

The ALMA Correlator Digital Filter Card

Ray Escoffier, 11-07-01

Overview

The ALMA correlator digital filter card provides bandwidth selection in the ALMA array. It accepts as its input the output of an ALMA digitizer that works at a constant sample rate of 4 GS/S. The digital filter card was designed to handle 4-bit samples from a system digitizer even though the final design of the ALMA digitizer will provide only 3-bit samples.

The input to the digital filter in the ALMA system is in the form of 32 parallel 3-bit wide data streams, each with a clock rate of 125 MHz matching the effective 4 GS/S rate of the digitizer. In this signal format, any one of the 32 X 3 data streams that drive a filter card carries bits from every 32nd sample. The filter card output is mainly in the form of 2-bit samples although some modes with 4-bit output samples are possible.

The filter card has 6 main modes of operation:

MODE	# TAPS	INPUT	OUTPUT FORMAT	# 125 MHz OUTPUTS
ALL PASS	1	3-BIT, 4 GS/S	2-BIT, 4 GS/S	32
1/2 BAND	128	3-BIT, 4 GS/S	2-BIT, 2 GS/S	16
1/4 BAND	256	3-BIT, 4 GS/S	2-BIT, 1 GS/S	8
1/8 BAND	512	3-BIT, 4 GS/S	2-BIT, 500 MS/S	4
1/16 BAND	1024	3-BIT, 4 GS/S	2-BIT, 250 MS/S	2
1/32 BAND	2048	3-BIT, 4 GS/S	2-BIT, 125 MS/S	1

All of the modes above are currently envisioned to be low pass filters. However, the filter card is completely programmable and other modes of operation including the use of band pass filters are possible. The card has 2048 hardware tap weight multipliers which must be distributed among the required outputs of the card. For example, when the card is programmed as a 1/2 band low pass filter, 16 parallel 125-MHz outputs are required to supply the aggregate 2 GS/S requirement. In this case, the 2048 hardware tap weight multipliers are distributed among the 16 card outputs and the card performs as a 2048/16 or 128-tap filter.

At the other extreme, a 1/32 band (62.5 MHz bandwidth) filter requires only a single (2-bit wide) 125 MHz output for Nyquist sampling and the card can have 2048-tap weigh performance.

All modes of the ALMA correlator digital filter card are possible using a single set of FPGA personalities. The various configurations possible are obtained by down loading appropriate mode programming bytes and unique tap weights.

Arithmetic in the digital filter is, for the most part, done with 10-bit arithmetic precision. The 3-bit or 4-bit input samples to the card are multiplied by programmed tap weights and yield 10-bit results. The first 5 stages of the following adder tree for summation of tap weight multiplier outputs adds with 10-bit precision. Tap weight scaling is controlled such that only a single bit of head room growth is permitted in these first 5-stages of the adder tree. The final stages of adder tree (the last 2 to 5 stages depending on mode) are done with 7-bit or 8-bit precision with another bit of head room growth permitted in these stages.

Re-sampling of the 8-bit filter adder tree output is done and yields 2-bit samples for compatibility with the correlator chips of the ALMA correlator. The final sampler at the filter card output can also be made to produce 4-bit samples from the 8-bit adder tree output. This mode of operation can be used to lessen the loss of sensitivity incurred in the second stage of quantization at the cost of a factor of 4 in frequency resolution in the correlator.

A small delay line is present in the input stage of the filter card. The delay function is required for two reasons. First, in modes of operation where sample decimation is done, delay is required before the

decimation in order to maintain 1/16 Nyquist sample delay resolution required by the ALMA specification. For example, when the filter card is programmed as a 1/32 band filter, the card output is a single 125 MHz output required for Nyquist sampling. Any delay applied after the decimation of the filter card can have a resolution of at best one sample time. Thus, delay must be applied before final sample decimation to achieve 1/16 final sample period resolution. Note that the phase shift mechanism of the ALMA digitizer (where the sample clock can be phase shifted and set to 1/16 of a 4 GS/S sample relative to the system clock) gives much more resolution than necessary in low bandwidth modes.

The second reason for the delay line at the filter card input is purely a correlator reality. Tracking delay in the ALMA correlator is set in the station card by offset addressing of a 64-bit wide RAM buffer. This action yields delay resolution to 64-bits only and putting the additional delay trim at the filter card input is just a convenience.

The filter card has state counters at both the card input and the card output. These state counters will count the frequency of each of the 8 input states or 4 output states (possible with the 3-bit input and 4-bit output). Another feature of the filter card is that it can either replace input samples with pseudo random data or it can test input data for adherence to a correlator 35-bit standard pseudo random pattern for testing purposes.

Block Diagram

A block diagram of the ALMA digital filter card is seen in Figure 1, below:

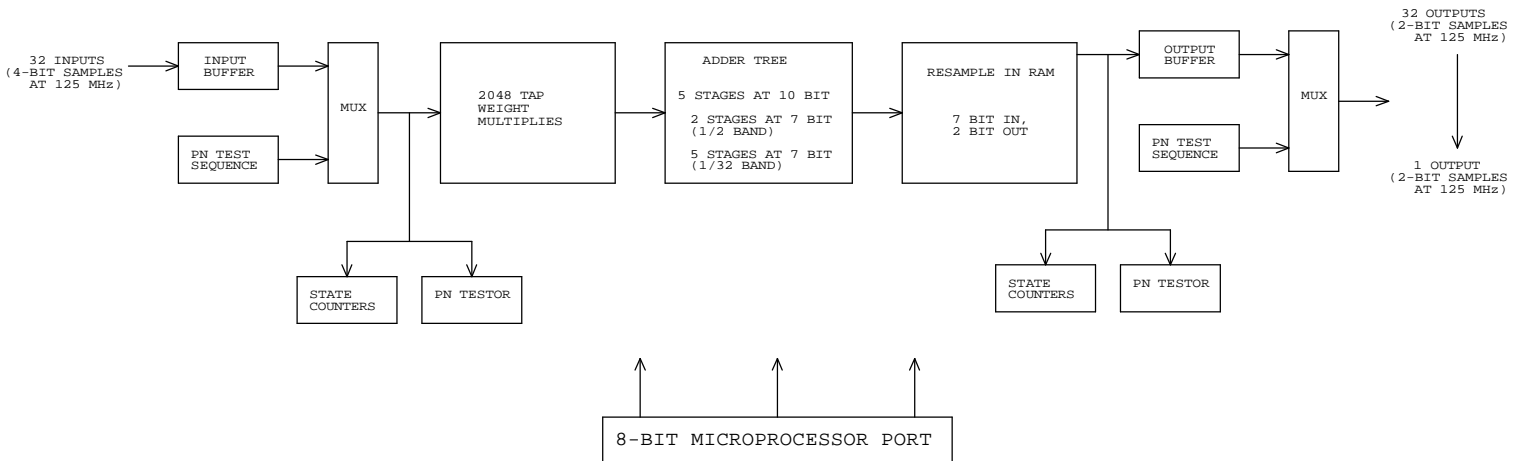


Figure 1, digital filter card block diagram

Figure 1 shows 32 parallel 4-bit samples driving the filter card input (in the ALMA system it will be driven by the 3-bit samples developed in the ALMA digitizer). The card output varies from 1 two-bit to 32 two-bit samples depending on the mode selected. Test logic that can substitute a pseudo random data pattern at either the card input stage or at the card output stage is seen. The card also has pseudo random data test circuits at both the card input and card output that can perform error rate measurements by counting deviations in the data from a 35-bit pseudo random data sequence.

The filter card also has an 8-bit ALMA correlator standard microprocessor port. This microprocessor port is used to load the FPGA personalities initially, to load tap weight and mode information in normal operation, or it can be used to monitor operation and collect measurements from the state counters or error counters.

Filter Card Schematic

The Orcad schematic of the ALMA filter card is seen in drawing number ALMA:07:001: SX:001. Sheet 1 of this schematic shows the data input connectors P2 and P3, the card output connectors P1 and P4. It also shows the CPLD microprocessor port IC, U6, and a sketch of the chip layout for the card. Most of the logic on the filter card is shown as a hierarchical block, FIR, detailed on sheet 2 of the schematic.

Other items shown on sheet 1 include:

- IC U2, the 125 MHz clock sinewave-to-LVCMOS converter
- IC U3, a differential line receiver for the microprocessor port clock
- IC U1, a reset chip (not used)
- The J1 test point connector

Sheet 2 of the filter schematic shows all of the 14 FPGAs that supply most of the logic on the card. Each FPGA appears on sheet 2 as a hierarchical block and each has its own schematic page. The FPGAs include 2 DELAY chips, 8 FIR chips and 4 ADDER chips. The DELAY chips of the digital filter card are driven by input samples to the card and provide sample fanout to the first rank of FIR (tap weight multiplier) FPGA chips (U7, U8, U9, and U10). FIR chips U7...U10 in turn provide sample fanout to the second rank of FIR chips (U11, U12, U13, and U14). Partial sum outputs from all FIR chips drive each of the 4 ADDER FPGA chips. Figures 2 and 3 below show the sample and partial sum interconnect on the filter card.

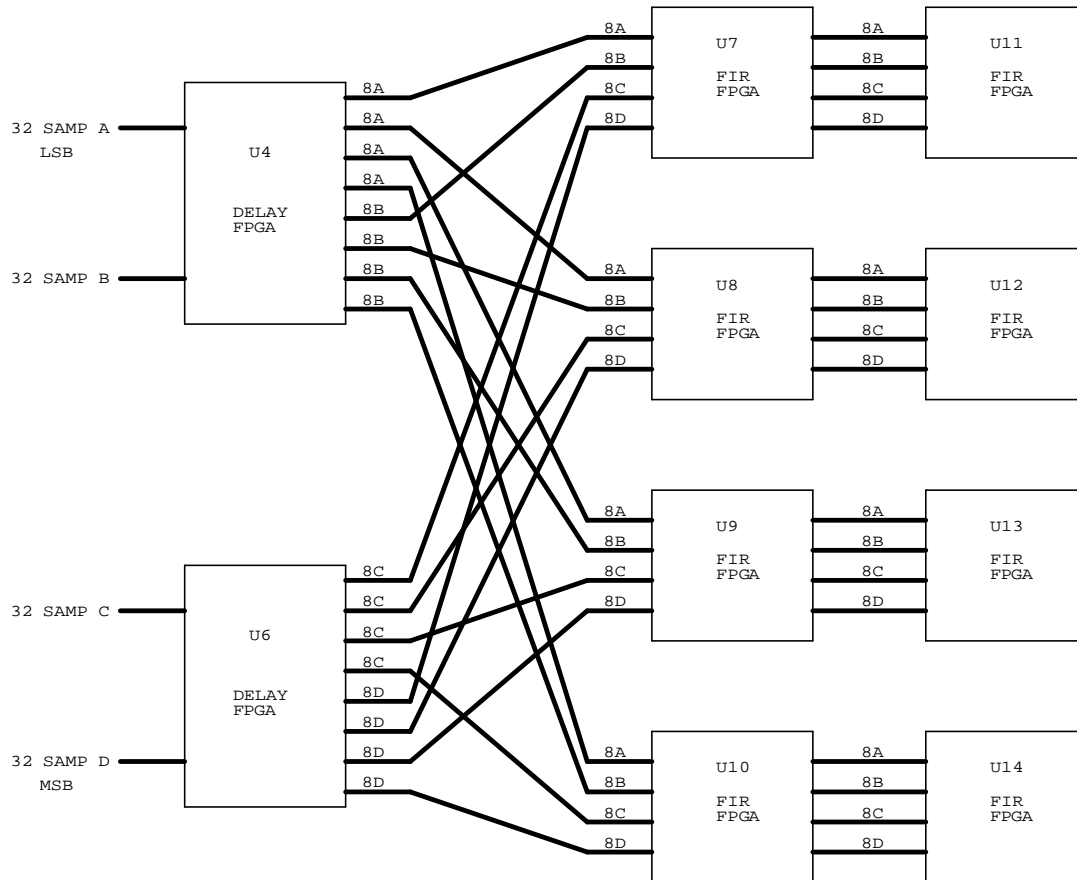


Figure 2, filter card sample interconnect

The four 32-bit busses that drive the DELAY FPGAs (A, B, C, and D) represent the 32-way demultiplexed output of a 4-bit digitizer. Each of the busses that drive into the FIR chips is 8-bits wide.

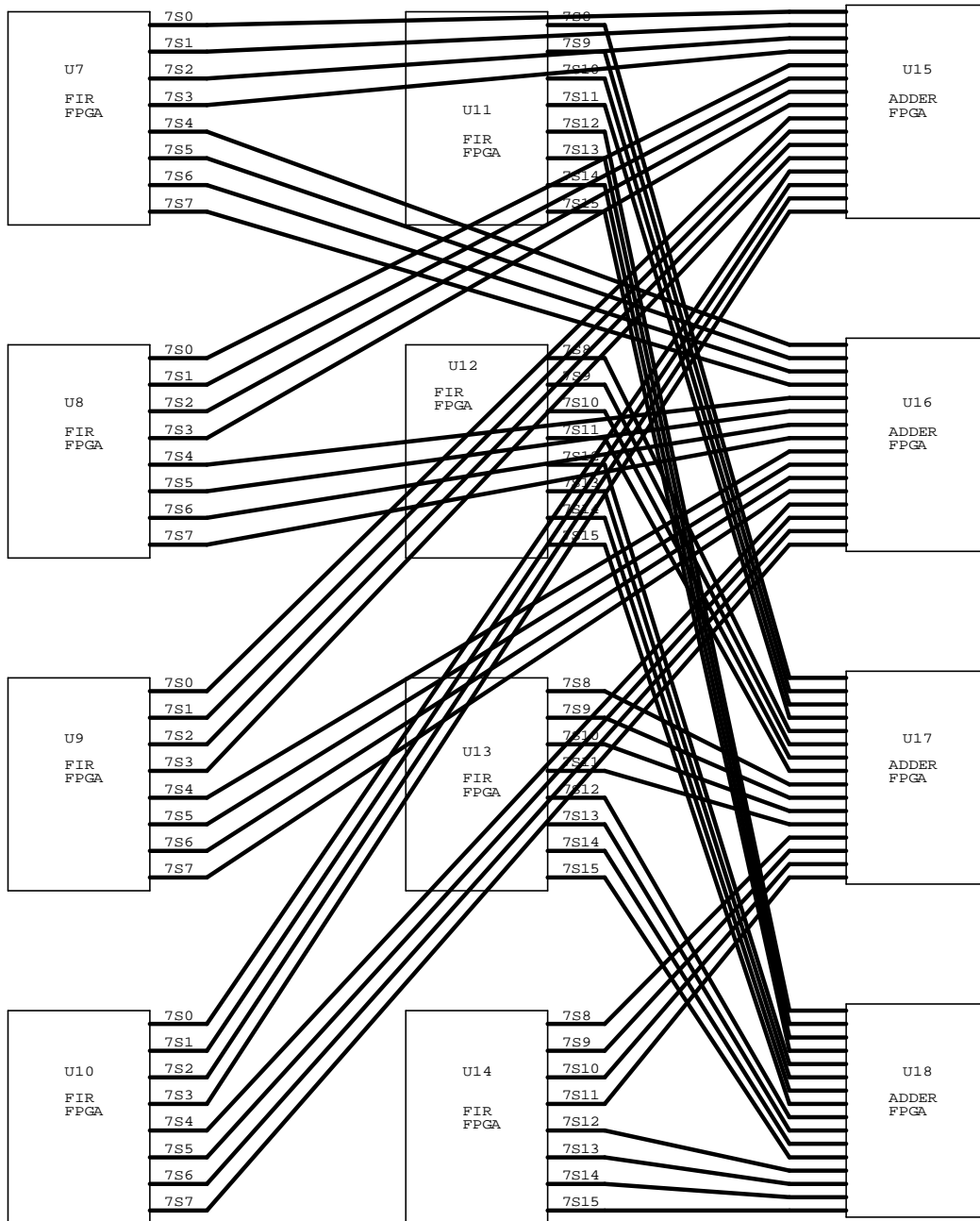


Figure 3, filter card partial sum interconnect

In Figure 3, each FIR chip outputs eight 7-bit partial sum filter results for final summation in the ADDER FPGA chips. In unity tap mode, each ADDER chip would output 8 two-bit samples, while in 1/2 band mode,

each ADDER chip outputs 4 two-bit samples (not shown in Figure 3). In each case the ADDER chip outputs is a Nyquist sampled signal at the bandwidth programmed.

DELAY FPGA

The DELAY FPGA had only 4 basic functions:

- Provide a small programmable delay line
- Provide input state counters
- Provide for self test capability
- Provide fanout of samples to the tap weight multiplier FIR FPGA chips

The Xilinx schematic of the digital filter card DELAY FPGA is seen in drawing number ALMA:07:001:SX:002. Sheet 1 of this schematic shows the FPGA microprocessor port, the delay lock loops, and chip test points. Also shown on sheet 1 in the upper right part of the page are the input and output pins. All input pins connect into the DELAY chip through MUX8 MACRO stages allowing outputs from a 35-bit pseudo random data generator on sheet 3 on the chip schematic to replace the card input samples.

The logic in the lower right part of sheet 1 shows capture and distribution of the 1 msec system memory cycle signal. This signal comes into the card as a square wave and is changed into a 1-bit wide pulse INIT here.

DELAY chip U1 captures the memory cycle signal onto the filter card (into P57) and distributes it to other FPGA chips (from P93, P174, and P126). U2 can also substitute an internally generated 1 msec signal for stand alone operation of the filter card.

Sheet 2 of the DELAY chip schematic shows the card delay line. Delay in the data input to the FIR chips can be set with a range from 0 to 63 bits in 1-bit increments (1-bit relative to the original 4 GS/S digitizer output). To get delays less than 32 bits where data is demultiplexed into 32 parallel streams requires a barrel roll operation as can be seen on sheet 2.

Sheet 3 of the DELAY FPGA schematic shows two 35-bit pseudo random data generators. Most of the logic for a data generator is in the form of the MACRO RAND_35 with its own schematic sheet. Each generator is a 35-bit EXOR feedback generator and generates a data pattern about 4.5 hours long when enabled. The only thing unusual about this circuit is that it contains 32 EXOR circuit instead of 1 and thus seems to shift by 32 bits for every 125 MHz clock instead of one. This strategy allows all 32 data output bits of the generator to be uncorrelated (for example, the second data stream is not a duplicate of the first data stream shifted in time by one bit).

The output of the 35-bit random data generators can be substituted for the normal card input bits at the input stage of the DELAY FPGA on sheet 1. This gives the capacity to work the card in a stand-alone mode in that known data patterns can be made to drive the card starting at the input stage.

At the bottom of sheet 4 of the DELAY chip schematic can be seen another 35-bit pseudo random generator (this time a conventional one with only a single EXOR), the predict data generator. The mux at the top of sheet 4 can select any of the 2-bit by 32 card inputs and test if it conforms to the 35-bit data pattern. When used, the predict data generator is loaded with any 35 consecutive samples from the line selected by the multiplexer to be tested. These 35-bits act as a seed to the data generator. Once the seed is loaded, the data generate is closed and will predict future bits from the selected line. The error counter will count deviations from the predicted data pattern. If the data line selected happens to be a constant zero, a 16-bit counter is used to inject occasional error counts into the error detector. (All zeros is the hang-up state of the 35-bit data generator, when loaded with a seed of all zeros it remains zero when closed and except for this counter would indicate error free operation in a dead input.)

The 35-bit pseudo random data generators seen in the DELAY chip schematic are standard to the ALMA correlator. Every stage in the data chain of the correlator has both a data generator that shifts by 32-bits per clock and data predict/checker logic. These test stages can be used to both test the corrector through each step of the system and to produce simulated data for end-to-end testing. The generator that shifts by 32 bits per clock and the predict generator that shifts by only one bit per clock are compatible in that any one of the 32 outputs of the first duplicates the pattern of the second (the predict generator).

Errors from the error counter and other monitor points in the DELAY chip can be read through the microprocessor port by using the monitor select logic seen in the lower right of sheet 4.

The EXOR gates in the upper right corner of sheet 4 allow the error counter to also count states in the input data. This operation also requires chip-to-chip connections between the two DELAY chips (DELAY chip outputs are from P235 and P80, inputs are from P221 and P72). With the total of 4 DELAY chip-to-chip connections, state counts can be done in either DELAY chip. The 4 program lines DC8, DC9, DC10, and DC11 can be set to any of 16 states possible with 4-bits and the number of times this state is input to the filter card DELAY chips can be counted by the 19-bit counter ERR[3:0],CNT[15:0].

FIR FPGA

The FIR FPGA has only 3 basic functions:

- Provide tap weight multipliers (256 per FIR chip)
- Provide the first 5 stages of an adder tree
- Provide for fanout of samples from the first rank of FIR chips into the second rank

The Xilinx schematic of the digital filter card FIR FPGA is seen in ALMA drawing number ALMA:07:001:SX:003. Sheet 1 of this schematic shows the FPGA microprocessor port, the delay lock loops, and chip test points. Also shown on sheet 1 in the right side of the page are the input and output pins. Eight 4-bit samples from the DELAY FPGA chips drive each of the FIR chips U7...U10 (the first rank of FIR chips). Drive into the second rank of FIR chips (U11...U14) comes from data flowing through the first rank. In most modes, samples flow through chips U7...U10 into U11...U14. When a 1/32 band filter (62.5 MHz bandwidth) is selected, however, samples that drive U11...U14 flow through 1024-stage shift registers in chips U7...U10 (see the block diagram in the lower right of sheet 1).

The 1-msec system memory cycle signal INIT is seen on sheet 1 of the FIR chip schematic driving into P118 and driving out of the chip from P216. The memory cycle signal is not used in the FIR chips but flows through both ranks of FIR chips for eventual use in the ADDER chips.

Sheets 3 and 4 of the FIR FPGA chip show the tap weight multipliers. Each TAP4 MACRO had 4 tap weight multipliers and the first two stages of the filter adder tree (MARO TAP4 has its own schematic sheet). Figure 4, below, gives a block diagram of the TAP4 MACRO. The MACRO had 4 tap weight multipliers in the form of four 16 X 10 look-up table RAMs and two stages of adder tree. A multiplexer on the MACRO input allows for mode selection.

In order to implement a 128-tap $\frac{1}{2}$ band digital filter, each of the 32 demultiplexed digitizer outputs must go down a 4-bit deep (4-bit wide for 4-bit sampling) shift register. The 4 X 32 or 128 four-bit samples stored in such an arrangement would be 128 consecutive samples in time of the ALMA 4 GS/S digitizer analog input. If these 128 consecutive samples are then multiplied by 128 appropriate tap weights and the 128 results summed, the result would be one data stream at 125 MHz (all 32 of the demultiplexed digitizer streams are on the system 125 MHz clock). This single 125 MHz data stream carries (assuming the appropriate 128 tap weights) every 32nd sample of a twice Nyquist sampled 1 GHz bandwidth (1/2 band) signal, or every 16th sample of a Nyquist sampled 1 GHz bandwidth signal.

A complete Nyquist sampled data stream with a 1-GHz bandwidth requires 16 such hardware sets described above, each working on 128 consecutive samples displaced from each other by two 4-GS/S samples.

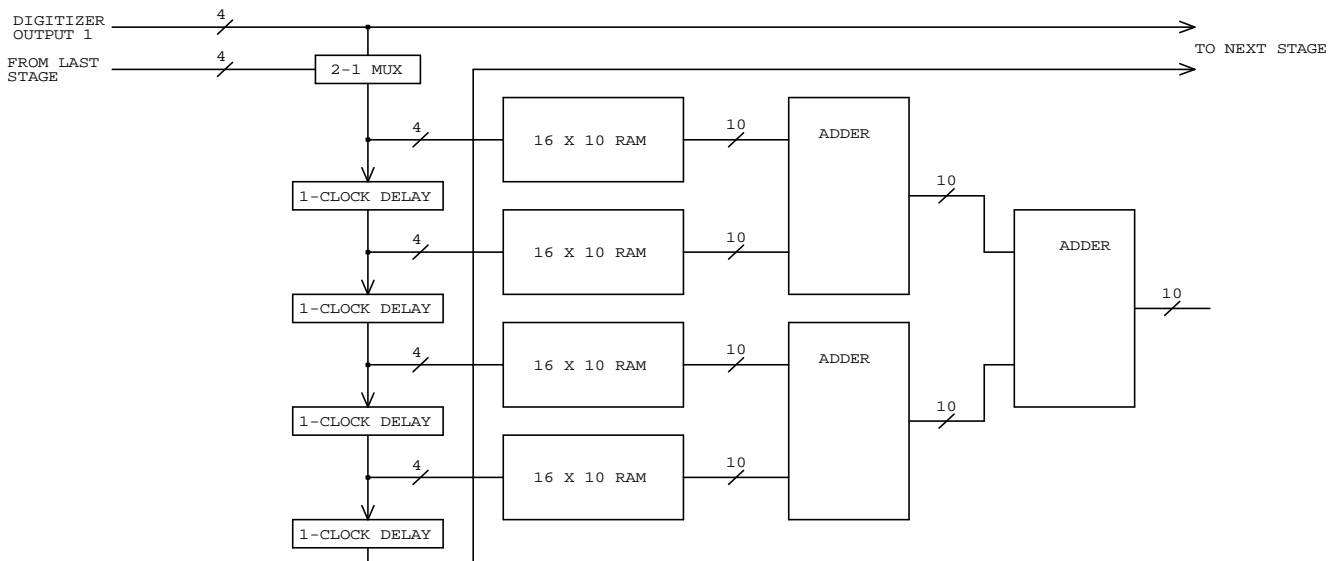


Figure 4, TAP4, the basic building block of the filter card

Referring to sheets 3 and 4 of the FIR FPGA chip schematic, TAP4 MACROS U1, U9, U17, U25 (sheet 3), and U69, U77, U85, U93 (sheet 4) in FIR chips U7, U8, U9, and U10 taken together provide the $4 \times 2 \times 4 \times 4 = 128$ tap weight multipliers as described in the paragraphs above. The outputs of these 128 tap weight multiplications summed together by the two adder stages in the TAP4 MACROS, and by adder stages U128, U139, U103, U138, U139, U103, and U61 (U61 is on sheet 2 of the chip schematic), and by two adder stages in the ADDER FPGA chips, produce 1/16 of a 1/2 band filter. The other 15/16 of the filter hardware are in the other 7 columns of TAP4 MACROS on sheets 3 and 4 of the chip schematic and on the 8 columns of TAP4 MACROS in FIR chips U11, U12, U13, and U14.

Shift registers and multiplexers in the FIR chips assure that each of the sixteen 128-tap weight summations distributed among the 8 FIR FPGA chips act 128 consecutive samples with 2-bit offsets from each other.

Making a 256-tap 1/4 band filter requires connecting TAP4 MACROS in series. With the shift registers of two TAP4 MACROS in series, 8×32 or 256 consecutive samples are produced from the 32 parallel digitizer data streams. When TAP4 MACROS such as U1 and U2 on sheet 3 of the chip schematic are connected in series (using the TD[3:0] to D[3:0] runs) the filter card FIR chips can be configured as eight 256 tap summations to produce the eight 125-MHz data streams required for a Nyquist sampled 500 MHz digital signal. This mode requires one additional adder tree stage in the ADDER chips.

TAP4 MACROS can be connected in series in sets of 1, 2, 4, 8, or 16 to produce sixteen 128-tap stacks, eight 256-tap stacks, four 512-tap stacks, two 1024-tap stacks, or one 2048-tap stack implementations for a 1/2 band, 1/4 band/ 1/8 band, 1/16 band, or 1/32 band digital filter respectively.

Block RAMs I542 and I536 and the logic producing signals TAP[15:0], TW[15:0], P[32:0], and WR0...WR3 on sheet 1 of the FIR FPGA chip schematic are used to load tap weights into the look-up table RAMs of the TAP4 MACROS. The actual tap weight to be written is the 10-bit bus TW[9:0]. The two block RAMs and the TAP[15:0] counter are programmed to go through a set sequence of tap weights, so the only requirement on the microprocessor in loading tap weights is to, load them in the correct sequence.

ADDER FPGA

The filter card ADDER FPGA chip has 5 basic functions:

- Providing the final stages of an adder tree (2 to 5, depending on mode) to implement the filter desired
- Re-sample the adder tree output
- Provide output state counters
- Provide for self test capability
- Provide a snap shot RAM buffer to store short bursts of output samples

The Xilinx schematic of the digital filter card ADDER FPGA is seen in ALMA drawing number ALMA:07:001:SX:004. Sheet 1 of this schematic shows the FPGA microprocessor port, the delay lock loops, and chip test points. Block diagrams on Sheet 1 show the adder tree connections for the various modes possible using the digital filter card. Note that some modes require ADDER chip-to-ADDER chip connections to sum partial sums from different ADDER chips. The actual chip-to-chip adder stages for this purpose are seen at the bottom right of sheet 1.

The 1-msec system memory cycle signal INIT is seen on sheet 1 of the ADDER schematic driving into P89. This signal is a square wave as in comes onto the chip and is turned into a one clock wide signal INIT using the logic seen on sheet 1.

Sheet 3 of the filter card ADDER FPGA chip shows the adder tree. All partial sums come onto the ADDER chip as 7-bit numbers. The first adder stage is in the INPUT-ADD MACRO with its own schematic page. This adder stage adds two of the 7-bit numbers that drive the ADDER chip to produce an 8-bit result. All other adder stages in the ADDER chip are 8-bit in and 8-bit out ADD-8 MACROs that allow no head room growth. Scaling of tap weights in the FIR chips is such that no head room growth is possible in these stages.

Pipeline stages are seen on the bottom of sheet 3 of the ADDER schematic to equalize partial sum streams participating in chip-to-chip adder stages (these compensate for off-chip and on-chip flip-flop capture stages).

Sheet 2 of the filter card ADDER FPGA shows the re-sample stages required to limit the filter card output to 2-bits so as to be compatible with the ALMA 4-level correlator chips. At the right side of sheet 2 can be seen mode selecting multiplexers and chip output stages.

Sheet 4 of the filter card ADDER FPGA schematic shows test logic and the state counters. A 35-bit pseudo random data generator, identical to that described in the filter card DELAY FGPA can generate test data that can be used to substitute for filter data at the chip, and hence, the filter card output. The 35-bit data generator tester is also identical to that in the ADDER chip. Any of the inputs to the ADDER chip may be selected using the chain of MUX8 MACROs to drive the tester.

The counter in the test logic is also utilized as a state counter as in the DELAY chip using the EXOR gates and the programmed DC10 and DC11 signals. The multiplexer logic seen at the bottom left of sheet 2 allows the microprocessor to read either the error/state counts or other chip monitor functions.

Sheet 5 of the filter card ADDER FPGA schematic shows a snap shot RAM. This logic can be used to store a short burst of chip and hence card output samples for readout by the microprocessor.

CPLD MICROPROCESSOR PORT

The digital filter card has a XC95144XL CPLD that provides the function of an 8-bit card microprocessor port. The CPLD interfaces to each FPGA on the card providing in turn each FPGA with an 8-bit microprocessor port. The CPLD design is an ALMA correlator standard design used on many card in the

ALMA correlator. The schematic for the CPLD can be seen in ALMA drawing ALMA:07:xxx:SX:xxxx. The filter card uses the ALMA correlator standard microprocessor interface, which includes:

- An 8-bit single ended bi-directional data bus
- A 4-bit single ended control bus card input
- One differential bus clock card input
- One spare differential card input

In addition to the microprocessor port, the filter card CPLD also has a 4-bit standard JTAG interface for programming the CPLD personality. The JTAG interface includes two single ended inputs (TDI and TMS), a differential clock input (for TCLK) and a single ended output (TDO).

The CPLD to FPGA interface (again, an ALMA correlator standard FPGA microprocessor port) includes:

- An 8-bit single ended bi-directional data bus
- One control/data FPGA input (used as the BUSY signal during FPGA personality loading)
- One port read/write FPGA input (used as the WRITE\ signal during FPGA personality loading)
- Individual FPGA microprocessor strobe (clock) signals (used as chip selects for personality load)
- One PROG\ FPGA input
- Four CCLK FPGA inputs
- One DONE line from FPGAs

The FPGAs also form a JTAG loop so the FPGA personalities can be loaded via JTAG (this is never expected to be used in operation but was provided in case a use arises some day). The FPGA JTAG loop goes through the CPLD and interfaces the card IO connector in signals XTDI, XTMS, XTCK, and XTDO.

In the system, the filter control card microprocessor will use the CPLD port to control or monitor either the CPLD itself or any of the filter card FPGAs. The CPLD port can work in either single target or broadcast modes. Use of the CPLD port can be either for initial FPGA personality loading (one at a time or to broadcast like personalities in any combination) or for operational control and monitoring of the card.

The filter has an input PBSW\ interface via a reset chip to the CPLD for possible use with a push button switch. The CPLD also has a Dallas serial number/temperature chip that can be read by the filter control card.

INTERFACE

The filter card interface includes:

- Four 32-input 125 MHz data busses
- Two 32-output 125 MHz data busses (some may be inactive, depending on mode)
- One 125 MHz sinewave clock input
- One 125 MHz clock rate memory cycle strobe input
- A multi-wire microprocessor port as described above.

All of the filter card 125-MHz data input and output lines and the memory cycle strobe input use the standard ALMA correlator interface seen in Figure 5 on the next page.

CARD TO CARD CONNECTIONS OVER A BACKPLANE

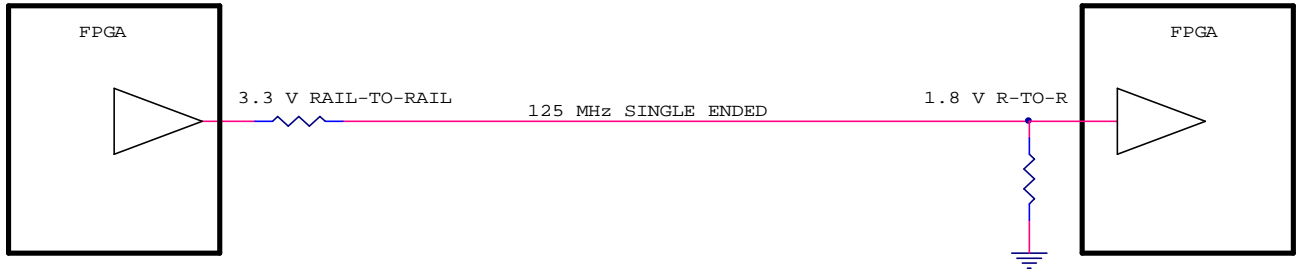


FIGURE 5, CARD-TO-CARD 125 MHz DATA INTERFACES

POWER REQUIREMENTS

The power requirements of the ALMA correlator filter card are:

3.3 VDC at 4.7 amps
1.8 VDC at 20.0 amps

FILTER CARD TESTING REQUIREMENTS

The ALMA correlator filter card may be tested using the station/filter card test fixture. This test fixture can supply simulated signals to drive all of the card inputs and has sufficient resources to test the card outputs for error free operation. Software run on the test fixture microprocessor will run anything from very primitive tests to see if minimal parts of the filter card being tested are functioning correctly to a complete end-to-end test of the entire card.

Test software for 10 main card tests is available. Description of the test selections can be found below. In each case a test can be initiated by typing TESTx or Tx on the test fixture terminal to execute test x (terminal commands are case insensitive, upper case is shown here for clarity). There are a number of application custom HELP screens available to assist the operator. The HC command is a summary custom help screen and gives a quick summary of commands available using the test fixture. Help screens H0, H1, H2, and H3 give more detailed help information.

Before testing a filter card in the test fixture both the fixture test card and the filter card being tested must have FPGA personalities loaded. Commands to perform these tasks are LTST for the filter card and LFIX for the fixture test card. To see if the FPGA delay lock loops are locked use the DLL command. Another requirement for most filter card tests is that filter card tap weights be loaded. The TAP command is a menu driven command to load filter card tap weights.

Many of the tests described below have a number of test options to help in testing and trouble shooting filter cards with logic problems. The help screens explain the test options.

T0 (or TEST0)

Test 0 is a primitive test that tests if the filter card CPLD and the CPLD on the fixture test card can be successfully communicated with. The test card microprocessor will write random bytes to registers in the CPLDs and read them back and test for errors.

T1 (or TEST1)

Test 1 is actually a test of the test fixture test card itself. The test card microprocessor will write random bytes to registers in each of the two test card FPGAs and read them back to test for errors.

T2 (or TEST2)

Test 2 is another primitive test of the filter card. It will tell if the filter card FPGA personalities are loaded correctly and if the microprocessor port to the filter card is working. The test will write random bytes to registers in each filter card FPGAs and read them back to test for errors.

T3 (or TEST3)

Test 3 is a test of block RAMs in the filter card ADDER FPGAs. Each ADDER FPGA on the filter card has 8 Xilinx block RAMs that can be both written into and read via the FPGA microprocessor port. Software for this test will fill the block RAMS and read them back to test for errors.

T4 (or TEST4)

Test 4 is a test of the DELAY to FIR FPGA chip data runs. This test will only run if one of two test personalities are loaded into the FIR FPGAs (the TST-FIR or CLK-FIR personalities).

T5 (or TEST5)

Test 5 is a test of the filter card internal data runs. The data source for the test is the DELAY FPGA 35-bit pseudo random data generators (a test option allows the data source to be moved to the test fixture OUTPUT FPGA). Data integrity is checked in the filter card ADDER chips. This test should be run with either the TST-FIR or CLK-FIR tap weight FPGA personalities loaded (some data runs will be error free when the normal FIR-FIR personality is loaded and unity tap weights are used).

T6 (or TEST6)

Test 6 is a test of the filter card internal data runs and the filter card output runs. The data source for the test is the DELAY FPGA 35-bit pseudo random data generators (options allows the data source to be moved to either the test fixture OUTPUT FPGA or the filter card ADDER chips). Data integrity is checked in the test fixture INPUT FPGA. This test should be run with the TST-FIR tap weight FPGA personalities loaded but will also run if the FIR-FIR personality is loaded and unity tap weights are used).

T7 (or TEST7)

Test 7 will perform integrations on the filter card output data using the test fixture card GBT 1024-lag correlator chip. Integration results are displayed on the terminal. The data source for the lag integrations is the test fixture OUTPUT FPGA.

T8 (or TEST8)

Test 8 will apply static inputs to the filter card from the test fixture OUTPUT FPGA and will read the final ADDER FPGA outputs. These ADDER FPGA outputs are compared to internally computed filter results using the tap weights loaded to test for correct operation. Possible overflows in each stage of the filter adder tree are reported.

T9 (or TEST9)

Test 9 will display the state counts made in the DELAY and ADDER FPGAs on the filter card.