

**A More Detailed Analysis of ‘Recirculation’  
Architecture, Algorithms, and Limitations  
in the Proposed WIDAR Correlator for the EVLA**

*NRC-EVLA Memo# 004*

Brent Carlson, July 6, 2000

**ABSTRACT**

The proposed WIDAR correlator for the EVLA aims to use the method of ‘recirculation’ to greatly increase the spectral resolution capabilities of the correlator at narrow bandwidths. This method involves buffering low sample rate data, and then bursting it at high data rates through the correlator with different relative station delays—or buffer address offsets—to effectively provide many more correlator lags for the hardware available. This memo investigates the basic recirculation architecture within the context of the proposed design, develops a buffer address offset algorithm, and discusses some fundamental limitations of the technique in terms of SNR degradation and integration time.

## **Introduction**

In [1], the basic architecture of the proposed WIDAR correlator for the EVLA was presented. In that document so-called recirculation is used to provide very high spectral resolution when correlating narrow bandwidths. This capability was included in the design by request of NRAO at the April 7-8, 2000 meeting in Penticton. The use of recirculation significantly improves the narrowband spectral line capability of the correlator, but it does come at an additional cost in terms of design complexity and performance demands on the correlator. This memo will investigate recirculation in more detail so that a more thorough understanding of the additional complexities and limitations of this method can be obtained.

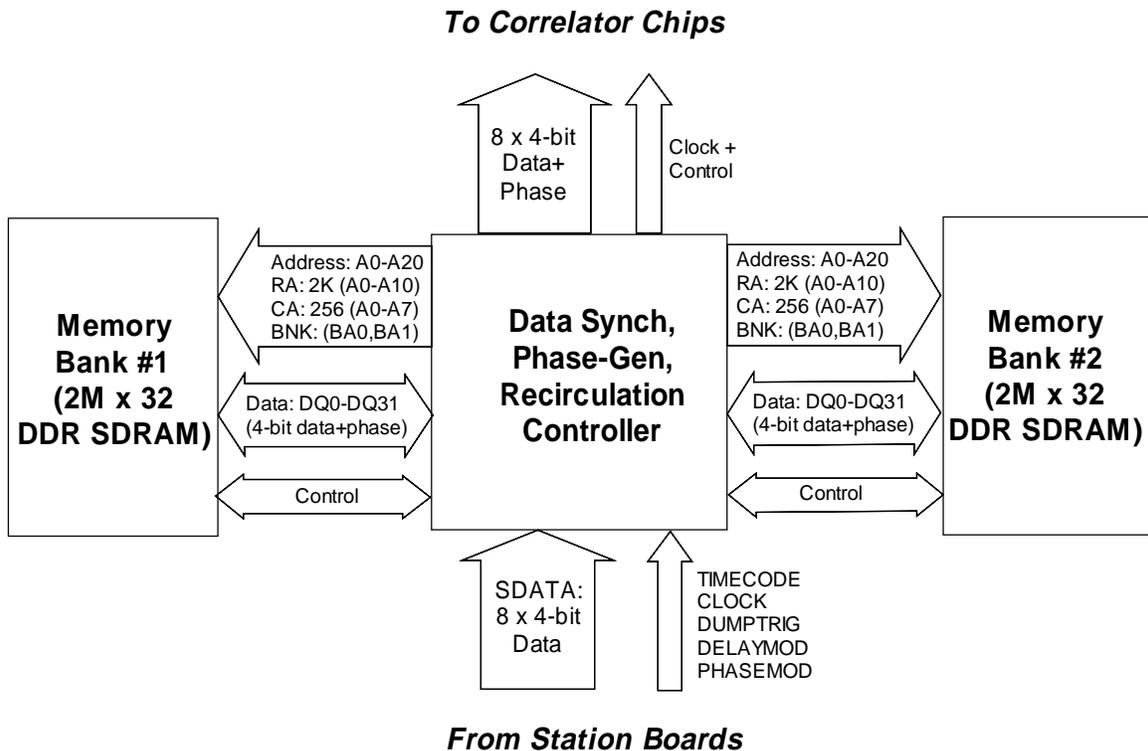
## **Recirculation Architecture**

Figure 1 is a block diagram of the station-based recirculation architecture. Each Baseline Board has 16 of these—8 for the ‘X’ antenna inputs, and 8 for the ‘Y’ antenna inputs. These are the “Rx/RMEM phi-gen Blocks” described in [1]. The controller will probably be implemented in a single FPGA (Xilinx Virtex-E<sup>1</sup>) and each of the memories is a chip in a 100 pin LQFP package. With double-sided surface mount boards, each of the blocks shown in Figure 1 would occupy a small amount of board space and it therefore seems feasible at this stage to include 16 of them on every Baseline Board. Depending on the

---

<sup>1</sup> These devices come packed with features that make them ideally suited for interfacing to DDR SDRAMs at high data rates.

total number of lags being generated, the real-time sample rate, and the size of the correlator chip, one controller can handle the entire lag correlation or just one section of the total lags—the other sections being handled by other controllers in other sub-band correlators [1]. Not shown is a small SRAM needed for the delay-to-phase lookup table for very fine delay tracking.



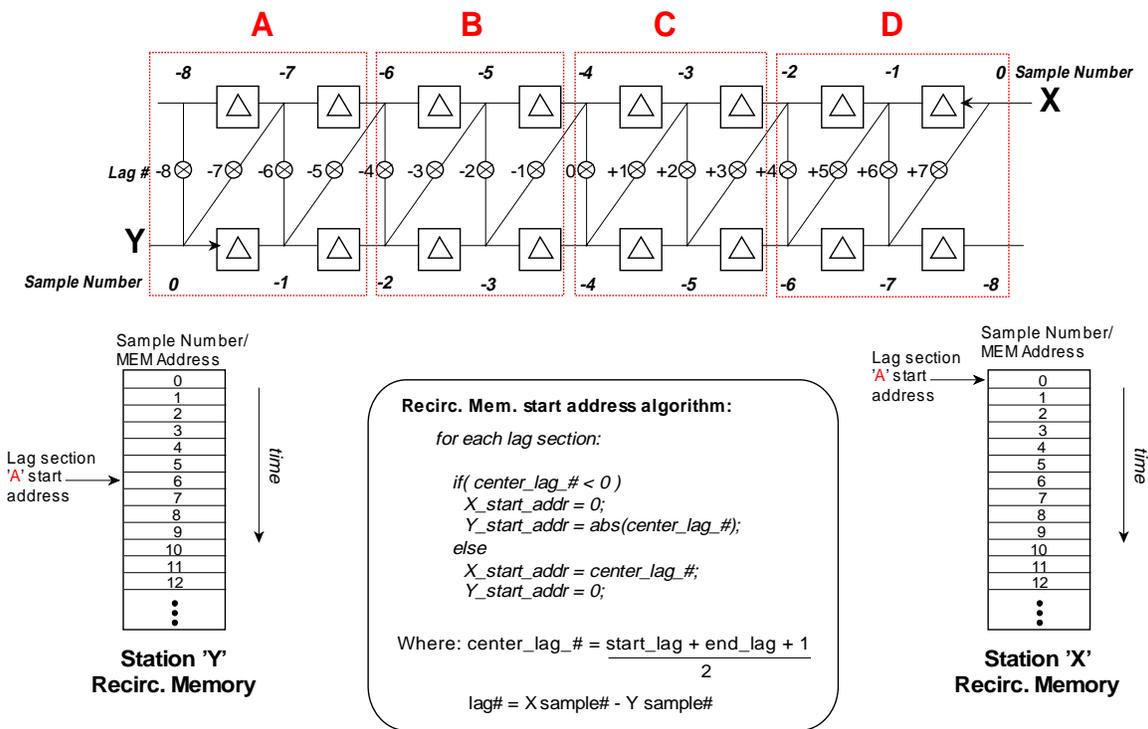
**Figure 1** Block diagram of the basic recirculation architecture for one station on a Baseline Board. There are potentially 16 of these on every Baseline Board—8 ‘X’ station controllers and 8 ‘Y’ station controllers. The controller receives and synchronizes SDATA and control signals coming from the Station Boards. Data and quantized phase is alternately written to memory banks implemented with 2M x 32 DDR SDRAM (Double Data Rate Synchronous Dynamic RAM). When data is being written to one RAM bank, data is being burst at the high sample rate and fed to the correlator chips from the other RAM bank. The maximum number of bursts that can be performed—and therefore the “lag-length multiplication factor”—is the ratio of the high sample rate to the current real-time sample rate. Double buffering is a necessary part of the design and ensures that only single-port RAM is required—enabling the use of high capacity dynamic RAM.

Data is alternately written into the memory banks and the bank that is full of data is burst at the high rate with varying start addresses into the correlator chips. Once each data burst into the correlator chips is complete, the correlator chips must be dumped and stored. With a 2M word burst, at 256 MHz clock rate, the correlator chips must be dumped every 8 milliseconds. This is a high performance dump requirement and so it would seem that the memory should not be any smaller than 2M. The controller chip will also generate 4-bit quantized phase from the phase (PHASEMOD) and delay models (DELAYMOD—for fractional sample delay tracking) that it receives. Phase is generated

in real time and written into the recirculation memory bank with the data. This eliminates complicated rewind control of the phase generators and ensures that precise phase for each sample is available. Because of the presence of phase data, the number of basebands that recirculation can be performed on is limited to 4 (i.e. 4 x [4-bit data + 4-bit phase])—unlikely to be a practical restriction on the correlator’s capability. If recirculation is not active then the memory banks are not used and data and generated phase is sent directly to the correlator chips.

### Memory Operation and Limitations

Figure 2 is a simplified diagram showing some correlator chip lags and the derived recirculation memory start address algorithm.



**Figure 2** Simple example used to derive the recirculation start memory address algorithm. The top of the figure shows a 16-lag correlator with ‘X’ and ‘Y’ station data inputs. The samples moving through the shift register are shown relative to time  $t=0$ . Older samples are those that have moved further along the shift register. To derive the recirculation memory offset equations, these 16 lags are split into 4 sections (A, B, C, D) and then the *relative* offsets of the data going into each lag section are considered. The middle bottom of the figure contains the key start address recirculation algorithm derived from this simple example. Also, X and Y station recirculation memory start addresses are shown for lag section ‘A’.

From Figure 2 it is possible to obtain the memory address offset required for any arbitrary lag section of an arbitrarily large complete lag correlation. The address offset compared to the size of the memory buffer will determine how much of the data is actually used in the burst to the correlator chips compared to how much is available in memory. At some point a significant amount of data is discarded and the SNR of the

associated cross-correlation is degraded. Note that the amount of data discarded is highest at the outer lags and becomes small near the center lag—no matter how long the overall lag chain.

### **SNR Degradation**

If the actual lag hardware available is much smaller than the total number of lags that are to be produced, the SNR degradation is<sup>2</sup>:

$$SNR_{loss} \cong 1 - \sqrt{\frac{\text{data used}}{\text{total data available}}} \cong 1 - \sqrt{\frac{\text{Memsize} - \text{addr\_offset}}{\text{Memsize}}}$$

In the worst case, at the ends of the lag chain, the *addr\_offset* is about the same as the number of spectral points that will be produced and so:

$$SNR_{loss_{max}} \cong 1 - \sqrt{\frac{\text{Memsize} - \text{tot.\#of freq pnts}}{\text{Memsize}}}$$

For 2M *Memsize* and 262k frequency points, the maximum SNR loss is ~6.5%. As mentioned, this occurs only at the edge lag channels and gets progressively better moving towards the center. If the lag data is windowed before FFT, this noise effect is attenuated somewhat and, in the above example, may indeed be negligible.

An alternative to the dual-memory design and SNR loss discussed above is to use triple buffering—requiring three memories. This facilitates a design whereby not only is the start address offset controlled on memory readout, but the *time* when data is written into a buffer is also controlled. This means that for lags near the edges of the lag chain, we delay writing data into the memory so that the required address offset depends only on the number of lags that this particular controller is generating rather than the total number of lags being produced by several controllers. Triple buffering is required because with a time offset, and double buffering, the memory buffers for the X and Y stations will not be finished being written to at the same time—preventing readout and losing data. However, given the number of lags that can be produced with double buffering, the size of the memory that is required anyway so the correlator chip dump time is not excessively small, and the small SNR loss, triple buffering is probably not worth further consideration.

### **Integration Time**

The integration time for a complete set of lags is fundamentally limited by the time it takes to fill the recirculation memory<sup>3</sup> at the real-time sub-band sampling rate and by the time it takes for the recirculation controller to perform all of its bursts. Since the number of bursts that are performed is dependent on the number of lags to be generated—and

<sup>2</sup> This is different than the equation presented in [1] which is incorrect!

<sup>3</sup> Recall that the size of the recirculation memory determines the dump time of the correlator chip and it cannot be too small or the dump rate becomes excessive.

could be less than what real-time allows—it is the size of the memory that is the most important governing factor. Thus:

$$T_{\min} = \frac{\text{Memsize}}{\text{subband sample rate}}$$

Additionally, any integration time must be an integer multiple of  $T_{\min}$ . For example, with a sub-band bandwidth of  $1/256^{\text{th}}$  of 2.048 GHz (8 MHz bandwidth with a sample rate of 16 Ms/s), and a memory size of 2M words, the minimum integration time  $T_{\min}$  is 0.125 seconds. According to the *SNRloss* equation, up to 262144 frequency points<sup>4</sup> could be obtained without a significant SNR degradation. If the bandwidth is reduced even further to improve spectral resolution, then the minimum integration time will get even larger. This effect can be offset by reducing the burst rate into the correlator chips or by reducing the amount of buffer memory actually used. The construction of the correlator chip will not permit an effective burst rate reduction to happen: correlation occurs on every high-rate clock cycle whether or not there is a new sample. This limitation is required for correlator chip performance reasons. The amount of buffer memory that is used *can* be reduced provided the correlator chip dump time does not exceed its minimum capability. Additionally, including dead time between bursts allows more time for correlator chip readout—reducing the amount of memory that is used and  $T_{\min}$ . In this case there is a tradeoff between  $T_{\min}$  and the number of lags that recirculation yields.

### **Other Subtle Effects**

#### **Time Skew**

The recirculation design that has been described ensures that all of the lag elements in each of the recirculation bursts see the correct X and Y relative station delay. However, compared to an ideal lag correlator without recirculation, at any given time *the actual times* of the data within an integration time that are being correlated will be different. For example, consider the case of a 524288 lag correlation with an 8 MHz bandwidth using recirculation. In the ideal lag correlator, the samples at the center are 131072 samples old (-8 msec), but with recirculation the samples are maybe a few thousand samples old. As we move towards the outer lags, the time skew moves in the opposite direction compared to the ideal correlator. To see this, consider lag -6 in Figure 2. The ideal correlator has the X sample at a time of -7 and the Y sample at a time of -1. The recirculation correlator (going by address offsets) has the X sample at a time of -1 (address 0 -1 delay) and the Y sample at a time of 5 (address +6 -1 delay). It is believed that this effect—particularly the reduction in time skew near the center lags—should not be problematic. Indeed, it could be advantageous since integrated data will be more accurately time-tagged.

<sup>4</sup> Spread across several controllers in several sub-band correlators.

## Blanking

The current plan for the correlator chip is to effectively have a single data valid counter at the center lag. Data valid is used for flagging bad or invalid data and for pulsar gating and it must be counted for correct data normalization. For each recirculation burst it is necessary to have a correlation inhibit control so that correlation does not occur until the X and Y delay lines have filled with (new) data. This is an important requirement to prevent systematic biases in the output data since the data valid counter is used to normalize the integer accumulator counts (after accumulator bias removal) to floating point quantities. A data valid counter at each lag would yield the best performance but it may be too costly to implement in the correlator chip.

## VLBI

As mentioned in [1], if the Baseline Board is used for VLBI it is necessary to get the real-time fractional-sample station delay model (DELAYMOD) to the correlator chip so that the baseline-based “vernier” delay can be formed<sup>5</sup>. This delay must also be stored with the data (2-bits) and phase (4-bits) into recirculation buffer memory so the correct baseline delay can be formed at the correct time. Since this mode of VLBI operation only uses 2-bit data, there is plenty of width in a x32 memory to hold the extra DELAYMOD signal and an associated framing clock.

## Dynamic RAM

Because of recirculation memory size requirements (nominally 2M x 32—or 64 Mbits) it is *necessary* to use (synchronous) dynamic RAM (SDRAM). Synchronous static RAM is much easier to work with, but it is not possible to obtain the same capacity in a single chip package. Double data rate (DDR) SDRAM further reduces the number of memory chips required over single data rate (SDR) SDRAM since it is capable of effectively operating at the 256 MHz clock rate that is currently being contemplated for the system.

Dynamic RAM has two complicating factors over static RAM. First, so-called CAS latency effectively means that for every row address there is a 3 clock cycle<sup>6</sup> delay or latency while a new CAS address is loaded<sup>7</sup>. Since the CAS address is 8-bits long, what this means is that every 256 samples, there will be 6 invalid samples going to the correlator chip. These samples can easily be flagged as invalid and result in an additional SNR loss<sup>8</sup> of ~1.2%. A subtle systematic effect may be introduced here if there is only one data valid counter at the center lag: the data valid counter sees 6 invalid samples, whereas a data valid counter away from the center sees 12 invalid samples. Mitigating

---

<sup>5</sup> In the case where narrower (e.g.  $\leq 128$  MHz) basebands are recorded on tape and so fine delay tracking, as normally done with WIDAR, cannot be performed. If however, the wideband data is recorded on tape—in some sort of time-demultiplexed fashion—then this is not a requirement and the correlator operates normally. Although in this case, an additional 4-bit requantization loss (1.5% SNR degradation) is incurred.

<sup>6</sup> Where the clock cycle is 1/128 MHz.

<sup>7</sup> Once the CAS address is loaded, an on-chip counter generates the addresses for an entire CAS page.

<sup>8</sup> Which does not manifest itself as an amplitude loss.



this effect may require more data valid counters in the correlator chip. The second complicating factor with DRAM is the need to refresh the memory cells on a regular basis otherwise they will lose charge and lose memory. There are several refresh modes that can be used but ROR (RAS Only Refresh) is probably most appropriate for this application. During recirculation burst reading, ROR will occur naturally since all of the row addresses are accessed within the refresh cycle time of the chip. During buffer writing, it will be necessary to perform additional RAS accesses for refresh since it can take quite some time to access all of the memory locations on the SDRAM. Because of the write requirements of DDR SDRAM, a 256 x 32 secondary buffer will most likely be required: slow speed data is written into the secondary buffer and, once full, it is burst written into the SDRAM. When the secondary buffer is filling, RAS accesses to the SDRAM will keep it refreshed. The 256 x 32 secondary buffer may be implemented in the recirculation controller FPGA itself or it may have to be a separate high-speed memory device.

## Conclusions

This memo investigated recirculation architecture, requirements, and limitations in reasonable detail. This was done keeping in mind the capabilities and nuances of real-world electronics so that there is a high probability of actually implementing what has been discussed. A general memory address algorithm was developed using a simple 16-lag example. The fundamental limitations in minimum integration time and maximum lag length—as it affects SNR degradation—were discussed so that an appreciation of the bounds within which recirculation must operate is gained. Other subtle effects such as integration timestamp skew, correlator chip blanking, VLBI requirements, and dynamic RAM operating requirements were discussed. Dynamic RAM CAS latency appears to introduce a systematic bias in the data that can be mitigated—but possibly at the expense of increased correlator chip hardware.

## References

[1] Carlson, Brent, A Proposed WIDAR Correlator for the Expansion Very Large Array Project: Discussion of Capabilities, Implementation, and Signal Processing, NRC-EVLA Memo# 001, May 18, 2000

