CARMA
Combined Array for Research in Millimeter-Wave Astronomy

CARMA Memorandum Series #7

CARMA/SZA Baseline Partitioning and FPGA Configuration Solutions

Kevin P. Rauch

University of Maryland

May 30, 2003

ABSTRACT

Optimal fanout-minimized baseline-to-correlator card partitioning solutions for the 15-station CARMA and 8-station SZA arrays are described, as well as the method used to obtain them. For CARMA, implementation of the new solution requires 11 digitizer cards per band assuming a digitizer-only fanout solution, or 8 digitizer cards and 5 digital fanout boards assuming the latter approach is used. The digital fanout boards require either 3 or 4 outputs depending on whether data is routed between correlator FPGAs at 125 MHz or 62.5 MHz, respectively. The boards should be able to use the same physical form factor as the current COBRA digitizer cards. The new CARMA solution requires only 3 distinct correlator FPGA configurations, for which board-level FPGA routing solutions are provided. No additional FPGA configurations are needed for the SZA correlator. Implications for the next-generation CARMA correlator are also discussed.

Change Record

Revision	Date	Author	Sections/Pages Affected
	Remarks		
1.0	2002-Dec-03	Kevin Rauch	
	Initial release.		
1.1	2003-May-30	Kevin Rauch	mainly Sec. 2,3
	Added routing diagrams for 62.5 MHz inter-FPGA buses and generalized discussion accordingly.		

1. The Interim CARMA Correlator

Beasley, Woody, and Hawkins (2002) have outlined a plan for the CARMA first-light correlator based on the existing COBRA design. In this plan additional digitizer and correlator cards, identical in design to the latest revision COBRA hardware, would be used to implement an interim correlator for CARMA with 4 GHz total bandwidth (8 bands of 500 MHz each). It is to be replaced later by a redesigned, "final" correlator. The SZA correlator will also be based on COBRA hardware. One of the relevant issues for the interim CARMA correlator is that of telescope data fanout. The COBRA digitizer cards contain two inputs and four outputs each; a minimum of 8 cards (per band) is therefore required for a 15-station CARMA array. However, the fanout limit of four is insufficient to allow calculation of all 105 baselines; additional fanout hardware is needed. To minimize hardware costs it is therefore necessary to determine the lowest possible fanout required to feed the 11 correlator cards per band needed for CARMA (assuming a maximum of 10 baselines per card). Given equal fanout ratios, it is also desirable to minimize the number of distinct FPGA routing configurations required to implement the baseline partitioning scheme. It is assumed that data demux-by-8 (16-bits at 125 MHz) is used to transfer data from the digitizer to the correlator cards (i.e., each 32-bit digitizer output contains both input telescope data streams).

2. Fanout- and Configuration-minimized Partitioning Solutions

A number of hand-crafted baseline-to-correlator mappings for CARMA have been examined, by both D. Hawkins and myself. Here a more systematic, computer-aided approach is taken, which allows the minimum possible fanout ratios to be determined. The number of FPGA configurations needed for each solution was also automatically computed and minimized (while keeping the overall fanout constant). Applying this approach, new solutions possessing superior fanout and configuration properties were found.

The method used is based on the basic-block decomposition proposed by D. Hawkins for use in the next-generation CARMA correlator. In the latter, baselines are partitioned into a uniform set of 2 x 2 blocks covering the entire space (blocks along the diagonal contain only three active baselines). For the current analysis three types of basic blocks are used, as shown in Figure 1: 2 x 2 blocks, 2 x 1 blocks, and 1 x 1 blocks. The advantage of this decomposition is that each 2 x 2 block requires only two input cables to calculate, which minimizes fanout at the basic-block level. Note also that the 1 x 1 blocks represent the cross-correlations between telescopes sharing the same input cable; these have no impact on fanout and can be calculated by any correlator card receiving the corresponding cable as input. The 2 x 1 blocks constitute the set of baselines involving telescope **F**, which is special in that it does not share its data cable with another telescope.

Since the 2 x 2 blocks total 21 in number, precisely 10 out of the 11 correlator cards must contain 2 such blocks, the 11th taking the remaining one. The critical strategy for minimizing fanout is to ensure that each pair of blocks in the first 10 cards share one of their inputs (i.e., one of the data cables is used by both blocks); since only three inputs are needed for those 8 baselines, each card will have one free input remaining. This allows telescope **F** to be input as well, and one of three 2 x 1 blocks to be calculated (filling the card to the 10

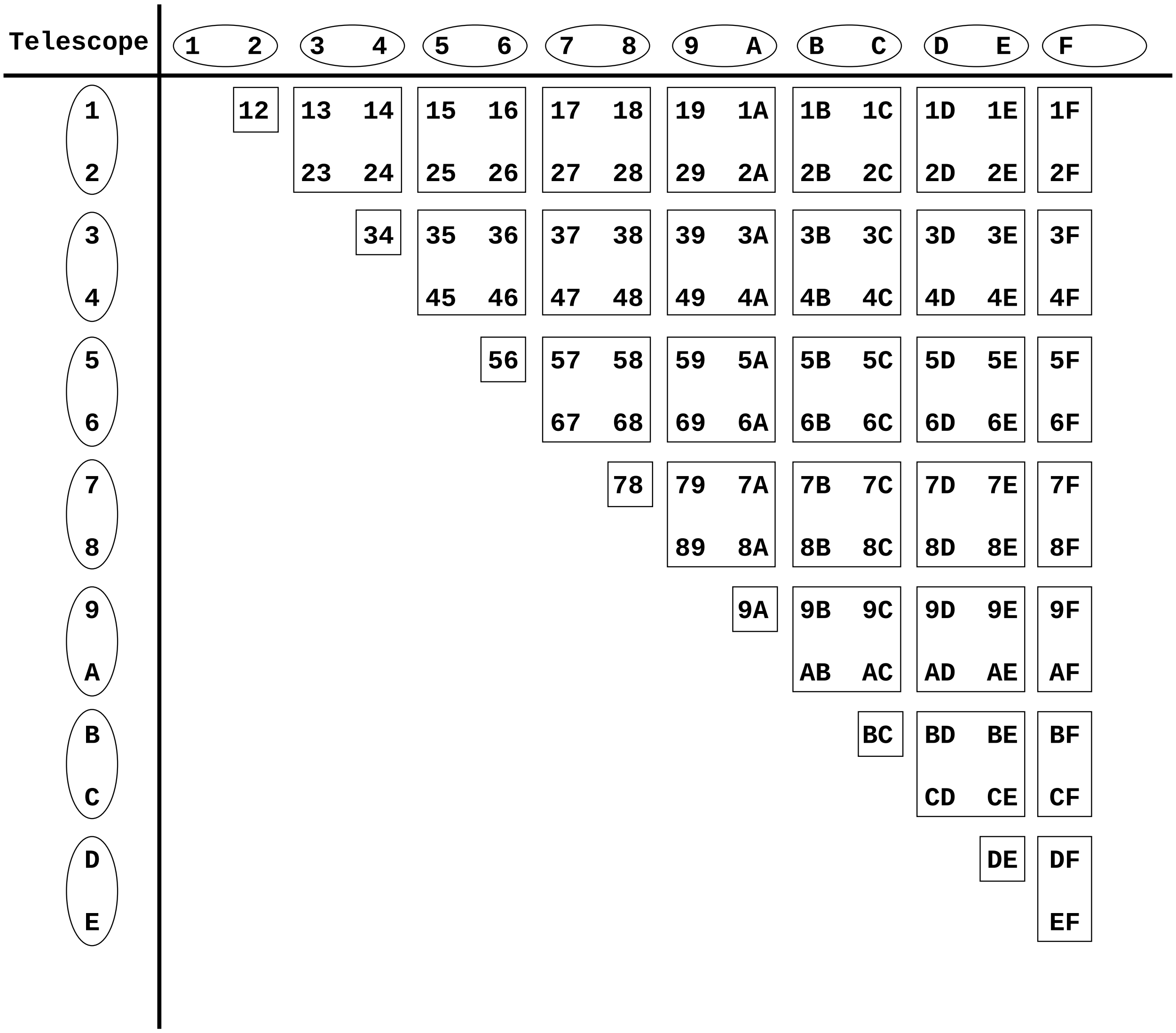

Fig. 1.— The basic-block decomposition of baseline space underlying the fanout-minimized partition solutions.

baseline maximum); alternatively, one or two 1 x 1 blocks (which never require an additional input) can be calculated. Applying this strategy yields a systematic method for generating "good" partitioning solutions, all of comparable quality to the best ones derived by hand using a more heuristic approach. However, there are still far too many possible combinations to check by hand—to have confidence that the global minimum fanout solution has been found, a computer-assisted search is needed. The stated rules for creating "good" mappings are easily coded. Adding automatic calculation of the required number of FPGA configurations (also easy to do in this approach) allowed exploration of this quantity as well.

In principle there are two kinds of fanout to optimize for, according to the type of hardware solution to be used (digitizer-only or digital fanout board). An additional constraint in the fanout board case is that all input to a correlator card must come either exclusively from fanout boards, or exclusively (i.e., directly) from digitizer cards; this is necessary to preserve phase between the input signals. The computer search tracked results for each hardware type separately. It turns out that many solutions exist which minimize both kinds of fanout simultaneously—as well as the number of FPGA configurations required. The one easiest to interpret visually is shown in Figure 2. This mapping requires 11 digitizer boards per band for a digitizer-only solution. For the digital fanout board solution, five fanout boards per band, each with three outputs, are needed; including the input, four LVDS connectors per board are therefore sufficient: the fanout boards can use the same physical form factor as the current COBRA digitizer cards. The number of outputs per fanout board was a free parameter in the simulations; three turns out to be the smallest number minimizing the total number of boards needed. In other words, the number of fanout boards required cannot be reduced below five by increasing the number of outputs per board above three; the fanout requirements for a 15-station array are simply too modest. The solution in Figure 2 contains only three distinct FPGA routing configurations; these are detailed in the following section.

Note that while we have determined the fanout *necessary* to feed the correlator cards, this fanout may not be *sufficient* (or feasible to implement) when the implied fanout within the correlator card itself is taken into account. It will be shown in the next section that when demux-by-8 (16-bits at 125 MHz) is used within the correlator cards for data transfer between FPGAs, the preceding minimum fanout solution can indeed be implemented. When demux-by-16 (32-bits at 62.5 MHz) is used, however, implementation demands one of the five fanout boards to have 4 outputs instead of 3. Since initial testing indicates that inter-FPGA buses cannot run reliably at 125 MHz, it is likely that the fanout boards will be a 4-output design. A 4-output fanout board conforming to the current COBRA form factor requires the use of "stacked" output connectors but is expected to be practical.

A corresponding optimal solution for the 8-station SZA array is shown in Figure 3. Note that since the number of telescopes is even, the basic-block decomposition contains only 2 x 2 and 1 x 1 blocks. The three correlator cards needed to calculate the 28 baselines all use the same basic FPGA configuration, and it is one of the three configurations already present in Figure 2—using the new solutions, no additional FPGA configurations are needed for the SZA.

Telescope	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1		12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2			23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
3				34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
4					45	46	47	48	49	4A	4B	4C	4D	4E	4F
5						56	57	58	59	5A	5B	5C	5D	5E	5F
6							67	68	69	6A	6B	6C	6D	6E	6F
7								78	79	7A	7B	7C	7D	7E	7F
8									89	8A	8B	8C	8D	8E	8F
9										9A	9B	9C	9D	9E	9F
A											AB	AC	AD	AE	AF
B												BC	BD	BE	BF
C													CD	CE	CF
D														DE	DF
E															EF

Fig. 2.— An optimum fanout- and configuration-minimized partitioning scheme for the 15-station CARMA array; the two dashed boxes go together in one correlator card. The first ten telescopes (**1-A**) require a fanout of 4; the last five (**B-F**), a fanout of 6.

3. FPGA Routing Configurations

It is straightforward to see that every (15-station) partitioning solution based on the preceding basic-block decomposition (Figure 1) contains at most six distinct FPGA configurations. One of these is for "card 11", the one containing only one 2 x 2 block. For the other 10 cards, which contain two 2 x 2 blocks, there are five possibilities. If the 2 x 2 blocks in the card share an input, then either the fourth input is unused (resulting in two sub-cases, depending on whether the 1 x 1 block corresponding to the shared input is calculated or not), or it is connected to telescope **F** (two more cases, depending on whether the 2 x 1 block calculated uses the shared input). The final case occurs when the 2 x 2 blocks *don't* share an input; here all four inputs are used (and equivalent), and the 1 x 1 blocks (if any) calculated by the card can all be accommodated by a single configuration. For fanout-minimized solutions the last case does not occur—for them, five configurations is the maximum. Similar reasoning shows that the minimum possible number of configurations is three; the solution in Figure 2 is therefore optimal in this regard as well.

Although several configurations are needed for any particular 15-station solution, they are closely related since the 2 x 2 blocks account for 84 of the 105 baselines. What varies between configurations are the 2 x 1 and 1 x 1 blocks, but at most two FPGAs per card are devoted to these (except for card 11). The basic task in developing a configuration is to route the signals to be cross-correlated to the appropriate FPGA with identical delays relative to the front-panel inputs. Although delay registers are sometimes needed to complete alignment of the data, their use should be minimized to avoid wasting logic cells. Another constraint particular to COBRA is that there is only one 32-bit bus connecting FPGA #5 to FPGA #6 (all others have two).

A useful concept for determining how to minimize the use of delay registers is that of input parity. Define the parity of two telescope signals to be even or odd according to whether the front-panel connectors they are attached to are separated by an even or odd number of inputs; for example, the signals sharing a single data cable are of even parity with respect to each other, while the parity of signals on adjacent connectors is odd. It is easy to show that signals of even parity can always be routed to an FPGA—arriving with the same net delay—without using any delay registers, whereas signals of odd parity *always* require them (two, to be precise). Thus to minimize the need for delay registers, the two cables defining a 2 x 2 block should have even parity whenever possible. For cards containing 2 x 2 blocks that share an input, it is also advantageous for the shared input to be attached to one of the card's middle two connectors; this aids in fanning out the "high-demand" signals across the board.

The rate at which data can be transferred between correlator FPGAs provides an additional constraint on the routing solutions. Two synchronous, board-level clocks, with frequencies of 125 MHz and 62.5 MHz, are available on COBRA correlator cards. Thus a single 32-bit inter-FPGA data bus can carry either two demux-by-8, 125 MHz telescope signals (each 16 bits wide) or a single demux-by-16, 62.5 MHz signal (32 bits wide); the former doubles the potential fanout available within a correlator card at the price of much tighter timing requirements compared to the latter. Although current testing indicates that the timing requirements for 125 MHz buses cannot be met in practice, routing configurations for both 125 MHz and 62.5 MHz buses are included for comparison.

Routing configurations based on these principles, for the three basic FPGA configurations needed to implement Figures 2 and 3, are given in Figures 4-6 (for 125 MHz inter-FPGA buses) and Figures 7-9 (for 62.5 MHz buses).

4. Application to the Next-Generation CARMA Correlator

D. Hawkins has proposed a plan for a final (23-station) CARMA correlator based on partitioning all baselines into 2 x 2 blocks. The favored correlator card layout appears to be one with 8 FPGAs arranged in a 2 x 4 array, enough to calculate two such blocks. For a 15-station array 14 cards are required, leading to an overall efficiency (fraction of FPGAs used) of 105/(14*8) = 93.75%. A major advantage of the proposed plan is that only one FPGA routing configuration is needed, allowing a great deal of flexibility in the creation of subarrays—there is only one type of basic block, which the card layout is designed to match. However, efficiency suffers somewhat for subarrays containing an even number of telescopes; an 8-station SZA subarray, for example, requires five cards in this approach, for an overall efficiency of 28/(5*8) = 70%. (Improving this efficiency requires the use of an additional FPGA configuration.) An alternative card layout, suggested by the preceding analysis, is shown in Figure 10. In this design, each card would contain 9 FPGAs laid out in a 3 x 3 array. All inputs are of even parity in this design (aside from the dashed input), and the data routing topology is simple. It is assumed that the central FPGA is twice as dense as the outer eight; the latter would process two 2 x 2 blocks, while the central FPGA would compute a 2 x 1 block (using the dashed input) or two 1 x 1 blocks. As shown below, this approach both reduces switchyard fanout and increases overall efficiency relative to an 8-baseline card design. Note that a double-density central FPGA will not complicate the design if it is pin-compatible with the remaining ones.

Since some reconfiguration of the switchyard output (and probably the Linux crate software) is required for each distinct subarray, reinitializing correlator cards at the same time is not a concern in itself; what does matter is that the set of FPGA configurations be fixed (and small), regardless of the size of the array. By induction, it is easy to see that precisely the same three configurations detailed in the previous section are sufficient to partition an array containing an arbitrary number of telescopes; hence explosion of configurations is not an issue in this approach. Notice also that the central FPGA is only used to compute baselines along the edge of the space (cf. Figure 1). Since edge effects diminish as the size of the space (i.e., number of telescopes) increases, for a 23-station array many of the cards do not use the central FPGA at all (13 out of 28, to be precise). To further increase hardware efficiency, one could therefore manufacture two types of boards, differing only in whether the central FPGA is present (for a 10-baseline card) or absent (for an 8-baseline card). Since the board design is identical in both cases, this should result in a pure cost savings; for an 8-band, 23-station array, 104 of the double-density FPGAs can be saved in this way. Assuming FPGA cost scales linearly with density, the effective number of base-density FPGAs in this hybrid approach is 254 for a 23-station array (an efficiency of 253/254 = 99.6%, 28 cards required per band), compared with 264 for a design consisting entirely of 8-baseline cards (an efficiency of 253/264 = 95.8%, 33 cards required per band). For 8- and 15-station arrays, the hybrid efficiencies are 100% and 99.1%, respectively. Fanout requirements are also noticeably reduced, from a total of 198 switchyard outputs using only 8-baseline cards to 176 for the hybrid design. If additional reduction in fanout is needed, this approach can be generalized to cards calculating more than 10 baselines each.

REFERENCES

Beasley, A.J., Woody, D.P., and Hawkins, D.W., 2002, CARMA Memo 3.

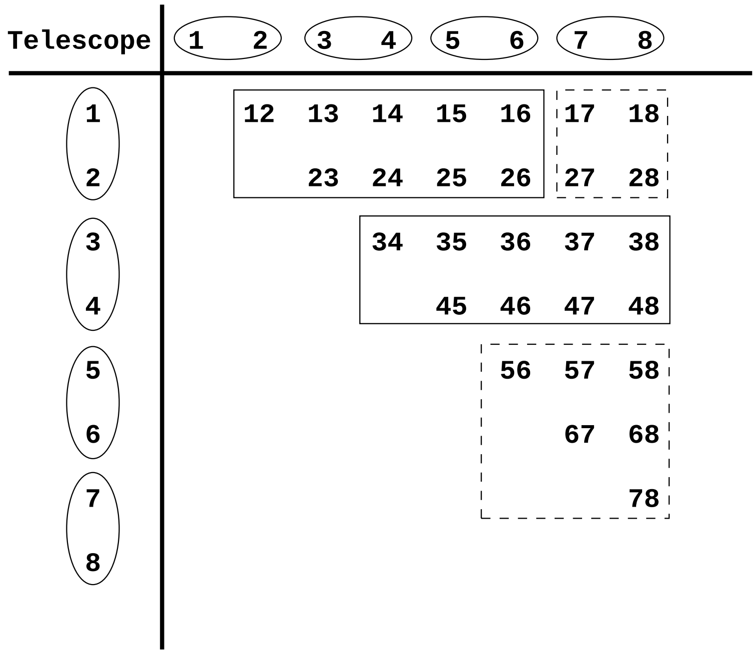

Fig. 3.— An optimum fanout- and configuration-minimized partitioning scheme for the 8-station SZA array; the two dashed boxes go together in one correlator card.

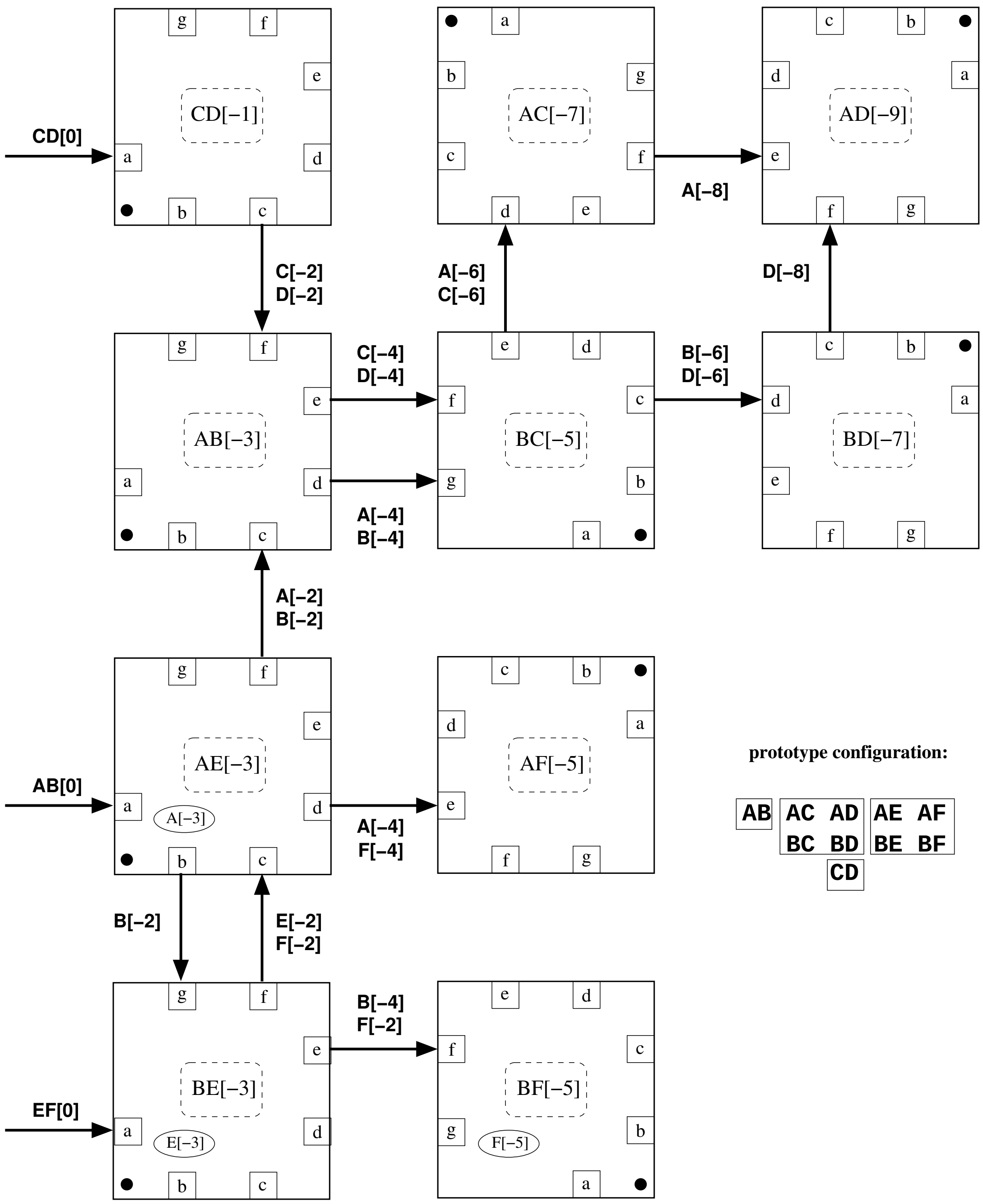

Fig. 4.— FPGA routing configuration #1 of the three needed by the fanout-minimized CARMA/SZA baseline partitioning solutions. Inter-FPGA data buses are assumed to run at 125 MHz (and thus can carry two telescope signals simultaneously). The notation `X[-d]` represents telescope signal(s) `X` delayed by `d` pipeline clock cycles relative to the front panel. Ovals represent pipeline registers adding two clocks of delay to the associated signal; dashed boxes indicate the baseline calculated in the corresponding FPGA.

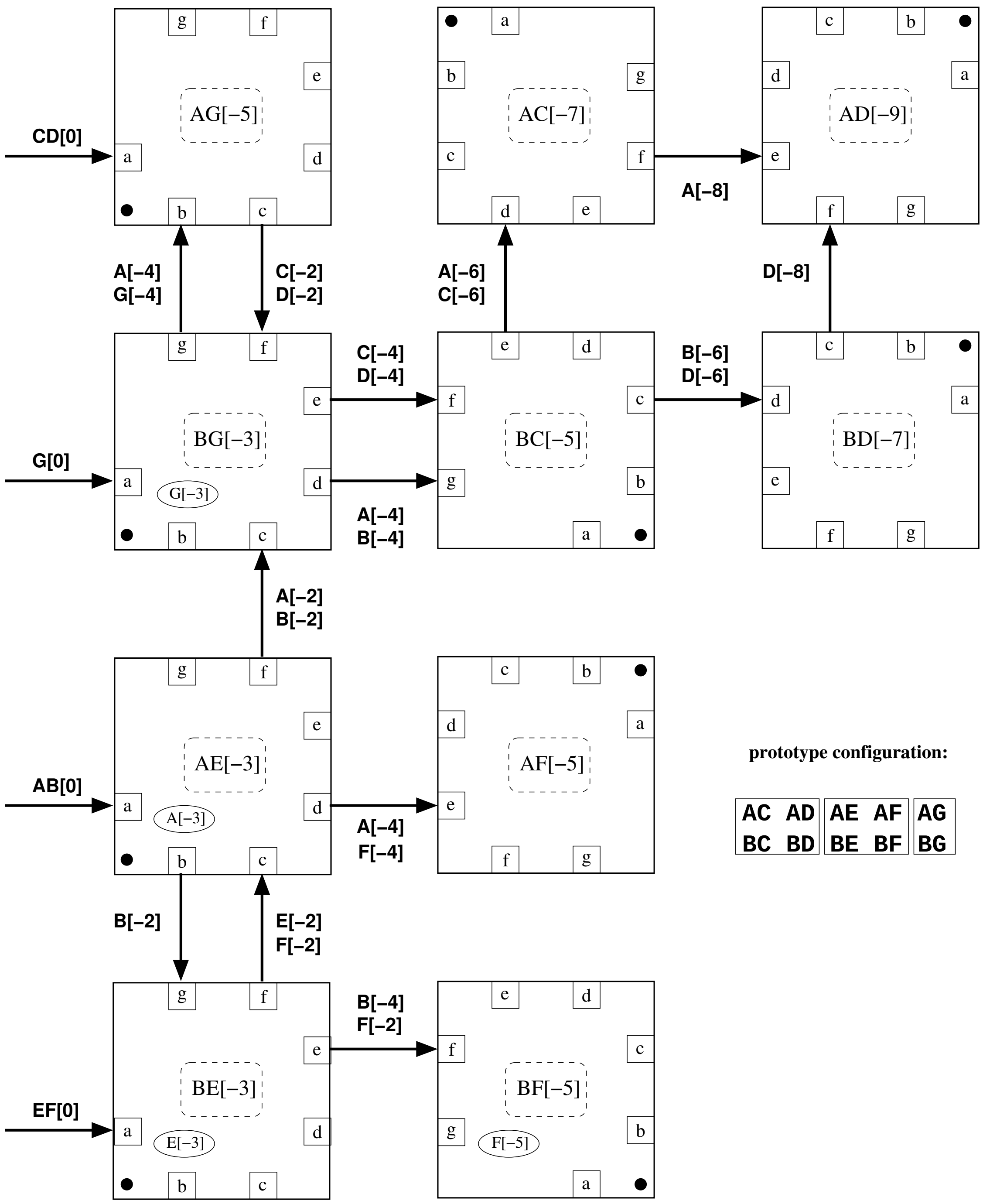

Fig. 5.— FPGA routing configuration #2 of the three needed by the fanout-minimized CARMA/SZA baseline partitioning solutions. Inter-FPGA data buses are assumed to run at 125 MHz (and thus can carry two telescope signals simultaneously). The notation `X[-d]` represents telescope signal(s) `X` delayed by `d` pipeline clock cycles relative to the front panel. Ovals represent pipeline registers adding two clocks of delay to the associated signal; dashed boxes indicate the baseline calculated in the corresponding FPGA.

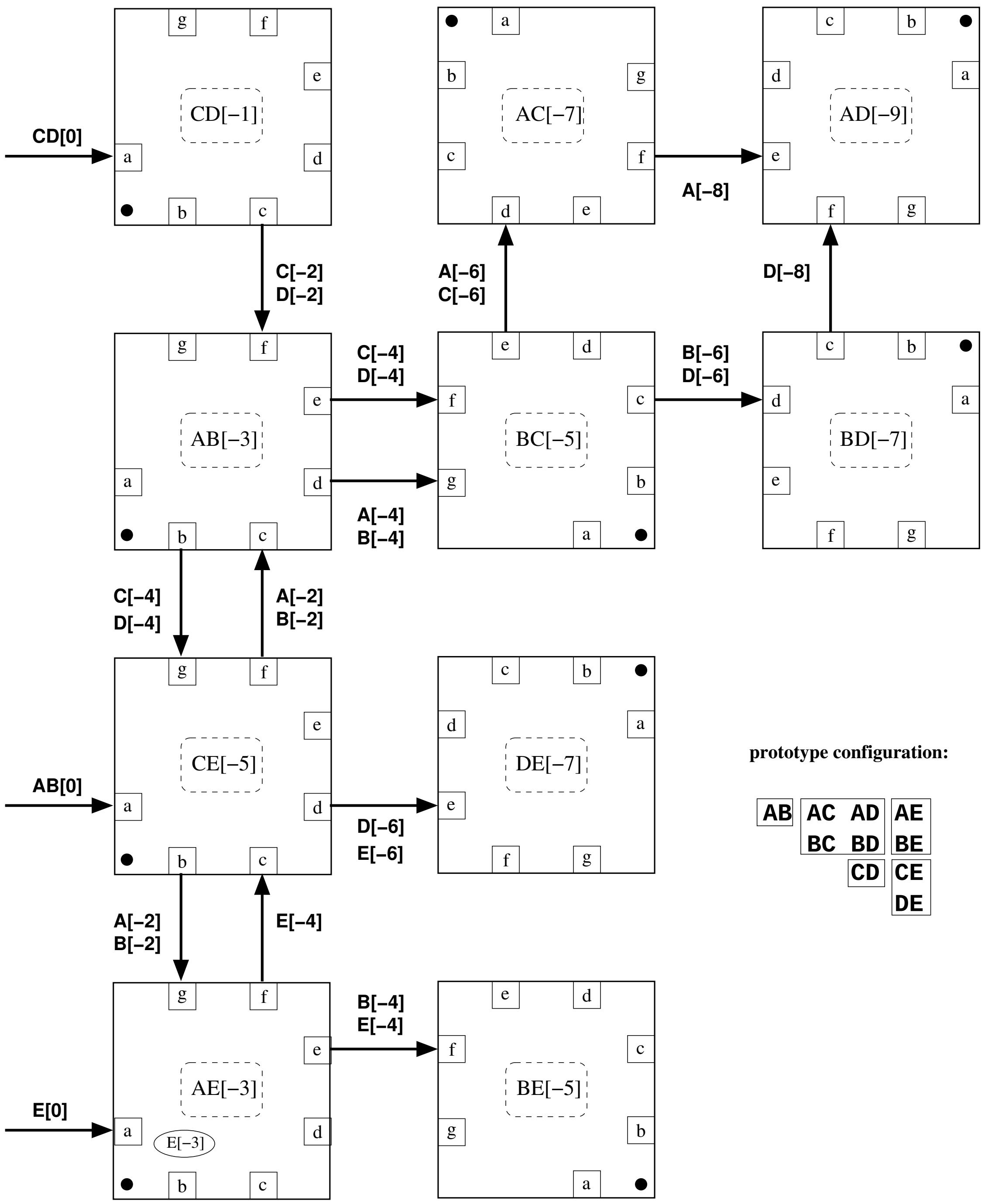

Fig. 6.— FPGA routing configuration #3 of the three needed by the fanout-minimized CARMA/SZA baseline partitioning solutions. Inter-FPGA data buses are assumed to run at 125 MHz (and thus can carry two telescope signals simultaneously). The notation `X[-d]` represents telescope signal(s) `X` delayed by `d` pipeline clock cycles relative to the front panel. Ovals represent pipeline registers adding two clocks of delay to the associated signal; dashed boxes indicate the baseline calculated in the corresponding FPGA.

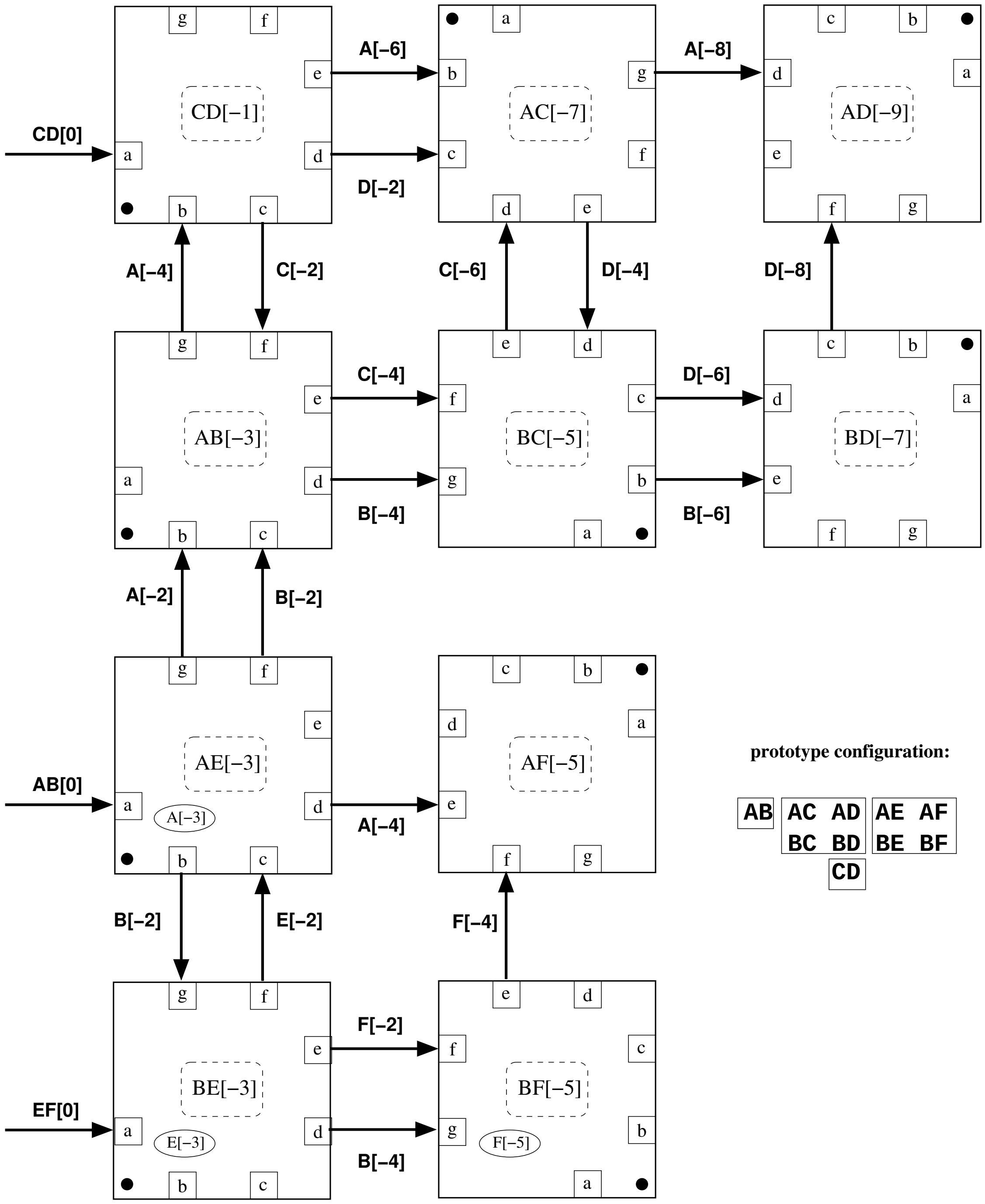

Fig. 7.— FPGA routing configuration #1 of the three needed by the fanout-minimized CARMA/SZA baseline partitioning solutions. Inter-FPGA data buses are assumed to run at 62.5 MHz (and thus can carry only one telescope signal). The notation `X[-d]` represents telescope signal(s) `X` delayed by `d` pipeline clock cycles relative to the front panel. Ovals represent pipeline registers adding two clocks of delay to the associated signal; dashed boxes indicate the baseline calculated in the corresponding FPGA.

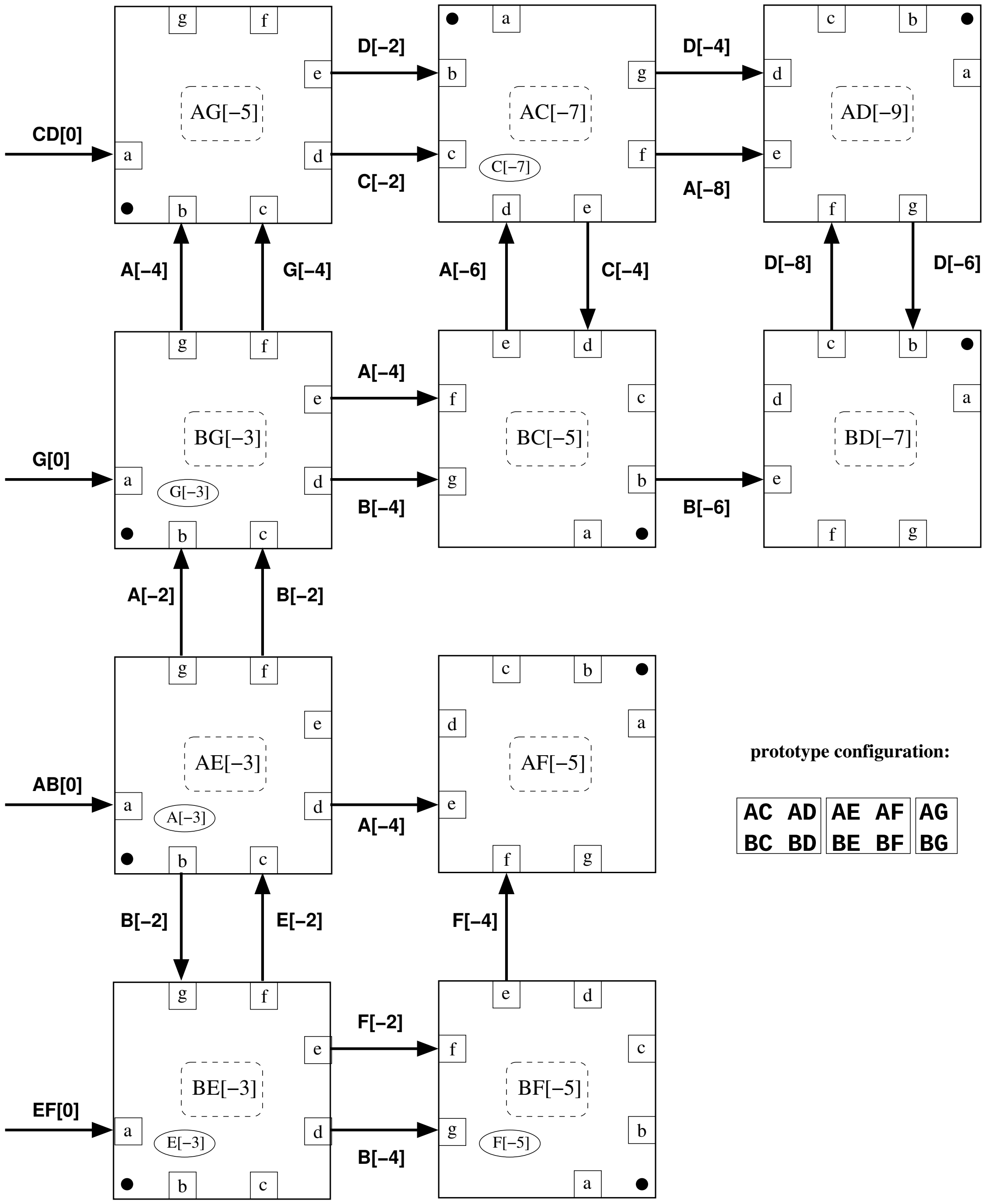

Fig. 8.— FPGA routing configuration #2 of the three needed by the fanout-minimized CARMA/SZA baseline partitioning solutions. Inter-FPGA data buses are assumed to run at 62.5 MHz (and thus can carry only one telescope signal). The notation `X[-d]` represents telescope signal(s) `X` delayed by `d` pipeline clock cycles relative to the front panel. Ovals represent pipeline registers adding two clocks of delay to the associated signal; dashed boxes indicate the baseline calculated in the corresponding FPGA.

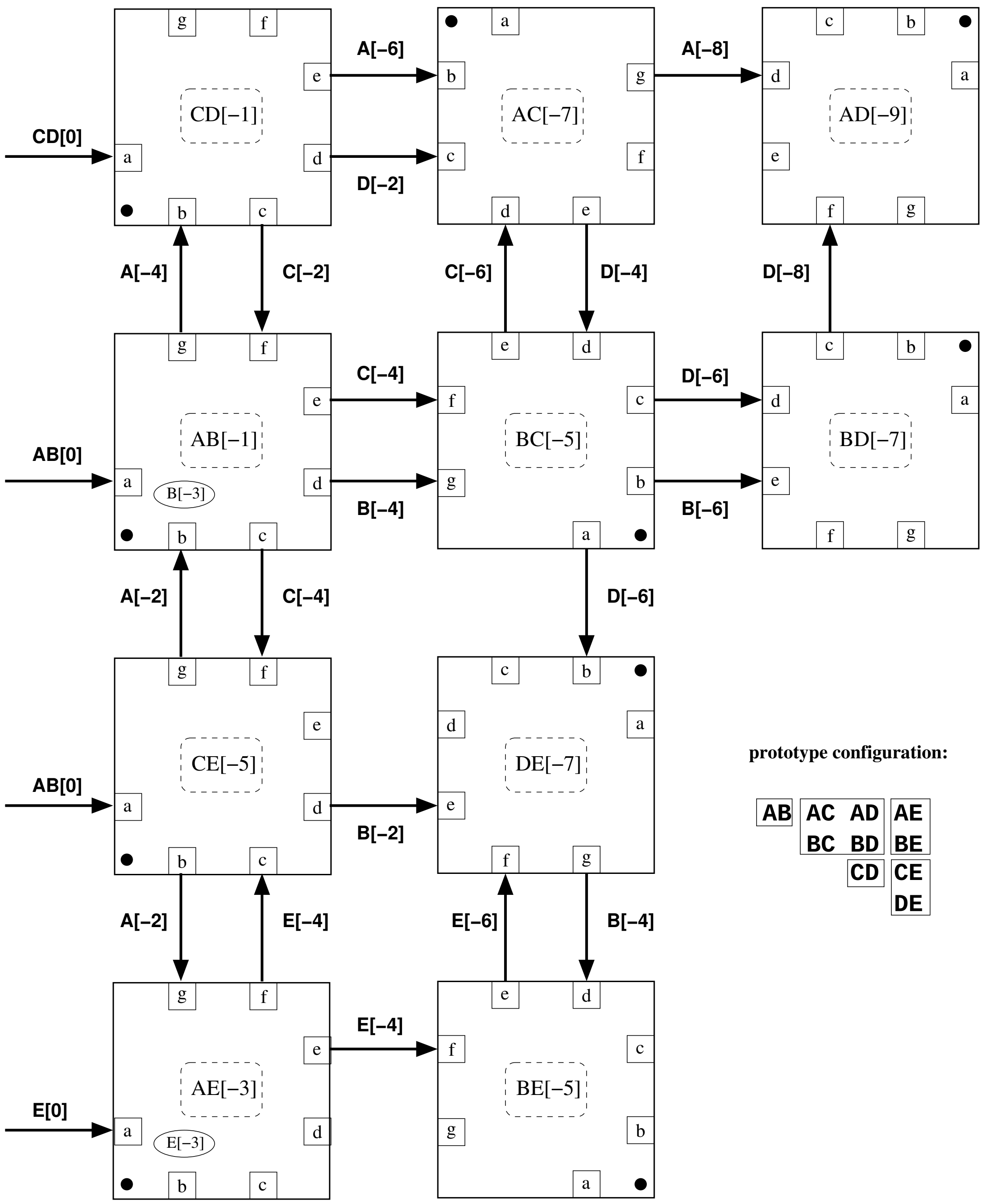

Fig. 9.— FPGA routing configuration #3 of the three needed by the fanout-minimized CARMA/SZA baseline partitioning solutions. Inter-FPGA data buses are assumed to run at 62.5 MHz (and thus can carry only one telescope signal). The notation `X[-d]` represents telescope signal(s) `X` delayed by `d` pipeline clock cycles relative to the front panel. Ovals represent pipeline registers adding two clocks of delay to the associated signal; dashed boxes indicate the baseline calculated in the corresponding FPGA.

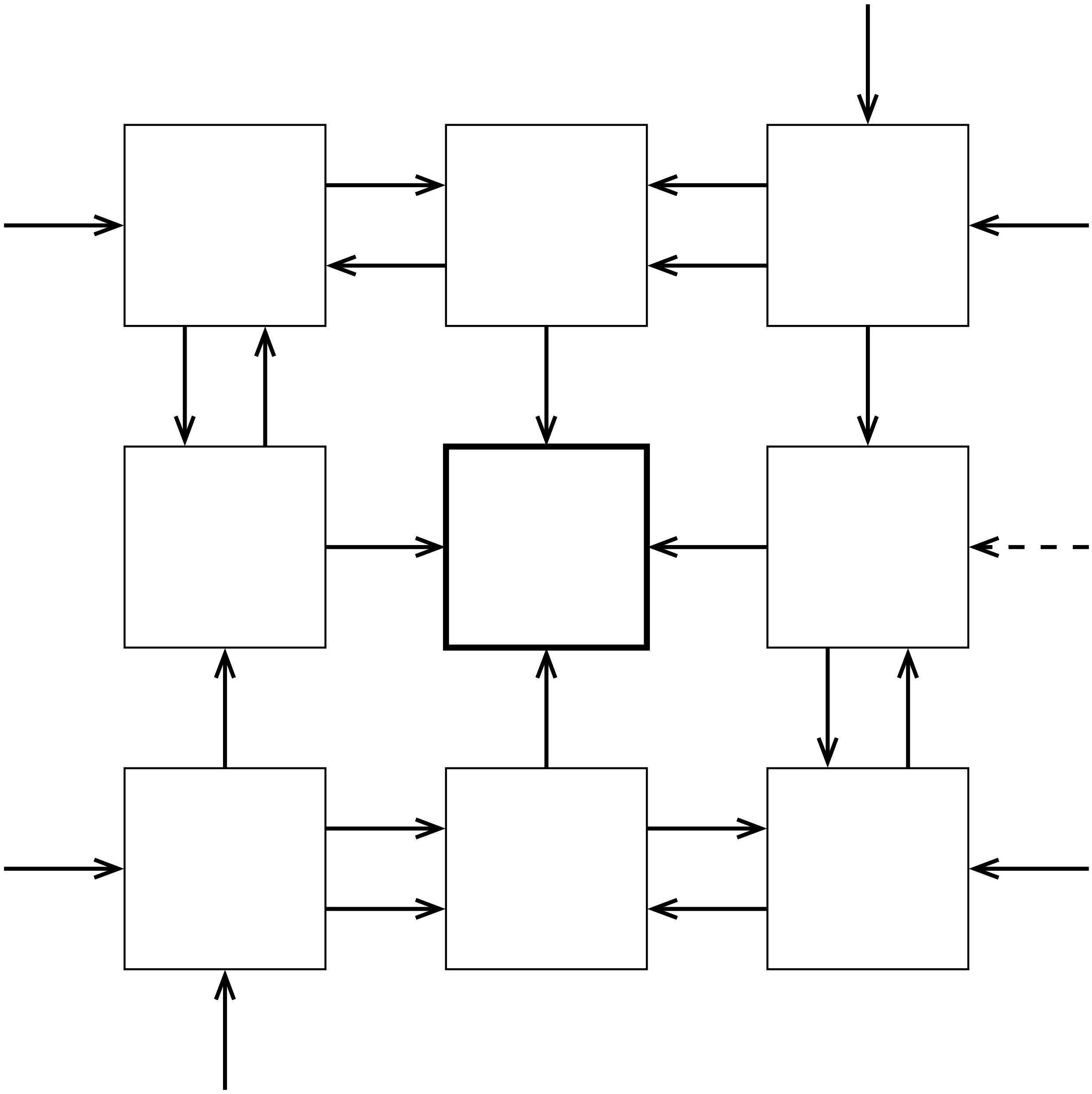

Fig. 10.— Proposed FPGA layout for the next-generation CARMA correlator card (see § 4). The central FPGA is assumed to be twice as dense as the other eight, allowing 10 baselines to be calculated per card; removing it (and the dashed input) results in an 8-baseline card with a simple routing configuration. Also shown are the data buses needed to implement the required FPGA configurations (§ 3); at most 5 (single telescope) buses are used by each FPGA.