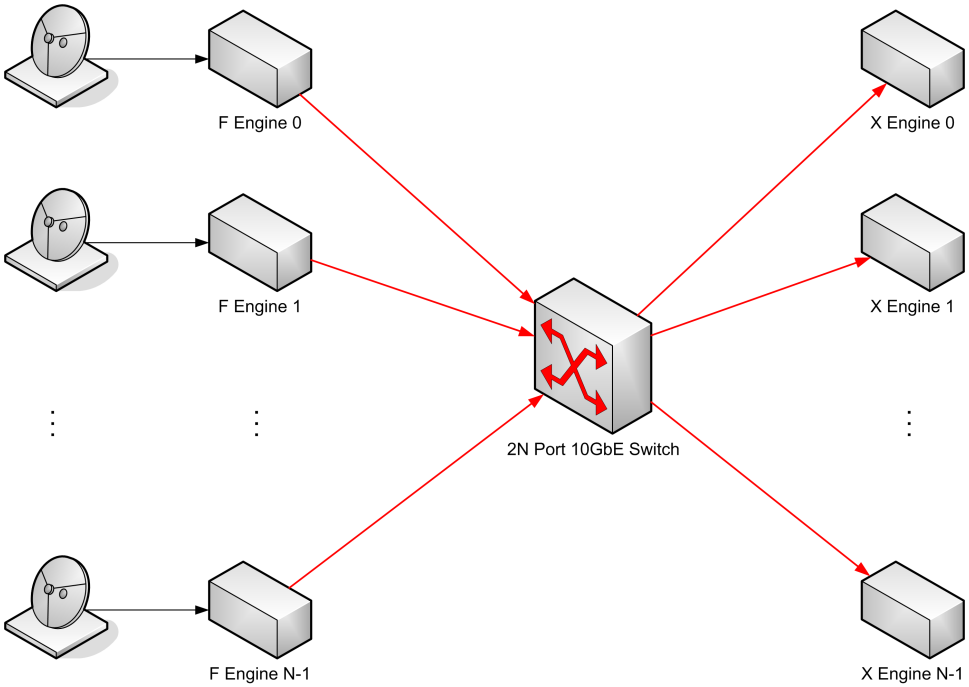

Figure 1: 10Gbit Ethernet.

each port on a switch that we purchase. In Architecture 1, we note that each connection has the capability of sending and receiving 10Gbps. However, each connection to the switch uses only 10Gbps in one direction - therefore only half the bandwidth that the switch is capable of, per port, is being used.

Given that the X engines have more than one 10GbE port (as is the case with the Parsons correlator, which uses the BEE2 board), we can build a correlator using a switch with only N ports (whereas Architecture 1 requires a $2N$-port switch), and each port is used to send and receive data at up to 10Gbps. Figure 2 shows this architecture.

The F engines now each send their output directly to a single X engine. Each X engine then creates the packets to distribute the frequency channel data to all the X engines (just as described for Architecture 1).

The use of 10GbE to transfer data from the F engines to the X engines is unnecessary, since there is only one endpoint, so in practice the XAUI protocol is used.

This architecture is also simple, and is no more complicated than Archi-

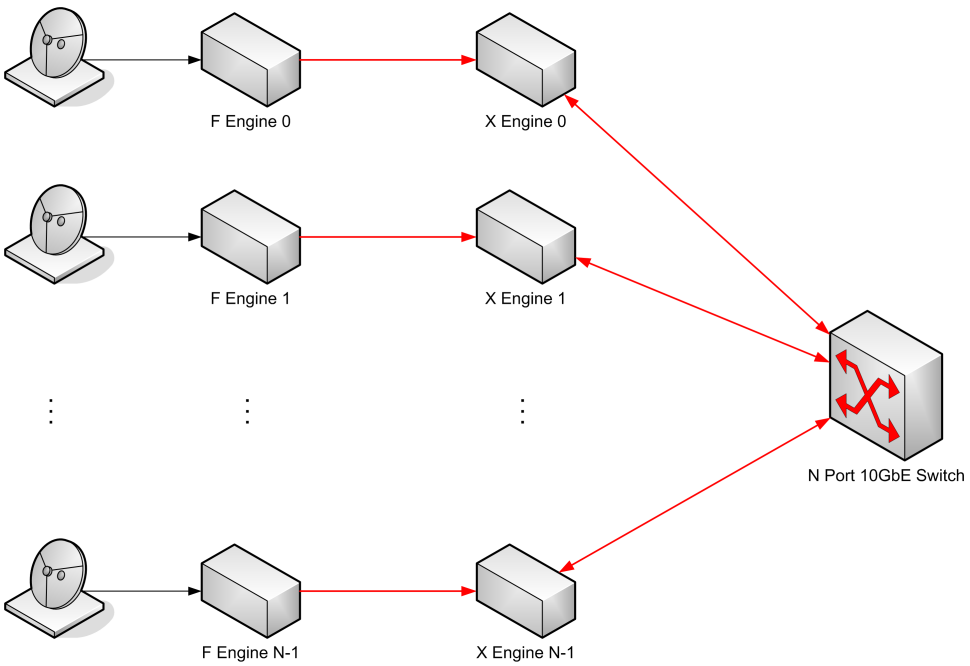

Figure 2: 10Gbit Ethernet, with better efficiency.

tecture 1 to implement, but takes full advantage of the capabilities of 10GbE switches, and in so doing reduces the number of ports required.

We assume that 10GbE switches are going to continue to be produced with ever-increasing numbers of ports. This is important to scale correlators using this architecture up to large N. This assumption appears to be reasonable: suitable mainstream commercial switches with 20 ports are readily available, and 10GbE switches with 256 ports, and Myrinet switches with thousands of ports, have been built. Current market pricing, perhaps surprisingly, is such that the price-per-port does not vary widely between switches with small and large numbers of ports.

In the Parsons correlator implementation, a single IBOB has two station inputs (each dual polarization), but only one XAUI output, so a 32-station dual-polarization correlator can be built using a 16-port 10GbE switch. To scale up, a 64-station correlator will require a 32-port switch, and so on.

In the event that suitable 10GbE switches with large numbers of ports (> 20) are not available, or become prohibitively expensive, Architecture 1.1 is no longer viable. It is not possible to scale up by stacking 10GbE switches

to effectively produce a larger switch, since the bandwidth between switches is limited to 10Gbps, which is not enough. As an example, suppose that the largest 10GbE switch you can purchase has 17 ports, and you wish to build a 32-station correlator (with one station connected to one F engine). You can purchase two switches, and connect them together using one port on each switch. You now have 16 free ports on each switch. Connect X engines 0 through 15 to switch 0, and X engines 16 through 31 to switch 1. X engine 0 outputs 10Gbps of data, half of which is addressed to nodes on switch 1. X engine 1 also outputs 10Gbps of data, and also needs to send 5Gbps of data to nodes on switch 1. Likewise for X engines 2 through 15. However, you can see that since the switches only have a 10Gbps connection between them, that connection becomes a bottleneck and severely restricts the performance of the correlator. In this example, the switch connection has 1/8th of the bandwidth that was required to maintain the same level of performance as the single-switch solution.

Architecture 2: 10Gbit/1Gbit Ethernet Hybrid

Architecture 2 was conceived as a contingency plan in the event that cost-effective large 10GbE switches that can meet the performance requirements for CASPER correlators do not materialize.

We make two observations: first, we note that switches with few 10GbE ports and many 1GbE ports (which we call 10GbE/1GbE switches) are widely available, and second, we note that in a correlator with > 10 F engines, no F engine needs a bandwidth of greater that 1Gbps to any individual X engine.

It is therefore possible to construct a suitable interconnect between F engines and X engines using multiple 10GbE/1GbE switches. Figure 3 illustrates such an architecture.

This architecture has the additional advantage that it can be scaled up by stacking 10GbE/1GbE switches. Currently commercial switch offerings include switches with four 10GbE ports and 48 1GbE ports. By using just one such switch per engine (96 switches in total), it is possible to build a 48-station correlator. However, you can connect two 10GbE/1GbE switches together using one of the four 10GbE ports. In this way, you can double the

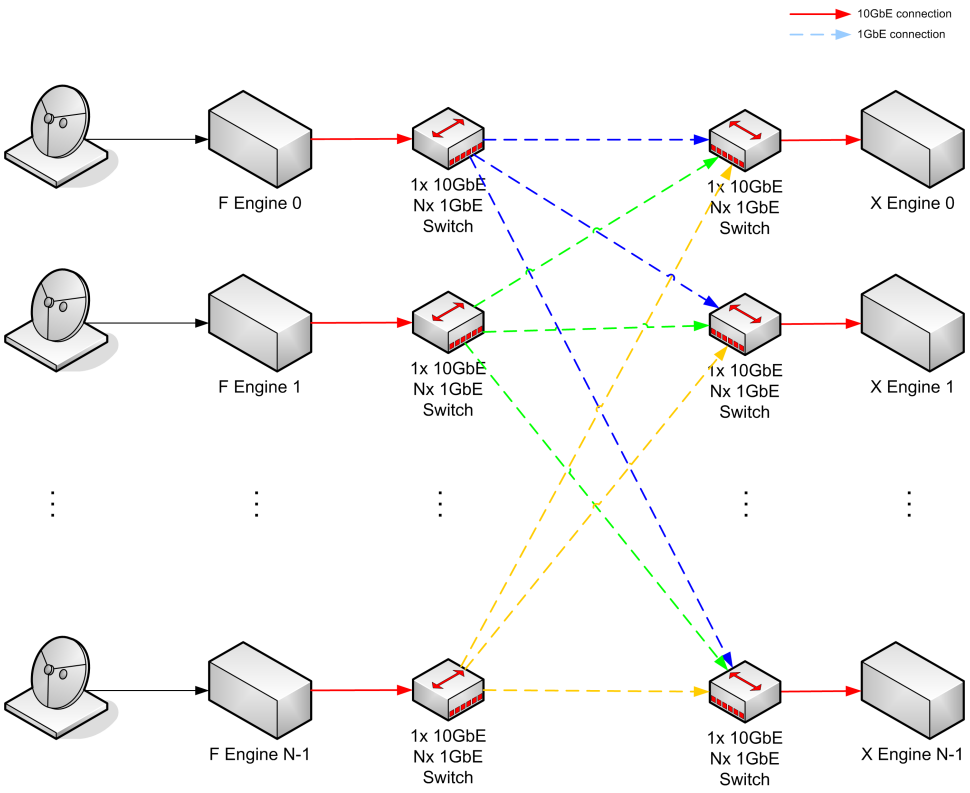

Figure 3: 10Gbit/1Gbit Ethernet Hybrid.

size of the correlator. You can connect four 10GbE/1GbE switches together by connecting each of three switches to a fourth switch's first three 10GbE ports. Its fourth port can then be connected to an engine. In this way a 192-station correlator can be built using switches with four 10GbE ports and 48 1GbE ports. It may be possible to stack the switches in multiple layers (using the three free 10GbE ports on each of three switches per engine), in which case an even greater number of stations can be supported.

The major disadvantages to this architecture are that a large number of switches is required ($2N$ switches for an N-station correlator, where each switch has N 1GbE ports; the latter requirement can be relaxed if more switches are used and are stacked), and that it is not immediately clear that the performance of a system built using this architecture in practice would be satisfactory. Specifically, it is important that the latency in the interconnect not vary excessively from node to node, and that packets not be excessively reordered. Another practical note is that switches should be programmable to allow their routing tables to be fixed, lest situations arise where packets take non-optimal routes (i.e. routing via multiple switches instead of just one) and hence cause excessive latency variance.

Architecture 2.1: 10Gbit/1Gbit Ethernet Hybrid, with fewer switches

We use a strategy similar to that which was used to halve the number of ports required on the 10GbE switch in Architecture 1 to halve the number of switches required in Architecture 2. Figure 4 illustrates Architecture 2.1, which takes advantage of the fact that two 10GbE unidirectional links can be combined into one bidirectional link. There is also a reduction in the number of 1GbE connections required, since two unidirectional 1GbE links can be combined into one bidirectional link.

Architecture 3: Multiple Per-Engine 10Gbit Ethernet for GHz Bandwidths

Correlators for upcoming telescope developments at meerKAT, GMRT and elsewhere are expected to process 1GHz bandwidth signals. The Nyquist sampling rate is thus 2GSa/sec, and with 8-bit ADCs, we can expect per an-

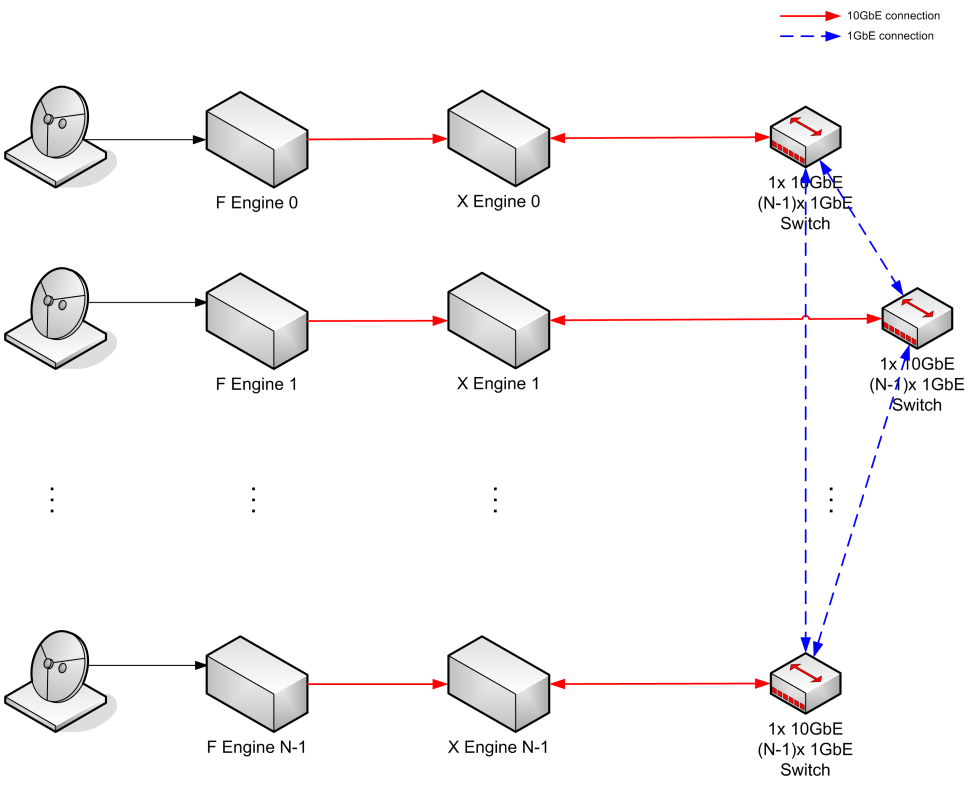

Figure 4: 10Gbit/1Gbit Ethernet Hybrid, with fewer switches.

tenna polarization data rates of 2GB/sec and upwards. The 10GbE standard cannot be used to transport data at these rates in a single cable, therefore the data needs to be divided amongst multiple 10GbE connections.

One obvious way to divide a signal up is by frequency channel. For example, we can take a 1GHz input signal, digitize it, channelize it, and output the channelized data by sending half the frequency channels over one 10GbE connection, and the other half over another 10GbE connection.

We can apply this idea to an N antenna correlator design in the following way: if each F engine has its output divided into s sets of frequency channels, you build s separate sets of X engines, and each set operates independently of the others. The first set of X engines performs the cross-multiplications on the first set of frequencies from each F engine, the second set of X engines performs the cross-multiplications on the second set of frequencies from each F engine, and so on.

Figure 5 shows an overview of this type of correlator design where $s = 4$, the bandwidth is 2GHz per input signal, each F engine is implemented on a separate board with only one signal input per board, and the number of frequency channels is 1024.

Figure 6 shows the architecture in more detail. F engine n sends its ith set of frequency channel data to the nth X engine in the ith set of X engines.

This architecture has the advantage that it allows one to process high bandwidth signals on a single board, with a single ADC. Other solutions, such as performing analogue "division" of the bandwidth and using multiple F engine boards per antenna, are less elegant in our opinion, especially when one considers that modern FPGAs can perform the F engine calculations for 2GHz bandwidths on a single chip.

The disadvantage of this architecture as presented in figures 5 and 6 is that it uses s sets of X engines, rather than just one set, and hence for some lower bandwidth applications may require an unnecessarily large number of X engines. This may also be the case if X engines are built with boards that can easily process bandwidths far larger than what 10GbE can carry (at least double). In these instances the presented architecture is unsuitable, and you may wish to instead have a single set of X engines, and connect all s sets of frequency channels into a single X engine. With this setup, only the interconnect from the F engines to the X engines is divided into sets of

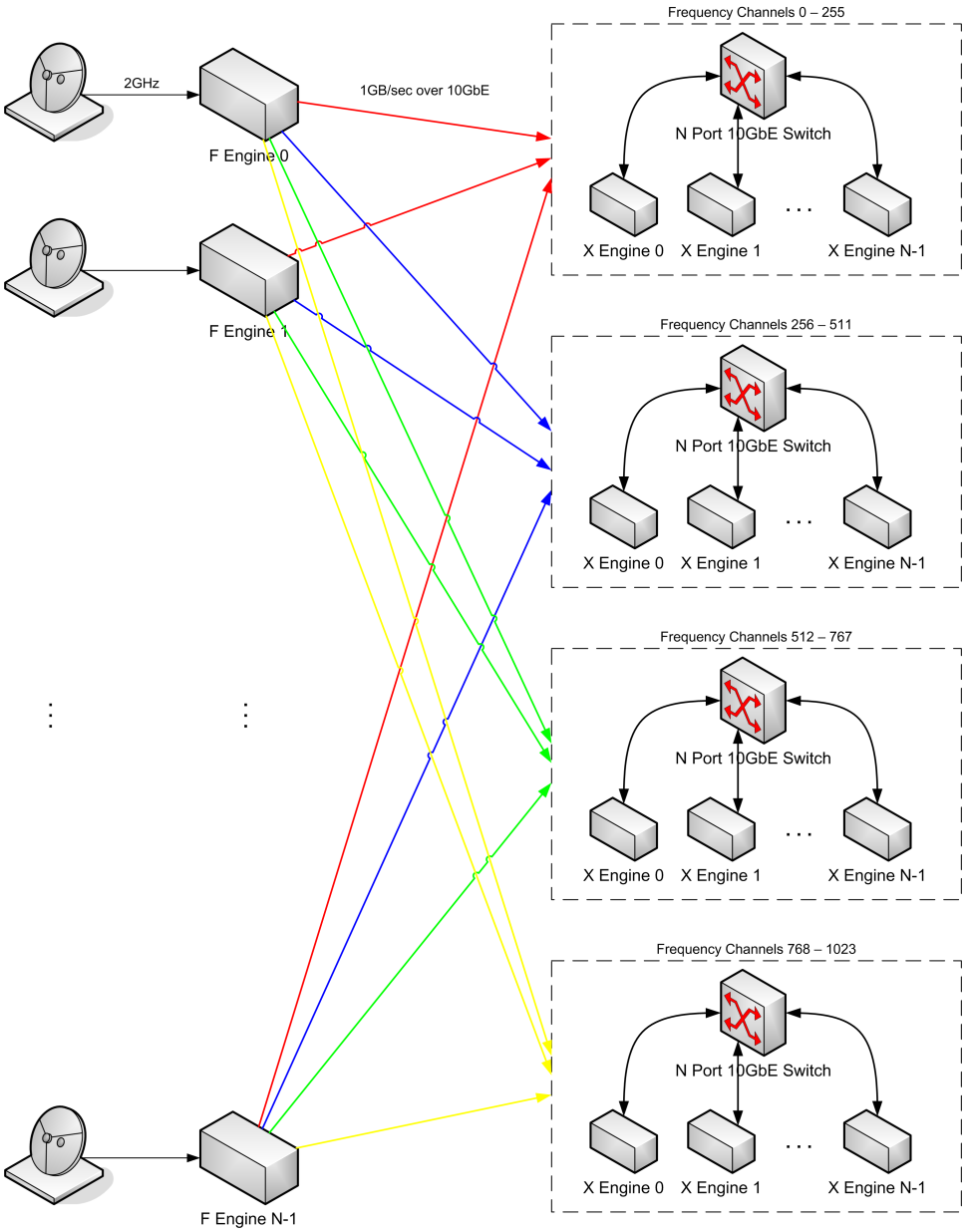

Figure 5: Multiple Port 10Gbit Ethernet (overview).

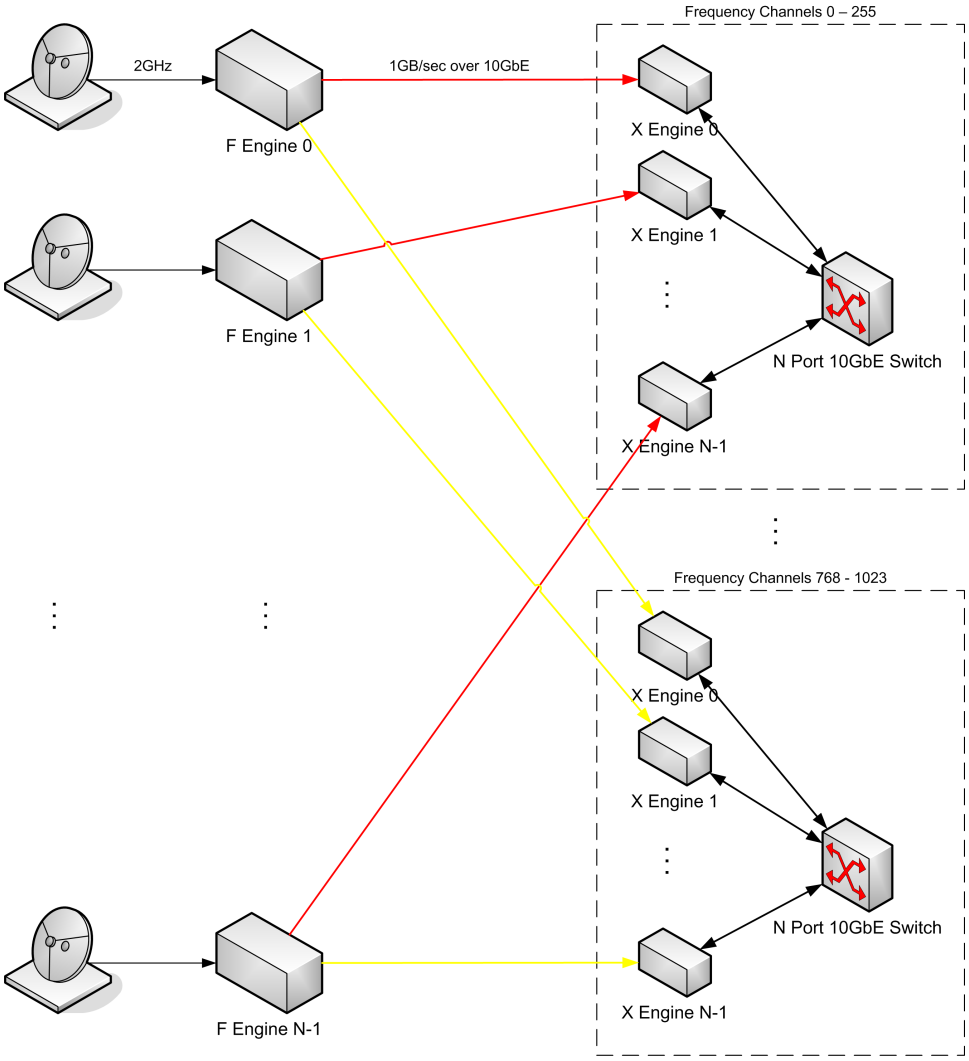

Figure 6: Multiple Port 10Gbit Ethernet (detail).

frequency channels, and the remainder of the architecture remains the same as Architecture 1.1.

Architecture mapped to implementation with ROACH for 1GHz bandwidths

Architecture 3 can be fruitfully used with signals that have only 1GHz bandwidth. The ROACH board has four 10GbE-capable CX4 connectors, and two ZDOK connectors to attach ADC boards. For applications that only require 1GHz bandwidths, the ROACH board can be used to implement F engines for two 1GHz signals on a single board. The X engines can be implemented by using a separate ROACH board for each X engine, since each X engine requires at least two 10GbE connections – one to an F engine, and one to a 10GbE switch.

There are multiple possibilities for implementing a scheme to output the correlator data. If the output datarate from each X engine is sufficiently low, it may be possible to simply add one port to each 10GbE switch, and connect this port to a data storage machine in some fashion. In this scheme, the output data is transported from an X engine to the switch over the same cable that it uses to transmit incoming F engine packets to the other X engines via the switch. This will only be possible if the data rate from an X engine to its neighbour X engines with incoming data is low enough that the sum of that data rate, and the output data rate per X engine is less than 10Gbps.

If the data rates in the interconnect 10GbE links are high enough that the existing link between an X engine and the switch cannot be used to transmit the output from an X engine as well, then an additional link may be required. The ROACH board has four CX4 connectors. Figure 6 shows that two connectors per X engine ROACH board are used – one for a 10GbE connection from an F engine, and another for a 10GbE connection to a switch. A third 10GbE link can be added, and this can be connected to the data storage machine in some fashion (possibly via another connection to the switch the X engine is already connected to, or via another, new 10GbE switch, or via a 10GbE/1GbE switch). Another possibility for output is to use the 1GbE port on the ROACH board – this can be connected to a standard 1GbE switch, or to a 10GbE/1GbE switch, or perhaps directly to the data storage machine.

Figure 7 shows how two 1GHz input signals can be processed on a single ROACH board, with 1024 channels used as an example. In this scheme, the

bandwidth is again divided into four. The frequency channels from both polarizations are concatenated, so the first output contains both frequency channels 0 – 255 for polarization 1, and frequency channels 0 – 255 for polarization 2.

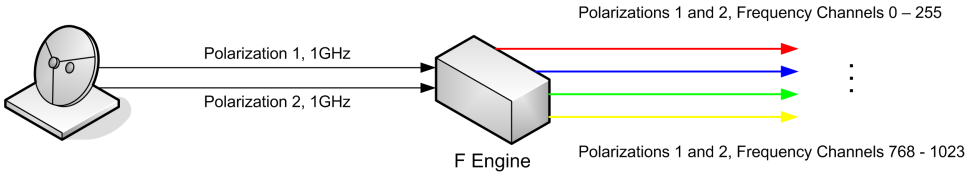

Figure 7: Multiple Port 10Gbit Ethernet for 1GHz bandwidth on ROACH.

An alternative scheme is to have each ROACH board output the data for each polarization separately – in this scheme, each polarization's bandwidth is only divided by two. This is shown in Figure 8.

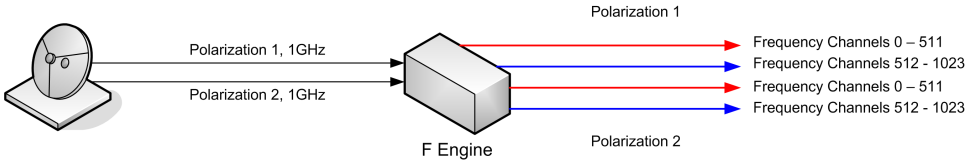

Figure 8: Multiple Port 10Gbit Ethernet for 1GHz bandwidth on ROACH alternative.

Architecture 4: Multiple F Engines per 10Gbit Ethernet Connection for Lower Bandwidths

Although CASPER's attention is primarily on developing technology for new high-bandwidth radio telescopes, there are still a significant number of projects that require only lower bandwidth signals, such as PAPER [3] (which, incidentally, will use a CASPER correlator).

A correlator for 250MHz bandwidths will require a sampling rate of 500MSa/sec. If 8-bit sampling is performed at the Nyquist sampling rate, than the data rate per F engine will be 500MB/sec. In the architectures we have already presented, we assumed that we could create X engines that can each process data at a rate of 1GB/sec. We note that we can build a

lower bandwidth correlator with an architecture that results in substantial resource savings by taking advantage of the fact that two F engines can feed data to a single X engine. Therefore a low bandwidth N antenna correlator can be built using N low bandwidth F engines, and $N/2$ X engines. Figure 9 shows an overview of an architecture for a lower bandwidth correlator based on Architecture 1.1.

As is shown in the figure, this architecture uses an $N/2$ port 10Gbit Ethernet switch. The resource saving of using this architecture for a lower bandwidth correlator is a reduction in the number of X engines by a factor of 2, and consequently a reduction in the number of ports on the 10Gbit Ethernet switch by a factor of 2.

The extension from Architecture 1.1 to Architecture 4 is fairly natural. Specifically, in architecture 1.1, each F engine effectively sends frequencies $f_0, f_N, f_{2N}, \ldots$ to X engine 0, frequencies f_1, f_{N+1}, f_{2N+1} to X engine 1, through frequencies $f_{N-1}, f_{2N-1}, \ldots$ to X engine $N-1$. In this architecture, each F engine must effectively send double the number of frequencies to each X engine. For example, we may choose to send adjacent frequencies, where each F engine sends frequencies $f_0, f_1, f_N, f_{N+1}, f_{2N}, f_{2N+1}, \ldots$ to X engine 0, frequencies $f_2, f_3, f_{N+2}, f_{N+3}, f_{2N+2}, f_{2N+3}, \ldots$ to X engine 1, through frequencies $f_{N-2}, f_{N-1}, f_{2N-2}, f_{2N-1}, f_{3N-2}, f_{3N-1}, \ldots$ to X engine $(N/2)-1$. As is depicted in the diagram, the F engines distribute their output frequencies via an X engine, in a similar fashion to that which was described in the summary of Architecture 1.1. However, in architecture 4, two F engines feed into a single X engine, as opposed to the one-to-one mapping in Architecture 1.1. From this view it is easy to see how the data rate is increased from, for example, 500MB/sec through each F engine to 1GB/sec through each X engine.

Architecture mapped to implementation with ROACH for 250MHz bandwidths

Architecture 4 can be efficiently mapped onto hardware such as the IBOB, BEE2 or ROACH. We briefly present a mapping to ROACH, but any hardware platform that has two 10Gbit Ethernet ports (such as CX4 ports) per device, and an FPGA capable of performing F and X engine calculations at a sustained cumulative rate of 1GB/sec should be sufficient.

Figure 10 shows how Architecture 4 may be mapped to ROACH in the

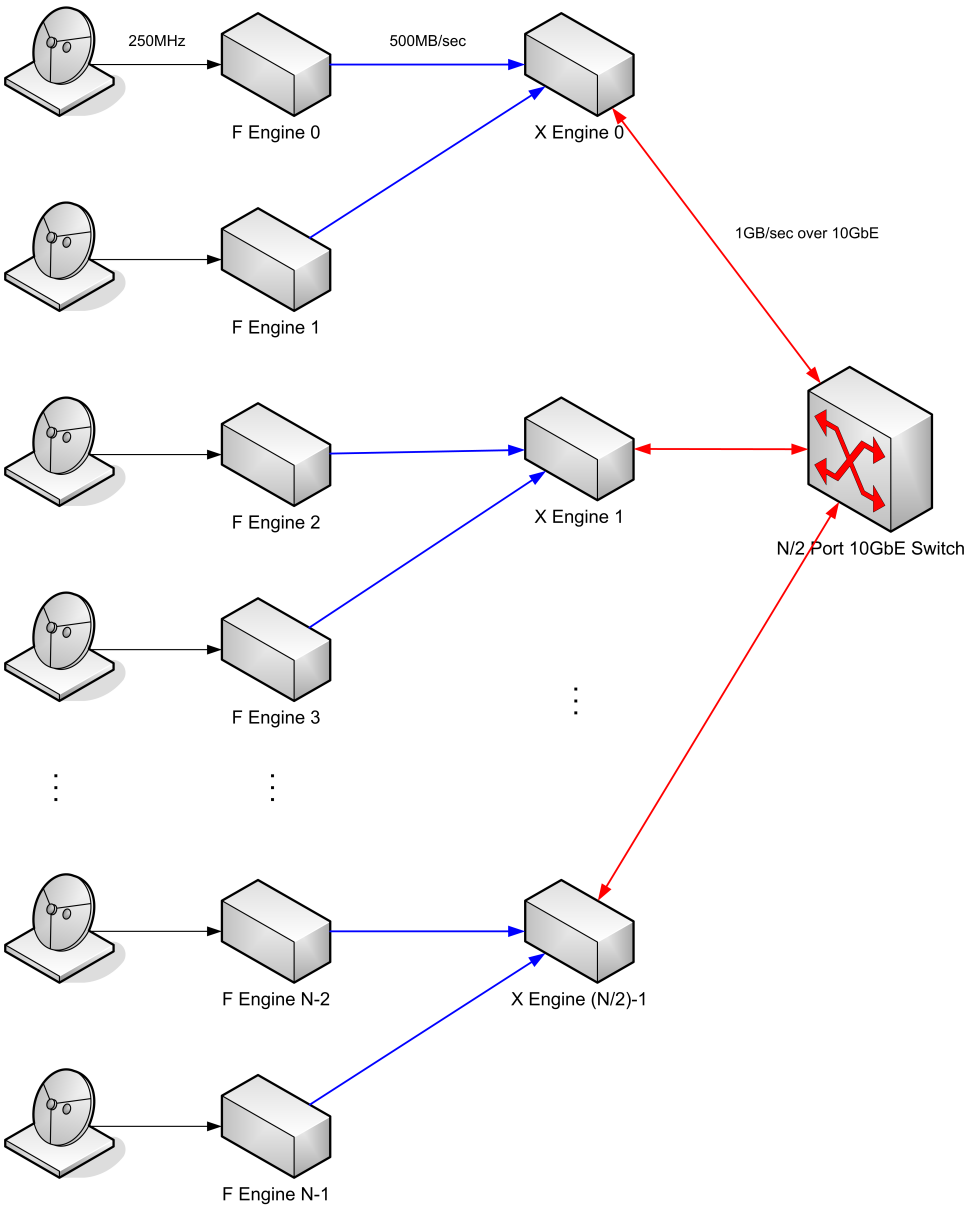

Figure 9: Lower bandwidth architecture using 10Gbit Ethernet.

case of an N antenna system, with 250MHz bandwidth per antenna. We expect that a single ROACH board will be able to perform F engine processing of at least 500MHz bandwidth data. Therefore we anticipate that a single ROACH board should be able to implement two F engines for 250MHz input bandwidths. The data from the two F engines on a single ROACH board is multiplexed to produce a single 1GB/sec output data stream, which is sent to a ROACH board that implements a single X engine over 10Gbit Ethernet. Therefore such an N antenna correlator will require only $N/2$ ROACH boards to implement F engines, and $N/2$ ROACH boards to implement X engines.

References

[1] Aaron Parsons. CASPER Memo 003: Correlator Development Plans. April 2006.

[2] Aaron Parsons et. al. PetaOp/Second FPGA Signal Processing for SETI and Radio Astronomy. *Proc. 10th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2006.

[3] Richard Bradley, Don Backer, Aaron Parsons, et. al. PAPER: A Precision Array to Probe the Epoch of Reionization. *American Astronomical Society, 207th Meeting* (Poster), 8 – 12 January 2006.

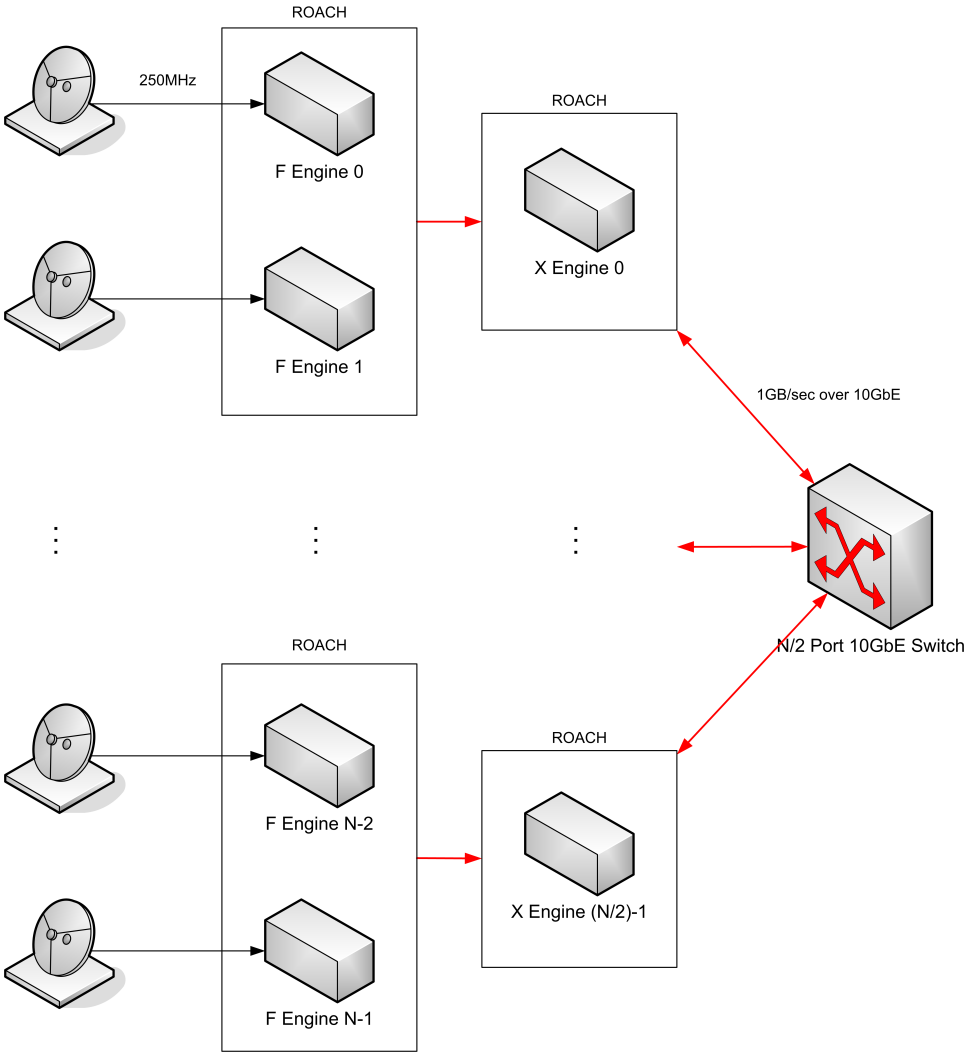

Figure 10: Lower bandwidth architecture using 10Gbit Ethernet, mapped to ROACH boards.