Low-Power Architectures For Large Radio Astronomy Correlators

Larry R. D'Addario[1]

[1]Jet Propulsion Laboratory, California Institute of Technology
M/S 238-333, 4800 Oak Grove Drive, Pasadena, California 91109, USA
Telephone: +1/818/393-0389. Email: ldaddario@jpl.nasa.gov.

Abstract

The architecture of a cross-correlator for a synthesis radio telescope with $N > 1000$ antennas is studied with the objective of minimizing power consumption. It is found that the optimum architecture minimizes memory operations, and this implies preference for a matrix structure over a pipeline structure and avoiding the use of memory banks as accumulation registers when sharing multiply-accumulators among baselines. A straw-man design for $N = 2000$ and bandwidth of 1 GHz, based on ASICs fabricated in a 90 nm CMOS process, is presented. The cross-correlator proper (excluding per-antenna processing) is estimated to consume less than 35 kW.

1. Introduction

Power consumption is a major limitation in the implementation of a very-large-N radio telescope, where $2N$ is the number of signals that must be cross-correlated (N antennas or stations, each receiving 2 polarizations). When N is sufficiently large, power consumption is dominated by the signal processing electronics for cross-correlation, which grows as N^2 while nearly everything else grows as N. In this paper, "very large" means $N > 1000$. The proposed Square Kilometer Array [1] will require such a correlator for its mid- and high-frequency components.

Considerable care must be taken in the design of a correlator for a very-large-N array if the power consumption (and perhaps also monetary cost) of that correlator is not to be a major obstacle to its construction and to the operation of the whole array. At its core, a correlator performs multiply-accumulate (MAC) operations on all pairs of input signals. The rate at which it does this is $N(2N+1)B$ (assuming complex signals and Nyquist-rate sampling), and that rate is the same regardless of architecture. However, a practical correlator must do other operations as well. In particular, it needs input and output (I/O) operations for external connections as well as for internal signals (since a large correlator cannot fit in a single device or box), and it needs memory operations because in most architectures it must store data temporarily in buffers. Memory and I/O operations typically use much more energy than MAC operations, and their rates are strongly dependent on architecture.

This paper considers only the cross-correlator proper, ignoring the processing that must be done on the signals separately before they are combined. This is because cross-correlation is the dominant power consumer at very large N. The cost of re-ordering samples between the filter banks and the cross correlator ("corner turner") is also ignored on the assumption that a memoryless corner turner [3] with negligible power consumption can be used. It is assumed that the per-signal processing includes a uniform filter bank so that the cross-correlation work can be partitioned by frequency ("FX" style). At the correlator's inputs, each signal is assumed to be a discrete-time sequence of coarsely quantized samples, each consisting of the real and imaginary parts of a complex number. The correlator then performs complex MAC (CMAC) operations.

2. Correlator Architectures

Figure 1 provides a classification of correlator architectures as a hierarchical tree, leading to five distinct architectures. At the top level, we distinguish between a matrix structure, where CMAC elements are arranged in a triangular array, and a pipeline structure, where CMAC elements are arranged in a chain with delays between them, as illustrated in Figure 2. In the matrix structure, it is possible for each signal pair (baseline) to be connected to a dedicated CMAC. This is the first of the architectures (#1). In all the others, each physical CMAC is time-shared among baselines so as to reduce the number of CMACs that must be built. This is possible if each CMAC can operate faster than the signal bandwidth that it is processing. Thus, if the bandwidth processed is b, where a filter bank has partitioned each antenna element's bandwidth B into multiple channels of bandwidth b, and a CMAC operates at clock rate $f = xb$, then that CMAC can be shared among x baselines.

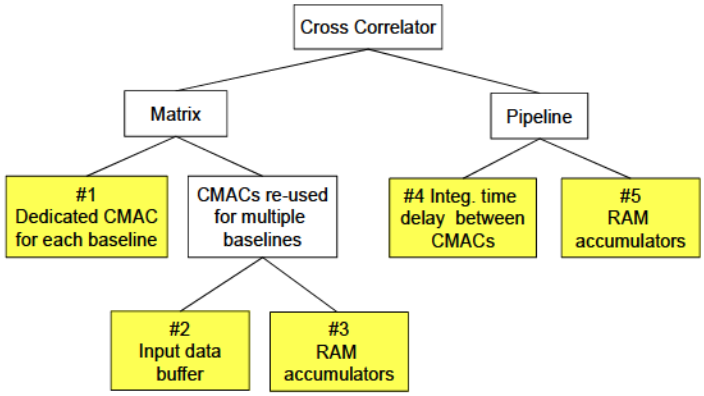

Figure 1: Correlator architecture tree.

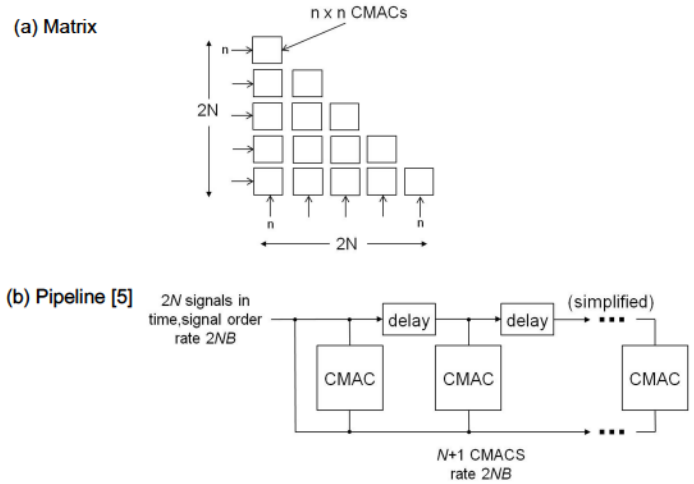

Figure 2: (a) Matrix structure. (b) Pipleline structure.

There are two ways of arranging the sharing of a CMAC among baselines. First, the input samples from all antennas of the shared baselines can be written to a memory that holds enough samples for one correlation (i.e., one short-term integrating time of the correlator). Samples for one pair of antennas are then read from the memory at rate xb and sent to the CMAC. At the end of the integrating time, the CMAC's accumulator is read out and then cleared. Then samples for a different pair of antennas are read from the memory and correlated, continuing in this way until all x pairs have been correlated. The second method involves providing a bank of accumulation registers for each CMAC. The CMAC uses a different accumulation register on each clock, so its inputs can be from a different pair of antennas on each clock and no input buffer is required. The most practical way to do this is to place the accumulation registers in a random-access memory (RAM) of depth x and to increment the RAM's address on each clock. After x clocks, the address returns to its original value and the input signals return to the original antenna pair, enabling the next product for that baseline to be accumulated. After enough cycles have elapsed to complete an integration time, results for all x baselines are available in the RAM and can be read out. The methods correspond to architectures #2 and #3 in Fig. 1.

In the pipeline structure, essentially the same two methods of CMAC sharing can be used, and these correspond to architectures #4 and #5 in Fig. 1. The pipeline concept is described in [4]. It involves sending samples from all $2N$ signals to the chain of CMACs serially, on a single bus, as shown in simplified form in Fig. 2b. It turns out that a chain of N+1 CMACs is sufficient to compute correlations for all $N(2N+1)$ pairs. On each clock, N+1 products are computed in parallel. After all $2N$ blocks of samples have been received, $2N(N+1) = N(2N+2)$ sets of products will have been computed; N of them are redundant. The sharing factor x is thus fixed by the architecture at $2N$. The difference between architectures #4 and #5 is in the length of the delay between CMACs. In #4, the delay is equal to the number of samples in one integration, so the block of samples from each signal must also be of that length. After each block, the accumulators of all CMACs are read out and cleared. In #5, each delay is for only one clock, implemented as a single register. A sample from a different antenna is received on each clock, and each CMAC has a bank of accumulators implemented in RAM as in #3. (This was first proposed by Sigman [5].)

For any of the architectures, if the correlator's clock rate is fast enough, then it can also be shared among multiple frequency channels. This is best done by processing samples from one channel for one integration time, accumulating results for all baselines, reading out those results, then processing the same number of samples from another channel. If the clock rate is xKb, then one CMAC can process x baselines and K channels (or bandwidth Kb) per integration interval. However, unlike the sharing of CMACs among baselines, we do not regard their sharing among channels as creating distinct architectures.

3. A Straw-Man Design

To evaluate the various architectures, consider a hypothetical correlator with these parameters: $N \approx 2000$, B = 1.0 GHz, b = 100 kHz, beams per antenna J = 100, minimum integrating time τ = .065 s, input sample quantization w_s = 4 (2 bits real + 2 bits imaginary). This is reasonably representative of the requirements for SKA-mid [1].

The value of J has no effect on the architecture, since a separate correlator must be built for each beam; we consider a $J = 1$ design and assume that it will simply be duplicated J times. We attempt to find a practical design for this correlator in each of the five architectures separately.

For such a large instrument the basic building block must be an application-specific integrated circuit (ASIC). A different ASIC is needed for each architecture, but for a fair comparison they should all be built using the same process. For this we choose CMOS with 90 nm gate length because it is readily available today from various foundries. The required ASICs have not been designed in detail, but instead are described by simple models based on the die area for each major component and the energy used per operation of that component. Three basic components are needed: CMACs, input/output (receivers and transmitters), and memories. The model parameters used are given in Table 1. These were derived in several ways. For CMACs, three existing correlator ASICs (GeoSTAR [6-7], ALMA [8-9], and EVLA [10-11]) were scaled from their original designs to the target 2b+2b CMACs and, if necessary, from their original process to the target 90 nm CMOS process. The results for all three were similar; the worst values were used in the model. For I/O, we assumed multi-Gb/s transceivers of the type size and power reported in [12]. For embedded memories, we relied on a simulation tool [13]. Details of the parameter derivations are given in [14]. Three constraints were imposed on all ASIC designs: Total die area was set to ≤ 200 mm^2, to ensure reasonable yield; power dissipation was set to ≤ 75W to allow air cooling; and input and output rates were each limited to 40 Gb/s, mostly to ensure reasonable board-level I/O with many ICs per board.

Table 1: Element Model Parameters For 90 nm Process Technology

Element (architecture)	*Die Area* mm^2	*Energy/op* pJ	*Static power* W	*Max. clock* MHz
AC (all)	0.006303	2.45	0	N/A
Transceiver, 6.25 Gb/s (all)	0.266	2.12	0	N/A
DRAM 600x177.5k (#2)	48.7	3750	0.593	127
SRAM 32x100 (#3)	.0130	1.87	.000143	1165
SRAM 4x6500 (#4)	.0488	2.97	.000102	695
SRAM 32x4000 (#5)	0.2428	14.0	.00499	661

Several things are neglected and simplified here. As mentioned, only the cross-correlator proper is considered. Also neglected is certain infrastructure, such as power supplies and control circuits. These can be implemented with off-the-shelf parts and should add only a few percent to the IC count, but they may add as much as 30% to the power consumption. All inter-chip I/O is assumed to be implemented in the same way (with high-speed serial links), without distinguishing board-to-board from inter-board connections.

Results are given in Table 2. For all architectures, the power dissipated by each IC remained far below the 75 W maximum because some other limit was reached first. Architecture #1 has the lowest system power because it requires no memories; for the others, almost all of the additional power is consumed by memory operations. For #3, which is a matrix structure with RAM accumulators, and for both pipeline structures (#4 and #5), the required memories not only dominate the power, but they dominate the die area and limit the number of CMACs that can be implemented in each IC. In Architecture 2, an input buffer is needed, but it uses only 26% of the die area and 19% of

Table 2: Summary of Designs By Architecture

Architecture:	*1*	*2*	*3*	*4*	*5*
Number of antennas, N	2024	2025	2040	2000	2000
CMACs/IC, m	30976	22500	10404	2001	667
Channels/IC, K	284	3	24	1	1
Clock frequency, f, Hz	2.84E+07	1.13E+08	2.40E+08	4.00E+08	4.00E+08
Input rate/IC, b/s	4.00E+10	4.86E+09	3.92E+10	6.40E+09	6.40E+09
Output rate/IC, b/s	4.33E+09	1.26E+10	1.23E+10	1.03E+10	7.71E+09
Power/IC, W	2.25	7.73	17.05	11.49	11.50
Die area, mm^2	195.24	190.50	200.51	208.06	166.18
Memory power, %	0.0%	18.7%	63.5%	82.6%	94.1%
I/O power, %	4.2%	0.5%	0.6%	0.3%	0.3%
Memory area %	0.0%	25.6%	67.3%	93.9%	97.5%
ICs to process all baselines, c_1	276	1	8	1	3
Total ICs in system, c	9,718	3,333	3,333	10,000	30,000
Total power, all ICs, W	21,859	25,771	56,823	114,873	344,873

the power. The buffer is implemented as a single large memory, and it is constructed as dynamic RAM in order to achieve high density in spite of a penalty in speed and static power compared with static RAM. (This was not possible for the other architectures.) Its system power is 18% higher than that of #1 but it uses 1/3 the number of ICs and its I/O rate 61% less. For the latter reasons, #2 is considered the best choice for our straw-man correlator, at least for an implementation in 90 nm CMOS under the adopted constraints.

To estimate the complete power consumption of the straw-man correlator, we add 35% to the 25.8 kW in Table 2 to account for power supplies, monitor/control, other infrastructure, and contingency, giving 34.8 kW.

5. Conclusion

It has been shown that minimizing power consumption in a cross correlator requires choosing an architecture that requires few memory operations. This means that the use of RAM accumulators should be avoided, and that the matrix structure is much better than the pipeline structure. Reasonable power (<35 kW/beam) for $N = 2000$ and $B = 1$ GHz can be achieved in 90 nm CMOS. This is about 200 times less than if it were built with the architecture and technology used for the ALMA correlator, the largest radio astronomy correlator to date. Part of this is the result of using a more modern technology (90 nm vs. 250 nm CMOS), but much of it is due to selecting the optimum architecture. Additional details of this work can be found in [14].

6. Acknowledgments

JPL is operated by the California Institute of Technology for NASA. This work was supported by a grant of internal JPL research funds. Ray Escoffier and Brent Carlson are thanked for providing unpublished details of the ALMA and EVLA correlator designs, respectively.

7. References

1. R. T. Schillizzi et al., "Preliminary Specifications for the Square Kilometre Array." SKA Memo 100, December 2007. Available: http://www.skatelescope.org/PDF/memos/100_Memo_Schilizzi.pdf.

2. R. Escoffier, private communication (email of 29 January 2010).

3. L. Urry, "A Corner Turner Architecture." ATA Memo 14, 17 November 2000. Available: http://ral.berkeley.edu/ata/memos/memo14.pdf. See also ATA Memo 22, 16 March 2001: http://ral.berkeley.edu/ata/memos/memo22.pdf.

4. L. Urry, "The ATA Correlator." ATA Memo 73, 16 February 2007. Available:

http://ral.berkeley.edu/ata/memos/memo73.pdf.

5. E. Sigman, private communication (April 2010).

6. B. Lambrigtsen, A. Tanner, T. Gaier, P. Kangaslahti, S. Brown, "A Microwave Sounder for GOES-R: Developing the GeoSTAR Mission." Geoscience and Remote Sensing Symposium, 2006.

7. Todd Gaier, private communication, 2010.

8. National Radio Astronomy Observatory, http://www.alma.nrao.edu/development/correlator/.

9. R. Escoffier, private communication (email of 1/29/2010).

10. B. Carlson, "Requirements and Functional Specification: EVLA Correlator Chip." NRC Canada, Hertzberg Inst. of Astroph., Dominion Radio Obs., RFS Document A25082N0000. Revision 2.5, January 20, 2010.

11. Brent Carlson, private communication (email of 7/16/2010).

12. R. Palmer, J. Poulton, W. J. Dally, J. Eyles, A. M. Fuller, T. Greer,M. Horowitz, M. Kellam, F. Quan, F. Zarkeshvari, "A 14mW 6.25Gb/s Transceiver in 90nm CMOS for Serial Chip-to-Chip Communications." IEEE Solid State Circuits Conference, 2007.

13. HP Labs, CACTI simulation software for embedded memories. Available: http://quid.hpl.hp.com:9081/cacti/.

14. L. D'Addario, "Low-Power Correlator Architecture For the Mid-Frequency SKA." Submitted to SKA Memo Series, 2011 February 19. http://www.skatelescope.org/pages/page_memos.htm.