

# Submillimeter Array Technical Memorandum

Number: 66  
Data: May 7, 1992  
From: Andrew Dowd

## Minimizing the Switching Requirements of the Correlator Board.

### I. Introduction

The conceptual specification of the Haystack/Goddard/SAO (et. al) correlator includes a significant number of 64 channel crosspoint switches. These switches are required to implement the various correlator modes (resolution modes) for a particular telescope that will use this board. These switches represent a significant cost element of the design. Also, the only available "true" 64 channel crosspoint switch (from LSI logic) operates marginally at 40 MHz. Therefore, a study of options was made to examine cheaper and faster alternatives.

The first step in this process is to examine the flexibility which is required to implement the necessary modes, then consider simpler switching methods. For my purposes, I limited this study to modes which are necessary for the SMA, primarily because this is the only instrument with which I am familiar. On the other hand, the requirements for the SMA seem to be more difficult to implement so hopefully this analysis will have some validity to the VLBI case.

### II. Reduction Proposal 1: On-chip switches

The first step in minimizing the switching involves the correlator chip itself. Consider a set of crosspoint switches which are implemented on the correlator chip to allow switching the 4 "X" inputs (X0,X1,X2,X3) into any suitable configuration (say for example: X0,X0,X3,X2). If this "internal" capability existed on the correlator chip, then the "external" crosspoint switches on the correlator board become simpler. For example, the external crosspoint switch must only get a specific signal to *any* input of a specific correlator chip. The necessary signal ordering after reaching the chip is handled by the "internal" switches. The consequences of this simplification is to reduce a single 64 channel crosspoint switch into 4 separate 16 channel crosspoint switches. Each 16 channel switches drives one input on the 16 correlator chips. This same analysis applies to both the "X" and "Y" inputs.

Four separate 16 channel crosspoint switches should be cheaper (at this point in time) than a single 64 point crosspoint switch. However, implementing the four separate on-chip 4 channel crosspoint switches may appear a difficult complication. (Note the chip requires 4 separate crosspoint switches to handle the X and Y inputs, with two bits each). In fact, by examining the correlator modes, it is possible to reduce these full cross-point switches into a tractable number of additional inputs to the existing on-chip multiplexing. In the worse case, a mux which exists in the version 1 (3/1/93) chip layout is increased from 5 inputs to 8 inputs. Table #1 lists the additional mux inputs which are necessary. The major constraint imposed by the implementing only a subset of switches options is to limit the resolution modes to factors of 2. (i.e. a chunk will always use 1,2,4,8 or 16 chips to measure a given correlation).

In the appendix to this memo are sketches of the correlator modes which are needed for the SMA. It was from these sketches that the proposed multiplexer inputs were derived. These modes assume the connection from one correlator chip to its neighbor (i.e. the cascade) will be implemented to allow real mode correlations to span chips. The chip-to-chip cascade is somewhat different depending if the data is real or complex. In the complex case (VLBI), the correlator cells are cascaded in pairs. In the real case (SMA), the pairs are daisy chained, with only one set of signals cascaded. Thus the right most output from the correlator chip (designated CXA#R) must connect into the left most input to the cascaded chip (designated XCA#L). (Similarly for the Y signals) If this connection is not made, it is not clear how to do multichip cascades in real mode.

## II. Reduction Proposal 2: 8 inputs not 16. (or 9 inputs)

It is possible for each correlator chip to handle the entire processing of a chunk. However, it is also possible to use two chips for each chunk without tossing out any chunks. Thus, there are, in fact two inputs to a correlator chip for every independent signal. The original proposal filled the "extra" inputs of the full 64 channel crosspoint switch with signals from the another antenna. These 32 extra inputs were available and "free", but not necessary. However, if we plan to forego the LSI crosspoint switches, it seems reasonable to reduce the inputs to the minimum necessary. As a result,

X inputs		Correlator Chip Row				Y inputs		Correlator Chip Row			
Correlator Column	0	1	2	3	Correlator Column	0	1	2	3		
0	<i>X0</i>	<i>X0</i>	<i>X0</i>	<i>X0</i>	0	<i>Y0</i>	<i>Y1</i>	<i>Y2</i>	<i>Y3</i>	<b>Y0</b>	<b>Y1</b>
							<b>Y2</b>			<b>Y2</b>	
	<i>X1</i>	<i>X1</i>	<i>X1</i>	<i>X1</i>							
	<i>X3</i>										
	<i>X2</i>	<i>X2</i>	<i>X3</i>	<i>X3</i>	1	<i>Y0</i>	<i>Y1</i>	<i>Y2</i>	<i>Y3</i>	<b>Y0</b>	<b>Y1</b>
	<b>X3</b>						<b>Y2</b>			<b>Y2</b>	
	<i>X3</i>	<i>X3</i>	<i>X2</i>	<i>X2</i>							
	<i>X1</i>										
	<b>X2</b>				2	<i>Y0</i>	<i>Y1</i>	<i>Y2</i>	<i>Y3</i>	<b>Y0</b>	<b>Y1</b>
							<b>Y2</b>			<b>Y2</b>	
3					3	<i>Y0</i>	<i>Y1</i>	<i>Y2</i>	<i>Y3</i>	<b>Y0</b>	<b>Y1</b>
							<b>Y2</b>			<b>Y2</b>	

Note -

*Italics* entries already exist in version 1 design

**Bold** entries are proposed additions.

Table 1 - On-chip additions to multiplexers to reduce 64 channel xpts switches to 4 - 16 channel xpts.

Modes	Number of correlator chips used on a specific chunk								Number of ignored chunks	Number of states for each mode
	1	2	3	4	5	6	7	8		
1	16								7	8
2	8	8							6	28
3	8	4	4						5	168
4	8	4	2	2					4	840
5	8	4	2	1	1				3	3360
6	8	4	1	1	1	1			2	840
7	8	2	2	1	1	1	1		1	840
8	8	2	1	1	1	1	1	1	0	56
9	4	4	4	4					4	70
10	4	4	4	2	2				3	336
11	4	4	4	2	1	1			2	378
12	4	4	4	1	1	1	1		1	280
13	4	4	2	2	1	1	1	1	0	420
14	4	2	2	2	2	2	2		1	56
15	4	2	2	2	2	2	1	1	0	168
16	2	2	2	2	2	2	2	2	0	1

7,849 = 13 bits

Table #2- Correlator chips per chunk for different modes : Crosscorrelation mode, No Polarization (identical, for autocorrelation modes)

Modes	Number of correlator chips used to a specific chunk								Number of ignored chunks	Number of states for each mode
	1	2	3	4	5	6	7	8		
1	16								7	8
2	8	8							6	28
3	8	4	4						5	168
4	8	4	2	2					4	840
5	4	4	4	4					4	70
6	4	4	4	2	2				3	560
7	4	4	2	2	2	2			2	420
8	4	2	2	2	2	2	2		1	56
9	2	2	2	2	2	2	2	2	0	1

2151 =

Table #3 - Correlator chips per chunk for different modes : Crosscorrelation with Full Polarization

only 8 inputs are required into the crosspoint switches (of the new design). This reduction does eliminate one potentially useful mode: using the correlator board to process a single chunk with all 32 chips. (The final word in frequency resolution!). In interferometric operation, this super high resolution mode would have limited utilization because it would require tossing baselines. However, during single dish operation, it might be nice. There are two methods to get back this mode without returning to a full 16 inputs on the switches. One possible method would need switches on the correlator chip to route the Y inputs to the X inputs. This extra switching seems unnecessary for the limited benefit this one mode produces.

A simpler method is possible by increasing the inputs to the external crosspoint switch to 9 and connect one signal from antenna A to the antenna B switch and vice-verse. This would limit a specific chunk to be the only possible "hi-res" chunk. The versatile downconverter which is planned for the SMA could easily contend with this limitation for the small number of cases where this mode is needed.

#### IV. Reduction Proposal 3: 12 outputs not 16.

This reduction proposal is the most complicated and produces a marginal improvement. However, when considering the actual implementation, this small improvement may be significant. Consequently, it does warrant some consideration.

The basis of this reduction is to hard-wire two correlator chip inputs to the same external crosspoint switch signal. Thus in all modes, these connected correlator inputs will have identical data. If the SAO correlator never used a single correlator chip to process a chunk, then it would be possible to reduce the outputs from 16 to 8. Thus every correlator input would be paired to another correlator input which has identical data. However, by requiring one-chip per chunk modes, it is only possible to reduce the independent signals to 12. Thus the switching matrix is composed of 8-to-12 channel crosspoint switches.

The wiring to tolerate this reduction is not obvious. The necessary connections are sketched in figure #1. The Y and X inputs must be connected differently to accommodate the crosscorrelation modes with and without polarization data. By wiring only half the correlator chips on a board in this reduced method, the SAO can use the other chips to handle the one-chip-per-chunk modes. From Table #2, At most 6 chips will operate in this one chip mode.

The specialized connections of figure #1 are only necessary to implement the "flat" mode, where the hardwired groups perform as 2 chips per chunk. The other modes (4 chips per chunk or more) can be implemented by driving the same signals into both sides of the pair.

The somewhat less critical configuration of the pairings on a correlator board are sketched in Figure #2.

#### V. Reduction Proposal 4: Baseline dependence

One final reduction can be achieved by forcing the Y antenna input to be identical for both baselines which exist on the correlator boards. This reduces two sets of switches on the Y's to single group. The only consequence for the SMA of this fixed connection is to force the two baselines (on the same correlator board) to have identical resolution modes during a given observation.

One complication of this reduction which may affect the chip is the handling of VLBI blanking. I believe the VLBI was going to use the Y inputs to distribute blanking. If the correlator chips expect this blanking on a specific pin (MS or LS) than the proposed reduction cannot be made. Ideally, the chip should be able to select which pin (LS or MS) of the Y-input to receive the blanking data. Thus the two bits can be used independently to drive the two separate baselines which exist on a correlator board

## VI. Implementation

There are lots of possible PAL-PLD-FPGA chips that can accommodate the 9 input to 12 output switching described by this memo. Lots of these will operate at speeds in excess of 100 MHz, with deterministic propagation delays. However, in general the cheapest methods do not have enough power to perform a full crosspoint switch. (This is easily understood when one considers an 8-to-12 crosspoint switch needs 36 bits of information to specify all its options.)

The most difficult limitation on these devices will be the large number of states (mode options) which they must implement. Although a full 36 bits is not necessary, its surprising how many different selections are necessary to implement the modes for the SMA. Referring to Table 2 and 3, the number of different "modes" (or states) which must be implemented in these 8-to-12 switches is over 17,000 (which translates into 15 bits, minimum).

In practice, a significant number of these states will be redundant (i.e. the same setting will work for more than one mode). Also, for many of the modes a significant number of correlator inputs are *don't cares*, which can be connected to any input. Still, the number of states is quite impressive, and requires more than a simple PAL circuit.

An additional complication is the limitations on the number of terms (AND-OR) which are available in programmable devices. If only the minimum number of bits were available, it greatly complicates the term reductions. Thus a PAL may have enough states (registers) to hold the necessary mode options, but not enough terms in its and-or array to produce the necessary modes. The *don't care* outputs will help this problem, somewhat.

## V. Conclusion

At the moment, the primary goal is to finalize the chip. Therefore, the actual implementation of the 9-to-12 switch is not critical. Suffice to say it should be an easier problem than the full 64 point option. That said, we would propose two changes to the initial design of the chip. First, add the multiplexer options described in table 1. Second, allow VLBI to accept blanking data on either the MS or LS bits of the Y input.

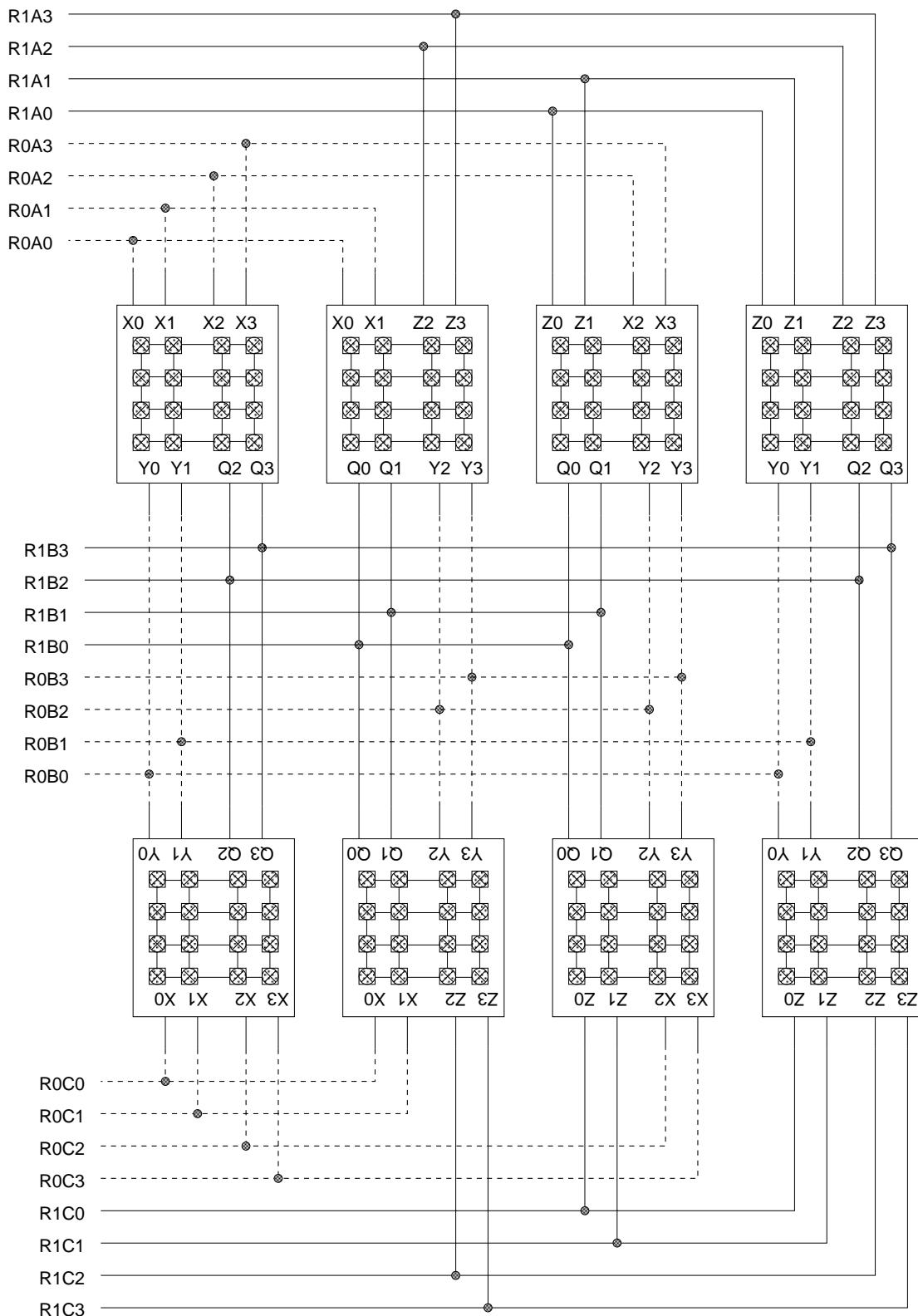


Figure 1 - Deterministic connection of correlator chips to reduce Xpt outputs

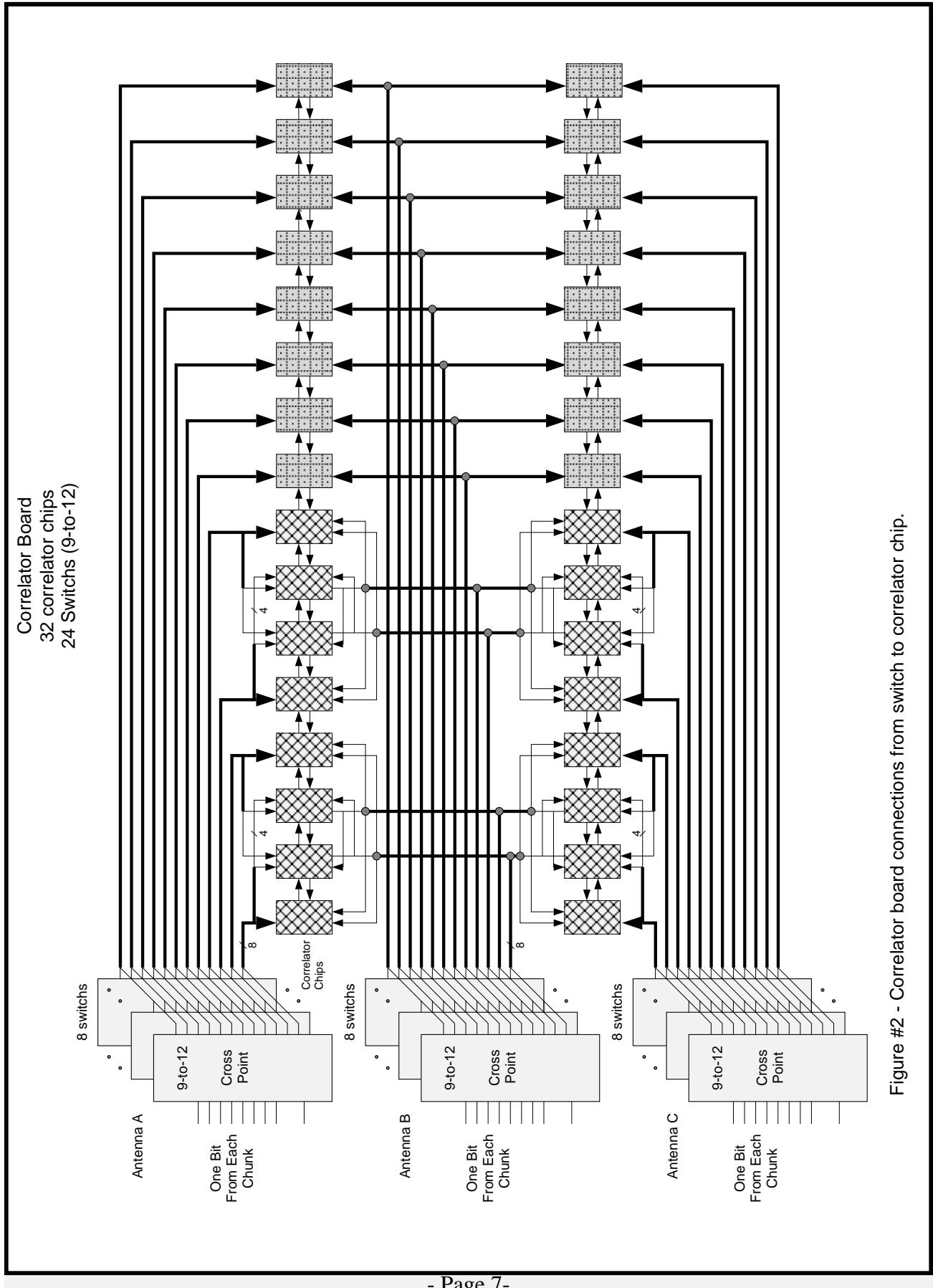


Figure #2 - Correlator board connections from switch to correlator chip.

## Appendix

The following tables show the multiplexing options which are needed to implement the SMA modes. In this case, the modes refer to specific resolution options from a single chunk. In most cases, several of these modes will be implemented on a given correlator card to measure the available 2 Ghz band with a given (possibly different) resolution at each chunk.

### Crosscorrelation Modes with No Cross Polarization

Chips per chunk	Pages	Description
1		Not shown (represents "native mode", all muxes = 0).
2 option A	A2-4	Note 1
2 option B	A5-6	Note 2
4	A7	
8	A8-9	Note 3
16		Note 3 Not shown (a direct expansion of previous mode)
32		Not supported

### Autocorrelation Modes

Chips per chunk	Pages	Description
1	A10	Only implemented with independently connected chips.
2 option A	A11-13	Note 1
2 option B	A14-15	Note 2
4	A16-19	
8		Not shown (a direct expansion of previous mode) Note 3
16		Not shown (a direct expansion of previous mode) Note 3
32		May be supported on specific chunk, see previous

### Crosscorrelation Modes AND Crosspolarization

Chips per chunk	Pages	Description
2 option A	A20-23	Note 1
2 option B		Note 2 -> Not shown (identical to Cross correlation with No Polar, 1 chip per chunk)
4		Not shown (identical to Crosscorr. with No Polar, 2 chips per chunk)
8		Not shown (identical to Crosscorr. with No Polar, 4 chips per chunk.)
16		Not shown (identical to Crosscorr. with No Polar, 8 chips per chunk.) Of questionable value since it can only produce two of four cross terms.

#### Note

1. This option implements this mode on the hardwired pairs section of the correlator board. (Figure 1 and Figure 2, left side)
2. This option implements this mode on the independently connected section of the correlator board (Figure 2, right side)
3. Requires a direct casacade from chip to chip, NOT pairwise

