

In j:/orcad/hwdwgs/chip/alma1spec.doc

**ALMA1 Correlator ASIC Specification**

**ALMA07004NX0001**

Thursday, February 01, 2001

Joseph H. Greenberg and Raymond P. Escoffier  
National Radio Astronomy Observatory (NRAO)

2015 Ivy Road, Suite 219  
Charlottesville, VA 22903

804-296-0355

[jgreenbe@nrao.edu](mailto:jgreenbe@nrao.edu)

Revisions:

- 3/28/00 Change XOE and YOE to X0E\ and Y0E\ (low true).
- 4/10/00 In Figure 8, add logic to limit Dump to Storage pulses to every other Blanking pulse. This included adding the bit ALTBLNK to the program word.
- 7/24/00 Delete RDCLKENBLOUT pin.  
Add Ring Oscillator.  
Add HIZ pin to make outputs high impedance, except the Ring Oscillator.  
Replace Tri-stating of 64 lag outputs with a 4 to 1 multiplexer.  
Replace the 16 to 1 multiplexer for the 256 lag outputs with Tri-stating.
- 8/3/00 Add RESETEN to Program Word to allow not FULLACC, Blanking, and no. resets for test.  
Add AUXEN program word control line on Figure 1.
- 8/15/00 Reorder data inputs on DBR, DBL, DTR, DTL, DL, and DR busses on Figure 1.
- 8/29/00 Change Ring Oscillator activation to be from PGM STB, instead of HIZ.
- 9/21/00 Change Program Word names to match those used by Innotech.  
Add Innotech Pin and signal names to Figures, as additional names shown in parentheses.  
Update Pin and Program Word list (alma1spec.xls) to reflect Innotech Names.  
Add Pipeline register after SELP[1:0] on Figure 4.  
Add Pipeline registers to the Vertical and Horizontal inputs to the Correlator Matrix on Figure 1.  
Redefine RC\_TSTE64 function so it does not inhibit the lower bits on Figure 6.  
Redefine logic so Dump to Storage creates a single RDCLK, instead of a double pulse. See Figure 5.  
On Figure 3, change the chip modes on the left and right sides of the card, to output data.
- \* 10/4/00 Correct numerous vestigial references to changed sections, including deleting all reference to chip pull-up resistors.
- \* 10/5/00 Invert sense of Control Word bit RESETEN to be come RESETENB.
- \* 10/6/00 Specify that OUT[15:0] be read out at half the clock speed.
- \* 10/25/00 Specify Program Word Clock at no more than half the main clock frequency.  
Remove Alternate Blanking (from Figure 8) and control word bit ALTBLNK.
- \* 1/23/01 Specify chip voltage as 1.8 VDC.  
Remove RESETOUT.

## **1.0) Introduction**

### **1.1 General Description**

This specification gives the requirements for the 4096-lag correlator ASIC to be used in the ALMA correlator. This chip runs on a 125 MHz clock. The chip operating voltage will be 1.8 VDC. The chip package will be a surface mount, industry standard package.

Each individual lag of the correlator chip consists of a 2-bit, 4-level, times 2-bit, 4-level, multiplier whose output is integrated in a 25-bit accumulator. Each accumulator has secondary storage for the 16 most significant bits. All 25-bits of the accumulator can be asynchronously cleared before an integration. A single synchronous Blanking signal input to the chip can be used to stop the accumulation in all correlator circuits.

The ALMA correlator chip has many modes of operation. A serial program word on the chip provides the configuration information for the chip. The serial program word register on the chip must be loaded by a microprocessor external to the chip before operation can begin. The program word register has a secondary storage register to prevent the chip from seeing the serial program word shift-through.

### **1.2 Doubling the Size from a 4096 Lag to a 8192 Lag Correlator Chip**

A larger version of the chip is also being considered. The primary difference is that each 64 lag sub-block, would be replaced with a 128 lag sub-block. Almost all other aspects of the chip will remain the same. This would approximately double the number of gates in the chip.

Due to cooling considerations, it is desirable to maintain the maximum chip power at 2 watts, despite the doubling of the number of gates. Thus, for the 8192 Lag Chip, 0.18 micron technology would probably be required.

### **1.3 Chip Quantities**

64 chips/correlator card \* 4 cards/plane \* 32 planes \* 4 Baseband pairs = 32,768 chips.  
If 20% spares are included, the total chip order becomes 1.2 \* 32,768 equals 39,322 chips.

### **1.4 Work Breakdown**

NRAO provides this detailed chip specification. Innotech will do the detailed design. NRAO will provide a close overview of the process, to assure the desired results. Innotech will do the simulation and test vector generation, with NRAO providing aid as required. Innotech will do the layout, and arrange fabrication. NRAO will evaluate the prototypes.

### **1.5 Chip Interface Timing**

The chip will interface with Xilinx Virtex E FPGAs.

For data from the ASIC to the Xilinx, the setup time  $T_{IOPICKP}$ , with respect to clock at an input register, from the pad, is 1.5 ns for the -6 part, with no delay programmed in the Xilinx input path.

The hold time  $T_{IOICK}$  is 0 or a negative hold time. If a delay is programmed, the setup is 2.9 to 3.6 ns, with zero or negative hold time.

For outputs from a Xilinx the CLK to Pad delay  $T_{IOCKP}$  is 2.9 ns, for the -6 part, LVTTTL with 12mA drive, and a fast slew rate.

## 2.0) Block diagram

Figures 1 and 2 give a block diagram of the ALMA correlator chip. This chip is a 4-by-4 matrix of correlator blocks where each block consists of 256 multiplier/accumulator/secondary storage circuits (referred to as lags). Each of the correlator blocks can be configured as a single 256-lag correlator, as two 128-lag correlators, or as four 64-lag correlators. Each 256-lag block is thus comprised of four 64-lag sub-blocks. There are a total of 64 of the 64-lag sub-blocks on the chip.

Figure 2 shows the 4-by-4 array of correlators being driven by samples from 4 antennas on each axis of the matrix (shown in the Figure as antennas Y, Y+1, Y+2, and Y+3 on the Vertical axis of the matrix, and antennas X, X+1, X+2, and X+3 on the Horizontal axis). Each antenna supplies 2-bit samples from each of two digitizers (M0 and M1 as seen in Figures 1 and 2).

The correlator chip thus requires eight, 2-bit driving signals (for a total of sixteen bits) for the Vertical axis of the internal 4-by-4 matrix, and eight, 2-bit driving signals on the Horizontal axis.

The correlator chip has two, separate sixteen bit busses entering the bottom of the chip, called the Main Bus and the Aux Bus (See Figure 1). These busses exit the top of the chip.

Figure 1 also shows eight, two bit, busses on the horizontal axis. There are bidirectional I/O blocks on the left and right side of these busses.

The Horizontal axis to the internal matrix, as seen in Figure 1, can be driven from one of the following four sources:

- Drive from the Main Bus input pins at the bottom of Figure 1.
- Drive from the Aux Bus inputs pins at the bottom of Figure 1.
- Drive from the Left I/O pins used as chip inputs, on the left side of Figure 1.
- Drive from the Right I/O pins used as chip inputs, on the right side of Figure 1.

The input busses are selected with tri-state drivers. The drivers are controlled by the CENTERBUS[1:0] bits of the program word (See Figure 3). When the Horizontal axis of the correlator matrix is driven from either the Main Bus or the Aux Bus, these signals can also drive both the right and left I/O pins out of the chip using the tri-state drivers seen in Figure 1. These modes are further explained later in this document.

Readout of the 16-bit accumulator secondary storage registers is over a 16-bit tri-state bus. Any of the 64, 64-lag sub-blocks may be selected for results readout with the application of a 6-bit code plus two permissive, output enable signals  $XOE\backslash$  and  $YOE\backslash$ . The 64-lag results shift out in lag order, lag zero first, lag one next, lag two next, etc. Any number from 1 to all 64 lags may be read from a sub-block at a time. Each enabling cycle of RDCLKENBL, outputs one lag. The maximum allowable output data rate is half the clock frequency. See the explanation of Figure 6, in Section 2.2.

## 2.1) Data and Clock Input-Output

Figure 3 shows data (samples) into and out of the correlator chip. The chip is to be connected in a matrix configuration on the correlator card in the ALMA application. The chip supports the Horizontal and Vertical matrix structure of the card.

Figure 3 is described further in the Card Level discussion, in Section 5.

The 125 MHz clock enters the chip on the C125 input pin. C125 is gated with the Program Word Bit CKPINEN to obtain C125OUT, as shown in Figure 1. The gating conserves power if the signal is not used. C125OUT provides a buffered clock output pin, for use on the correlator card.

The HIZ Pin shown on Figure 1, puts all outputs to high impedance, for leakage tests.

## 2.2) The Basic Correlator

Figure 6 shows one of the 4096 (64x64) correlator lag circuits on the correlator chip. A correlator lag includes a 2-bit, 4-level, by 2-bit, 4-level, multiplier, using the Biased Multiplication Table shown. The multiplier output is summed in a 25-bit accumulator, at the 125 MHz chip clock rate (that is, one multiplication and summation into the accumulator occurs on each positive clock transition). The 25-bit accumulator consists of a 5-bit synchronous stage, a 4-bit ripple-through pre-scalar, and a 16-bit ripple through counter.

The 5 bit synchronous stage consists of the Multiplier, 5 Bit Adder, and 5 Bit Accumulator. Since the maximum output from the Multiplier is 9, that makes the maximum toggle rate of the MSB of the 5 bit stage to be approximately every 4<sup>th</sup> clock cycle. Shown below the 5 bits are incrementing by 9.

00000, 01001, 10010, 11011, 00100, 01101, 10110, 11111, 01000

The MSB transitioning from 1 to 0 provides the toggle edge for the subsequent ripple counter. Each T flip-flop asynchronously toggles on the 1 to 0 transition of the previous stage. Since there is an interval between the BLANKING, and the DUMP TO STORAGE signal, the ripples will have time to settle (See the timing diagram on Figure 6).

The Accumulator can function as a 25 bit or 21 Bit Accumulator, as determined by the FULLACC line at each block. FULLACC is logically defined on Figure 4, since it effects the whole 256 lag Block. The FULLACC line is a function of only the FULLACC<sub>x</sub> control. In FULLACC<sub>x</sub>, x equals 0, 1, 2, or 3 to indicate the row number. Figure 2 illustrates the Row definition.

If FULLACC=0, the 4-bit pre-scalar is bypassed and the accumulator acts as a 21-bit accumulator. In this mode, the pre-scalar is held static to reduce the power consumption of the chip. If FULLACC=1, the pre-scalar is used, and the correlator has the full 25-bit accumulator for integration.

21 bit accumulate mode implies one ms dump times. Only the diagonal blocks are of interest in this mode, so more rapid data retrieval is possible.

The RC\_TSTE64 program word bit is provided to allow for faster functional test. Otherwise it would take too large a number of test vectors to increment the 16 bit counter through all states. RC\_TSTE64=1 has the upper eight bits and lower eight bits increment in parallel (See Figure 6).

Figure 6 shows the 16-bit secondary storage register for the 16 most significant bits of the accumulator. The contents of the 16 bit secondary storage registers are shifted out of the chip on a

16 bit bus, when clock enabled. During read-out, the 16-bit secondary storage registers shift broadside off the chip. All 64 lags from a lag sub-block connect as a 16-bit wide, 64-bit deep, broadside shift register (lag 0 first).

In typical operation, the BLANKING signal to the chip will be high (inhibiting correlation/integration) for 64, 128, 256, or 512 clock cycles every blanking period (typically one millisecond) (See Figure 6 and Figure 8). BLANKING controls a gated clock (See Figure 5).

Occasional Blanking cycles will be bracketed by an externally generated DUMP ENABLE signal. DUMP ENABLE is an input pin to the chip, typically occurring every 16 ms. (Logic shown on Figure 4).

The signals SEQ ACCUMULATOR RESET and SEQ DUMP TO STORAGE go from Figure 8, to Figure 4. There they are gated with the DUMP ENABLE signal, and the FULLACC<sub>x</sub> signal to generate two signals, the DUMP TO STORAGE signal to the secondary storage register, and the ACCUMULATOR RESET. These will be generated from the rising edge of the marked BLANKING signal (See Figure 8 and Section 2.4). These signals implement dumping the sixteen, upper accumulator bits to secondary storage, and resetting the accumulator.

The overall effect is that in FULLACC (25 bit) mode, there is a Dump to Storage and Accumulator Reset every 16 ms. If it is not FULLACC (21 bit mode), then there is a Dump to Storage and Accumulator Reset every 1.0 ms. There is a BLANKING every 1.0 ms in both cases.

### **2.3) 64-Lag Correlator Sub-Block**

Figure 5 shows how 64 of the basic correlator circuits are connected together to build a 64-lag correlator sub-block. All 64, 2-bit P (prompt) inputs to the 64 correlators are connected in parallel. The 64 2-bit D (delayed) inputs are connected to the correlators from a 64 stage (1 bit or 2 bit) lag generator. The 2 bit output of the sub-block lag generator connects to an adjacent 64-lag correlator sub-block input multiplexer (See Figure 2).

The lag generating shift register has two modes of operation (See Figure 5). In the normal mode, there is one stage of shift register between each correlator of the 64-lag correlator sub-block (for each bit of the 2-bit D signals). When the input signal is twice Nyquist sampled, the lag generator supplies two stages of shift register per signal bit between each correlator. The OVERSAMP<sub>0</sub> through OVERSAMP<sub>3</sub> bits (four bits for the four rows of the chip) from the serial program register generate the OVERSAMP signal for each row. OVERSAMP determines the one-or-two bit operation of lag generators for the respective row of correlators.

There is a digital delay line at the input end of the Delayed (D-DATA) Input. The delay line can be set for 1, 2, or 3 clock cycle(s) delay, as programmed by the M<sub>0</sub> and M<sub>1</sub> bits. The table at the top of Figure 5, defines these bits.

- Note that a M<sub>0</sub> M<sub>1</sub> of 1 0 could be used, instead of the shown 0 1. This would still give the required 2 delays.
- LEAD-BLK is derived from the program word bits LEADLL and LEADUR as defined in the lower right of Figure 4.
- CONCAT is equal to the Program Word bit that controls the 2 to 1 Mux (the Mux which gives delay data coming from another 64 lag block) as shown on Figure 4.

- OVERSAMP is derived from the Program Word Bit OVERSAMPx for the specific Row x.

The LEAD-BLK signals can add an extra one (or two bits delay if Oversampled), if the 256 lag block is a Lead (As shown in the truth table at the top of Figure 5). This is discussed further in the next section. A pipeline register is required, since the parallel, PROMPT line has one. This makes for a 2 or 3 bit delay for a LEAD section.

The purpose of the LEAD-BLK signal is as follows. For a cross product of two antennas, the leads and lags are concatenated, and an FFT performed. Without the LEAD-BLK signal present, the leads and lags share a common signal. The common signal is the product of the two antennas with no delays. Removing the common signal leaves an odd number of components, which are difficult to do an FFT on. The extra delay inserted by LEAD-BLK solves the problem.

The CONCAT signal indicates the D-DATA INPUT is concatenated from a previous 64 lag block. An extra 1 bit (or 2 bit if Oversampled) delay is required. This takes the value that went into Lag 63 of the previous block, delays it and continues the delay chain. Note, as seen on Figure 4, CONCAT equals the same Program Word Bit that controls the last element in the respective 5-1 MUX. There is no pipeline register, since the prompt data would be in parallel with that from the previous, 64 lag block. If CONCAT is 1, LEAD-BLK is don't care (X), since the LEAD bits are only inserted at the beginning of the Delayed chain, not in the middle.

Referring to Figure 2, note that for Block 0, the CONCAT data comes from off the chip via D0-X[1:0]. D0-X[1:0] has registered inputs. The extra delays would have to be accounted for by delaying the prompt input into the chip. This feature is not anticipated to be used in the ALMA project. Similarly Block 15 has its delay chain leave the chip via D4-X[1:0], which uses registered outputs. These delays would also have to be allowed for.

If LEAD-BLK=0 and CONCAT=0, then only the pipeline register is required. This matches the delay due to the pipeline register in the Prompt data path. OVERSAMP is don't care, since the pipeline register does not count as a delay element.

The BLANKING is not applied to the D-DATA chain of shift registers. This allows new data to shift into the registers for the next 1 ms. period. The BLANKING pulse width correlates with how many registers require data shifted into them. For example the width would be at least 512 clocks for four 64 lag blocks, in series, in oversample mode. The BLANKING pulse width can be greater than required. That would entail the unnecessary loss of a small fraction of the data.

## **2.4) 256-Lag Correlator Block**

Figure 4 shows how four, 64-lag sub-blocks interconnect to make a 256-lag correlator block. Each 256-lag correlator block is driven by four 2-bit digitizer signals. The exact mode of operation is controlled by 8 multiplexer stages that are programmed by bits from the program word register. Each row of 256-lag correlator blocks in the 4-by-4 chip matrix (See Figure 2) receives the same 19 program word bits, with the exception that each 5-1 MUX in an upper 64 lag block, receives an independent bit(WRAP-BLK[15:0]). The three lower 64 lag block, 5-1 MUXs are controlled by 3 bits(Rx-W[2:0]), duplicated across the Row. Figure 4 shows the detailed implementation of the 5-1 MUX. Figure 2 shows the multiplexers for the whole chip. A total of  $4*(16 + 3 + 4) = 92$  program word bits are used to control all input multiplexers on the chip.

The X-M0, X-M1, Y-M0 and Y-M1 inputs connect to every one of the input multiplexers, allowing any pair of inputs to drive either the prompt or delayed input of any 64 lag correlator block. The

source of inputs to the prompt and delayed inputs of the 64 lag correlator block, could be swapped for test purposes.

The LEAD-BLK signal into the 64 lag blocks can provide an extra stage of delay, for the LEAD portion of the array. For each 256 lag block, LEAD-BLK is derived from the table in the lower right of Figure 4, based on the Program Word Bits, LEADLL and LEADUR. The equations for all four, 64 lag blocks share the element LEAD-BLK.

The CONCAT signal into each 64 lag block specifies if there is a wrap around input from a previous stage. For 64-LAG CORRELATOR 0, CONCAT = WRAP-BLK0 indicating the delay data comes from the previous 256 lag block (See Figure 2). For 64-LAG CORRELATORS 1, 2, and 3, CONCAT=R0-W1, 2, or 3 indicates the mux setting for the delay data coming from the previous 64-LAG CORRELATOR (See Figure 4). CONCAT is used in the 64 LAG CORRELATOR to adjust the delay, as previously explained.

In Figure 8, the signals SEQ DUMP TO STORAGE and SEQ ACCUMULATOR RESET are generated. These provide for transferring the accumulator contents to secondary storage, then resetting the accumulator. In Figure 4, these signals are used in each 256 lag block.

If FULLACC = 0, the 21 bit accumulator is used. This forces a DUMP TO STORAGE and ACCUMULATOR RESET every BLANKING cycle, otherwise the accumulator would overflow. The multiplication table is biased, since they are all positive numbers (See Figure 6). Thus even totally uncorrelated inputs could cause an overflow, if the integration period is too long.

If FULLACC = 1, the 25 bit accumulator is used. Then the DUMP ENABLE signal (See Figure 4) controls for which Blanking cycles, the DUMP TO STORAGE and ACCUMULATOR RESET are asserted.

A 16-bit wide, 4 to 1 multiplexer is used to allow selection of any of the four, 64-lag correlator sub-blocks for read-out (See Figure 4). The two bits to control which Sub-Block is selected, come from chip input pins, SEL[1:0]. BLOCKSELECTED enables a tri-state bus from the selected, 256 lag block. The sixteen, 256 lag blocks are tri-stated together(See Figure 7).

The input pin RDCLKENBL is used to clock enable the shifting out of the accumulator data. Figure 5 shows how RCLKENBL gates a clock to shift out the data.

DUMP TO STORAGE must also generate a RDCLK, to clock in the selected correlated data. See Figure 5. The DUMP TO STORAGE' signal is generated, as DUMP TO STORAGE delayed one clocks. DUMP TO STORAGE' shifts the mux in front of the shift out registers to select the clocked data.

Do not attempt a DUMP TO STORAGE while reading the outputs. DUMP TO STORAGE is triggered by blanking. Note the blanking is delayed by the internal variable delay line, while RDCLKENBL is not delayed. Thus the timing of RDCLKENBL relative to blanking, must be allowed for externally.

SEL[5:0] selects which of the 64 lag sub-blocks to route to the 16 bit output bus. SEL[0..5] also determines which of the 64 Results Shift Out Registers will be clock enabled. This serially shifts out the 64 lags in the block, Lag 0 first, over the 16 bit parallel bus. Note there are two pipeline registers in the SEL[5:0] path. One pipeline register is at the input to the card. The other pipeline register is on Figure 5, preceding the gating of RDCLKENBL' with the 125 MHz clock. A pipeline register is provided on Figure 4 for SELP[1:0], preceding the 4-1 Mux, to keep the data

lined up. Note BLOCKSELECTED (based on SELP[5:2]) does not have an equivalent pipeline register, since that could lead to bus contention problems. Thus an extra clock delay must follow the switching of the selected, 256 lag block.

## **2.5) Chip Results Output Bus**

Figure 7 shows the final 16-bit wide, 16-to-1 tri-stating required for results read-out. Sixteen bit broadside outputs, from each of the sixteen, 256 lag correlator blocks, come together here. The selected bus is enabled by the SEL[5:2] designating that 256 lag block (See Figure 4). The four, correlator block select input pins select one of the sixteen baseline correlator blocks for readout. A baseline correlator block is a 256 lag block, obtained when an antenna  $x$  is multiplied by an antenna  $y$ .

There is a pipeline register in the output pads. SETOUT from the program word, allows selection of alternate outputs from OUTPIN<15:0> and NXTPIN<1:0> for test purposes.

The XOE\ and YOE\ input pins output enable a single correlator chip onto a correlator card sixteen bit, results output bus. HIZ give high impedance of all outputs for test purposes. XOE\ and YOE\ are low true. Multiple correlator chips can have their output busses tri-stated together.

## **2.6) Control Logic**

Figure 8 shows control logic required for the correlator chip.

### **2.6.1) Program Word**

A serial-in program register can be programmed from an external source to set the mode of operation for the chip. The interface to this program register is via the PGM DATA and PGM CLK input pins. The PGM CLK frequency shall be no more than half the frequency (62.5 MHz), of the main chip clock pin (125 MHz). Bit 0 is shifted in first. Data clocks on the clock rising edge. The serial clock is buffered and output from the PGM CLK OUT pin. Data from the last stage of the shift register is output from the PGM DATA OUT pin, so correlator chips on the correlator card may be connected in series. Care must be taken in the chip design, so the setup and hold times into subsequent chips are satisfied.

The program register has a secondary register so all program bits to the chip can be made to change at once. The secondary storage register is loaded via the rising edge of the PGM STB input pin. This strobe input is buffered and drives the PGM STB OUT output pin.

### **2.6.2) Storing the Results for Readout**

The BLANKING and the DUMP ENABLE signals into the chip control the correlator accumulator secondary storage registers. A logic high on BLANKING will synchronously stop all correlators on the chip. BLANKING is implemented via gating the clock (See Figure 5).

Dump of the correlator accumulator results, into the correlator secondary storage, will occur after the assertion of BLANKING, if DUMP ENABLE is high or if FULLACC $x$  is low (See Figure 4). SEQ DUMP TO STORAGE (See Figure 12 for timing) and SEQ ACCUMULATOR RESET are implemented via delays in the Control Logic of Figure 8. Note RESETENB, from the program word, being zero enables these signals. These signals are gated on Figure 4, with FULLACC $x$  and

DUMP ENABLE. ACCUMULATOR RESET can be seen on Figure 6, to reset the 25 bit accumulator.

### **3.0) Input-Output Pins**

See alma1spec.xls Sheet 1 for a detailed listing of the input-output pins.

### **4.0) Serial Program Word**

See alma1spec.xls Sheet 2 for a detailed listing of the program word elements.

### **5.0) Card Level Considerations**

#### **5.1) Reason for Card Considerations**

Understanding how the chip is used in the Correlator Card will allow the chip design to take into account card level effects. It could also allow the recognition of errors, or improvements in the specification of the chip.

#### **5.2) Signal Distribution to Set of Four Correlator Cards**

See Figure 9. Each Correlator Card services the correlation of 32 by 32 antennas. For a plane of 64 antennas, four cards are needed as shown.

The dashed diagonal line on Figure 9, represents the 64 self products, or autocorrelations. The Self Cards, Cards 0 and 3, are along the diagonal. Card 0 multiplies antennas 0-31 times antennas 0-31. Card 3 multiplies antennas 32-63 times 32-63.

Cards 1 and 2 are called Cross Cards. Card 1 multiplies 32-63 times 0-31, for the LEADS. Card 2 multiplies 0-31 times 32-63 for the LAGS. The Self Cards contain LAGS above the diagonal and LEADS below the diagonal.

The upper right of Figure 9 shows the signal distribution to the cards. Note the Cross Cards (Card 1 and Card 2) have twice as many input signals as the Self Cards. The Cross Cards and Self Cards will be physically identical and interchangeable. Card functionality will be determined by its position in the rack, and chip program words. The Self Cards only have signals coming in on the Main Bus, with the Aux Bus unused. The pulldown termination resistors on the card pull low the Chip Aux bus inputs, when they are not connected. The Cross cards use both busses. Different chip modes are necessary to distribute the signals from the Main Bus in the Self Cards, and to use the Aux Bus in the Cross Cards.

#### **5.3) Chip Modes**

Figure 3 portrays the chip data bus structure. Each bus represents sixteen identical data paths. Sixteen is obtained by multiplying four antennas, times two digitizers, times two bits. The chip Program Word Bits determine the flow of data between the six busses. Figure 3 identifies the six, sixteen bit busses as DBL[15:0], DBR[15:0], DTL[15:0], DTR[15:0], DL[15:0], and DR[15:0]. These data paths connect into (or out of) the correlator chip, and into the Horizontal and Vertical axes of the internal array of correlators. All sixteen of the data paths in a bus are programmed identically in all cases.

Horizontal data flow between chips is controlled by the LTOR (Left to Right) bit of the Program Word. LTOR controls the Right Output Bus via OE-RO. LTOR is output via the LTOR-OUT pin. This is input into the next chip to the right via the LTOR-IN Pin. LTOR-IN controls the Left Output Bus via OE-LO. This arrangement prevents bus contention, regardless of the programming of the LTOR bits in the chips.

The CENTER BUS in the chip can be driven from one of four sources. In the Mode Logic, the two bit CENTERBUS code from the Program Word is decoded to specify which of the four sources drives the CENTER BUS. This arrangement precludes bus contention. The CENTERBUS decoder outputs should be break before make to prevent a momentary bus contention on transitions. That is there will be a period when all four outputs are zero, whenever the inputs change states.

The remaining Program Word bit, which determines the chip mode, is AUXEN. This enables data passage out the top of the Aux Bus. Zeroing this data when not used saves power. On the Self Card, AUXEN would always be zero, since the Aux Bus is not used.

Thus the input signal LTOR-IN, and the program word bits LTOR, CTRBUS[1:0], and AUXEN determine the data flow in the chip. The state of these bits for useful modes is tabulated in the lower left of Figure 3.

Above the table are data flow diagrams for a partial, horizontal row of chips in the Self Card and Cross Card.

## **5.4) Self Card Description**

### **5.4.1) Data Distribution**

See Figure 10. The signals from 32 antennas enter the card. Sets of four antennas enter into each of eight correlator chips. The signals are passed chip to chip vertically up the card. Along the diagonal, the signals spread out horizontally as well. To the left of the diagonal, chips pass data horizontally to the left. To the right of the diagonal, chips pass data horizontally to the right.

The Vertical axis signals into the internal correlator matrix must be applied through adjustable delay lines (See Figure 1). A delay of from zero to 31 bits is available to compensate for the position of the correlator chip in the correlator card chip matrix. Delay is set by SELDLY[4:0] from the Program Word. The current card design would use delays of 0,2,4,6,8,10,12, and 14 bits. The higher delay values would allow for a card containing 256 Correlator Chips.

### **5.4.2) Self Card Chip Modes**

The three different chip modes required are shown on the right side of Figure 10. The chip modes are also shown on the left in Figure 3. They are labeled T1, T2, and T4. There are more possible modes, but they are not useful.

Program word bit LTOR determines the horizontal data flow. LTOR leaves the chip as LTOR-OUT, to become LTOR-IN of the next chip to the right. This arrangement guarantees there will not be bus contention between chip outputs. Tying chip pin LTOR-IN low on the far left turns on the left output bus, driving test points. Leaving the unconnected pins as inputs is not advised, since it could cause high current conditions. Setting program word bit LTOR = 1 on the far right chip, similarly turns on the output to the right.

CENTERBUS (a two bit field from the program word) selects the tri-state drivers to drive the Center Bus (See Figure 3). Also this determines the Horizontal input into the correlator array in the chip.

## **5.5) Cross Card Description**

### **5.5.1) Data Distribution**

See Figure 11. One set of 32 antennas enters the card into the chip Main Busses. Each data bus shown contains the signals from four antennas. The signals are passed chip to chip vertically up the card. These signals provide the drive to the Vertical (Y) axis of the internal correlator matrix.

A second set of 32 antennas enters the chip Aux Busses. These pins were unused on the Self Card. The signals propagate vertically using the Aux Bus, until they reach the diagonal. The signals are not processed in chips that they travel vertically through. Upon reaching the diagonal, the signals spread out horizontally. To the left of the diagonal, chips pass data horizontally to the left. To the right of the diagonal, chips pass data horizontally to the right. These signals provide the Horizontal (X) axis drive to the internal correlator matrix.

The Vertical (Y) axis signals into the actual correlator matrix must be applied through adjustable delay lines. A delay of from zero to 31 bits is provided to compensate for the position of the correlator chip in the correlator card chip matrix.

### **5.5.2) Cross Card Chip Modes**

The three different chip modes required are shown on the right side of Figure 11. The chip modes are also shown on the left in Figure 3.

As in the Self Card Modes, LTOR determines the horizontal data flow, and CENTERBUS determines the drive source for the tri-state drivers to drive the Center Bus (See Figure 3). The difference from the Self Card modes is that now the Aux Bus is used for the Horizontal input. Also, Program Word bit AUXEN is set to propagate the data upward. After the diagonal, AUXEN is not set, to save power. Note that even though the Aux Bus data is not needed past the diagonal, it will still be wired on the card, for the sake of versatility.

## **5.6) Test Modes**

The left most chips will output their left bus to test points.

The right most chips will output their right bus to test points. Setting LTOR =1 in these chips program word will enable the outputs.

The top chips will have their Main Bus and Aux Bus outputs to test points. The Main Bus outputs will be enabled all the time. The Aux Bus outputs can be enabled by setting AUXEN in the Program Word. Note that AUXEN would have to be set for the chips between the diagonal and the top to give full data propagation.

## **5.7) HIZ Test**

The HIZ pin puts all outputs to a high impedance state. This is for test purposes.

## 5.8) Ring Oscillator

The Program Word Strobe Pin (PGM STB) being high, also turns on the Ring Oscillator, which has its own output pin RINGOSC. The Ring oscillator gives a measurement of the process speed.

## 5.9) 32 K Lag Card Configuration

This configuration breaks the card into four 32K Lag correlators, plus four 28 K correlators. The card will have each row's, two bit delay chain proceed across the row from right to left. The programmable delays in the card could be configured:

28 26 24 22 20 18 16	28K Lags. Prompt fed by Ant 5-7 Aux Bus M1
26 24 22 20 18 16 14	28K Lags. Prompt fed by Ant 5-7 Aux Bus M0
24 22 20 18 16 14 12	28K Lags. Prompt fed by Ant 5-7 Main Bus M1
22 20 18 16 14 12 10	28K Lags. Prompt fed by Ant 5-7 Main Bus M0
20 18 16 14 12 10 08 06	32K Lags. Prompt fed by Ant 0-4 Aux Bus M1
18 16 14 12 10 08 06 04	32K Lags. Prompt fed by Ant 0-4 Aux Bus M0
16 14 12 10 08 06 04 02	32K Lags. Prompt fed by Ant 0-4 Main Bus M1
14 12 10 08 06 04 02 00	32K Lags. Prompt fed by Ant 0-4 Main Bus M0

Note there is an extra two bits pipeline delay getting the delay from one chip to the next. The prompt signals will be sent Right to Left also, having an identical delay. Note the programmable delay line is only used to time the Blanking pulse, since the data all avoids the delay lines, using the Center Bus in the chip.

## 6. Production Test

### 6.1) Purpose

This section provides an outline of a possible methodology to be used to sequence the chip through its test vectors.

### 6.9) Test Sequences

#### MAIN ACCUMULATOR CHECK

- Initialize the Random Number Generators
- Have a random number generator generate independent, data inputs X M0, X M1, Y M0, Y M1, and D0-X[1:0] inputs (all separate inputs).
- In Program word have the mux addresses Wrap-BLK[15:0], and R?-W[2:0] and R?-M[15:0], OVERSAMPX, and LEADLL and LEADUR, SELDLY[4:0] all set to 0, FULLACCx to 1s. RC\_TSTE64=1, RESETENB=0.
- Have Dump Enable pin to 1.
- Load and Strobe this initial Control Word.
- Then Have Blanking high for two or more clocks to allow the delay chains to fill with data. This also resets the accumulators. Note there is at least 17 clocks delay till the Reset Accumulator takes effect .
- Set RESETENB=1, in the next control word (shifted in below) so subsequent blanking won't cause resets or dumps.

- MUXLOOP For(delay\_value=0 to 31, by 1) /\* for the 32 delay values \*/
  - Begin Muxloop
    - DATALOOP
    - Begin DATALOOP
      - For (clock=0 to 128, by 1) (125 MHz Clock)
        - Begin
        - Clock in the next random number into the data inputs.
        - Throughout the test, be checking D4-X[1:0] for appropriate outputs.
        - BLANKPIN= 1 if (0<clock<32), else =0; /\* tests blanking delay \*/
        - Simultaneously, be shifting in a new control word
        - End
      - End Dataloop
      - Strobe in New Control Word
    - End Muxloop

In the progression of control words have the following subloops, where a new control word being strobe in increments the loop.

- The below takes 20 control words to complete, then starts over
- For (4-1 Mux positions=0 to 3 by 1) Move all 4-1 Muxes together
  - For(5-1 Mux positions=0 to 4 by 1) Move All 5-1 Muxes together
  - By together, I mean every 4-1 Mux in the chip is in simultaneously in position 0, next iteration every 4-1 Mux would be in position 1, etc.
- Each control word, increment LEADLL, LEADUR as a 2 bit number, modulo 4.
- For delay\_value > 29, have RC\_TSTE64=0.
- Each control word, set the next OVERSAMPx (OVERSAMP(3,2,1,0)=0000, then 0001, 0011, then 0111, then 1111, then start the cycle over with 0000).
- Each control word, set the next FULLACCx (FULLACC(3,2,1,0)=0000, then 0001, 0011, then 0111, then 1111, then LEAVE 0000 FOR THE DURATION OF THE SIMULATION since this gives better resolution).
- delay SELDLY[4:0]= delay\_value
- To test the DUMPENB and FULLACC inhibiting resets and dumps:  
 For the initial reset pulse, after i\_iter6, have FULLACCx=0000, have DUMPENPIN=0, and RESETENB=1. The FULLACCx signal should enable the reset pulses.  
 Somewhere in the middle of the sequence of tests, when FULLACCx = 1111 has been been strobed in such as when i\_iter = 10, have DUMPENPIN = 0, and RESETENB=1. Then it would test that it is DUMPENPIN that is inhibiting the reset pulses.

If there were unwanted reset pulses, then the total accumulated would be wrong.

When all done with MUXLOOP, only once:

- Dump to Storage by setting RESETENB=0, pulsing a 1 to Blanking.
- Read out all 64, 64 lag blocks and check for correct values.

### **BUS CHECKS**

**(These can occur simultaneously with some of the above checks so the same random numbers are going through the Main and Aux Busses)**

- Mode T1 LTOR-IN=1, LTOR=1, CENTERBUS=0, AUXEN=0.
  - Check Main Bus to DTL
  - Check Random Numbers in DL to DR
  - Check DTL=0.
- Mode T2 LTOR-IN=0, LTOR=1, CENTERBUS=1
  - Check Main Bus to DTL, DL, and DR
- Mode T3 LTOR-IN=0, LTOR=1, CENTERBUS=2, AUXEN=1.
  - Check Aux Bus to DTR, DL and DR
- Mode T5 LTOR-IN=0, LTOR=0, CENTERBUS=3
  - Check Main Bus to DTL
  - Check Random Numbers in DR to DL
  - Check Aux Bus to DTR
- Be checking that LTOR-OUT is following the state of LTOR.

### **SECONDARY REGISTERS TEST**

- RC\_TSTE64=1, FULLACCx=0
- Enter data to get all 5s in the Secondary Registers.
- Check all 64 Registers.
- Enter data to get all As in the Secondary Registers.
- Check all 64 Registers.

### **PROGRAM WORD TEST**

- Whenever a Program word is shifted in, verify that the previous program word is shifted out, along with the valid clock.

### **PROGRAM WORD STROBE TEST**

- By shifting in the program word ahead of time during the test, then strobing it in at the appropriate moment, that adequately test the Program word strobe. The strobe should be noted on the PGM STB OUT pin.

### **CKPINEN TEST**

- Check that it gates the C125OUT.

### **XOE YOE TEST**

- Check that XOE\ and YOE\ control the OUT[15:0] bus.