

**CSIRO Division of Radiophysics**  
The Australia Telescope

**Hybrid FFT Correlator - Design Outline**

Martin Ewing  
28 April 1986

*Like cold water to a weary soul  
is good news from a distant land.*

**Summary.** A cost-efficient correlator incorporating fringe rotation can be built using a 64-point Fourier transform in front of a conventional correlator. This technique may be used with the off-the-shelf AT XCELL chip to satisfy the needs of either the Australia Telescope Long Baseline Array (AT LBA) or the interim US Very Long Baseline Array (VLBA). Some VLSI development would be useful, but is not required. For a larger version, suitable for the ultimate VLBA, VLSI implementation of the FFT processor will be cost justified.

**Introduction.** I have been looking for a cost-effective way to use the CSIRO/Austek XCELL chip in a correlator design that could be used for either the AT LBA or the interim VLBA. The task is complicated by the fact that both Arrays are likely to require fringe rotation in the correlator which is not provided in the XCELL.

Two methods of post-recording antenna-based fringe rotation have been proposed: (1) translate each data stream with an SSB mixer before correlation, using digital phase shifting (Hilbert transforms), and (2) multiply each data stream by an approximate in-phase and quadrature sinewave, preserving a 4-bit complex product, with post-correlation image suppression (the "conventional" VLBI approach). This memo concentrates on the second option, looking for an economical and flexible implementation.

Correlation of two phase-rotated data streams suffers from double the SNR loss of a single rotation and from other possible intermodulation effects. Such problems are minimized by keeping more bits per data sample. In particular, preserving and correlating 4 bits per sample seems to reduce losses and spurious responses to an acceptable level (<1%). The XCELL can support 4-bit multiplication with a suitable allocation of its multipliers (cf AT Memo AT/24.2/007 = VLBA Correlator Memo 057).

**The Hybrid Approach.** I have explored the possibility of using a moderate size DFT in front of the correlator to reduce the number of multipliers required. It is well known that by splitting the signal band into N subbands before correlating, a factor of N reduction is obtained in the number of multiplies per second required to reach a certain resolution. (In fact, the AT and VLBA already realize some of this gain by dividing their IFs into 2-8 channels before recording.)

Specifically, I have looked at the possibility of using 64 point transforms to reduce the multiplier requirement, as shown in Figure 1. In this I am using the approach taken by John O'Sullivan in Note 375 of the NFfRA (Dwingeloo) "*Efficient Digital Spectrometers - a survey of possibilities*", Sept., 1982. N data samples are buffered and padded with zeroes and transformed with a 2N point DFT. After the 64-point spectra of two 32-sample buffers are multiplied and accumulated, the precise crosscorrelation function may be recovered. Moreover, the correlation amplitude can be corrected by the standard "Van Vleck" methods. Essentially, we synthesize a large "time-domain" correlator using the fast convolution (overlap and add) algorithm, DFT processors, and a relatively small time-domain correlator.

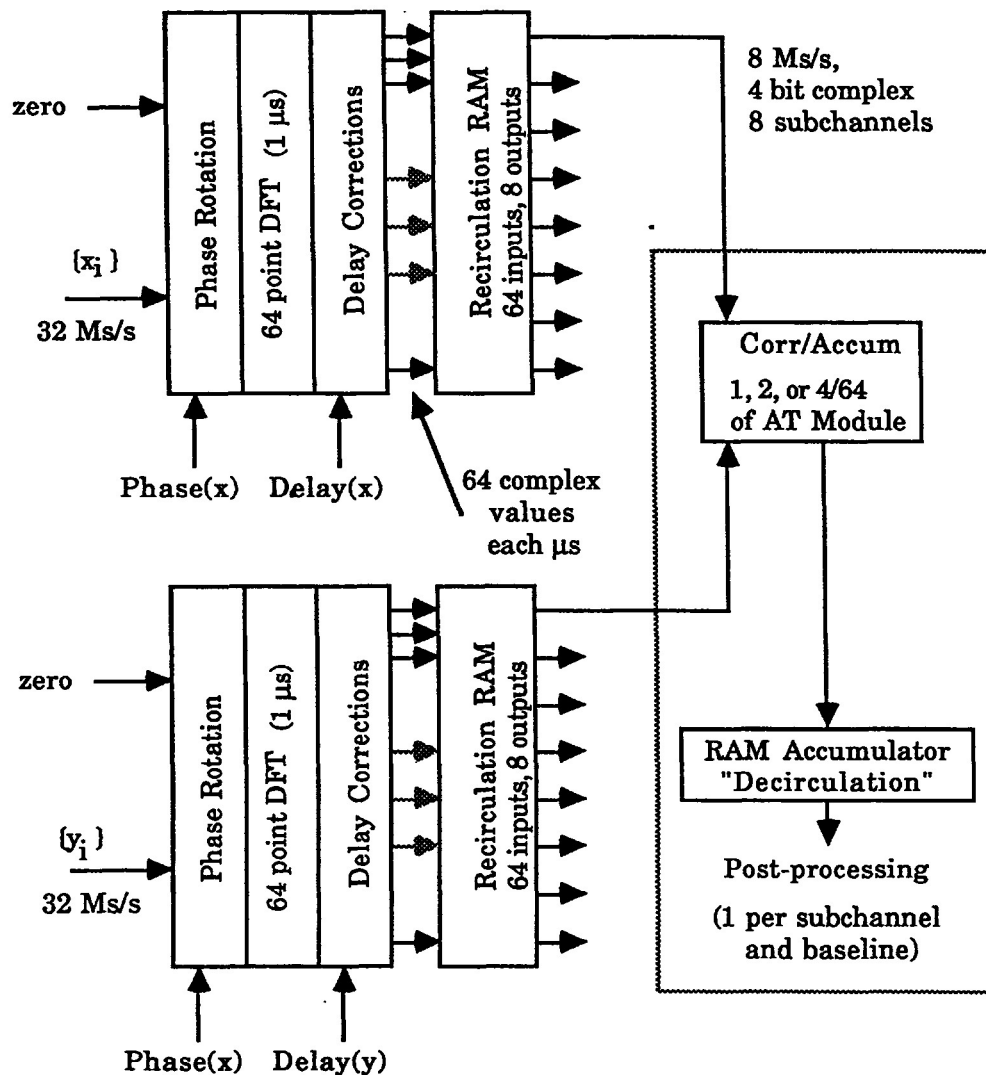


Figure 1. Hybrid Correlator overall.

Why choose 64 point transforms? It is advantageous to avoid large transforms, so that dynamic range may be preserved in a small word length, i.e., to minimize the "processing gain." Longer transforms also increase the damage done by small sections of invalid data, etc. Longer transforms minimize the size of the time domain correlator, but short transforms require less transform hardware. I chose 64 point FFTs as a happy compromise among these considerations; it is also about the smallest FFT length that compensates for the factor of 16 loss in effective correlator density caused by 4-bit complex arithmetic. The FFT output dynamic range should fit within an 8 bit field, a convenient size for practical design.

We can take advantage of the FFT processor to perform the fringe rotation function. This amounts to a "phase winding" at the input. Rather than explicitly generating sine and cosine waveforms, we can provide 3 or 4 bits of phase to the first FFT stage, which can be implemented in ROM. At the FFT output, a similar phase winding can be applied to apply the vernier delay and fractional bit shift correction.

One remaining consideration is that, though the FFT should probably be computed to full dynamic range, it must be requantized to 4 bits before correlation. This is no problem for continuum or weak spectral line observations, but must be examined carefully in the

case of strong maser lines. It would be possible to set requantization thresholds to a high value in the maser case, but SNR would be lost on weak features in the band. It would even be possible to have a logarithmic response. The quantization is specified in ROM, and different types of experiments could use different quantizing "laws."

**FFT Implementation.** The requirement is to carry out a 64-point DFT with pre- and post-processing phase winding. The first 32 data points are real numbers, and the upper 32 input points are zero. Although the input data are 2-bit samples, the bits are weighted in a ratio of about 4:1. Also, the validity bit may be considered a third data bit, multiplying the data by 1 or 0. The transform must be completed in 1  $\mu$ s.

In order to estimate costs I have made a rough design of such an FFT processor as shown in Figure 2. In each of the 4 radix 2 stages, we must perform 32 butterfly operations per  $\mu$ s - about 31 ns per butterfly. This is quite fast for TTL RAMs, especially since one needs two reads or writes per butterfly cycle. I assume we would choose to implement 93 ns butterflies. In this case, 3 separate 3  $\mu$ s FFT processors would operate in parallel to satisfy the 1  $\mu$ s specification. Perhaps in the end a faster implementation can be found so that only one or two processors would be required.

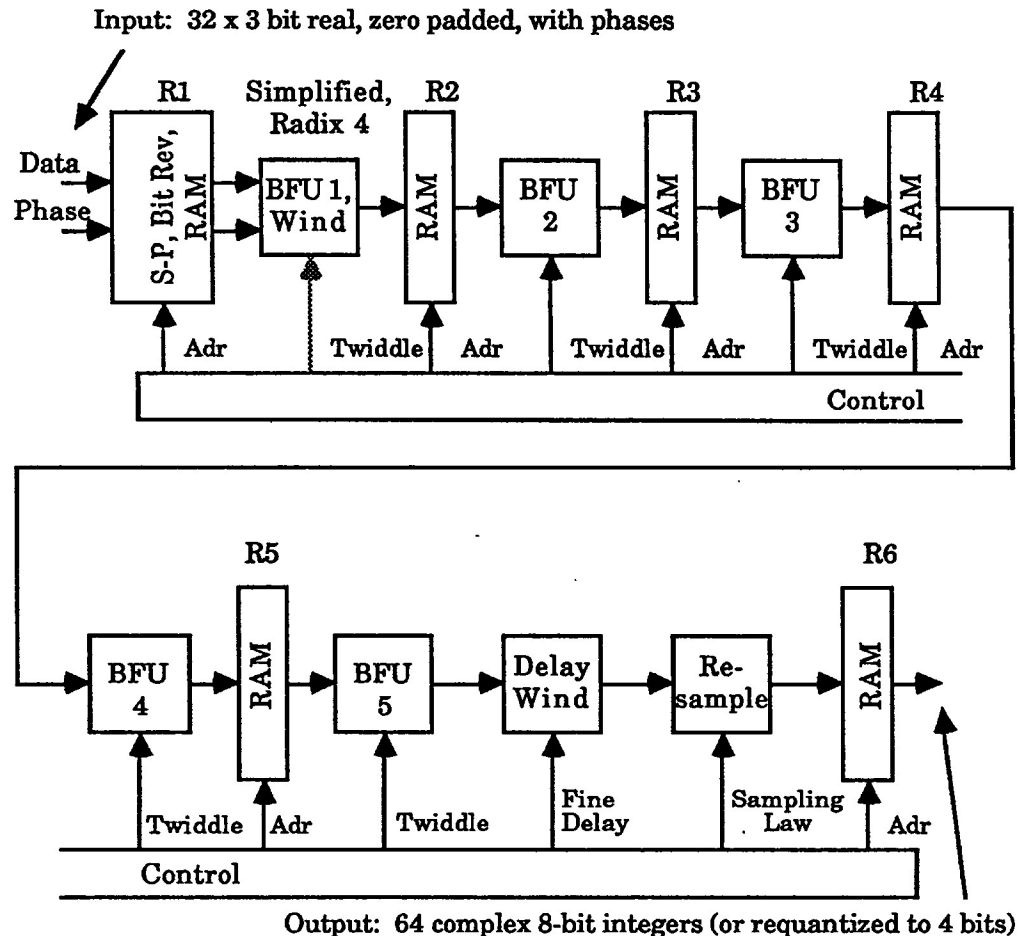


Figure 2. 64-point FFT Implementation

In a front end section, shift registers store up data and phase rotator values of 32 samples. (In fact, fewer phase values need to be stored, since the extreme fringe rate is

$\pm 128$  kHz, and phase does not have to be updated every sample.) The input bit reversal is accomplished through address reordering.

The first processing stage (radix 4) is particularly simple since half the inputs are zero, the non-zero inputs are real, the data dynamic range is small (effectively about 3 bits), and only trivial twiddle factors ( $\pm 1$ ) are required. If we wish to use 4 bits of phase input, this stage may be implemented in a  $16K \times 8$  ROM.

The successive radix 2 stages are built up using a ROM-based "twiddler", which multiplies a complex sample by  $\exp(j\phi)$ . The ROM (or pair of ROMs) accepts a complex sample and a phase setting to produce a rotated sample. Since the data dynamic range increases from one stage to the next, the ROM size increases substantially, but should still be manageable in the final stage. The final stage takes an additional phase input that specifies the delay offset (fractional bit plus vernier). Storage between stages is accomplished with fast (35 ns) RAMs.

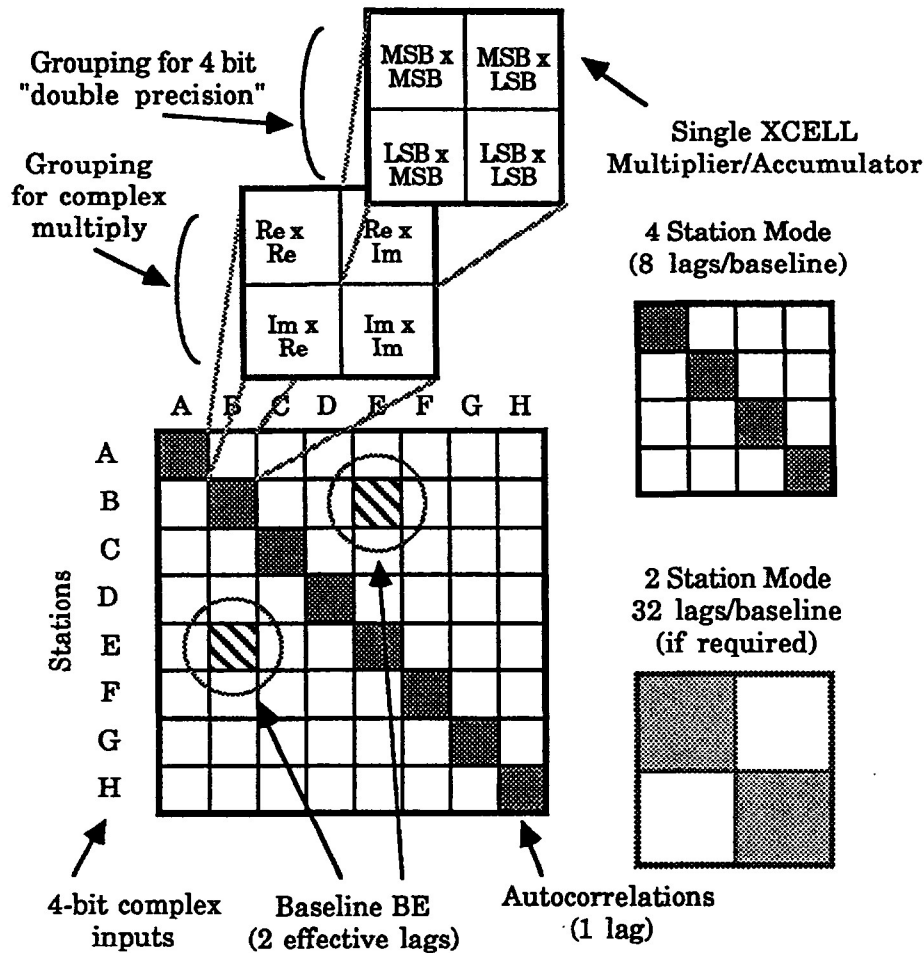
After the final output stage, a full 8-bit wide RAM stage would be provided. The correlator, however, will accept only a 4-bit input. The RAM output must be "requantized" via a ROM, logically considered an output stage on the last butterfly processor. The transfer function might be chosen from the following: (1) no change, straight-through 8 bits for tests, etc., (2) a "normal" law, for best sensitivity in weak line or continuum observing, or (3) a "special" law, for strong line observing, where there is some risk of data overflow.

**Recirculation Memory.** After the FFT, the various data corrections, and requantization, we are left with 64-point complex spectra being generated each  $\mu s$ . Since the "sample rate" of each FFT output channel is 1 MHz, we immediately face the need to rearrange the data streams so that the XCELL-based correlator can be used at its maximum data rate (8 Ms/s in 2-bit mode). This can be achieved in a "recirculation memory," which takes in 64 Ms/s (64 complex points each  $\mu s$ ) and puts out 8 Ms/s streams suitable for correlation in XCELL arrays. The size of memory required depends on the minimum acceptable XCELL integration time (and RAM dump interval). With 40 ms integration, about 5 MB of memory is required, allowing for double buffering. (That's only 40 chips nowadays.)

The recirculation RAM can also be organized to output delayed and undelayed versions of the data, so that when the input data rate is less than 32 Ms/s/channel, correspondingly higher numbers of frequency channels may be generated.

**The Correlator Proper.** The output of the recirculation RAM is in 8 "subchannel" streams which are crosscorrelated with streams from other antennas. Only corresponding subchannels are ever correlated, so the subchannels must be correlated on different modules. I assume that we would use the AT Compact Array module, which is effectively a "super XCELL" making an array of  $32 \times 32$  simple multiplier / accumulators, or  $8 \times 8$  complex double precision multiplier / accumulators.

What are the appropriate "X" and "Y" inputs to the module? If subchannels cannot be mixed, the only other useable dimension is antenna number; see Figure 3. Thus we find that this *must* be an 8-station correlator! Actually, by "reconfiguring" we may choose to apply all the module's multipliers to 4 station or 2 station processing. The 4 station mode may be quite useful in many AT/LBA experiments. Just as for the AT Compact Array correlator, it is convenient to group the modules into 8-module (8-product) "blocks;" this is illustrated in Figure 4. Each block correlates a one subchannel "slice" of all baselines and input channels.

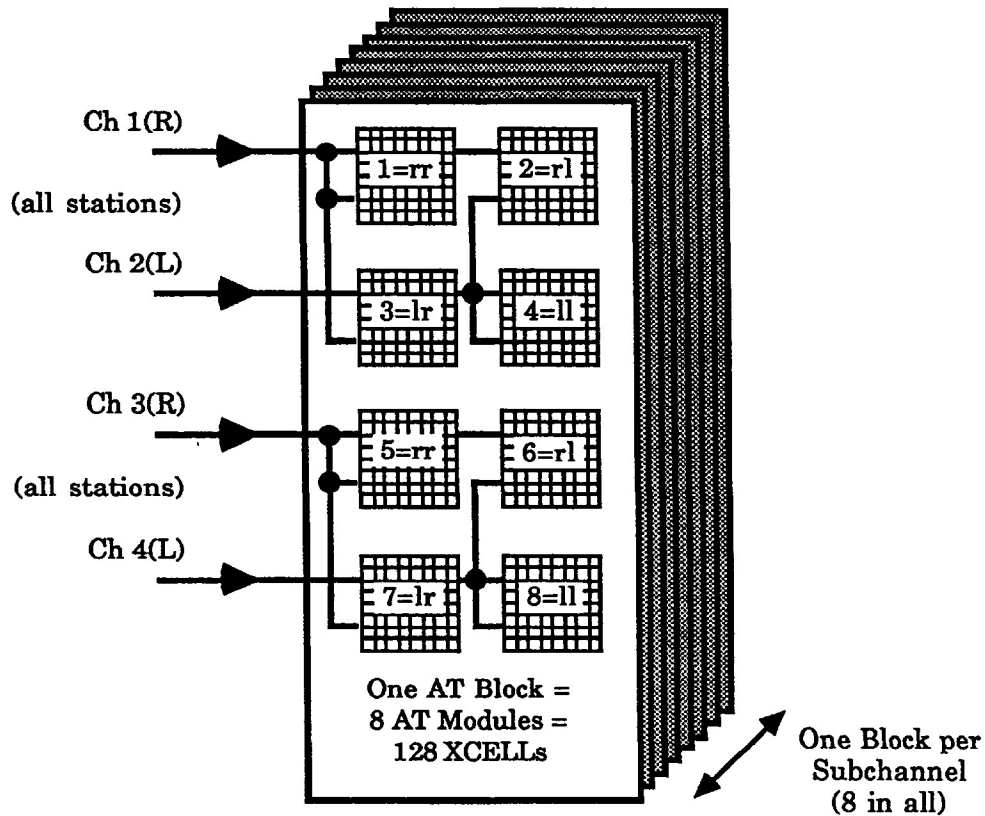


Note: Each "lag" in a correlator module becomes equivalent to ~32 effective lags after combination of subchannels.

**Figure 3. Allocation of Module Inputs to Stations.**

This design can be expanded by increasing the size of the "module" array in steps of 1 XCELL = 2 stations, i.e., to 10, 12, ... stations. In practice, of course, a large upgrade would probably justify an upgrade of the XCELL itself.

**Cost Estimates.** The design of the FFT processor and recirculation memory has only been made in a very preliminary way to see how feasible the hybrid approach might be. In rather round numbers, the cost figures of Table 1 apply to the various elements.



**Figure 4. Eight Product Correlator  
(arranged for polarization analysis).**

**Table 1. Hybrid Correlator Subsystems Costs**

Item	No. of ICs	Cost
<i>Multiply the following by the total number of input channels:</i>		
1 $\mu$ s FFT (3 3- $\mu$ s units)	700	US\$ 5.7 K
Recirculation Memory	80	1.2
<i>Multiply the following by 8 times the number of products:</i>		
AT XCELL Corr. Module	172	2.4*

\*assumes US\$100 per XCELL chip. (The XCELL chips account for about 75% of the module cost.)

Note that these are simple fabrication costs for the boards that must be constructed. They would include the ICs, printed circuit boards, and connectors, but they would not include playback systems, power supplies, racks, cabling, computers, or engineering time. They should only be used as guides to intercompare alternative architectures or configurations.

Taking these rough estimates as given, we can look at the costs of some possible correlator requirements. Table 2 indicates three possible scenarios: (1) a system for the

Australia Telescope LBA, which according to current specifications will be a 5-station, 2-channel correlator, (2) the VLBA interim correlator which should handle 7 stations and 8 channels, and (3) an "ultimate" VLBA system for 16 stations. Note that in cases (1) and (2), the correlator actually is capable of 8-station processing; the number of input channels (FFTs and Recirculation RAMs) is reduced.

Table 2. Hybrid Correlator Cost Analysis

	AT LBA 5 stn	VLBA 7 stn	VLBA 16 stn
No. Input Stations	5	7	16
No. Channels / Station	2	8	8
No. products	4	8	8
Cost	\$145.8 K	\$540.0 K	\$1,036.8 K
No. XCELLs	576	1,152	4,608
Cost of XCELLs	\$57.6 K	\$115.2 K	\$460.8 K
Percent cost in XCELLs	40	21	44
<i>Comparison: AT real correlator: 1 block = N products per baseline No Fringe Rotation or Autocorrelation!</i>			
No. Baselines	10	21	120
No. Corr. Modules	40	168	960
Comparison Cost	\$96.0 K	\$403.2 K	\$2,304.0 K
Comparison No. XCELLs	720	3,024	17,280

The comparison against a "standard" AT correlator with similar input specifications offers some perspective. Of course, such a correlator does not meet the requirement of FTC—fringe tracking in the correlator.

In general, the hybrid correlator requires fewer correlator ICs, but costs somewhat more than the AT "real" correlator. The FFT processors are quite a significant cost item for all 3 systems, and a more refined (less expensive) FFT design will be well worthwhile. But the significant result is that the FTC function can be achieved at roughly the same cost as a similar non-FTC XCELL-based machine.

**Larger Systems.** For the "ultimate" VLBA system, and possibly for the 7-station interim correlator, the cost model described above is probably too pessimistic. One can afford to improve several parts of the system using custom or semi-custom VLSI.

We already know of two possible upgrades to the XCELL chip: an increase to 16 Ms/s for 2-bit sampling (with a mask charge of \$40K), and a reimplementation in CMOS which should yield 32 MHz operation. (Refer to Ewing, *Austek quote for XCELL ICs*, 23 April 1986, VLBA Correlator Memo, in press.) These would help reduce the cost of the correlator proper, although this is not an overwhelming cost in the hybrid design. A further factor of 4 in correlator circuit density would come through implementing 4-bit multiplication in VLSI.

An even more productive area for VLSI is the FFT processor. CMOS gate array technology should bring down the IC counts by a factor of 10 or more. From a cost viewpoint, a VLSI development would clearly be justified for the larger VLBA machine. A VLSI version of the FFT could be viewed as an upgrade to be implemented *after* the interim machine was completed.

**Statistics.** Table 3 presents a general summary of the capabilities of an 8 station hybrid correlator, with all 8 station inputs implemented. Two-bit sampling is assumed throughout. Table 4 lists the modes available in an 8 product hybrid correlator assuming that 8 input channels per station are implemented.

**Table 3. Summary Statistics**

Number of Stations/Antennas:	8 or 4
Number of Baselines:	28 or 6
Number of Autocorrelations:	8 or 4
Number of products/baseline:	8*
Number of Channels/Antenna:	8 (non-polarized) 4 (polarization modes, supporting two frequencies)
Number of subchannels/channel:	8
Number of modules/subchannel/product:	1
Number of modules total:	64
Bits/sample:	2 data, 1 validity
Sample Rate:	32, 16, 8, ... Ms/s (taking every $2^n$ -th bit from the DPS)
Phase Setting Precision:	4 bits
Minimum Phase update interval:	8 samples
Instantaneous Delay Error:	$\leq 1/16$ sample
Minimum Delay update interval:	32 samples (1 FFT time)
Recirculation Factors vs BW:	
16 MHz	1 (no increase in channels)
8	2 (2x increase)
—	--
125 kHz	128
Reconfiguration Factors vs Stations:	
8 stations	1 (no increase in channels)
4	4 (4x increase)

\* In AT parlance, a "product" is a complete correlator unit, able to process one channel and one baseline. Four products are required to measure all polarization parameters on one baseline, but products can be used singly or in pairs to increase the number of frequency channels when polarization analysis is not required.

**Table 4. Processing Modes**

Polarized/ Nonpol.	No. Stations	BW/IF ch, MHz	Tot BW MHz	N IF ch*	Freq. ch /IF ch.	Freq. Res. kHz	Recirc. Factor
Nonpol.	8	16	128	8	32	500.	1
Nonpol.	8	16	64	4	64	250.	1
Nonpol.	8	16	32	2	128	125.	1
Nonpol.	8	16	16	1	256	62.5	1
Nonpol.	4	16	128	8	128	125.	1
Nonpol.	4	16	64	4	256	62.5	1
Nonpol.	4	16	32	2	512	31.25	1
Nonpol.	4	16	16	1	1024	15.62	1
Nonpol.	8	8	64	8	64	125.	2
Nonpol.	8	8	32	4	128	62.5	2
Nonpol.	8	8	16	2	256	31.25	2
Nonpol.	8	8	8	1	512	15.62	2
Nonpol.	4	8	64	8	256	31.25	2
Nonpol.	4	8	32	4	512	15.62	2
Nonpol.	4	8	16	2	1024	7.81	2
Nonpol.	4	8	8	1	2048	3.90	2
Nonpol.	8	4	128	8	128	31.25	4
Nonpol.	8	4	64	4	256	15.62	4
Nonpol.	8	4	32	2	512	7.81	4
Nonpol.	8	4	16	1	1024	3.90	4
Nonpol.	4	4	128	8	512	7.81	4
Nonpol.	4	4	64	4	1024	3.90	4
Nonpol.	4	4	32	2	2048	1.95	4
Nonpol.	4	4	16	1	4096	0.97	4
Polarized	8	16	32	4	32	500.	1
Polarized	8	16	16	2	64	250.	1
Polarized	4	16	32	4	128	125.	1
Polarized	4	16	16	2	256	62.5	1
Polarized	8	8	16	4	64	125.	2
Polarized	8	8	8	2	128	62.5	2
Polarized	4	8	16	4	256	31.25	2
Polarized	4	8	8	2	512	15.62	2
Polarized	8	4	32	4	128	31.25	4
Polarized	8	4	16	2	256	15.62	4
Polarized	4	4	32	4	512	7.81	4
Polarized	4	4	16	2	1024	3.90	4

\* Polarized modes use channels in pairs. With 8 products, we can have 2 pairs using 4 products each or 1 pair using 8 products arranged for doubled lags.

