

ATA Memo No. 43
Digital Processing Architecture of the ATA

Larry R. D'Addario
2002 February 15

At the preliminary design review of the ATA digital subsystem [1], a “baseline” design [2] that satisfies the identified requirements [3] was presented. Since then, considerable discussion of alternatives has occurred, important additional requirements have been introduced, and further study of available components and technologies has been accomplished. This report is an attempt to summarize the changes to requirements and technologies and to recommend a specific architecture for implementation in the production test array (PTA).

New Requirements

Rapid tracking of gain magnitude

In the baseline design [2], each beam is provided with separate and independent phase tracking for each antenna. The phase is adjusted for every data sample at a programmable rate, and the rate may be updated at least 1000 times per second by a local microprocessor. This is accomplished by a DDS-style phase generator driving sine and cosine lookup tables to generate a complex gain value, which is then multiplied by the current complex data sample. In this design, the magnitude of the complex gain can be varied by re-loading the lookup table so that all entries are scaled by the desired magnitude. Provisions are made for doing this from the local microprocessor, but it requires that the magnitude be updated far more slowly than the phase. The arrangement is more than sufficient for accurate tracking of a target source, even if the source is a low earth orbit satellite and even in the case of an expanded (3 km) array. The gain magnitudes can be different among the antennas, thereby allowing some beam shaping, but it was expected that they can remain fixed for a given observation.

It has now been suggested that more general tracking capability is desirable, where the full complex gain (phase and magnitude) can be varied rapidly. It would then be possible, using known algorithms, to generate an array beam with its peak tracking a target source and simultaneously with a null tracking an interfering source. However, due to the fact that the delay tracking is still in the direction of the target, the null would be effective only over a narrow bandwidth.

We therefore take it as a requirement that rapid tracking of the full complex gain be provided, although limiting the magnitude variation to about $\pm 10\%$ is probably acceptable.

Phase Switching

At least for the PTA, it has been decided to implement 180-deg phase switching at the RF Converter. This will be done in either the first or second LO. The sign reversal must then be synchronously removed in the digital system. It is best if this is done at the Digitizer, since then the switching waveform generator might be common to the RF Converter and Digitizers of each antenna.

Technology Changes

Since the PDR, changes have occurred in available serializer-deserializer chip sets; in the availability of inexpensive VCSEL arrays for optical transmitters; and in FPGAs.

Serializers and Deserializers

Devices designed for speeds near 2.5 Gb/s have been readily available for several years, but many new ones have been introduced in the past year. It turns out that the newest offerings are by far the least expensive, so it is necessary for us to adopt one of these in spite of their having some undesirable characteristics for our application. Also, the newest devices tend to be implemented in CMOS, which has far less power dissipation than other technologies; some older devices used GaAs.

A spreadsheet printout listing selected specifications of known devices is given in Table 1.

Devices are now available for operation over a wide range of line speeds, e.g. from 1.6 to 2.7 Gb/s or from 2.5 to 3.125 Gb/s, by locking to the user's clock. This allows us considerable design freedom compared

with being tied to an industry-standard rate like OC48 (2.48832 Gb/s). The down side is that these wide-range devices are expected to have much worse clock jitter than narrow-range ones; the latter are usually designed to comply with SONET jitter specifications, whereas the former have no explicit specifications. To overcome this, the receiver's PLL must have large bandwidth, but in turn this means that it requires a higher transition density in order to maintain lock—it cannot tolerate long runs of 1s or 0s. SONET devices often have run length tolerance of hundreds to thousands of bits. The others typically rely on 8b/10b encoding (25% overhead) to ensure low run length, but this is not necessary for us (as explained in [4]).

The least expensive devices are almost always built as transceivers, i.e. with a serializer and deserializer in the same device (known as a SERDES), usually operating independently. We have no use for a transmitter and receiver in the same place, so adopting one of these requires that half of it be idle. More board space is needed because of the larger pin count, even though many pins are unused. Although this is wasteful, all of the known transmit-only/receive-only chip sets are far more expensive, so we must accept the inefficiency.

There are a few quad devices available, with 4 transceivers per chip, at very attractive prices per channel. But all of those so far identified have low multiplexing factors (4 or 10), so that their parallel-side clock is > 300 MHz. This is too high for our FPGA processing, so we would have to provide additional levels of mux/demux, along with word alignment circuitry, in order to use them.

Some of the chips have built-in circuitry for word alignment on the receive side. This always requires transmission of a known bit pattern, and it is usually the user's responsibility to generate this on the transmit side when initialization or re-alignment is desired. There are two industry-standard patterns in common use: one is from SONET and is known as A1/A2, and the other is a special 10b character from the 8b/10b code. Chips that recognize the 10b character are more common, but SONET chips are also available. Unfortunately, all manufacturers have apparently used an implementation that produces an uncertainty in latency, as discussed below.

Unstable Latency in High Speed Links

We have come to have a more detailed understanding of the operation of the SERDES devices, including both older and newer ones, that reveals a significant difficulty. On the receive side, deserialization necessarily involves dividing the line clock frequency by the multiplexing factor; the phase of this divider is initially underdetermined, producing two effects: underdetermined phase of the word clock at the receiver's output, and undetermined alignment of the bits of each word. The word alignment can be corrected upon detection of a pattern whose alignment is known. It is possible to implement this by resetting the divider phase or by causing the divider to skip an appropriate number of clocks, thus also removing its indeterminacy. Such a system, once aligned, would have a fixed latency between the clocking in of a given word at the transmitter and the clocking out of the same word at the receiver. But this implementation is technically difficult because it involves manipulating circuitry (the divider) that operates at the multi-Gb/s line speed. For chips that do not have word alignment built in, external word alignment circuitry has no access to the fast components. For those that do have it built in, the manufacturers have apparently chosen to let the fast circuits free-run and to implement the alignment in a way that makes the latency even less well determined.

There are some devices that also include a divider of indeterminate phase on the transmitting side, and there are some that also include FIFO buffers in the transmitter and/or receiver where the buffer latency is variable. These features increase the indeterminacy of the overall latency.

Such uncertainty is perfectly acceptable in most communications applications, where all channels are independent. For us it is not, since we must combine the signals from many antennas where their mutual timing is crucial. The absolute latencies need not be known nor equal among antennas, since they are included in astronomical calibration, provided that each is stable. The problem is that any power cycle, temporary loss of signal, or re-initialization is likely to produce a latency that is different from before.

To overcome this difficulty, it is necessary for us to add a FIFO after the deserializer and to adjust the delay through this FIFO during an alignment sequence so as to produce a stable overall latency. This requires that we provide a stable reference path in hardware, such as by distributing a periodic signal to all modules (transmit and receive sides) where the period is greater than the largest possible latency variation. Given the need to do this, we may be better off including the word alignment feature in this circuitry that we design, rather than relying on the built-in word alignment of some chips. This gives us a larger selection of suitable chips and provides us with better control of the process, at at cost of some additional engineering time.

Optical Transmitters and Receivers

At the time of the PDR, several companies (Agilent, Molex, and Zarlink) were offering 12-channel arrays of 2.5 Gb/s optical transmitters and matching arrays of optical receivers. The transmitters were to be based on VCSELs (vertical cavity lasers) and included integrated drivers accepting differential logic. The receivers were to include integrated buffers with differential logic outputs. In view of the multi-channel integration, the price per channel seemed very attractive.

In fact, none of the companies can actually deliver these products. It is possible that some will be available later in 2002, but no firm prices are available and the estimated prices are substantially higher than the quotes I received in mid-2001.

The most suitable and economical products actually available now seem to be individual VCSELs and optical receivers from Honeywell. It is possible to get these packaged as dual units (two VCSELs or two receivers per package). The VCSELs are simply diodes, so they need an external driver. Nevertheless, the total cost of an optical channel (driver IC, VCSEL package, receiver package) is lower than the per-channel cost of the arrays considered earlier.

Additional data is given in Table 1.

FPGAs

At the time of the PDR, the Xilinx Vertex-II series and Spartan-II series of FPGAs were under consideration for the main signal processing. Further study of the logic requirements (see a later section of this report) shows that either a large and expensive Vertex-II or several Spartan-II devices would be needed for each IF Processor module. There is uncertainty about whether the Spartan-II devices will operate at the desired clock rates. Meanwhile, another series called Spartan-IIe has been introduced. It is available in somewhat larger size than Spartan-II, appears to be faster, and includes some new and useful features (like direct support of differential I/O). These devices should be considered.

Furthermore, other FPGA manufacturers, especially Altera, have devices that may be better than Xilinx devices for our purposes. This is especially true if faster I/O is needed, such as to support SERDES chips with less multiplexing and thus higher parallel-side clock rates, or to reduce the number of backplane connections between boards for the phased-array summation chain.

Alternatives

Although various options differing in detail have been considered, it is now possible to reduce them to two serious possibilities. The difference is primarily in the way that two fundamental difficulties of digital delay tracking are addressed: first, the straightforward variable delay line has resolution that is too coarse, namely one sample. This leads to loss of gain for frequencies away from the center of the channel. Second, if the delay tracking (however fine or coarse) is common to all beams, then any beam whose direction is different from that of the delay tracking center suffers chromatic aberration. This is a substantial effect for the ATA, even within the primary beam of the antenna.

In the baseline architecture, both problems are avoided by following the common coarse delay with a filter bank that breaks the channel into narrow subchannels. Phase-only tracking is then applied to each subchannel, separately for each beam. Among the subchannels of one beam, the relative phase is linear with frequency, and may be understood as synthesizing a time-varying correction to the coarse delay. For directions within the primary beam, 16 subchannels are sufficient to make the residual gain loss and aberration negligible. This approach is now known as the frequency domain method, or F architecture.

The main alternative involves reducing the coarseness of the variable delay by a combination of oversampling and FIR interpolation. This eliminates the gain loss due to the delay error. By implementing separate delays for each beam, chromatic aberration is also eliminated; if all of the coarse delay is implemented separately, then each beam can be anywhere in the sky, not just within the primary beam. No filter bank is necessary. Phase tracking is still necessary because of fringe rotation. This is known as the time domain method, or T architecture.

To provide a concrete and fair comparison of the two methods, consider designs that achieve the same bandwidth. Such a pair of designs is summarized in Table 2. The two designs are designated T1.5, which

Table 2: Two Strawman Designs

	T1.5	F16
Nominal bandwidth	103.68	103.68
Sampling rate	155.52	103.68
Quantization: nom/eff bits	8/7	7/6.5
Data rate/chan, digitizer to IFP	2488.32	1658.88
Optical links per antenna	8	6
Processing rates		
IFP input	155.52	103.68
IFP output; back ends	103.68	127.61
Data rate/chan IFP to BE	1658.88	2041.70
Optical links per antenna	6	8

(All rates in MHz or Mb/s.)

denotes the T architecture with oversampling factor of 1.5; and F16, which denotes the F architecture with 16 subbands. In each case, the bandwidth per channel is 103.68 MHz.

Block diagrams of the two approaches are shown in Figures 1 and 2. The F16 design was described in some detail in the PDR report [2], so the present discussion will concentrate on the T1.5 design.

The 103.68 MHz bandwidth results from using one OC48 optical link (2488.32 Mb/s) for each IF channel on the Digitizer to IF Processor connection for the T1.5 design. The corresponding F16 design uses less bandwidth since it is not oversampled, and this allows a higher level of multiplexing and fewer optical links. On the other hand, the F16 design requires a speedup in the IF Processor, resulting in a higher rate on the IFP to Back End connections. At that point the T1.5 design is back to Nyquist, so it can then take advantage of the higher multiplexing. Since the numbers of Digitizer to IFP connections and IFP to BE connections are nearly equal, the two designs have equal interconnection cost. (Actually, if the F16 design were chosen, we would probably elect to increase the bandwidth and accept a somewhat higher interconnection cost. By running the IFP to BE links at OC48, we would obtain 126.36 MHz bandwidth, as in [2], and we would use 8 optical links per channel for each interconnection stage.)

In the T1.5 architecture of Figure 1, the four beams are fully independent and could be placed anywhere on the sky. If the beams can be restricted to a smaller part of the sky, some saving might be achieved by making part (but not all) of the coarse delay common. For a 1 km array extent, the memory for the coarse delay is small (2048 samples) so it is easy to have fully independent tracking. The array size is limited only by the size of this memory. In the F16 design, the array size is also limited by chromatic aberration and that limit is determined by the number of subbands. By fixing that number at 16, the chromatic aberration loss is limited to about 1% at 1.4 GHz over the primary beam for a 700 m array. At lower frequencies or for longer baselines or for offsets beyond the primary beam, the loss increases.

Both designs provide for complex gain tracking (magnitude and phase) so as to allow null tracking. This turns out not to be difficult or expensive in the T1.5 design. Even for LEO satellites, the complex gain update rate is more than 100 times slower than the sampling rate, so a single complex gain generator can be time shared across many beams and channels; we can therefore make the gain generator somewhat complicated at low cost. It is more difficult in the F16 design because all subchannels must be tracked, making the update rate 16 times larger. Additional details are given in [5].

Optimization of the Oversampling/Interpolation Tradeoff

The main concept of the T architecture is to obtain sufficiently accurate fine delay tracking by interpolation with an FIR filter. A design procedure for such filters is given in [6]. For a given number of taps, it yields a filter that produces minimum mean square interpolation error for a flat-spectrum noise signal covering frequency range $(0, \alpha B)$ where B is the Nyquist bandwidth and $0 < \alpha < 1$. I have written a Matlab program to calculate the coefficients and analyze the responses of such filters. For example, Figures 3 and 4 show the interpolation error vs. frequency for 6-tap filters with $\alpha = 0.9$ and $\alpha = 0.5$, respectively. The various curves in each figure are for different fractional delays (interpolation times) over a range of 1 sample in steps of 1/8 sample. There is a dramatic improvement in accuracy as α is decreased. Equivalently, the

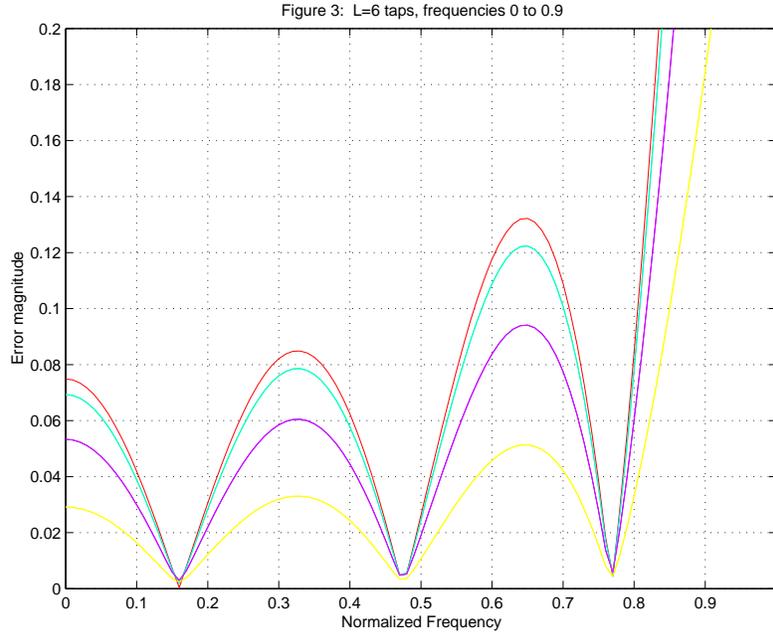


Figure 3. Interpolation filter of length 6 taps, designed for MMSE over 90% of the Nyquist bandwidth. The ordinate is the magnitude of the difference in complex gain between this filter and an ideal interpolation filter. The various curves are for different delays over the range 2.125 sample to 2.875 sample in steps of 0.125 sample.

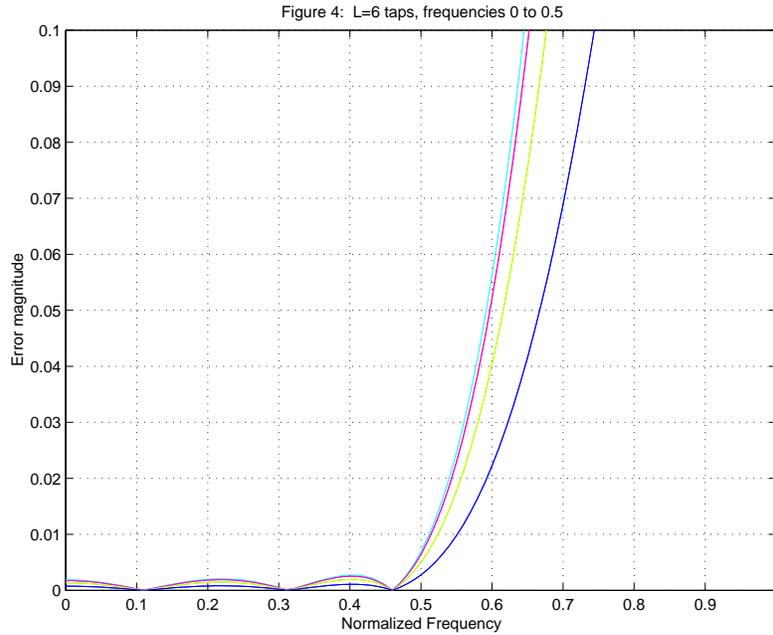


Figure 4. Interpolation filter of length 6 taps, designed for MMSE over 50% of the Nyquist bandwidth. Note change of vertical scale compared with Fig. 3.

number of taps required to reach a given accuracy increases strongly with α . Since the signal bandwidth is $B_1 = \alpha B$, we are oversampling by $1/\alpha$.

Table 3 shows the results of a more detailed exploration of the tradeoff. For various oversampling factors, we find the filter length L needed to keep the maximum error below 1% or below 2%. For the filters in the table, the value of α was allowed to vary slightly from $1/M$ in order to achieve the specified accuracy over

Table 3: Oversampling Tradeoff

Oversample factor M	1% pk err L (α)	2% pk err L (α)
2	5 (0.52)	4 (0.54)
3/2	7 (0.67)	6 (0.67)
4/3	10 (0.76)	8 (0.755)
5/4	12 (0.80)	11 (0.82)
8/7	17 (0.855)	14 (0.855)

95% of the oversampled Nyquist bandwidth (i.e., for frequencies f such that $|f| < 0.95f_s/M$ where f_s is the sampling rate).

There are other considerations that affect the overall cost and complexity as a function of oversampling factor M . At the outputs of the interpolation filters it is important to reduce the sampling rate back to Nyquist so as not to burden downstream circuitry with operating at the oversampled rate. This causes the filters to be more complicated in proportion to the denominator of the oversampling factor $M = m/k$, where m and k are relatively prime integers, as further explained below, even though the overall computation rate (multiplies and adds per second) is independent of M . So the filter complexity increases strongly with $1/M$.

On the other hand, the Digitizer to IFP connection cost, for a given bandwidth, generally increases with M . The connection data rate is directly proportional to M , but the cost remains nearly constant over a wide range corresponding to the capabilities of available parts. For this reason, we choose to fix the transmission rate at the largest value supported by parts that are currently known to be suitable, namely about 2.5 Gb/s per optical link (see Table 1). The multiplexing factor is fixed at 16, so this allows transmission of one channel (8b real and 8b imaginary) at a sampling rate of $(2.5 \text{ Gb/s})/16 = 156 \text{ Mb/s}$. Next, based on Table 3, we select the smallest oversampling factor that allows construction of interpolating filters that are not too complex, as estimated in the next subsection, resulting in the choice of $M = 3/2$.

We select the interpolation filter design that limits the error to 2%, namely $L = 6$ and $\alpha = 0.67$. Performance of the filter is plotted in Figure 5. This shows (like Figs. 3 and 4) the magnitude of the complex gain difference between this filter and an ideal interpolator (unity magnitude, linear phase); it applies to a sinusoidal signal at each frequency. Although the error reaches 2% at the band edge, it is well below this for most other frequencies. Remember that the filter was actually designed for MMSE with broadband noise. The average error for broadband signals should be much less than 1%.

Of importance to our application is the “closure” error in interferometry. This is the residual from an attempt to represent the broadband gain of every interferometer in an array as the product of the complex gains of two antennas. Usually a continuum calibrator is observed, and the $N(N - 1)/2$ measured visibilities are used to find the N antenna gains that provide the best fit. If the complex frequency responses of all antennas were identical, then the residual of this fit would be due to noise alone and could be reduced by additional integrating time. In practice, differences in the frequency responses cause systematic residuals called closure errors. These limit the calibration accuracy, which affects the dynamic range of images, among other things. (For additional discussion, see [7].) When interpolation filters are used for fine delay tracking, the variation of response with delay setting is one cause of closure errors. I have not attempted an accurate calculation of those errors, but an upper bound is provided by the square of the maximum gain difference, $.02^2 = .0004$. Published calculations [8] of the effect of sinusoidal ripple in the frequency responses show that the closure error is less than 1% for ripple amplitude of 1 dB (13%). If it is shown that better accuracy is needed, the $L = 7$ filter can be considered; whether the cost of the additional logic is significant will not be known until a detailed design is available.

Logic Requirements

Table 4 is a spreadsheet in which I estimate the logic required to implement the IF Processor functions for each architecture. This covers one IF channel (out of 8) of one antenna. Only the main signal processing functions are considered, omitting some auxiliary features such as capturing data into a buffer. Some functions are common to both architectures and therefore use the same logic, but must operate at different speeds. The complex gain generator, which is a major function common to both architectures, is also omitted.

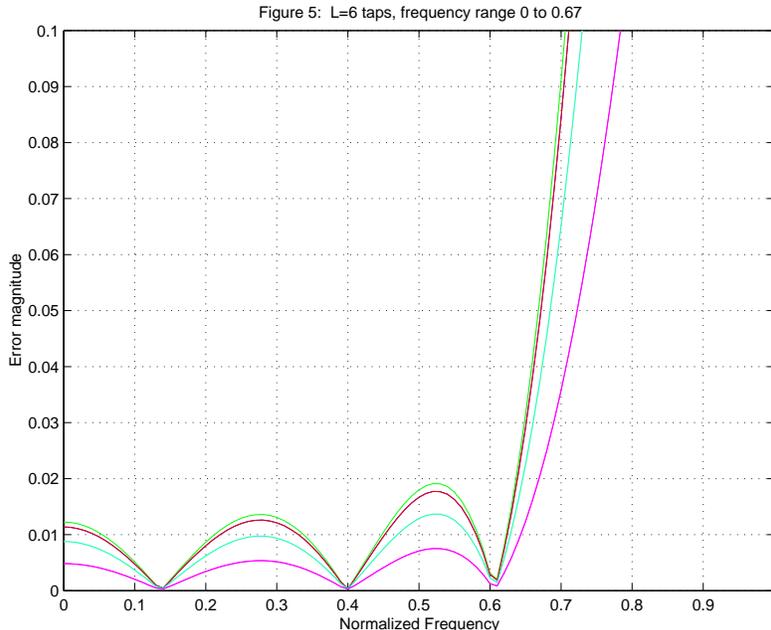


Figure 5. Performance of the selected interpolation filter, $L = 6$ taps and $\alpha = 0.67$.

Logic is measured in two units: Xilinx “slices” and 4096-bit blocks of RAM. The latter are used only to implement the coarse delay. A slice consists roughly of two 4-input boolean function generators and two flip-flops, along with certain additional logic and flexible interconnection paths. It is a basic unit of all Xilinx FPGAs, and very similar building blocks are used by other FPGA manufacturers. The size of each major functional element is determined either from a rough design that I have done, or from a pre-designed “core” available from Xilinx. It can be seen from Table 4 that the F16 design is dominated by the 16-channel polyphase filter bank, even though this is common to all beams, and the T1.5 design is dominated by the FIR interpolation filters. Therefore, the designs of these are broken down in a bit more detail.

The interpolating filter is complicated by the need for $2/3$ decimation. The delay through the filter may be regarded as consisting of the interpolation delay, measured relative to the time of the first input sample used in the computation and determined by the filter coefficients; and the intrinsic delay, which is the time between the loading of the first input sample and the generation of the output sample. The difficulty is that the intrinsic delay is not constant. For example, at $L = 6$ the smallest possible intrinsic delay is 6 input samples (with no latency or combinatorial delay in the filter). But if it is 6 samples for one output, then it is 6.5 samples for the next output, which occurs 1.5 input samples later. In general, for $M = 3/2$, the intrinsic delay alternates between two values 0.5 input sample apart. In order that the total delay be constant, it is necessary that the interpolation delay alternate as well. Thus we must really implement two filters, with different coefficients, where each is needed only half of the time.

An efficient implementation of an FIR filter with constant coefficients uses lookup tables for the multipliers. Xilinx has a core for such constant-coefficient multipliers. In Table 4, I have extrapolated the slice count of the Xilinx core by including two lookup tables, used on alternate cycles, to allow for two sets of coefficients. Other elements of the multipliers and filters, such as adders, are not duplicated.

The overall slice counts for the two architectures are remarkably close. The T1.5 architecture uses more RAM because the coarse delay is implemented separately for each beam, but it falls well within the capacity of FPGAs having sufficient logic. Examples of Xilinx FPGAs that might be used for the implementation are also listed in Table 4. Each design would almost fit into a large Vertex-II series chip (XC2V1000), but these are expensive. Each would fit more comfortably into two smaller Spartan-IIe chips (XC2S300E).

A possible design involves constructing the processing for two IF channels on one board using a total of four XC2S300E FPGAs. The complex gain generator, along with some auxiliary functions and control, could be implemented in a fifth, much smaller, FPGA.

Recommendation

It has been shown that the T1.5 architecture is feasible, and it is that arrangement that I now recommend. The overall cost of implementation is very nearly the same as for the F16 architecture, provided that the same bandwidth is processed. Some of the considerations that favor each choice are:

Favoring F architecture:

- can support larger bandwidth with known link components
- possible lower cost for digitizer to processor data links ($\sim 75\%$)

Favoring T architecture:

- output at Nyquist, no rate penalty downstream
- simpler back end interface, no need for block clock to align subchannels
- supports larger array size (not limited to 1 km)
- supports simultaneous beams in any directions (not limited to primary beam)
- rapid tracking of gain magnitude is easier
- possible add-on FIR filters for improved alias rejection

An earlier concern about the SNR loss caused by delay errors in the T architecture has been eliminated by the interpolating filters.

Although T remains simpler, the difference is not large. Both schemes now require a non-integer rate change during processing. Overall, I believe that the T architecture is significantly, but not overwhelmingly, better.

Unresolved Design Issues

This report has discussed only the high-level architecture, although it has included a few design details. Many other details need to be worked out before a complete design can be realized. Issues that are currently unresolved include:

High speed backplane connections. The beam partial sums for phased array back ends are intended to be passed from one IF Processor module to the next over a backplane. In the baseline design, this was done with differential signals at a speed 4 times the sample rate. But the multiplexing/demultiplexing required for the rate change adds substantial complexity, and the clock rate is high compared with the experience of any of the engineers involved. Therefore, this approach is considered risky. If the speedup is omitted, with signaling at the sample rate, then the number of wires is uncomfortably large. This number is cut in half if single-ended connections are used, but signal quality suffers. It is further reduced if only 3 (dual polarization) beams are included in the summation chain, rather than all 4; this still leaves 448 signals. (The fourth beam goes to the correlator, but in the baseline design it was also available for a phased array back end when the correlator is not using it.) The design needs to be finalized.

Possible integration of correlator F engine with IF processors. It has been proposed that the Back End Transmitter modules be replaced with the antenna-based processing (mostly filter banks or “F” engines) of the correlator. This would require all such processing for two channels of six antennas to fit into the same space as a BET module, and it is currently uncertain whether this is feasible. It would save the cost of the BET/BER modules, which is substantial. Integration of the two very different subsystems in the same chassis would cause difficulties during development and testing, in view of their different schedules, funding, and personnel. Coordinating command and monitor interfaces would also be difficult. Packaging of the IF Processors might have to be modified to accommodate the F modules. Decisions about future upgrading of the two subsystems would have to be closely coordinated, rather than mostly independent.

Monitor and Command interfaces. Very little work has been done so far on control and monitoring. This is rather simple for the Digitizers, but some sort of computer interface still needs to be designed. It is made slightly more complicated by the requirement to support phase switching. For the IF Processors, it is considerably more complicated because of the need to update continually the tracking information. It can be assumed that each IF Processor assembly carries a microcontroller that does the internal control and supports an interface to a higher level computer of the Monitor/Command subsystem. A design is needed for this interface. Since the present design calls for 1400 IF Processor modules, putting them on a single bus may be difficult.

REFERENCES

- [1] ATA Project, Digital Subsystem Preliminary Design Review. Meeting held on 2001-Jul-31. Documents available at
<http://intranet.seti.org/docs/ata/PDR/#Digital>
- [2] L. D'Addario, "ATA IF Processor: Description of the preliminary baseline design," 2001-Aug-02.
<http://astron.berkeley.edu/~ldaddari/ata/baselineDesignAll.pdf>
- [3] L. D'Addario, "ATA IF Processor: Requirements," rev 2.1, 2001-Jul-14.
<http://astron.berkeley.edu/~ldaddari/ata/ifpRequirements.pdf>
- [4] L. D'Addario, "Notes on 8b/10b coding for serial data transmission," report dated 2001-Aug-09.
<http://astron.berkeley.edu/~ldaddari/ata/8b10b.pdf>
- [5] L. D'Addario, "Processing architectures for complex gain tracking," ATA Memo No. 40, 2001-Oct-25.
<http://intranet.seti.org/docs/ata/Memo/memo40.pdf>
- [6] R. Crochier and L. Rabiner, *Multirate Digital Signal Processing*, p 162. Prentice Hall, New York: 1983.
- [7] A. R. Thompson, J. M. Moran, and G. W. Swenson, Jr., *Interferometry and synthesis in radio astronomy*, second edition, pp 235–240 and Tables 7.1 and 7.2. Wiley & Sons, New York: 2001.
- [8] A. R. Thompson and L. R. D'Addario, "Frequency response of a synthesis array: performance tolerances." *Radio Sci.*, vol 17, pp 357–369, 1982.

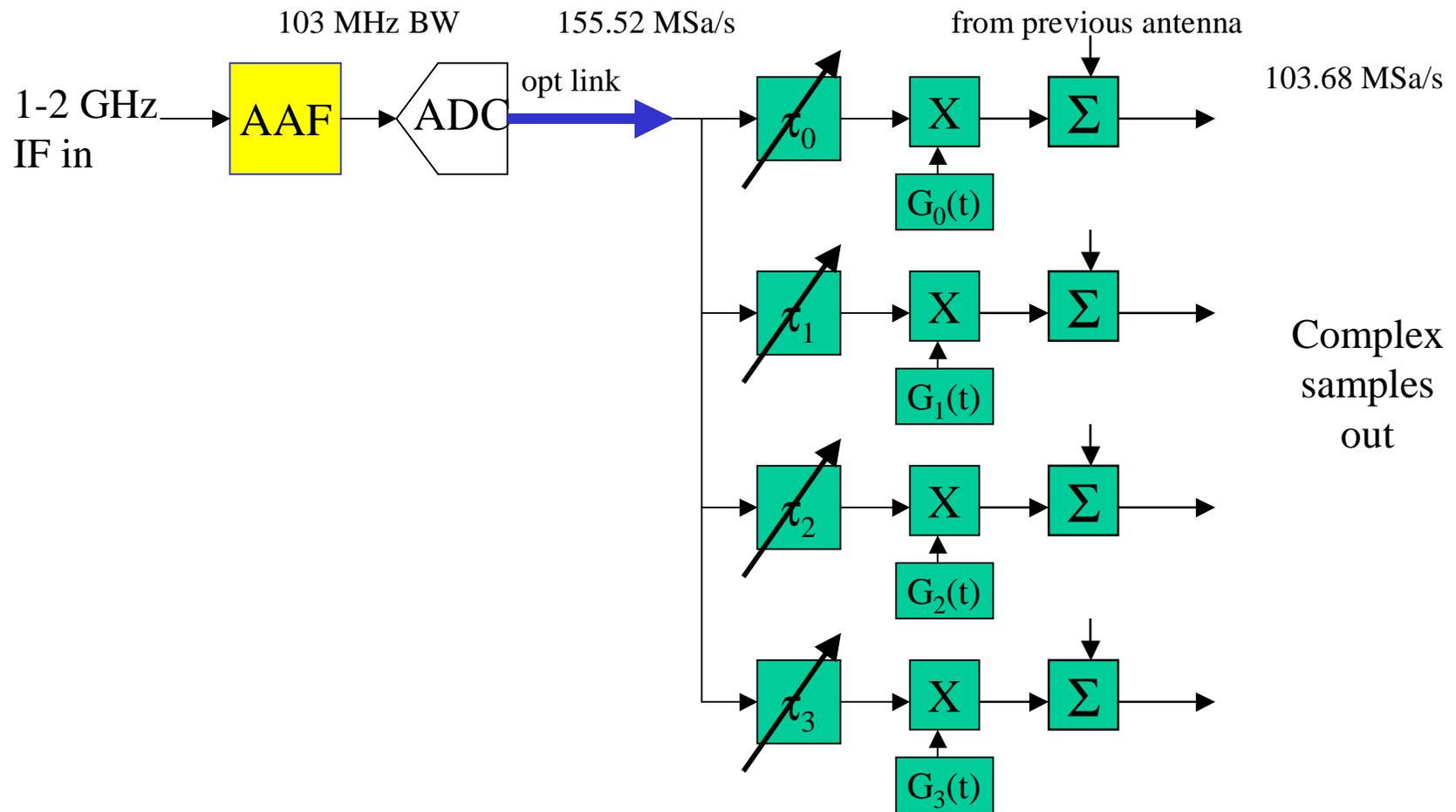


Figure 1. Block diagram of the T1.5 architecture, showing the processing for one IF channel (out of 8 channels per antenna). Here AAF=anti-aliasing filter (analog); ADC=analog-to-digital converter; X=complex multiplier.

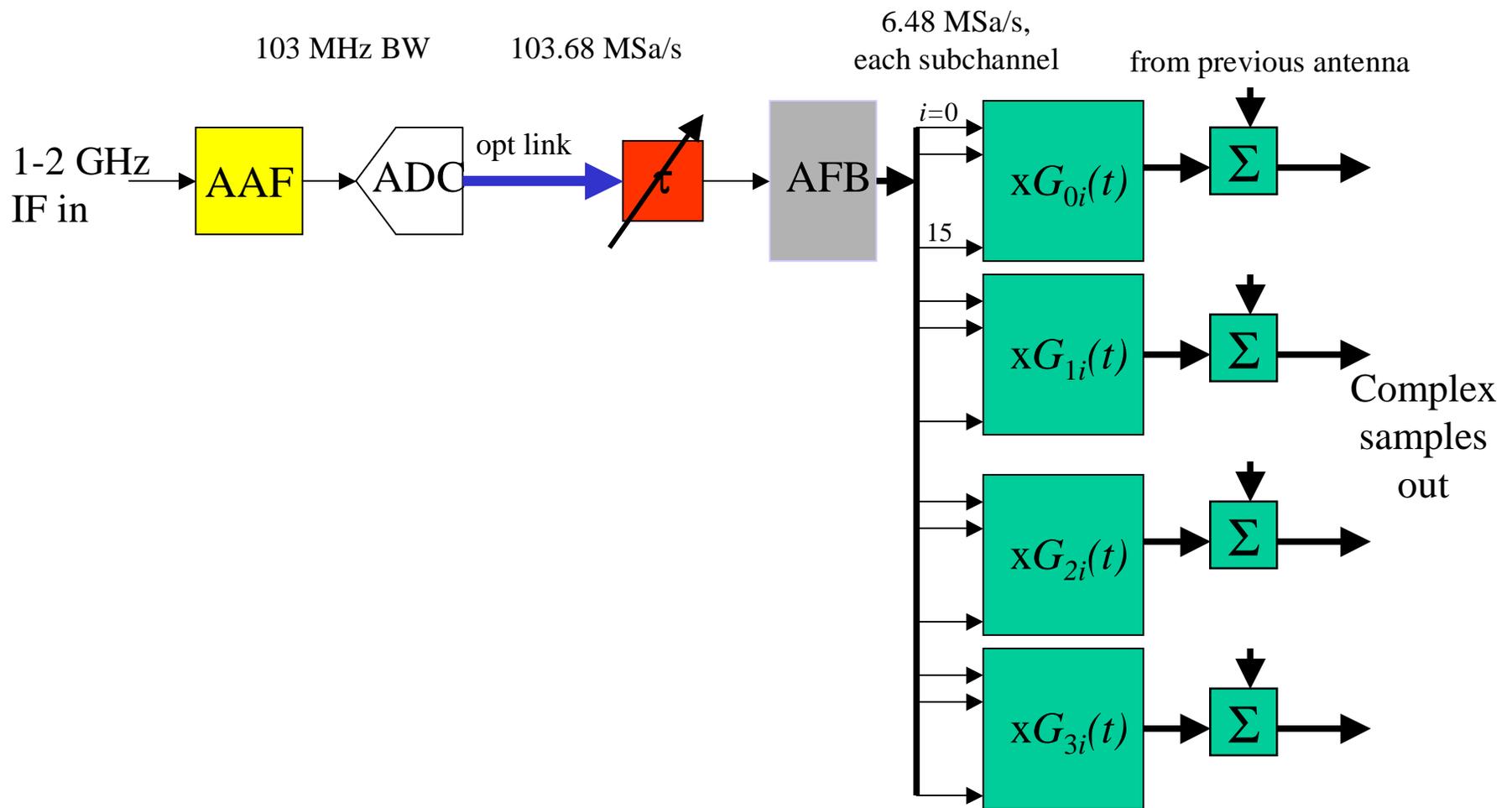


Figure 2. Block diagram of the F16 architecture, showing the processing for one IF channel (out of 8 channels per antenna).

AFB=analysis filter bank: separates channel into 16 subchannels.

Table 4: Estimates of Required Logic

Main Signal Processing Path, one single-polarization IF channel, 4 beams

27-Dec-01 revised 31-Jan-02

bandwidth 103.68 MHz

	<i>rate</i>	<i>slices</i>	<i>Blk RAMs</i>	<i>tot slices</i>	<i>tot BRAMs</i>
<u>F Architecture</u>					
1 Sample statistics computer	103.68	184	0	184	0
1 QDC corrections	103.68	176	0	176	0
1 Coarse delay	103.68	48	8	48	8
1 Polyphase filter bank, 16 ch	127.6062	3674	0	3674	0
4 Complex mpy (8,8)x(8,8)	127.6062	229	0	916	0
4 Cplx add (16,16)	127.6062	16	0	64	0
Totals				5062	8

<u>T Architecture</u>					
1 Sample statistics computer	155.52	184	0	184	0
1 QDC corrections	155.52	176	0	176	0
4 Coarse delay	155.52	48	8	192	32
4 FIR8 filters for delay interpolation	103.68	912.5	0	3650	0
4 Complex mpy (8,8)x(8,8)	103.68	229	0	916	0
4 Cplx add (16,16)	103.68	16	0	64	0
Totals				5182	32

Dec'01/#25

Xilinx XC2V1000-6FG456C	\$559.90			5120	40
Xilinx XC2S300E-6BG456C	\$67.50			3072	16

Polyphase filter bank, 16 complex ch, L=8

16 8x8 mult		40		640	
14 add 16+16		8		112	
128 input SR words, 8+8b wide		8		1024	
128 coef SR words, 8b wde		4		512	
1 FFT16, Xilinx core		1386		1386	
				3674	0

FIR filter, length 6, data 8b, coef 10b reloadable

Custom design with two LUT sets per multiplier to allow alternating 2 coef sets

12 Const coef mult 8x10, modified Xilinx core		68.5		822	
5 Add16		8.5		42.5	
16 6x1 SR (8b for I, 8b for Q)		3		48	
				912.5	0