CARMA Memorandum Series #46

Revised CARMA Correlator FPGA Configurations

Kevin P. Rauch (UMD)

April 19, 2011

ABSTRACT

This memo provides technical information on a variety of FPGA-level design details for the revised CARMA correlator digital hardware. This includes descriptions of the board-level FPGA layout, communication paths, and baseline partitioning; specifics of fundamental VHDL design components, and their distribution by FPGA; and the FPGA memory map and control register specification.

Change Record

Revision	Date	Author	Sections/Pages Affected
	Remarks		
1.0	2008-September-3	Kevin Rauch	
	Documented delay/phase coefficient table format.		
1.1	2009-April-13	Kevin Rauch	
	Detailed description of pipeline metadata. Updated memory map to match current configurations.		
1.2	2009-May-22	Kevin Rauch	
	Added MMAP_REG_STATUS_OLD.		
1.3	2009-June-23	Kevin Rauch	
	Added description of pipeline data tap memory.		
1.4	2009-July-24	Kevin Rauch	
	Updated descriptions of MMAP_REG_CORL_MODE and MMAP_REG_SAMP_DELAY. Added description of special test configurations (with figures).		
1.5	2009-August-25	Kevin Rauch	
	Updated register descriptions.		
1.6	2009-October-27	Kevin Rauch	
	Updated register descriptions.		
1.7	2009-November-28	Kevin Rauch	
	More register tweaks. Updated phase offset definition and NCO parameters.		
1.8	2010-February-18	Kevin Rauch	
	More register tweaks. Updated metadata description.		
1.9	2010-May-18	Kevin Rauch	
	Updated resolution tables.		
1.10	2010-June-25	Kevin Rauch	
	Updated resolution tables.		
1.11	2011-January-18	Kevin Rauch	
	Full-polarization configurations and related changes.		
1.12	2011-Apr-18	Kevin Rauch	
	Updated to version 2.4.0.		

1. The Revised CARMA Correlator

The initial CARMA correlator system, consisting of three bands (up to 1.5 GHz bandwidth) of recycled COBRA hardware, became fully operational in 2006. At that time, development proceeded on revised and upgraded digital hardware, which will replace the original hardware and expand the correlator system to eight bands (up to 4 GHz total bandwidth). The upgraded CARMA hardware will provide $\sim$ 6x the channel resolution per baseline compared to the existing COBRA hardware when cross-correlating 2-bit samples, and will also support new capabilities, such as FPGA-based fractional sample delays, cross-correlation of 3- and 4-bit samples, and the ability to be reconfigured for dual-polarization observations. This document discusses several FPGA-specific design details of the revised hardware, such as layout of the FPGAs communication buses, FPGA memory map and location definitions, and the specifics of pertinent VHDL component implementations. When referencing specific files in the carmacorl CVS module, $CCORL is used to represent the top-level project directory.

2. Physical Configuration

In contrast to COBRA, the revised CARMA digitizer and correlator boards share a unified PCB design. Each board contains 4 correlation/signal processing FPGAs, and one system controller FPGA acting as "glue logic" between the correlation FPGAs and the board CPU. All FPGAs are Altera Stratix II devices; specifically, an EP2S60F1020C3 device for the controller FPGA, EP2S90F1020C3 devices for digitizer card data FPGAs, and EP2S130F1020C3 chips for correlator card data FPGAs. Each digitizer card receives RF input from two antennas. The correlator boards can also be used as digitizers, once the PCB is loaded with A/D conversion logic. The unified PCB design also simplifies the HDL coding; since all digitizer and correlator data FPGAs share the same signal pinout, the same high-level VHDL components can be used in both instances. All new and upgraded VHDL components were designed to be shareable in this way.

Although the physical communication bus layout is identical for the digitizer and correlator boards, the bus directions and signal processing tasks differ between the two. Figures 1 and 2 display the data pipeline and shorthand VHDL bus naming conventions for the digitizer and correlator data FPGAs, respectively, as well as the high-level data processing components present in each FPGA, for the original single-polarization (minor version 0) FPGA configurations. All buses shown are 32-bits wide (excluding bus clocks, where applicable), except for the digitizer input bus (`dig`), which is 17-bit. The Stratix II FPGAs integrate support for a multitude of single-ended and differential I/O standards. Two such I/O standards are used in the revised correlator design: a differential standard (2.5 V LVDS) is used by the front panel I/O buses (`ext`) and rear digitizer input (`dig`); the remaining buses are single-ended, 2.5 V LVCMOS connections. Of the latter, eight buses, `1a` through `1d` and `2a` through `2d`, are used to transfer digitized samples (in various stages of processing) between FPGAs; two, `cpu1_ad` and `cpu2_ad` (not shown), are used to transfer data to the board (or host) CPU via the system controller FPGA.

Support for full-Stokes observations requires a different bus configuration to transfer alternate polarizations between adjoining pairs of correlator crates. This also impacts the single-polarization configurations as the LVDS cable fanout is fixed in both cases. In this 'recabled' setup one of the digitizer front-panel inputs is used to transfer signals between the corresponding digitizer in the neighboring crate, and a correlator card is used to provide extra fanout lost as a result. As a result two bus configurations are required per card type and polarization option. These are shown in figures 3-10.

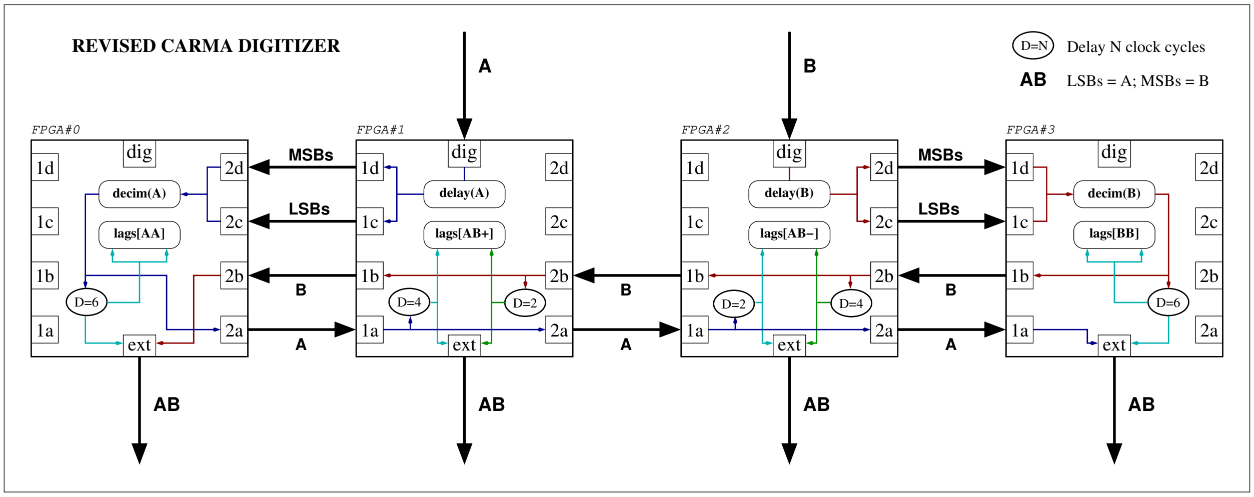

Fig. 1.— Revised CARMA digitizer board data pipelines and signal processing component layout (minor version 0).

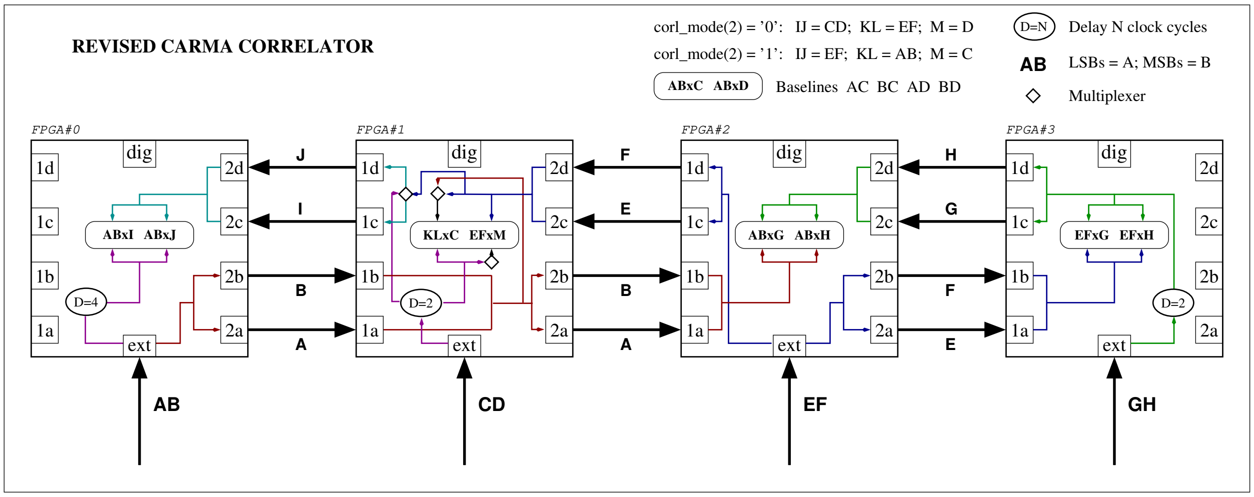

Fig. 2.— Revised CARMA correlator board data pipelines and signal processing component layout (minor version 0).

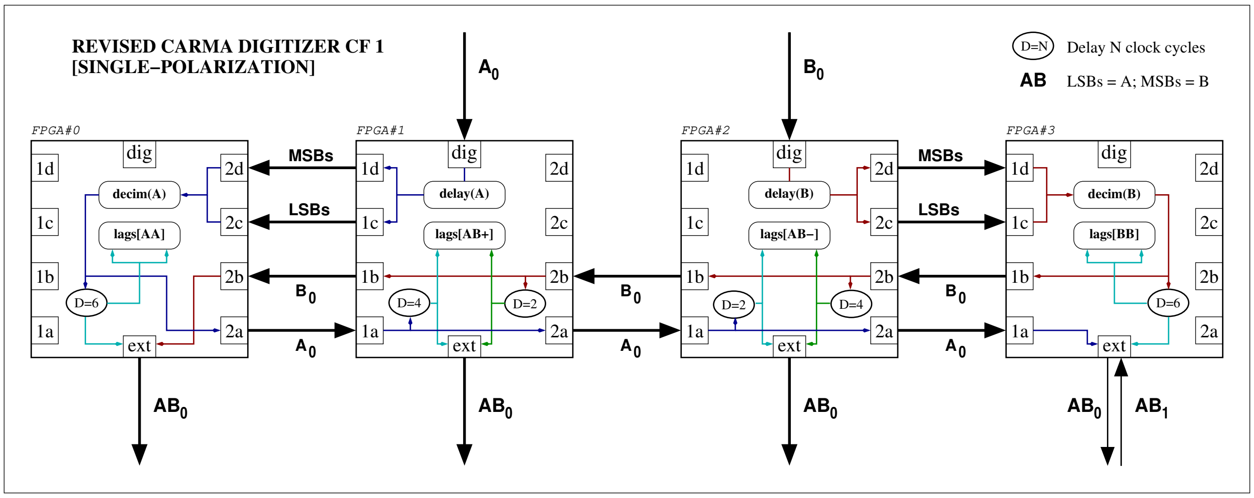

Fig. 3.— Revised CARMA digitizer board data pipelines and signal processing component layout (bus configuration 1).

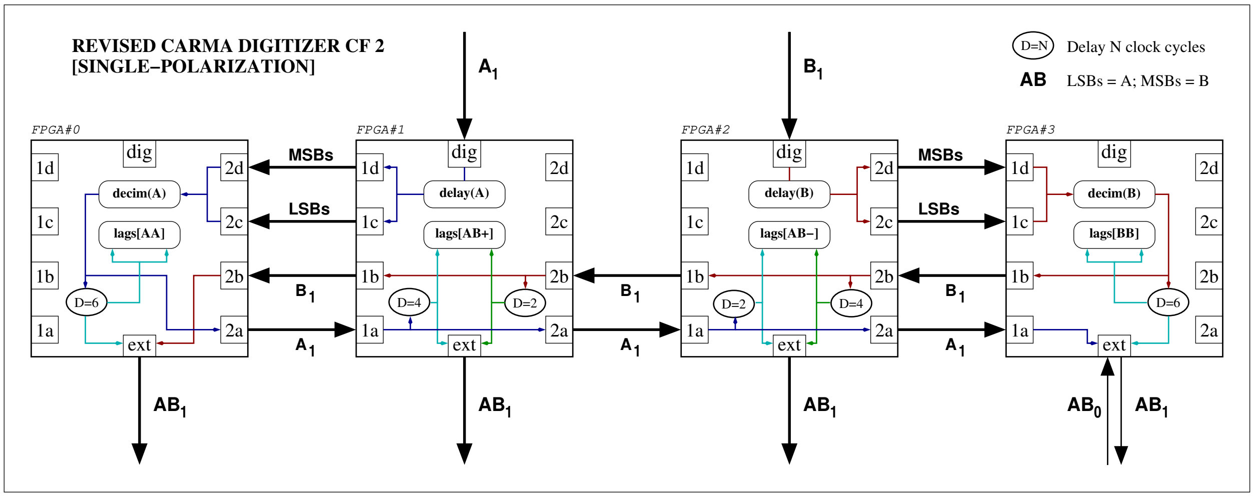

Fig. 4.— Revised CARMA digitizer board data pipelines and signal processing component layout (bus configuration 2).

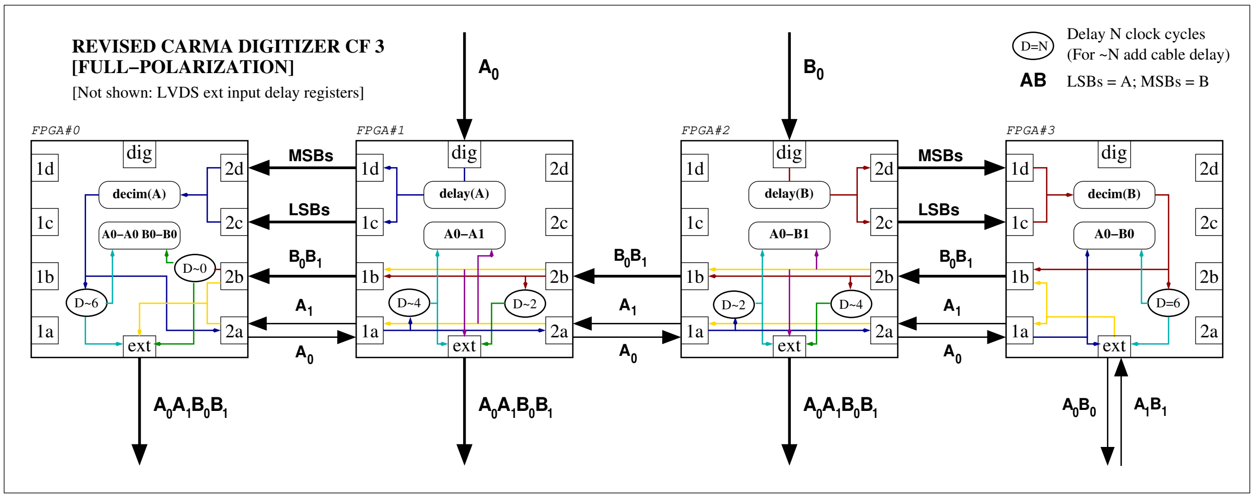

Fig. 5.— Revised CARMA digitizer board data pipelines and signal processing component layout (bus configuration 3).

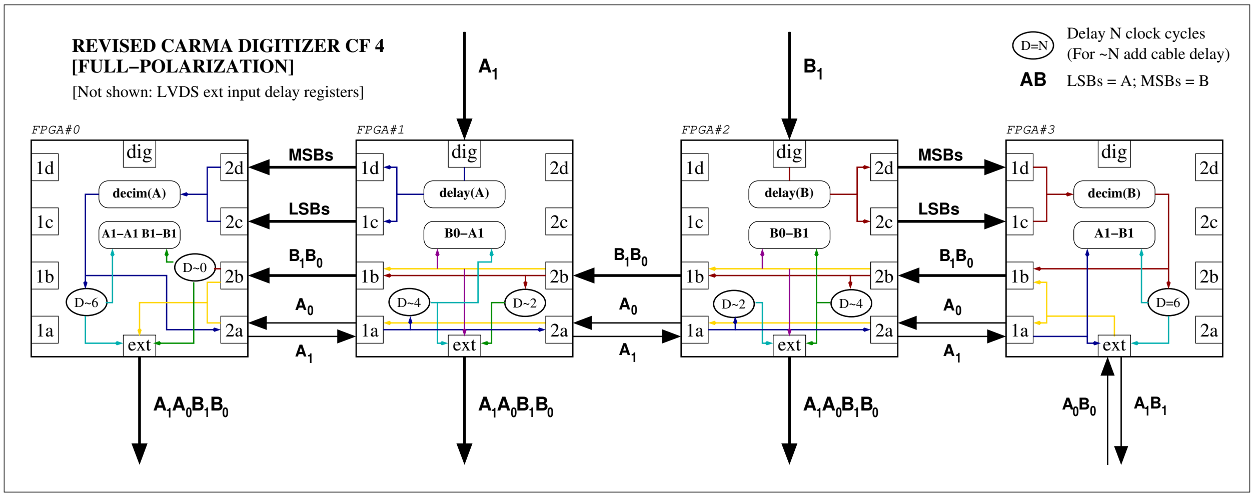

Fig. 6.— Revised CARMA digitizer board data pipelines and signal processing component layout (bus configuration 4).

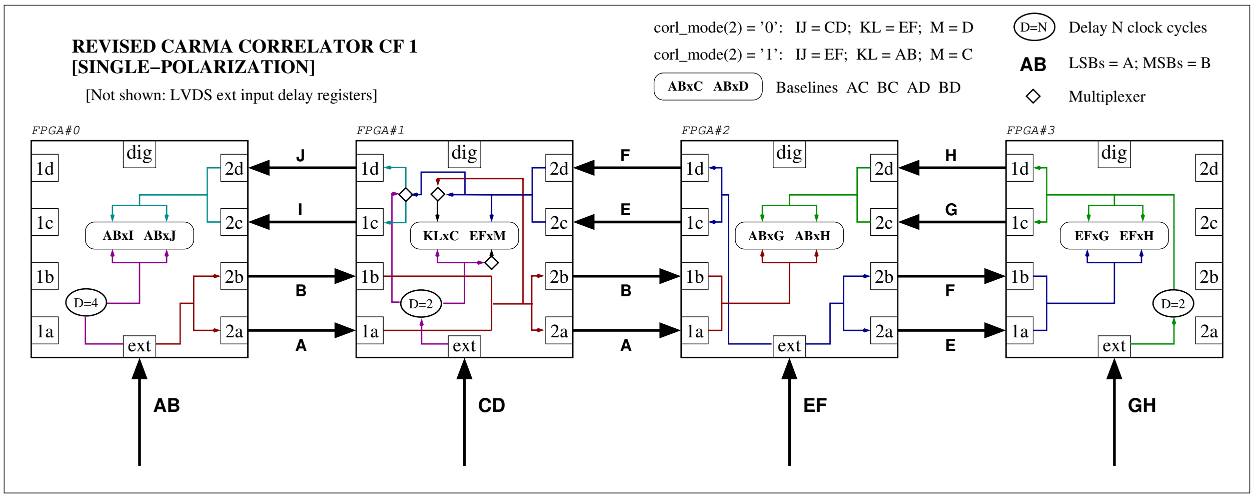

Fig. 7.— Revised CARMA correlator board data pipelines and signal processing component layout (bus configuration 1).

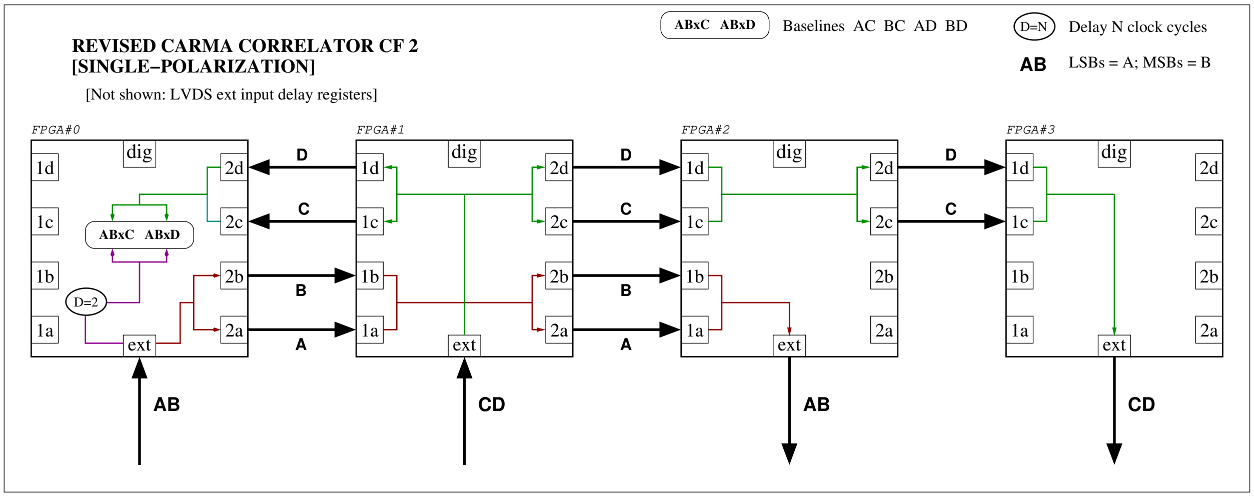

Fig. 8.— Revised CARMA correlator board data pipelines and signal processing component layout (bus configuration 2).

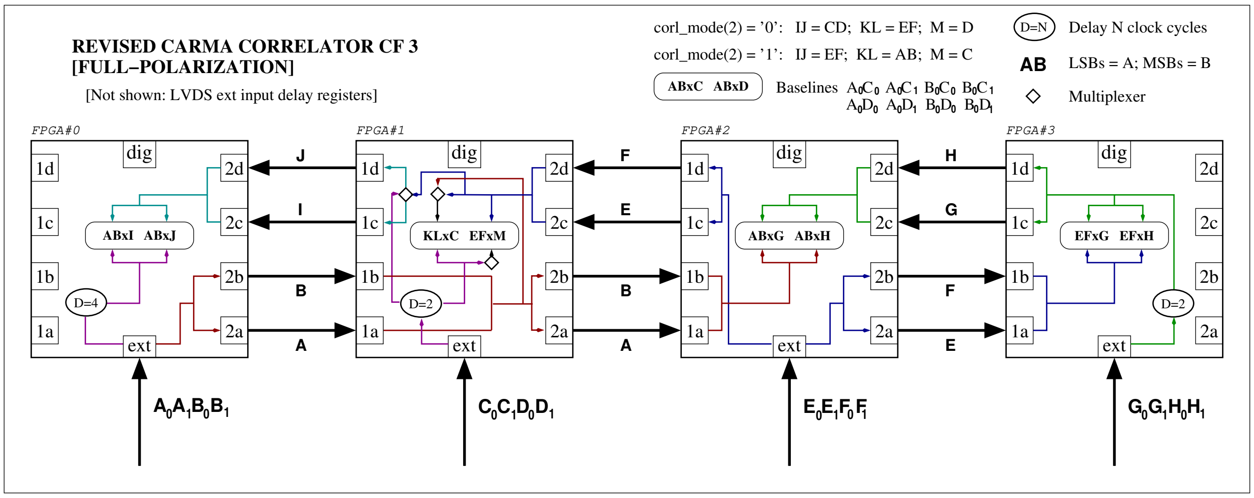

Fig. 9.— Revised CARMA correlator board data pipelines and signal processing component layout (bus configuration 3).

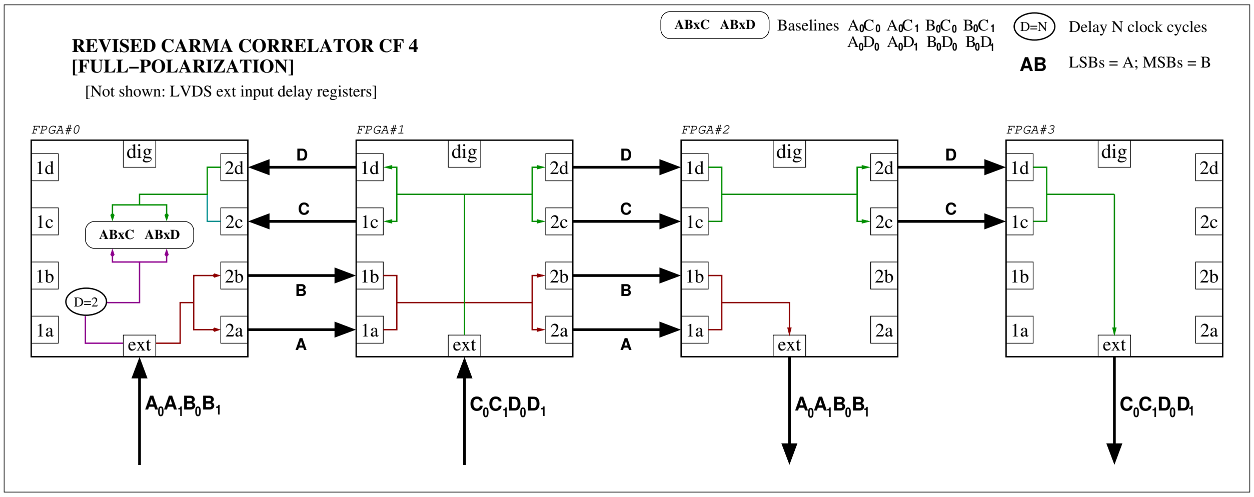

Fig. 10.— Revised CARMA correlator board data pipelines and signal processing component layout (bus configuration 4).

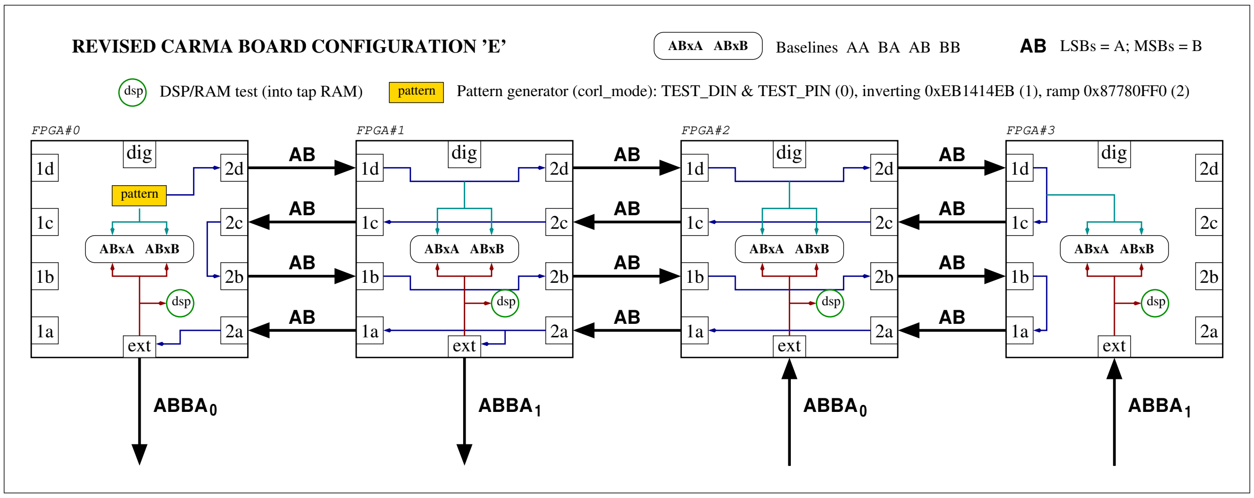

Fig. 11.— Revised CARMA board stress test configuration 'E'.

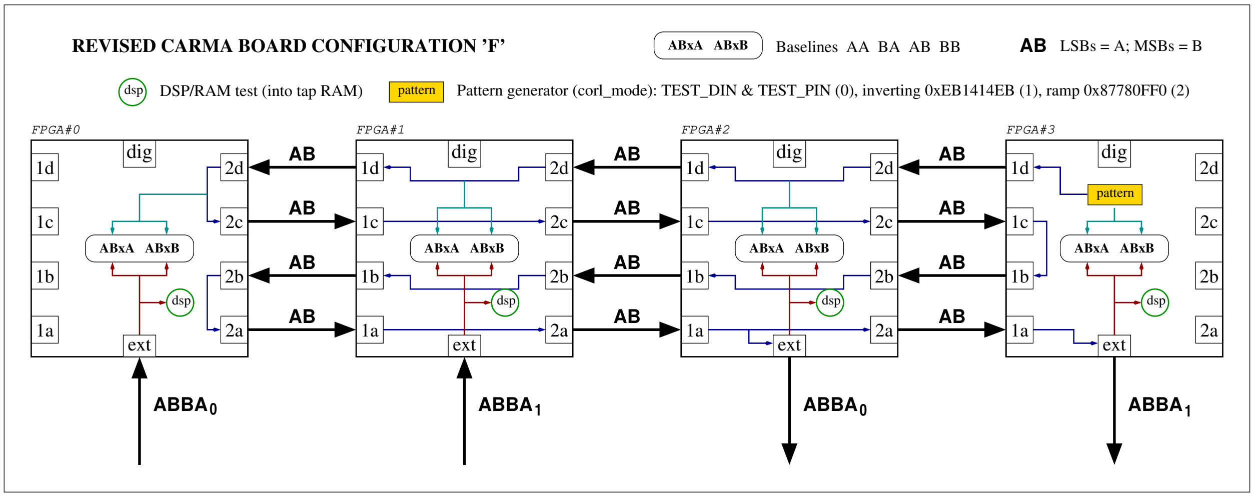

Fig. 12.— Revised CARMA board stress test configuration 'F'.

Two special pipeline configurations are used to gauge overall health of the FPGAs components; see Figures 11 and 12. The purpose of these configurations is to stress test a superset of the FPGA resources required by the normal configurations—if these test configurations perform correctly, the probability of failure with the normal configurations will be very low. Together the two configurations exercise all inter-FPGA buses and front-panel connections (as both transmitters and receivers), as well as $\sim$90% or more of the interal logic, RAM, and DSP blocks in each device. As an additional diagnostic aid, the normal inter-FPGA bus readback registers (see Table 1) are replaced with status bits indicating the health of the associated pins, determined by monitoring bus contents during testing. These configurations exist for both digitizer and correlator boards.

3. Baseline Partitioning

The revised correlator boards contain four data FPGAs, each calculating the cross-correlation of four individual baselines, for a total of 16 baselines per board. For CARMA-15 (15 antennas), there are 105 unique baselines. Eight digitizer cards (each with 2 RF antenna inputs) are needed per band. Due to increased logic availability in the digitizers, in the revised hardware each digitizer card can be responsible for computing the cross-correlation between its own antenna input pair (cf. Figure 1). This leaves 98 baselines to be computed by the correlator cards; hence 7 correlator cards are required per band—although one of them calculates only two unique baselines. As shown in Figure 13, only two distinct baseline-to-FPGA partitioning geometries are needed to distribute the baselines among the 7 cards. The two geometries are similar enough that a single FPGA configuration can be used for both; which one to use is determined by the setting of a `corl_mode` bit, and can be changed dynamically. The maximum fan-out any any antenna is four, meaning that no fan-out boards are required for the revised hardware. In contrast, the COBRA boards required three distinct partitioning geometries, each necessitating a unique FPGA configuration, and several fan-out boards. Conceptually, the revised hardware moves card-to-card cabling fan-out into PCB (and internal FPGA) traces. Note that if digitizer cards are not used to cross-correlate their own baselines, fan-out boards would be required.

4. FPGA Memory Map

The revised FPGA memory map consists of a set of control registers and M-RAM blocks organized into a contiguous address space, as seen by the external CPU interface. Internally, each M-RAM possesses a dedicated read/write bus. Stratix II M-RAM blocks support true dual-ported, mixed-width configurations; the new memory map component (`mmap`) instantiates a 32-bit port to communicate with the external interface and a 64-bit port for use by internal logic. Memory configuration is little-endian in this regard: writing a 64-bit value to memory via the 64-bit port to an address A is equivalent to writing the 32-bit LSBs and MSBs to addresses $2A$ and $2A+1$, respectively, via the 32-bit port.

The memory map implements a set of 32 x 32-bit control registers, beginning at address 0x0; the first 16 of these are read-only (cf. Table 1). The remainder of physical memory is provided by M-RAM blocks. Each M-RAM contains 2^{19} bits of memory, equivalent to `MMAP_BLOCK_SIZE` = 0x4000 32-bit words. The memory map reserves address space for 8 M-RAM blocks per FPGA, `MMAP_FPGA_SIZE` = 0x20000 (17-bit local addresses), and allocates three additional address bits (MSBs) for the chip select, for a total of

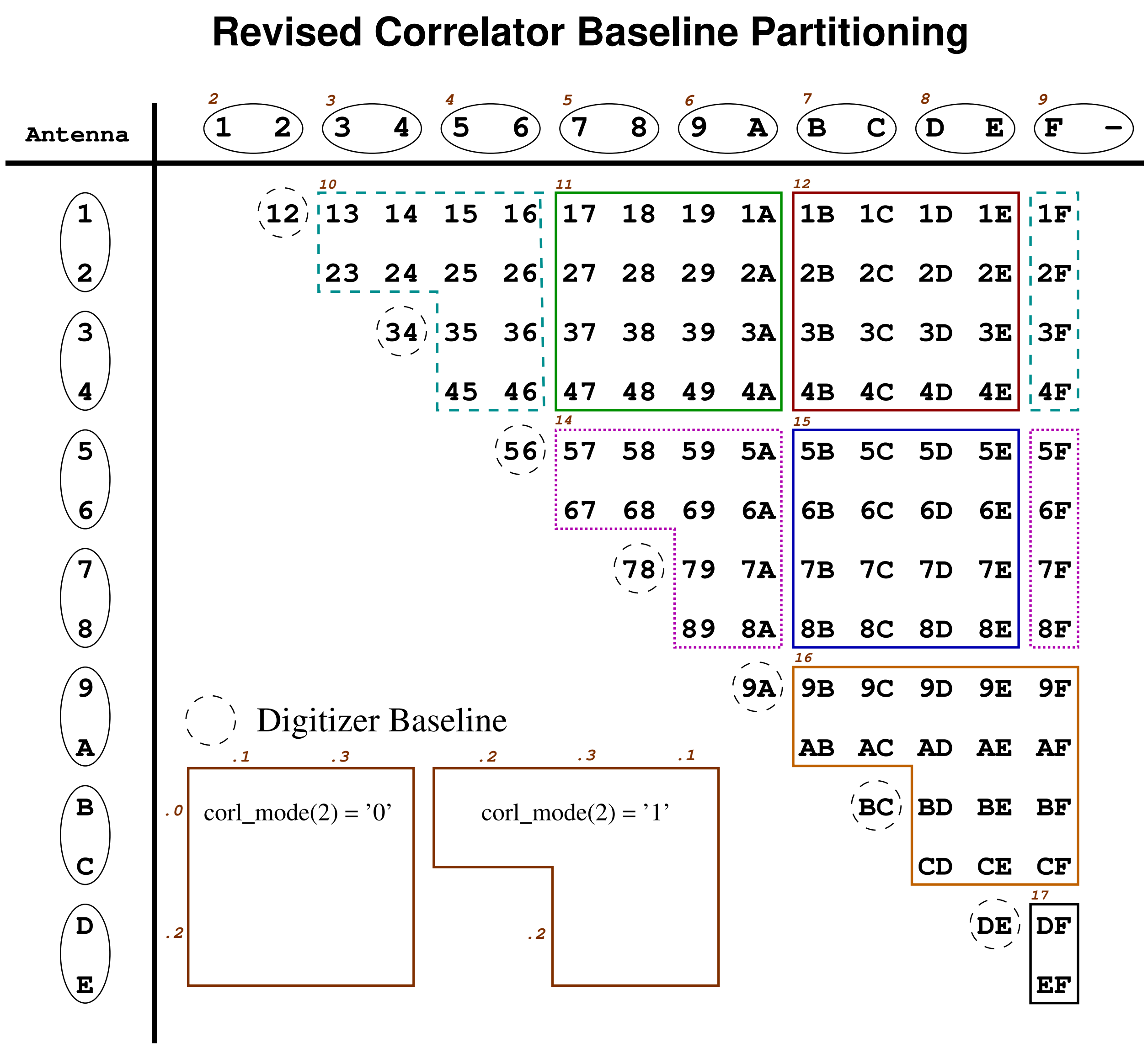

Fig. 13.— Revised correlator baseline partition map for CARMA-15. Each band requires 8 digitizer cards and 7 correlator cards. Board slot numbers and front-panel inputs (UL → .0, UR → .1, LL → .2, LR → .3) noted in small type. Note: in the ‘recabled’ setup slot 17 provides the fanout to slot 16 for inputs D-F.

SYS_ADDR_WIDTH = 20 bits of FPGA address space visible to the CPU. This exceeds the physical limits of 5 FPGAs and (up to) 6 M-RAM blocks per FPGA present on the revised boards. A generic (NUM_BLOCKS) determines the actual number of M-RAMs consumed by the memory map in any particular FPGA; reads from addresses for which no physical RAM has been allocated return a pre-defined hex string (currently 0xDEADBEEF).

The detailed memory map is listed in Table 1. Control registers occupy the first 32 locations, and shadow the corresponding region (= 0.2%) of the first M-RAM block, which physically begins at address 0x0. Subsequent M-RAMs are contiguous—the starting address for block i is $A = i \times \mathtt{MMAP\_BLOCK\_SIZE}$. The identical layout is used in both digitizer and correlator FPGAs. The first M-RAM block is reserved for the phase/delay coefficient tables used in digitizer FPGAs #1 and #2 (cf. Figure 1); the table contents are described in § 6.2. Subsequent blocks, one per baseline, are reserved for lag readout data; see § 5.

The control register contents are as follows (omitted bitfields are undefined):

- MMAP_REG_VERSION [Version (read-only)]
  This register contains board-level configuration settings.
  - **bits 27-24:** hardware revision; 0x0 = COBRA final, 0x1 = revised prototype
  - **bits 23-20:** FPGA type; 0xC = correlator, 0xD = digitizer
  - **bits 19-16:** requantized sample width
  - **bits 15-12:** bandwidth mode number; 1 = 500 MHz, 9 = 2 MHz
  - **bits 11-8:** bus configuration number
  - **bits 7-0:** minor version number
- MMAP_REG_COMPAT [Configuration compatibility (read-only)]
  This register contains FPGA-specific configuration settings.
  - **bits 31-18:** *reserved* (configuration type field)
  - **bit 17:** set iff configured for 250 MHz digitizer input (else 1 GHz)
  - **bit 16:** set iff configured for 125 MHz external reference (else 31.25 MHz)
  - **bits 15-0:** bit n is set iff the configuration is compatible with FPGA #n.
- MMAP_REG_CORL_CONF1 [Correlation configuration register 1 (read-only)]
  For minor versions 2 and later:
  - **bits 30-24:** Metadata elements per lag stream (NUM_META)
  - **bits 23-12:** Number of lags per stream (NUM_LAGS)
  - **bits 11-8:** Number of baselines per M-RAM block (NUM_PACK)
  - **bits 7-4:** Number of correlation baselines (NUM_CORL)
  - **bits 3-0:** Correlation type; 0 = auto, 1 = cross +lags only , 2 = cross -lags only, 3 = cross +lags and -lags

  Minor versions 0 and 1 only (NUM_PACK = 1 implied):
  - **bits 30-20:** Metadata elements per lag stream (NUM_META)

- **bits 19-8:** Number of lags per stream (NUM_LAGS)
- **bits 7-4:** Number of correlation baselines (NUM_CORL)
- **bits 3-0:** Correlation type

- MMAP_REG_CORL_CONF2 [Correlation configuration register 2 (read-only)]
  - **bits 30-27:** Written lag block address width (1+DUMP_ADDR_WIDTH)
  - **bits 26-12:** Written lag block locations ($2 \times$ DUMP_COUNT)
  - **bits 11-0:** Number of quantization state counters (NUM_QCNT)

- MMAP_REG_STATUS_OLD [Previous correlation status (read-only)]
  This register contains the value of MMAP_REG_STATUS when the most recent lag dump completed. It can be used determine the final status of the previous correlation while the next is in progress.

- MMAP_REG_TD_1A to MMAP_REG_TD_2E [Inter-FPGA bus readback registers (read-only)]
  These control registers contain the current (or "recent") contents of the associated IOE registers (cf. Figs. 1-2).

- MMAP_REG_CTRL_TAP [Control bit readback (read-only)]
  - **bits 31-22:** Reserved (cleared).
  - **bit 21:** dig_rx_err
  - **bit 20:** dig_rx_lock
  - **bit 19:** ext_rx_err[1]
  - **bit 18:** ext_rx_err[0]
  - **bit 17:** ext_rx_lock[1]
  - **bit 16:** ext_rx_lock[0]
  - **bits 15-13:** Reserved (cleared).
  - **bit 12:** demod_state (current phase demodulation bit)
  - **bit 11:** corl_doneN (SYSCTL fpga_ctl_io[3:0])
  - **bit 10:** corl_dump (SYSCTL fpga_ctl_out[11:8])
  - **bit 9:** ctrl_correlate (SYSCTL fpga_ctl_out[7:4])
  - **bit 8:** ctrl_1pps (SYSCTL fpga_ctl_out[3:0])
  - **bit 7:** pll_inclk[1] (SYSCTL fpga_data[7])
  - **bit 6:** pll_clksw (SYSCTL fpga_data[6])
  - **bit 5:** pll_rst (SYSCTL fpga_data[5])
  - **bit 4:** pipe_tap_enable (SYSCTL fpga_data[4])
  - **bits 3-0:** corl_ctrl[3:0] (SYSCTL fpga_data[3:0])

- MMAP_REG_TD_EXT_LSBS and MMAP_REG_TD_EXT_MSBS [Front-panel readback registers]
  In addition to simple bus readback, these registers are connected to a continuous verification component which monitors incoming samples for erroneous dynamic behavior (such as counters which fail

to advance monotonically, or changes in fields which should remain fixed). Writes to these registers control the behavior of this component and the register output. By default the registers output the current (or "recent") contents of the front-panel input IOE registers. Writing 0xBADFEED switches output to the cumulative mask of erroneous bits; 0x1BADFEED switches to the latest instantaneous mask containing bad bits; 0x*n*FEED switches output to the cumulative number of input samples for which bit *n* was bad; 0xB0DE switches to the cumulative number of samples examined (32-bit range); 0xBADF1F0 outputs the last erroneous input sample (0x0 if none seen); writing any other value switches output to normal bus readback. In addition, writing 0x0 resets all indicators to zero. These options can be used to analyze bad bits and determine their absolute bit error rates. Note: due to logic constraints, most production FPGA configurations support simple bus readback only. Digitizer and 4-bit correlator configurations also enable 0xBADFEED support.

- `MMAP_REG_STATUS` [Correlation status]
  - **bits 30-8:** Counter indicating the number of clock cycles the correlation (specifically, multiply-adder) logic was active (enabled) since the last correlation dump. Cannot be reset externally (cleared automatically).
  - **bits 7-4:** Metadata error indicator for correlation baselines 0 (bit 4) to 3 (bit 7), if present. A set bit means that while correlation was active at least one sample from that baseline suffered overflow prior to requantization (hence becoming garbage), or a glitch was detected in the metadata time sequence. For `NUM_PACK` = 2, `NUM_CORL` = 8 (CARMA23/FULLPOL correlator boards), the four baselines chosen for status output, indices 0, 3, 4, and 7 (slot 10 FPGA #0 example: 1-3, 2-18, 1-4, 2-19), sample every input to help disambiguate faulty inputs (considering status from all FPGAs). Bits stay set until cleared.
  - **bit 3:** Front input error indicator; if set, the front-panel LVDS input PLL lost lock, or its FIFO suffered underflow/overflow, while correlation was active. Stays set until cleared.
  - **bit 2:** Digitizer input error indicator; if set, the digitizer LVDS input PLL lost lock, or its FIFO suffered underflow/overflow, while correlation was active. Stays set until cleared.
  - **bit 1:** Correlation error indicator; if set, the correlate signal went high before lag data was successfully transferred to FPGA RAM. Asserts the `corl_doneN` interrupt when set.
  - **bit 0:** Correlation done indicator; this bit is set once lag data (including metadata and quantization counts) for the most recent integration has been successfully transferred to FPGA RAM. Asserts the `corl_doneN` interrupt when set.
- `MMAP_REG_INPUT_DELAY` [LVDS input delay control]
  Adjusts the LVDS input delays. Writing a non-zero value *n* to the 16-bit LSBs/MSBs of this register delays the 32-bit LSBs/MSBs of front-panel input by *n* clock cycles ($n = 0$ is a valid no-op). The applied delay is 7-bit (i.e, modulo 128). The special value 0xDEAD puts the corresponding half of the bus (including its input PLL) into reset until another delay value is written. The readback from this register indicates the current *total* delay applied to each bus, and is automatically cleared when the associated input PLL loses lock (as this resets the `ext_rx` component); the MSB is set in this case to indicate the loss of lock (including those due to 0xDEAD requests).
- `MMAP_REG_CORL_MODE` [Correlator operating mode]

- **bit 6:** Impulse response test; when set, the input of the fractional delay filter or decimation pipeline in digitizer FPGAs is replaced with a periodic delta function, enabling filter impulse response testing. The scale and period of the delta function is set using `MMAP_REG_IMPULSE`. In FPGAs #1 and #2, setting this bit also replaces prompt and delay correlation inputs with the fractional delay output LSBs and MSBs, respectively (causing them to be dumped to RAM during integrations).
- **bit 5:** When set, the output of the decimation pipeline, which normally consists of requantized (and possibly multiplexed) output samples, is replaced by a single 32-bit, full-precision output sample. This mode enables precise decimator testing. In digitizer FPGAs #1 and #2, quantization counts are based on sub-ns filter output instead of raw samples.
- **bit 4:** When set in digitizer FPGAs, the normal prompt/delay sample input streams are replaced by the LVDS front-panel readback (prompt with the 32-bit LSBs, delay with the 32-bit MSBs). In correlator FPGAs, secondary metadata is replaced with local information.
- **bit 3:** When set in correlator FPGAs, an FPGA normally calculating multiple baselines will calculate a single correlation (correlation 0) at `NUM_CORL` times normal resolution. In digitizer FPGAs, the digitally rescaled samples override the raw samples for quantization counting and bypass the sub-ns filter for pipeline input.
- **bit 2:** When set in digitizer FPGAs, the decimation and sub-ns delay components (including digital rescaling) are bypassed, and the (LSBs of) the input sample bus is transmitted intact. In correlator FPGAs, this bit selects between the two basic pipeline configurations required to implement baseline partitioning (cf. Figure 2).
- **bits 1-0:**
  - Mode `"00"`: normal operation of the pipeline and correlation logic.
  - Mode `"01"`: correlation logic replaces the prompt and delay inputs with test patterns based on `MMAP_REG_TEST_PIN` and `MMAP_REG_TEST_DIN`, respectively.
  - Mode `"10"`: in digitizer FPGAs, digitizer input is replaced with the ramp pattern {0x0FF0F00FF00F0FF0, 0x1EE1E11EE11E1EE1, ...} (offset by 8 between DIGA/DIGB), and decimator output is replaced with test patterns `MMAP_REG_TEST_PIN` (for DIGA output) or `MMAP_REG_TEST_DIN` (for DIGB output); in single-polarization correlator FPGAs (only), the test patterns replace the front-panel input data (LSBs considered 'prompt').
  - Mode `"11"`: in digitizer FPGAs, the 64-bit digitizer input bus is replaced with the concatenation TD2B & TD1A (supports sub-ns filter testing)

- `MMAP_REG_DEMOD` [Phase-switch demodulation state]
  Encodes the 180 degree phase-switch demodulation sequence for up to 32 consecutive integrations (bits 31-0, with 0 being read first). A set bit indicates that samples should be negated during the corresponding integration cycle. Registers exist in all FPGAs; in FPGA #1 and #2 they apply to the raw 8-bit digitizer samples, and in FPGA #0 and #3 to the requantized samples. Aside from testing, only one pair or the other should be used.

- `MMAP_REG_IMPULSE` [Impulse response pattern parameters]
  Determines the frequency of sample generation for impulse response filter testing (cf. `CORL_MODE`, bit 6). The (up to) 16-bit LSBs encode the impulse sample value (limited to `SAMP_WIDTH`); the

16-bit MSBs represent the number of (pipeline) clock cycles between impulse sample output in the pattern generator, with all zeroes output in between.

- `MMAP_REG_TEST_PIN` and `MMAP_REG_TEST_DIN` [Correlation sample test patterns]
  The 16-bit LSBs of each register defines the test patterns fed to the prompt (`PIN`) and delay (`DIN`) cross-correlation inputs in self-test mode (cf. `CORL_MODE`). Reset default patterns are `PIN` = 0xEB14 and `DIN` = 0x14EB.

- `MMAP_REG_OUT_ENABLE` [Inter-FPGA bus output enables]
  - **bit 9:** Output enable bit for bus 2E
  - **bit 8:** Output enable bit for bus 2D
  - **bit 7:** Output enable bit for bus 2C
  - **bit 6:** Output enable bit for bus 2B
  - **bit 5:** Output enable bit for bus 2A
  - **bit 4:** Output enable bit for bus 1E
  - **bit 3:** Output enable bit for bus 1D
  - **bit 2:** Output enable bit for bus 1C
  - **bit 1:** Output enable bit for bus 1B
  - **bit 0:** Output enable bit for bus 1A

  Output enable bits apply only to buses actually used for output by the configuration; they are ignored by input buses.

- `MMAP_REG_DIG_DELAY` and `MMAP_REG_DIG_PHASE` [Digitizer sample delay and phase (read-only)]
  These registers provide readback of the currently employed sample delay and phase offset values. Each is a fixed-point value with a 16-bit fraction. The delay value is in ns and the phase is in revolutions. Due to peculiarities of the sub-ns delay filter, if the fractional delay is at least 0.5 ns, then the true (requested) delay is 1 ns less than the register value; however, the latter does indicate the actual whole-ns delay used by the delay line. In addition, the integer part of the phase (ignored by the FPGAs) normally contains origin information consisting of the associated digitizer input (0xA or 0xB) and the tap table index from which the value was read (0 to 47).

- `MMAP_REG_DIG_GAIN` and `MMAP_REG_DIG_OFFSET` [Digitizer sample gain and offset]
  These registers define a linear transformation applied to the raw 8-bit digitizer input samples before they are rounded to fewer (currently 6) bits prior to processing by the sub-ns delay filter:

$$x \to x' = (\text{GAIN} \times x + \text{OFFSET})/1024.$$

  For example, GAIN = 1126, OFFSET = 3072 corresponds to $x' = 1.100 \cdot x + 3$. Both factors are 16-bit signed quantities (the 16-bit LSBs of each register).

- `MMAP_REG_DIG_CTRL` [Digitizer initialization and control bits]
  - **bits 31-3:** Reserved (cleared).

  - **bit 2:** Digitizer data ready reset enable.
  - **bit 1:** Clock divider reset enable.
  - **bit 0:** FPGA digitizer LVDS receiver PLL enable.

- `MMAP_REG_DIG_PHASOR` [Digitizer phasor enable control]
  Writing a non-zero value n to this register causes the rotating phasor used to remodulate decimated samples to be disabled for n clock cycles. This is used by the phase flattening algorithm to help achieve phase alignment.

Table 1. Revised CARMA correlator FPGA memory map

Symbolic Name[a]	Hex Address[b]	Description
MMAP_REG_VERSION	0x00000[c]	FPGA configuration version.
MMAP_REG_COMPAT	0x00001[c]	FPGA configuration compatibility.
MMAP_REG_CORL_CONF1	0x00002[c]	Correlation logic specification.
MMAP_REG_CORL_CONF2	0x00003[c]	Correlation logic specification.
MMAP_REG_STATUS_OLD	0x00004[c]	Correlation status of last integration.
MMAP_REG_TD_1A	0x00005[c]	Inter-FPGA bus 1A readback.
MMAP_REG_TD_1B	0x00006[c]	Inter-FPGA bus 1B readback.
MMAP_REG_TD_1C	0x00007[c]	Inter-FPGA bus 1C readback.
MMAP_REG_TD_1D	0x00008[c]	Inter-FPGA bus 1D readback.
MMAP_REG_TD_1E	0x00009[c]	Inter-FPGA bus 1E readback.
MMAP_REG_TD_2A	0x0000A[c]	Inter-FPGA bus 2A readback.
MMAP_REG_TD_2B	0x0000B[c]	Inter-FPGA bus 2B readback.
MMAP_REG_TD_2C	0x0000C[c]	Inter-FPGA bus 2C readback.
MMAP_REG_TD_2D	0x0000D[c]	Inter-FPGA bus 2D readback.
MMAP_REG_TD_2E	0x0000E[c]	Inter-FPGA bus 2E readback.
MMAP_REG_CTRL_TAP	0x0000F[c]	Control bit readback.
MMAP_REG_TD_EXT_LSBS	0x00010	Front-panel LVDS readback (LSBs).
MMAP_REG_TD_EXT_MSBS	0x00011	Front-panel LVDS readback (MSBs).
MMAP_REG_STATUS	0x00012	Correlation status.
MMAP_REG_INPUT_DELAY	0x00013	LVDS input delay control.
MMAP_REG_CORL_MODE	0x00014	Correlation/pipeline operating mode.
MMAP_REG_DEMOD	0x00015	Phase-switch demodulation state.
MMAP_REG_IMPULSE	0x00016	Impulse response cycles & sample.
MMAP_REG_TEST_PIN	0x00017	Prompt input test pattern.
MMAP_REG_TEST_DIN	0x00018	Delay input test pattern.
MMAP_REG_OUT_ENABLE	0x00019	Inter-FPGA bus output enable.
MMAP_REG_DIG_DELAY	0x0001A[c]	Digitizer sample delay readback.
MMAP_REG_DIG_PHASE	0x0001B[c]	Digitizer sample phase readback.
MMAP_REG_DIG_GAIN	0x0001C	Digitizer sample gain.
MMAP_REG_DIG_OFFSET	0x0001D	Digitizer sample offset.
MMAP_REG_DIG_CTRL	0x0001E	Digitizer initialization and control.
MMAP_REG_DIG_PHASOR	0x0001F	Decimation phasor enable control.
MMAP_DELAY_BEGIN	0x00080	Start of delay/phase table buffers.
···	0x00080	Start of ring buffer 0, integration 0.
···	0x00080	Start of sub-ns coded tap table.
···	0x001CB	End of sub-ns coded tap table.
···	0x001CC	Fixed-point delay (16-bit frac).

5. Revised VHDL Components

Channel resolution in each bandwidth mode for the COBRA-based hardware and (15-input) revised correlator hardware is given in Tables 2-5. Each revised hardware table is for a specific (requantized) input sample bit width (COBRA hardware is limited to 2-bit samples). For 30-input operating modes (CARMA23/FULLPOL), channel resolution is precisely half that for the corresponding 15-input mode (e.g., 49 channels for 2-bit 500 MHz mode). Note however that the 3-bit and 4-bit 500 MHz modes are not available for CARMA23/FULLPOL (due to LVDS cable transmission limits).

Each correlator card data FPGA calculates four cross-correlations, for a total of 16 baselines per card. For a typical resolution of 257 channels per sideband (512 lags per baseline), the average FPGA-to-CPU data transfer rate is 2 MB/s (16 x 512 x 32-bit words = 32 KB each 15.625 ms). Optimizing the correlation logic should permit twice this resolution in narrowband modes (62 MHz and below); an upper-limit on the required data transfer rate is therefore $\approx$ 4 MB/s.

Numerous improvements were made to the correlation logic to take advantage of the revised hardware's capabilities and to streamline code maintenance. A single VHDL component (`correlation`) now covers all correlation logic instantiations required; a generic (`CORL_TYPE`) controls the type of correlation performed (auto, cross: positive lags, cross: negative lags, or cross: all lags). This component also manages the processing of multiple baselines by a single chip via the `NUM_CORL` and `NUM_PACK` generics. Digitizer quantization state counters are also handled by the `correlation` component, controlled by the generic (`NUM_QCNT`). The values of these generics are available in new control registers (see § 4) to enable automated lag retrieval and processing without the need to hard-wire configuration information into external initialization files, which is error-prone and hard to maintain (particularly during testing).

The correlation logic used in the first-light configurations introduced the concept of a "meta-lag", a word of arbitrary metadata written along with the primary lag and quantization state data during a lag dump. This facility has been significantly enhanced in the new `correlation` component. A generic (`NUM_META`) now controls the number of (32-bit) words of metadata read out during a RAM dump; in addition, a continuous stream of data is written to unused high memory while a correlation is active, providing integrated SignalTap-like functionality. The metadata as currently defined consists of the (prompt and delayed) input samples and—subject to bit-width constraints—the 16 bits per sample of metadata (such as the decimation phasor alignment bits) transported with the data by the pipeline.

For single-polarization configurations (`NUM_PACK` = 1), the 16-bit MSBs of each 32-bit data bus contains metadata for the sample(s), unless occupied by sample data; the lower byte is the primary metadata and the upper byte is the secondary metadata. Metadata is attached to the sample packet in the digitizer FPGA following decimation. The primary metadata consists of decimator phase information that can be used to verify phasor alignment between digitizers, and a single bit overflow/invalid indicator for that input stream (the MSB of the primary byte—or nibble, for the 3-bit 500 MHz mode). The phasor state counter $pstate(pBits-1 \text{downto} 0)$, where $pBits = \log_2(decRatio/2)$ and $decRatio$ is the decimation ratio relative to 500 MHz, occupies the primary metadata LSBs. It is not present (and unused bits are cleared) in the 250 MHz and 500 MHz modes. Secondary metadata consists of an *origin* nibble (4-bit LSBs) and a *sequence* nibble (4-bit MSBs), where *origin*(3 downto 2) = corl_mode(1 downto 0), *origin*(1 downto 0) = `FPGA_NUM` (0 for DigA and 3 for DigB), and *sequence* is a simple 4-bit counter. The latter can be used to verify clock synchronicity and relative pipeline alignments. Secondary metadata is not present in the 3-bit and 4-bit 500 MHz modes. For testing, secondary metadata can be replaced with local information by the

Table 1—Continued

Symbolic Name[a]	Hex Address[b]	Description
...	0x001CD	Fixed-point phase offset (16-bit).
...	0x001CE	Start of ring buffer 0, integration 1.
...	0x0155F	End of ring buffer 0, integration 15.
...	0x01560	Start of ring buffer 1.
...	0x02A40	Start of ring buffer 2.
MMAP_DELAY_END	0x03F1F	End of delay/phase table buffers.
MMAP_LAGS	0x04000	Start of lag data for correlation 0.
...	0x04000	Lag 0.
...	0x04001	Lag -1.
...	0x04002	Lag 1.
...	0x04003	Lag -2.
MMAP_LAGS +		
2 × NUM_PACK × NUM_LAGS	$A_{\rm meta}$[d]	Start of metadata for lag RAM 0.
...	$A_{\rm meta}+0$	Prompt input metadata word 0.
...	$A_{\rm meta}+1$	Delay input metadata word 0.
...	...	...
$A_{\rm meta}+2\times$ NUM_META	$A_{\rm qcnt}$[d]	Start of quantization state counters for lag RAM 0.
...	$A_{\rm qcnt}+0$	Sample count for quantization state 0x00.
...	$A_{\rm qcnt}+1$	*Cleared.*
...	$A_{\rm qcnt}+2$	Sample count for quantization state 0x01.
...	$A_{\rm qcnt}+3$	*Cleared.*
...	...	...
$A_{\rm qcnt}+2\times$ NUM_QCNT	$A_{\rm samp}$[d]	Start of continuous sample dump area for lag RAM 0.
...	...	...
$2^{1+\rm DUMP\_ADDR\_WIDTH}-1$	$A_{\rm end0}$[d]	End of continuous sample dump area for lag RAM 0.
2 × MMAP_LAGS	0x08000	Start of lag data for lag RAM 1.
...	...	...
(1+NUM_LAG_RAM) × MMAP_LAGS	$A_{\rm ptap}$[d]	Start of pipeline data tap block.
...	...	...
MMAP_FGPA_SIZE - 1	0x1FFFF	Top of local FPGA memory.

[a]Defined in $CCORL/revised/share/fpga/src/revised_components.vhd.

[b]As seen by the 32-bit external CPU memory interface.

[c]These locations are read-only.

[d]Calculable using the contents of MMAP_REG_CORL_CONF1 and MMAP_REG_CORL_CONF2.

Table 2. Spectral resolutions for COBRA-based correlator bands

Bandwidth (MHz)	Channels (per sideband)	δV[3 mm] (km/s)	V_{tot}[3 mm] (km/s)	δV[1 mm] (km/s)	V_{tot}[1 mm] (km/s)
500	17	94	1500	31	500
62	61	3.4	188	1.1	62.5
31	65	1.7	93.8	0.56	31.2
8	65	0.42	23.4	0.14	7.81
2	65	0.10	5.86	0.03	1.95

Table 3. Revised CARMA correlator spectral resolution [2-bit samples]

Bandwidth (MHz)	Channels (per sideband)	δV[3 mm] (km/s)	V_{tot}[3 mm] (km/s)	δV[1 mm] (km/s)	V_{tot}[1 mm] (km/s)
500	97	16	1500	5.2	500
250	193	4	750	1.3	250
125	321	1.2	375	0.39	125
62	385	0.49	188	0.16	62.5
31	385	0.24	93.8	0.081	31.2
8	385	0.061	23.4	0.020	7.81
2	385	0.015	5.86	0.005	1.95

Table 4. Revised CARMA correlator spectral resolution [3-bit samples]

Bandwidth (MHz)	Channels (per sideband)	δV[3 mm] (km/s)	V_{tot}[3 mm] (km/s)	δV[1 mm] (km/s)	V_{tot}[1 mm] (km/s)
500	41	38	1500	12	500
250	81	9.4	750	3.1	250
125	161	2.3	375	0.78	125
62	257	0.73	188	0.24	62.5
31	321	0.29	93.8	0.10	31.2
8	321	0.073	23.4	0.024	7.81
2	321	0.018	5.86	0.006	1.95

correlator on reception through TDEXT input; when activated (by setting corl_mode(4) = '1'), *origin*(3) = '1' when (corl_mode(1 downto 0) = "10") else '0', *origin*(2) = '1' or '0' for MSB- or LSB-derived TDEXT input, and *origin*(1 downto 0) = `FPGA_NUM` (of the correlator FPGA). The sequence nibble is also replaced with a locally-generated counter.

For dual-polarization configurations (`NUM_PACK` = 2), metadata is the 8-bit MSBs of the 16-bit data bus, and contains only the primary metadata. (The 2-bit 500 MHz and 4-bit 250 MHz modes have no metadata; the 3-bit 250 MHz mode contains only the upper nibble). In this case each 32-bit word of metadata dumped to RAM contains a pair of prompt-delay bus samples (the "prompt" metadata containing the prompt-delay pair for the first correlation and the "delay" metadata containing the inputs to the second correlation in that RAM block).

For FPGA minor version 2 and later, digitizer sample data on the data bus can be replaced by a simple counter by selecting `CORL_MODE` = 2 and setting `TEST_PIN` (FPGA #0) or `TEST_DIN` (FPGA #3) to 0xB0DE (this is known as "Bode mode"). For FPGA minor version 3 and later, unused sample bits on the data bus are filled with a simple counter (instead of cleared), in whole-nibble (MSB) increments; any partially used nibble is (MSB) zero-padded. The various counters available on the data bus are useful for determining data alignment (required for phase flattening). They are also used by the `td_bus_check` VHDL component to continuously verify the integrity of incoming front panel data.

To prevent latency problems with high resolutions, lag readout is organized for maximum throughput. The memory map allocates one M-RAM block per `NUM_PACK` baselines, each connected by a dedicated 64-bit bus. Hence (using integer arithmetic) there are `NUM_LAG_RAM=(NUM_CORL+NUM_PACK-1)/NUM_PACK` M-RAM blocks allocated to correlation logic for a particular FPGA. Since readout data (lags, quantization counts, and metadata) is 32-bit or less, the positive and negative lag/metadata streams are dumped in parallel, with the positive stream occupying the 32-bit LSBs of each 64-bit word. Readout for each baseline begins at the base of the corresponding M-RAM. Hence as seen from the external (32-bit) CPU interface, positive lags occupy even addresses and negative lags occupy the interleaving odd addresses, each followed by their metadata streams. The quantization state counts, if any, are appended to the positive lag/metadata channel. There is only one metadata and/or quantization stream per M-RAM block, even when `NUM_PACK` > 1 (cf. Table 1). The readout logic (like all components) runs at 125 MHz; hence a 1024-channel spectrum (`NUM_LAGS=1024`) can be dumped to memory in $\approx 8.2\,\mu$s, comfortably within the phase-switch settling

Table 5. Revised CARMA correlator spectral resolution [4-bit samples]

Bandwidth (MHz)	Channels (per sideband)	δV[3 mm] (km/s)	$V_{\rm tot}$[3 mm] (km/s)	δV[1 mm] (km/s)	$V_{\rm tot}$[1 mm] (km/s)
500	17	94	1500	31	500
250	33	23	750	7.8	250
125	65	5.9	375	2.0	125
62	129	1.5	188	0.49	62.5
31	161	0.59	93.8	0.20	31.2
8	161	0.15	23.4	0.049	7.81
2	161	0.037	5.86	0.012	1.95

interval of $\approx 20\,\mu$s. The region of each M-RAM following the main readout buffer is used to hold a continuous record of the incoming samples/metadata, which is useful for diagnostic purposes. The region is written (filled) only once per integration—i.e., it is *not* a circular buffer—and (depending on the lag count) can hold hundreds to thousands of samples. During normal operation its contents will not be read by the CPU, but it is always available for inspection should problems arise. The correlation logic accepts a `DUMP_ADDR_WIDTH` generic determining the number of (64-bit) locations per block it can write. Currently, the configurations write to only the first half of each block; the remaining half is unused.

Following the baseline data RAM blocks is a single M-RAM block dedicated to pipeline data tap output. The pipeline component includes a generic 64-bit output bus that can be used to inspect arbitrary data passing through it. (The data connection is fixed for a particular FPGA configuration.) A bit on the `corl_ctrl` input bus (part of the `FPGA_DATA` bus) coming from the system controller is used to control when to dump the bus contents to RAM. Samples are dumped when the control bit is high, starting at the beginning of the block, until either the block is filled or the control bit is de-asserted, whichever comes first. It is not possible to restart a dump in the middle of a block—clearing the control bit resets the write address to the beginning of the block. The block holds 8192 64-bit samples, which requires 65.536 μs to fill at a clock frequency of 125 MHz. This is a diagnostic facility and is not used in normal operation.

6. FIR Filter Design

6.1. Digital Decimation Filters

The revised correlator hardware implements all sub-500 MHz bandwidth modes via digital decimation and band-shaping. Decimation is a multi-stage process involving a series of simple half-band filters, each followed by decimation by a factor of two. Successive filters gradually increase in complexity as the final sample rate is approached. Following decimation, a single high-precision edge-defining filter is used to produce a sharp final bandpass. For additional discussion of the method, see Rauch (2003).

All revised correlator bandwidth modes were designed to meet the following criteria:

1. Peak-to-peak bandpass ripple of 0.1 dB.
2. Peak sidelobe power below -40 dB.
3. Two edge channels aliased in excess of -40 dB (@ nominal resolution).
4. Three edge channels aliased in excess of -40 dB (@ maximum resolution).

The anti-aliasing specification implicitly depends on the channel resolution. Nominal resolution refers to the channel resolutions achieved as of July 2007; maximum resolution was defined to be 50% higher than this. The general recommendation for observers is to clip the outer three edge channels to avoid aliasing artifacts (one of these being the phaseless, half-width edge channel). The preceding figures refer to the digital decimation filters only; in the wider bandwidth modes, the analog filter in the spectral downconverter limits band flatness to 1-3 dB peak-to-peak and, for 500 MHz, edge channel anti-aliasing (-20 dB at 475 MHz and -15 dB at 480 MHz).

Figures 14-17 plot the net decimation filtering response for the revised 500, 250, 125, and 62.5 MHz bands, respectively. The spectral downconverter analog filter response is *not* included. In each case the band is centered on zero frequency (DC) when the filter is applied, and subsequently re-modulated into the positive frequency band; hence only half of each band (and residual out-of-band artifacts) is shown. Filter response is symmetric about zero frequency. The heavy (green) line is the signal level and the thinner (blue) line is the aliased noise level. Vertical lines denote the location of spectral channel boundaries for the mode. The number of filter taps used increases as the bandwidth decreases (and number of channels increases; cf. Tables 3-5), in such a way that only ~ 2 edge channels suffer noticeable out-of-band aliasing, regardless of the absolute channel resolution. The corresponding plots for the narrowband modes (31, 8, and 2 MHz) are very similar to Figure 17 as the same edge-defining filter is used in all four modes.

6.2. Fractional Sample Delay Filters

Figure 18 displays the amplitude and phase performance of the set of sub-ns (fractional sample) delay filters used in the digitizer cards to align antenna input signals. The dashed vertical lines indicate the limiting "usable" analog bandwidth, defined here as the frequency beyond which out-of-band aliasing exceeds -18 dB. Filter performance depends on the delay (each delay corresponds to a unique set of filter coefficients). The dashed lines denote the worst-case amplitude and phase variations over all possible delays; the solid curve plots the response for one particular delay. Amplitude response is flat to within 0.1 dB over the lower 94% of the 500 MHz band, and phase delay accuracy meets the nominal CARMA specification over the lower 97% of the 500 MHz band. All remaining bandwidth modes—carved from the center of the 500 MHz band—exceed the desired accuracy over 100% of the band. The width of the region near Nyquist exhibiting degraded delay performance scales inversely with the filter length; logic usage increases linearly with the filter size. The implemented filter (discussed in detail below) balances the two competing constraints.

Delay filter coefficients are computed using a windowed sinc function method, which has a closed-form solution for each coefficient. Given a fractional delay δ, $0 \leq \delta < 1$ and a filter containing N coefficients, the formula for coefficient c_i, $i \in \{0,\ldots,N-1\}$ is

$$c_i = \mathrm{W}(i-\delta,N)\,\mathrm{sinc}[\pi(1-\varepsilon)(i-D)],$$

where W is the windowing function, ε is a fixed parameter chosen to minimize delay errors, and $D = \delta + N/2$ ($\delta < 1/2$) or $D = \delta + N/2 - 1$ ($\delta \geq 1/2$) is the effective filter delay. The filters developed for the CARMA digitizers employ a Hann window function, $\mathrm{W}(x,N) = [1-\cos\{2\pi x/(N-1)\}]/2$, with $N = 86$ and $\varepsilon = 0.01165$. The coefficients are quantized to 15-bit precision for use by the digitizer FPGA sub-ns delay filter component (`frac_delay`). Quantization reduces the net number of coefficients to a maximum of $N = 80$, as several edge coefficients underflow to zero. An optimized subroutine to compute CARMA digitizer delay coefficients, `fd_carma_subns_coef()`, can be found in $CCORL/share/fpga/test/frac_delay.c. See Laakso, Valimaki, & Karjalainen (1996) for a review of fractional delay filter design methods.

The Altera FIR filter component requires that reloadable filter coefficients be stored in (private) RAM blocks. The contents of this tap RAM must be loaded serially into the component using a hard-wired interface; the RAM is not directly accessible. Updating delay coefficients in the digitizer FPGAs is done by sequentially reading an area in the main (PPC-visible) FPGA memory map, treated as a circular buffer. The address limits for this buffer are `MMAP_DELAY_BEGIN` and `MMAP_DELAY_END` (cf. Table 1). The output of this buffer is then fed into the appropriate filter components. The Altera FIR components can only receive

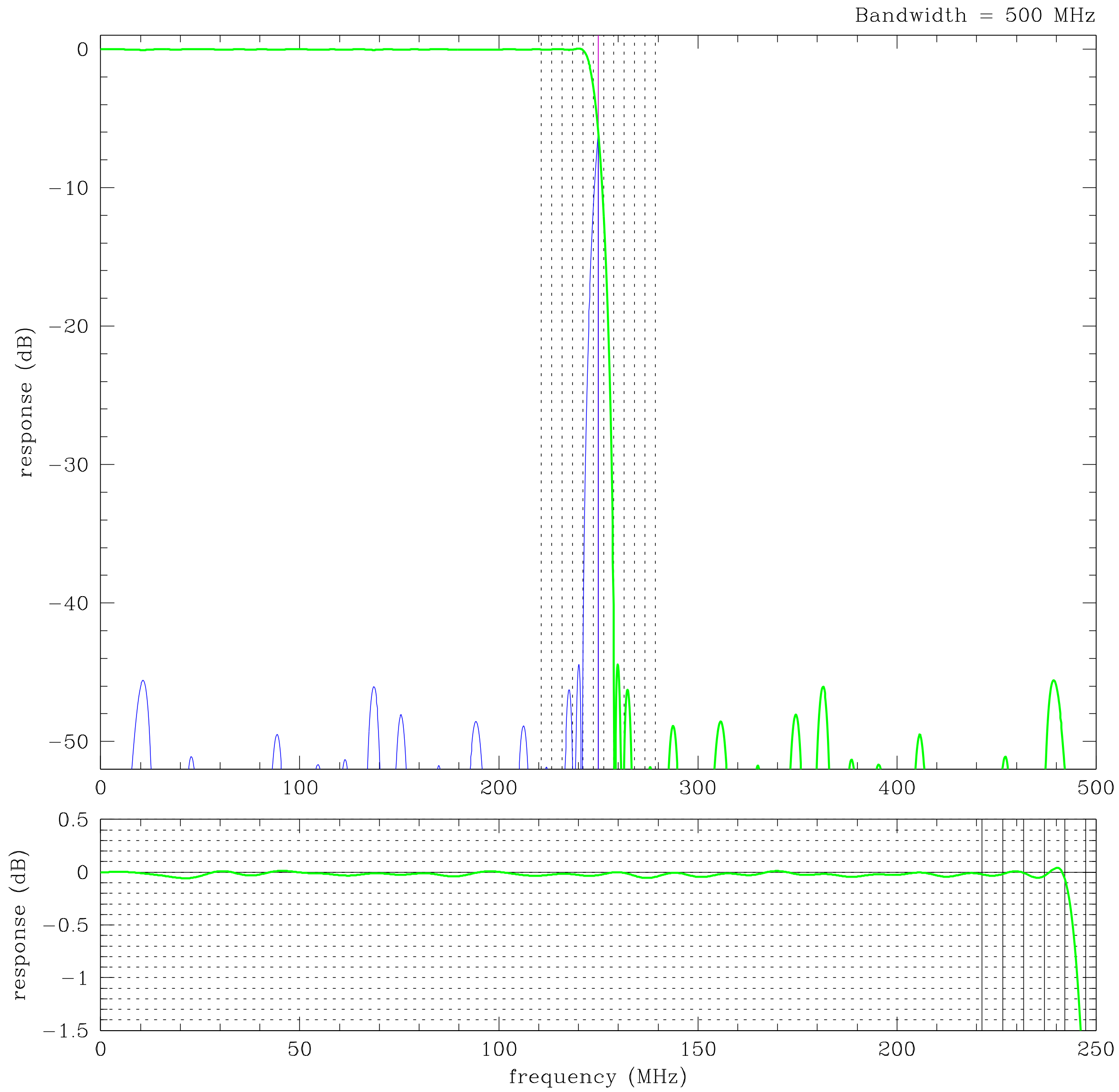

Fig. 14.— Revised CARMA correlator decimation filter performance for the 500 MHz band.

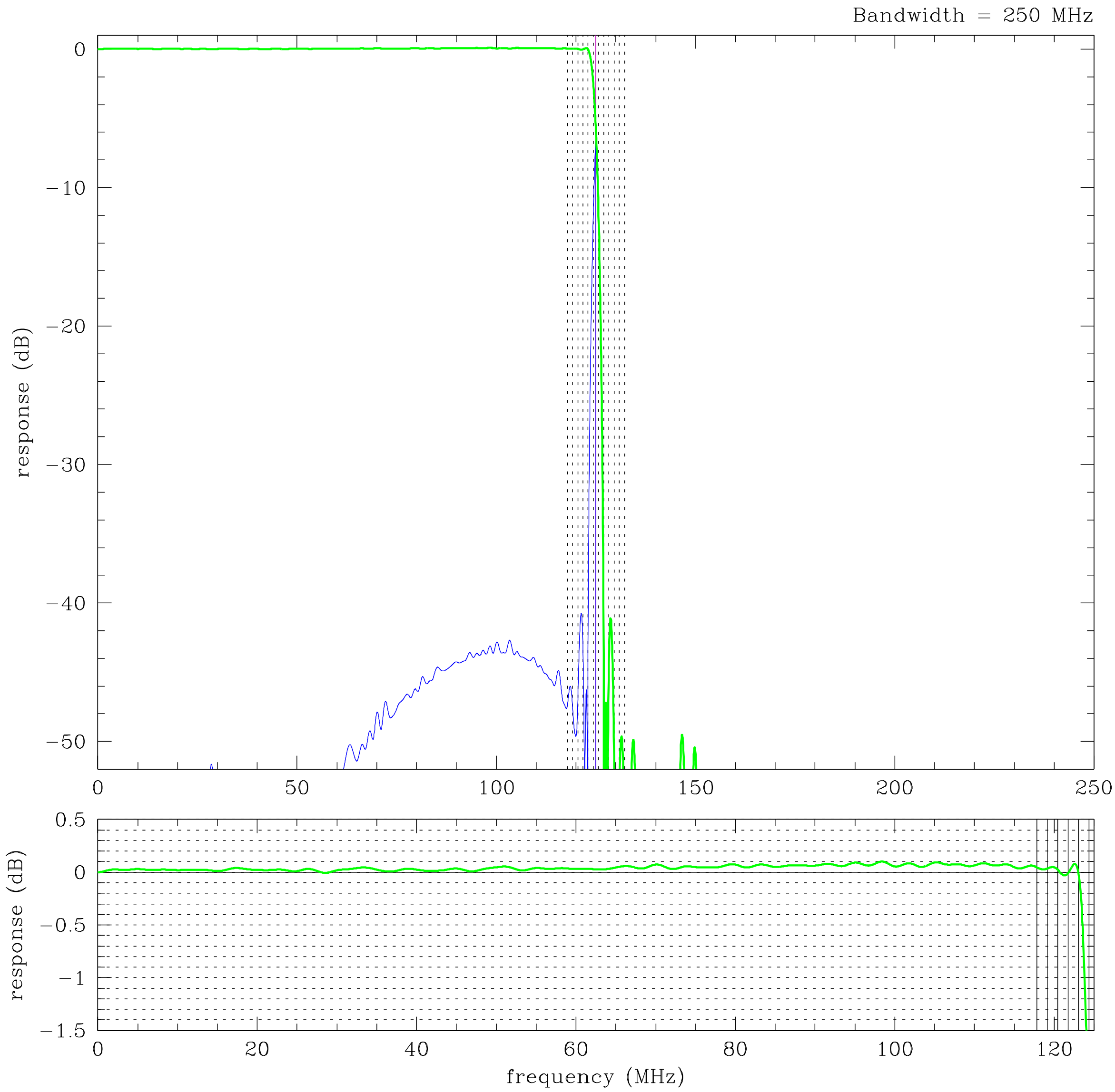

Fig. 15.— Revised CARMA correlator decimation filter performance for the 250 MHz band.

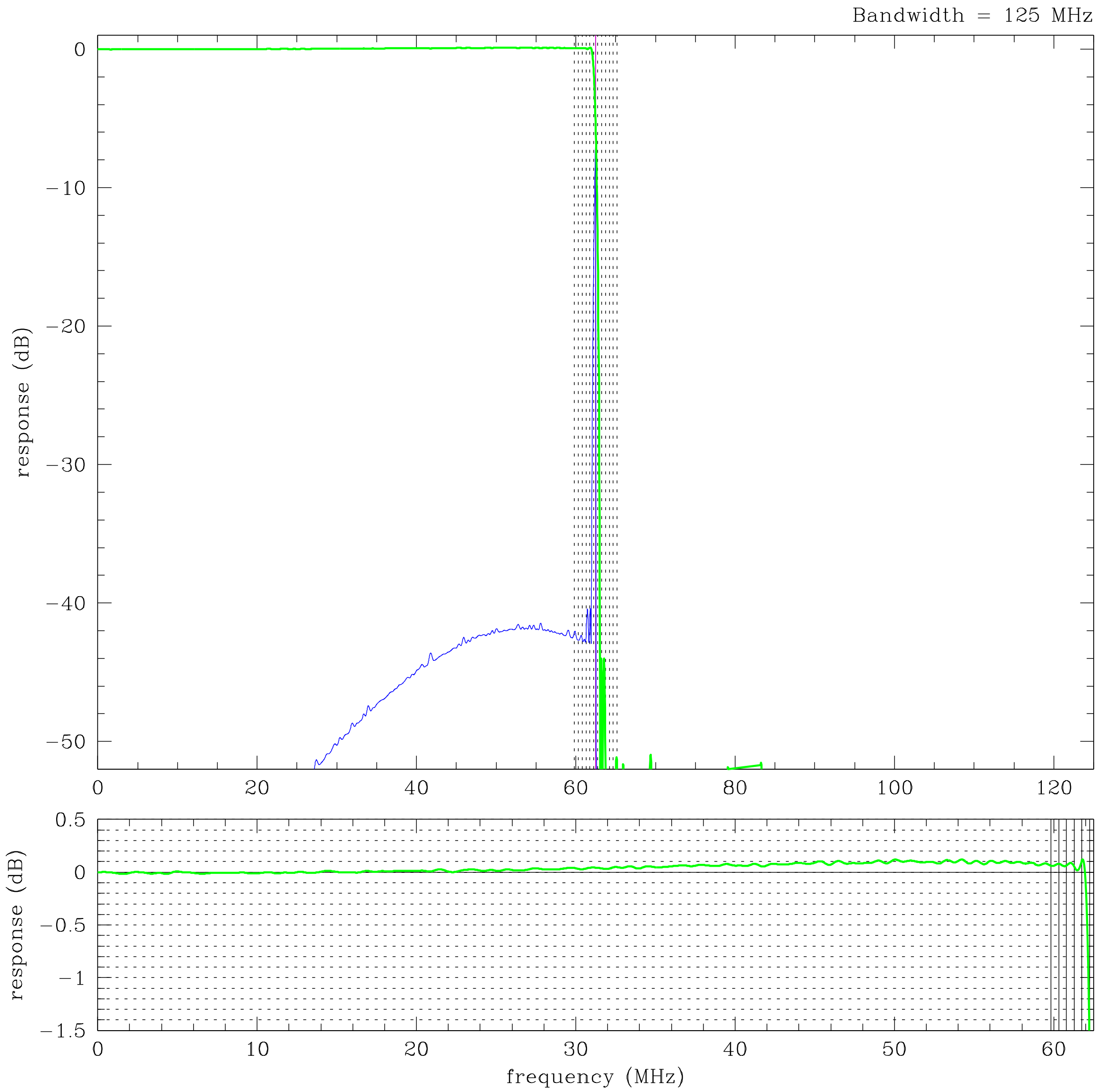

Fig. 16.— Revised CARMA correlator decimation filter performance for the 125 MHz band.

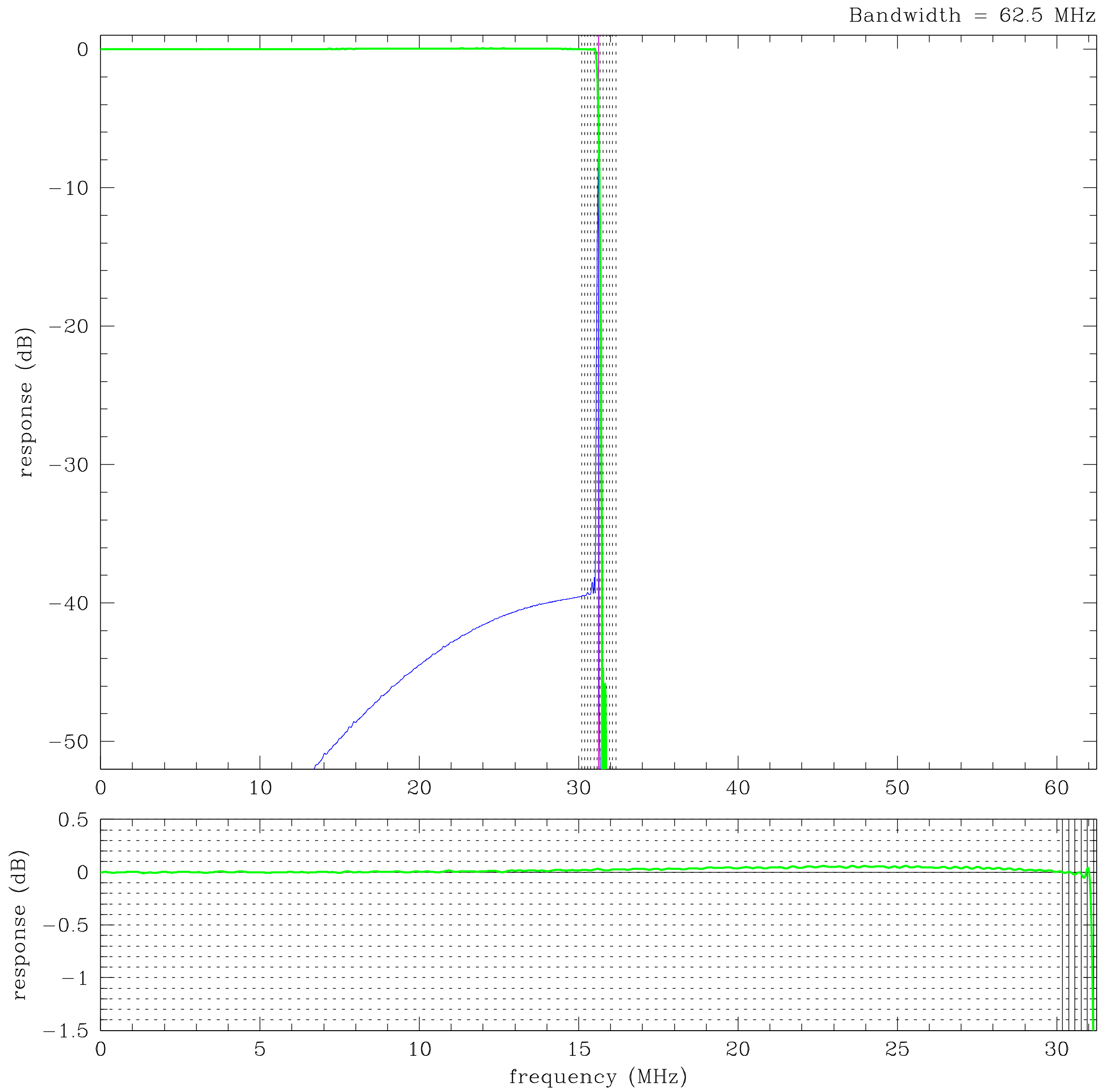

Fig. 17.— Revised CARMA correlator decimation filter performance for the 62 MHz band.

one input sample and produce one output sample per clock. The digitization logic outputs demux-by-8 @ 125 MHz samples (say x_0 to x_7); hence in practice the sub-ns delay component needs to instantiate a vector of 8 identical filters, each accepting 8 input samples and producing one output sample per clock (y_0 to y_7). The input samples fed to each filter are skewed by one clock cycle between filters, so that the phase of each output sample matches that of its youngest input sample. Specifically, filter $i \in \{0,\ldots,7\}$ receives input samples x_{i-7} to x_i; implementing this requires caching the 7 youngest input samples from the previous clock cycle.

Since each filter must consume 8 input samples per clock, these filters themselves need to be split into 8 sub-filters, each containing 1/8 of the filter coefficients, whose individual outputs are summed to produce a single output sample. To fully utilize logic resources, the number of filter taps should therefore be a multiple of 8. For the CARMA digitizers $N = 80$ and each sub-filter contains 10 taps. In terms of the taps c_i, $i \in \{0,\ldots,N-1\}$ of the original filter, the taps a^k_j, $j \in \{0,\ldots,(N/8)-1\}$ of sub-filter k, $k \in \{0,\ldots,7\}$, are $a^k_j = c_{8j+k}$; the input to filter k is x_k.

Altera FIR filters with reloadable taps do not accept the actual coefficients as input, but rather a coded data stream produced by their `coef_seq` command line utility. A copy of the C++ source code for `coef_seq` can be found in $CCORL/share/fpga/test. The format of the coded data is not specified by Altera; empirically, for a filter with N x m-bit taps, it consists of 2^N x $(m+3)$-bit values (the first of which is always zero) representing some kind of pre-computed multiplication table. The critical thing to note is that the amount of coded data increases exponentially with the number of taps and linearly with the bit-width of the input (for a fully parallel filter, as required here). A Stratix II M512 block has a capacity of 32 x 18-bit elements, precisely enough to hold the coded data for a filter with 5 x 15-bit taps. The EP2S90 FPGAs loaded into digitizer cards contain 488 M512 blocks and 408 M4K blocks (with 256 x 18-bit capacity). To reduce RAM usage to an acceptable level, the 10-tap sub-filters are further sub-divided into two 5-tap filters. In total the sub-ns delay filter instantiates 128 divided sub-filters, requiring 128 RAM blocks per bit of input. The implemented component accepts 6-bit input samples—in practice, the number of meaningful bits expected from the digitizers—and consumes 480 M512 blocks and 288 M4K blocks (as the latter are poorly utilized in this context, their use should be minimized). Output samples are rounded to 8-bit precision.

The coded coefficient stream for a single delay filter therefore consists of $(N/5) \times (2^5 - 1) = 496$ x 18-bit values (the leading zero in the coded streams are inserted on-the-fly by the filter reload state machine). Since the FPGA memory map is 64-bit wide internally, it is convenient to group values into triplets occupying the lower 3 x 18 = 54 bits of each quadword of the circular buffer in FPGA RAM. This nominally amounts to 166 quadwords of FPGA RAM per delay coefficient set; however, an extra quadword containing the whole-ns delay value (32-bit LSBs) and normalized phase-offset (32-bit MSBs) is appended to each coefficient set. Hence each phase/delay update occupies 167 quadwords (334 longwords) of FPGA RAM. The phase/delay update rate is 64 Hz (1/15.625 ms), which implies an average data transmission bandwidth of $\sim$ 83.5 KB/s to each of the two digitizer FPGAs receiving this information (FPGA #1 and #2); since the CPU bus is shared, the effective load is $\approx$ 167 KB/s, a very small fraction of the available bandwidth. The memory map allocates one Stratix II M-RAM block (8192 quadwords) for the delay buffer, enough to hold 750 ms of phase/delay update data (48 sets x 167 = 8016 quads).

The precise contents of the delay/phase coefficient table are as follows. The `frac_delay` component receives two coded coefficient streams in parallel, one for each sub-channel (5-tap) filter of the pair which together implement a particular 10-tap sub-filter, as described above. To support parallel readout, the delay table therefore interleaves coefficients from two individual filters. Let $a^{k.0}_l = a^k_l$ and $a^{k.1}_l = a^k_{l+5}$, $l \in \{0,\ldots,4\}$, be the coefficients for the two sub-channel filters implementing sub-filter k, and denote the corresponding `coef_seq`-coded tap streams by $s_m^{k.0}$ and $s_m^{k.1}$, $m \in \{0,\ldots,31\}$, where $s_0^{k.0} = s_0^{k.1} = 0$. One delay/phase table consists of 167 quadwords; except for the final quad (containing the whole-ns delay and phase offset), each quadword contains three 18-bit coded taps, packed contiguously into the LSBs. Let $T_i = \{s_2, s_1, s_0\}$ represent table quadword i, where s_m are the coded tap values (in MSB to LSB order), and unused MSBs are cleared (though ignored by the FPGA). FPGA RAM addresses increase with i, and quadwords are stored in little-endian order (RAM addresses refer to 32-bit words). Then

$$\begin{aligned}
T_0 &= \{s_2^{0.0}, s_1^{0.1}, s_1^{0.0}\} \\
T_1 &= \{s_3^{0.1}, s_3^{0.0}, s_2^{0.1}\} \\
&\vdots \\
T_{20} &= \{s_1^{1.0}, s_{31}^{0.1}, s_{31}^{0.0}\} \\
T_{21} &= \{s_2^{1.1}, s_2^{1.0}, s_1^{1.1}\} \\
&\vdots \\
T_{164} &= \{s_{31}^{7.0}, s_{30}^{7.1}, s_{30}^{7.0}\} \\
T_{165} &= \{0, 0, s_{31}^{7.1}\} \\
T_{166} &= \{\Phi, \Delta_{\mathrm{ns}}\},
\end{aligned}$$

where $\Phi = \lfloor 2^{16} \cdot [\phi/(2\pi)] \rfloor$ is the 16-bit fixed-point phase offset for absolute phase offset ϕ, and Δ_{ns} is the 16-bit fixed-point ns delay value (cf. the description of `MMAP_REG_DIG_DELAY`). Both are 32-bit quantities. Note that the null ($s_0^{k.x} = 0$) coded taps are not stored in the table, but inserted on-the-fly during the reload process.

7. Phase Offset Correction

In contrast to the first-light CARMA correlator, the revised correlator applies phase corrections continuously to incoming samples as part of the signal decimation process; the corrections remove antenna-based phase offsets arising from bulk downconversion and lobe-rotation differentials. As described in Rauch (2003), downconversion entails shifting the band center from half the Nyquist frequency (250 MHz) to DC by multiplying the input samples $\{x_k\}$ by $e^{-i\pi f_0 k}$, where $f_0 = -1/2$. (Note that in the CARMA band definition, which like FFTW assumes the forward FFT uses the e^{-i} sign convention, this corresponds to centering *negative* frequencies onto DC. This fact motivates the sign choice in defining ϕ below.) The resulting frequency modulated samples are $\{x_0, +ix_1, -x_2, -ix_3, x_4, \ldots\}$. Phase correction in the revised correlator is implemented by further multiplying the modulated samples by $e^{-i\phi}$, where ϕ is the phase offset parameter (identical to the antenna-based phase offset correction in the CARMA band definition). The corresponding phase-corrected, frequency modulated samples are $\{x_0\cos\phi - ix_0\sin\phi, x_1\sin\phi + ix_1\cos\phi, -x_2\cos\phi + ix_2\sin\phi, -x_3\sin\phi - ix_3\cos\phi, x_4\cos\phi - ix_4\sin\phi, \ldots\}$.

A new offset ϕ is applied each integration by the decimation logic in digitizer FPGA #0; the value of ϕ to use is driven by FGPA #1 onto the "E" data bus connecting FPGA #1 to #0 (and similarly by FPGA #2 for FPGA #3) and remains static for the duration of each integration. The values are stored in FPGA RAM as part of the delay/phase tables (see Table 1); each time the phase/delay reload occurs between integrations, a new value of ϕ appears on the bus. Applying the offset requires calculation of the $\cos\phi$ and $\sin\phi$ factors.

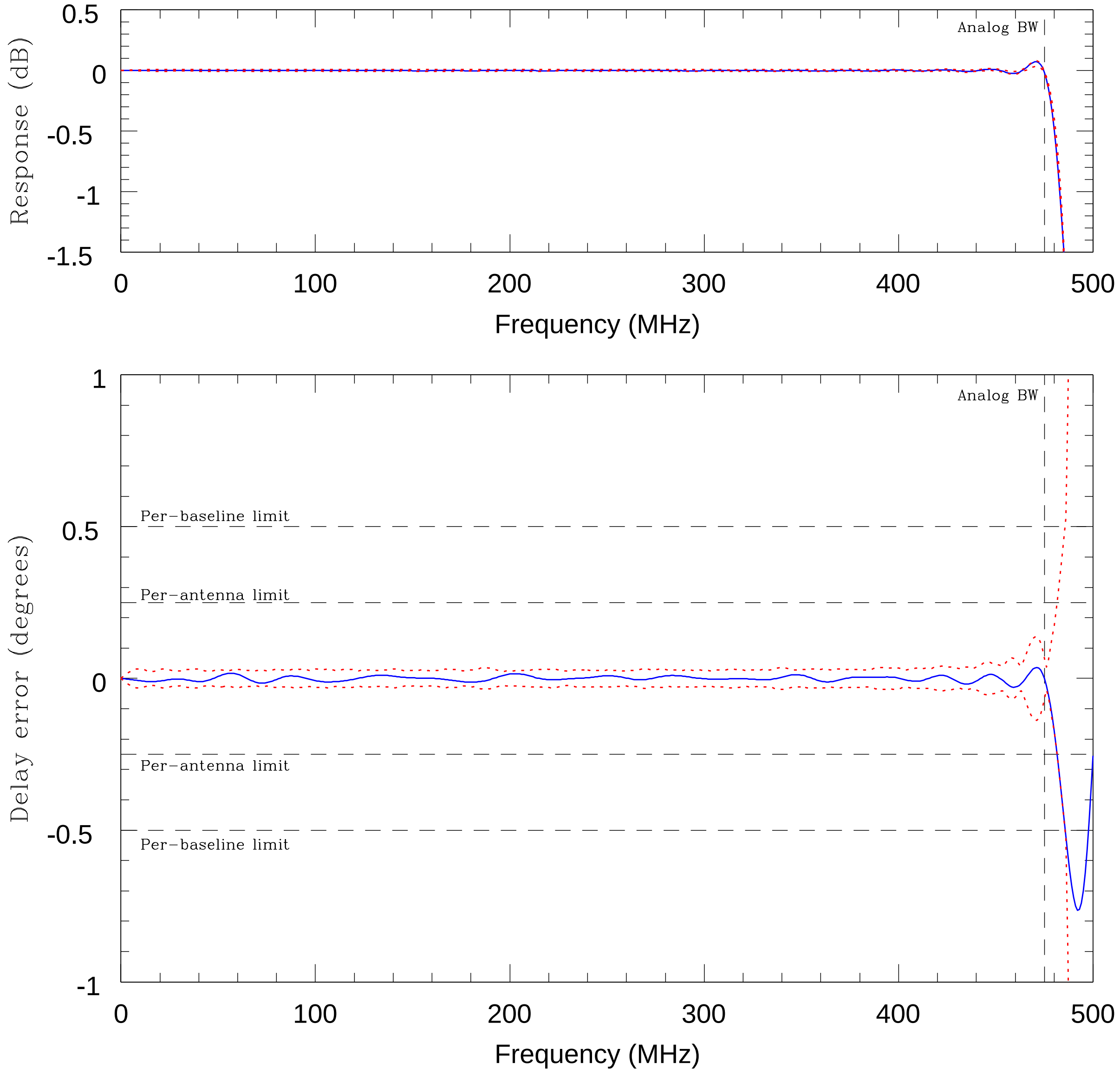

Fig. 18.— Revised CARMA correlator fractional sample delay filter performance. The top and bottom windows display frequency and delay responses, respectively. CARMA-defined per-antenna and per-baseline limits are also indicated.

An Altera numerically controlled oscillator (NCO) megacore component is instantiated for this purpose by the decimation logic. The NCO component accepts scaled output frequency and phase parameters, $\phi_{\rm INC}$ and $\phi_{\rm PM}$ respectively, where the physical output frequency $f_{\rm o} = \phi_{\rm INC} f_{\rm clk}/2^M$ and phase $\phi = 2\pi\phi_{\rm PM}/2^P$. Here $f_{\rm clk}$ is the actual frequency of the NCO input clock, M is the internal accumulator precision, and P is the angular precision. The NCO component outputs two waveforms in two's-complement format, $2^{N-1}\sin(2\pi f_{\rm o}t+\phi)$ and $2^{N-1}\cos(2\pi f_{\rm o}t+\phi)$, where N is the magnitude precision. The parameters M, N, and P are fixed for each NCO instantiation; the $\phi_{\rm INC}$ and $\phi_{\rm PM}$ inputs are fully dynamic (can change each clock cycle). In the present case the NCO does not actually oscillate ($\phi_{\rm INC} = 0$) as we are only interested in computing the (quasi-)static values $\sin\phi$ and $\cos\phi$. The CARMA implementation uses $M = 24$, $N = 18$, $P = 18$, and rounds the resulting phase-corrected, frequency modulated samples to 12-bit precision. The choice $P = 18$ corresponds to a phase resolution of $360/2^{18} = 1.4\times 10^{-3}$ deg.

8. Host Integration

The downloadable configuration bitfiles produced by the TCL synthesis scripts are named using a convention closely related to the version control register available in the memory map. In particular, bitfiles are named according to the following template:

```
carma_v${HW}${TYPE}${BITS}${BW}${CF}${MINOR}_${TYPE}${NUM}.rbf
```

where `HW` is the hardware revision (0 for COBRA, 1 for revised CARMA); `TYPE` is the FPGA type (0xC/0xD for correlator/digitizer chips); `BITS` is the requantized (post-decimation) sample bit width; `BW` is the bandwidth mode number (1 for 500 MHz, 2 for 250 MHz, etc.); `CF` is the bus configuration number; `MINOR` is the minor version; and `NUM` is the FPGA chip number (chip #0 connects to the top left front-panel connector, #3 to the bottom right).

REFERENCES

Laakso, T.I., Valimaki, V., and Karjalainen, M. 1996, IEEE Signal Processing Magazine, vol. 13, no. 1, pp. 30-60.

Rauch, K.P. 2003, CARMA Memo 12.