Interoffice

VLBP Memo No. 15

National Radio Astronomy Observatory

Charlottesville, Virginia

July 14, 1981

To: VLBP Project Group

From: R. P. Escoffier

Subject: Parallel Correlator

This memo will describe an alternate approach in designing a large VLBI correlator system. As in my last memo, I will consider only the correlator portion of such a system ignoring such items as recorder control, data buffers, long term integration, etc. The last memo described a recirculating correlator in which astronomical data from either a small number of high data rate inputs or a larger number of lower data rate inputs were stored in a recirculating memory. This recirculating memory then played back the recorded data thru the correlator system multiple times at a high data rate, thus increasing the effective number of correlator channels to some factor of recirculation higher than the actual hardware channels provided. One disadvantage to the approach outlined in Memo No. 14 is that it requires a substantial wiring matrix to drive the many correlator cards with the proper ECL data signals. For example, a 256 MHz implementation of a 15 antenna correlator as described in Memo 14 would require something like 90 P.C. cards, each with a 5 X 1 matrix of 4 sin lag and 4 cos lag correlator circuits (40 correlators per card). Such cards would each require 11 256-MHz data inputs (excluding blanking terms) for a total data distribution of 990 256-MHz data interfaces (also excluding quadrature signal generation and other factors). A 128 MHz implementation would require twice the number of P.C. cards and hence twice this number of 128 MHz data interfaces.

Figure 1 illustrates how a parallel correlating approach would eliminate this high speed signal distribution problem. A parallel correlator that could, say, simultaneously multiply 16 pairs of inputs would reduce the interface data rate requirement to 16 MHz (of course, the number of data lines goes up by a factor of 16, but I feel the degree of difficulty still goes down significantly). Such an implementation would stay complete in the TTL logic domain and would probably result in cost reduction. Doing 16 pair of correlations simultaneously with enough hardware to generate 4 lags at a time (equivalent to 256 M bit data processing) or 8 pair correlations with enough hardware to generate 8 lags at a time (equivalent to 128 M bit data processing) turned out to require about the same number of IC's, so from this point I will consider only taking 8 bits at a time from the recirculator every 62.5 nsec.

For a parallel correlating element, a look-up table ROM seems to be the best approach although 62.5 nsec is pushing presently available TTL ROM's (but no more than 256 MHz pushes available ECL). A second advantage to a look-up table multiplier approach turns out to be circuit efficiency in fringe rotation and generation of sin and cos quadrature outputs. In principle a ROM could simultaneously multiply 8 pairs of bits and, if addressed also by the fringe phase, look up sin and cos results that would be numerically the same as results obtained by sequentially multiplying the 8 pairs of bits in a conventional serial fringe rotator and multiplier (the only exception to this statement would be instances where the 3-level fringe rotator states change over the 8 data bits considered). By this statement, I mean that over 8 bits a conventional correlator will develop two numbers N_S and N_C that are added respectively to the sin and cos integrations. Each N is either 0, 1, 2, 3, 4, 5, 6, 7 or 8, the result of eight 0 or 1 correlator products. A ROM given the 8 pairs of inputs and the fringe phase could look up these exact same numbers N_S and N_C previously worked out for all possible combinations and stored in the ROM. In reality, a better fringe rotator could be obtained since an 8 or 16 level fringe rotator could be efficiently designed instead of the present 3 level rotator. Thus, a parallel correlator approach might have some technical advantages as well as cost and engineering advantages.

Figure 2 illustrates the most efficient circuit, in terms of cost and IC count, for a parallel correlator I have been able to come up with. The 8 EX-OR gates multiply the 8 pair of inputs reducing the 16 inputs to 8 outputs that allow use of a smaller, less expensive ROM. This ROM then takes the 8 multiplier outputs and looks up the number N = 0, 1, 2, 3, 4, 5, 6, 7 or 8 that corresponds to the number of input pairs that were at the same logic level. The second ROM takes this 4 bit N and applies the fringe phase to it to look up N_S and N_C, the numbers to be added into the sin and cos integrators. The reason for using two stages of ROM's is to take advantage of the low cost 256 X 4 and 256 X 8 PROM's available ($2 and $4 each respectively). In the future, larger high-speed PROM's might be available that will make a one ROM circuit cost effective. Another possibility at this time is a "PAL" (programmable array logic) circuit to replace the EX-OR and first ROM stages of Figure 2.

Blanking will produce some inefficiency in that if any of the 16 bits involved in a single integration must be blanked, all 8 pairs must be discarded. However, since the fringe rotation blanking terms have disappeared, the only blanking terms left, headswitching, drop-outs, etc., should be of a sufficiently long term nature that this added inefficiency will not be significant.

The only large problem I have discovered thus far in studying such a correlator design is in data integration. The correlator results come out so fast that I have not been able to come up with an IC efficient way of integration. The large number of IC's required (although inexpensive IC's) tends to indicate a larger system than any ECL configuration I have studied (90 correlator cards at 256 MHz, 180 cards at 128 MHz, and 450 cards with this parallel approach). Probably the best way to solve the size problem would be a semi-custom approach. There are now several companies that offer semi-custom IC's that consist of uncommitted gate arrays to be wired together via the last stages of IC metalization to implement a customer's specific logic requirement. I feel that the trend to such circuits will result in even more cost attractive circuits in the future and that any correlator design, but in particular this unfortunately large parallel correlator, will benefit from semi-custom circuits.

One additional feature of the parallel correlator I have presented so far needs to be explored. Since no element in the ROM correlator possesses storage, there is no real reason for having the recirculator. If the incoming data were presented to the correlators instead of being written into the recirculating memory, proper correlator operation still takes place. Such a system then becomes a time-shared correlator instead of a recirculating correlator. The problem to this approach is that contiguous correlations yield results that must be integrated into different integrator locations. In the Mark III case, for example, products from the 28 tracks will be produced sequentially, one each 62.5 nsec, so that in the time it takes to shift in 8 bits per track at the 4 MHz data rate, 28 4-bit correlation results must be integrated into the 28 track integrators per baseline (times 2 for sin/cos). This requirement multiplies the integration problem already mentioned by a factor of 32. Logic is available to perform the required integration but the IC count and cost make this approach totally impractical for the present and thus, for the present at least, the recirculators must stay. If in the future, however, a semi-custom IC, of sufficient density, becomes available, a significant simplication to the system can occur by elimination of the recirculators.

Figure 3 will give some comparison of the cost of a parallel correlator relative to a high-speed ECL correlator. These two tables present circuits that do essentially the same thing and give their respective IC counts and IC costs. Many factors that will affect the overall cost of the two approaches, such as the P.C. card costs, the fringe rotation logic, etc., are not considered here so these tables may be of limited use. However, they give some idea of the relative costs involved.

Attachments

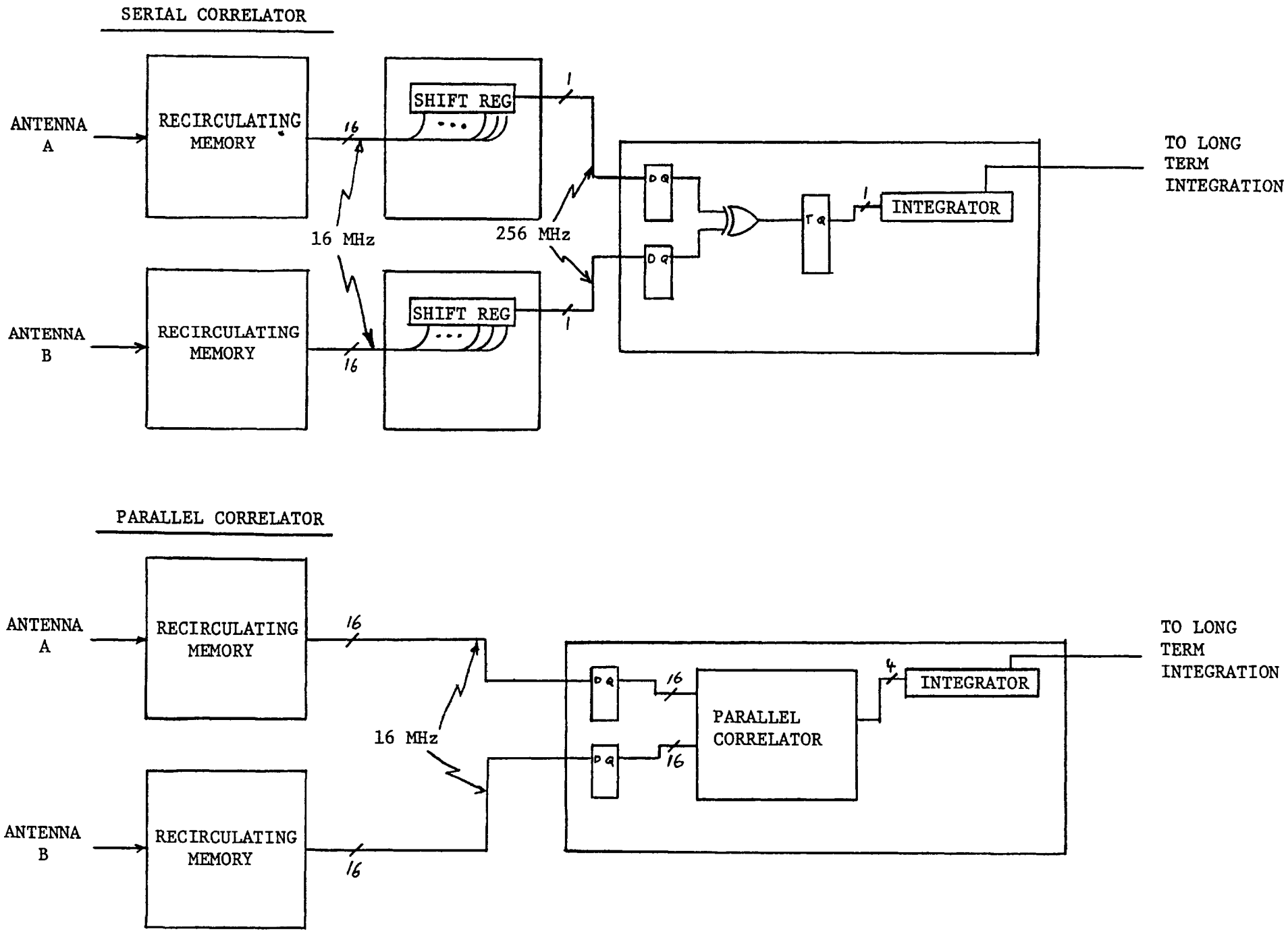
SERIAL CORRELATOR
ANTENNA A
RECIRCULATING MEMORY
16
SHIFT REG
1
16 MHz
256 MHz
ANTENNA B
RECIRCULATING MEMORY
16
SHIFT REG
1
D Q
D Q
T Q
1
INTEGRATOR
TO LONG TERM INTEGRATION
PARALLEL CORRELATOR
ANTENNA A
RECIRCULATING MEMORY
16
16 MHz
ANTENNA B
RECIRCULATING MEMORY
16
D Q
D Q
16
16
PARALLEL CORRELATOR
4
INTEGRATOR
TO LONG TERM INTEGRATION

FIGURE 1

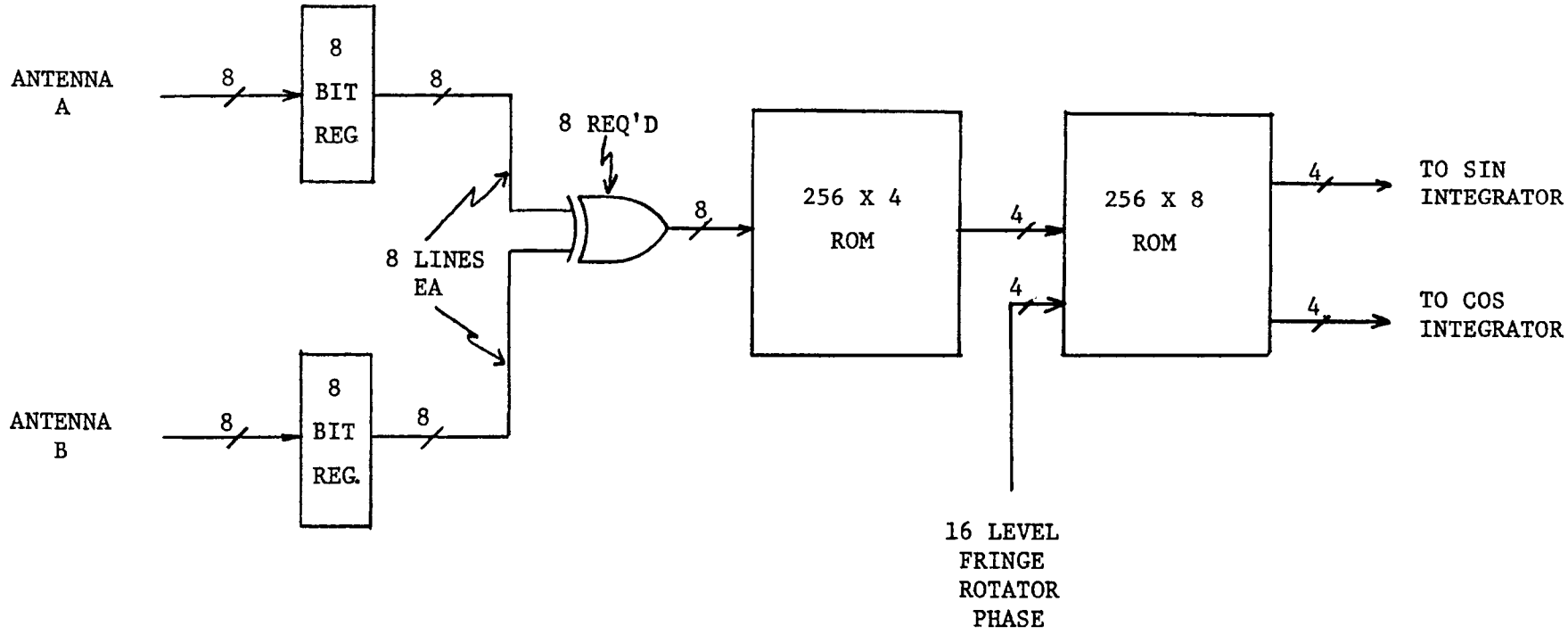
ANTENNA A
8
8 BIT REG
8
8 REQ'D
8 LINES EA
8
256 X 4 ROM
4
256 X 8 ROM
4
TO SIN INTEGRATOR
4
TO COS INTEGRATOR
ANTENNA B
8
8 BIT REG.
8
4
16 LEVEL FRINGE ROTATOR PHASE

FIGURE 2

Conventional Serial Correlator

(4 sin and 4 cos channels at clock rate of 256 MHz)

IC	Quan.	Price
100141	1	$ 17.65
100107	4	6.40
100151	2	12.05
10231	4	2.83
10125	2	1.37
VLA-2	8	10.00
	21	$160.

Parallel Correlator

(8 sin and 8 cos channels at 16 MHz clock rate)

IC	Quan.	Price	Possible PAL/Custom Design
8 bit reg.	3	$ 1.11	3
Ex-or	16	.37	8
256 X 4 PROM	8	2.05	
256 X 8 PROM	8	3.96	8
Pipeline reg.	12	1.11	4
8 bit reg.	16	1.11	8
Adder	16	.72	
	79	$100.	31

FIGURE 3