

MASTER

AT/10.4/002 MUST WE RECIRCULATE THE AT CORRELATOR?

J.D. O'SULLIVAN

DIVISION OF
AT/10.4/002
RADIOPHYSICS

COMMONWEALTH SCIENTIFIC AND INDUSTRIAL RESEARCH ORGANIZATION

DIVISION OF RADIOPHYSICS

MUST WE RECIRCULATE THE AT CORRELATOR?

J. O'Sullivan 12.7.83

1. Introduction

Previous reference has been made to the necessity to recirculate the AT correlator system to gain larger numbers of channels at low bandwidths (AT/05.4/001). Recirculation is a well known technique for time multiplexing a high speed correlator to provide an equivalent number of low speed tasks, such as more lag channels at lower bandwidths. Some first look recirculation parameters were given in AT/10.1/027.

The XCELL chip proposed and also described in AT/10.1/027 achieves a large bandwidth by parallel operation of large numbers of low speed correlators (perhaps 10 MHz sampling rate in 1-bit). This suggests a certain redundancy of effort if recirculation is used to increase the number of channels which are present in the XCELL chip but combined in post-correlator processing.

This note examines the possibilities in the light of up to date information.

2. Recirculation parameters

AT/10.1/027 assumed that the fastest rate that all correlators could be read was 5msec. Using this rate and a maximum input data rate of 320 Mbit, we require a 3.2Mbit buffer per input with double buffering and 6 telescopes x 2 polarisations of inputs, the total is 4.8Mbyte of recirculator input memory.

The situation may in fact be somewhat worse if cheap memory doesn't come in the right shape to be useful. Consider 64K x 1bit chips with a read cycle time of 200 ns (not too expensive). We need a lagged and an unlagged output (because we want to correlate all outputs with each other) so the maximum buffer readout is 400 ns. The buffer must be 128 bits

wide to accommodate 320Mbit/s and only 12.5K locations are used. We must supply 2 x 64K locations so the total memory becomes 40Mbyte and we might as well cycle at 52ms instead of 5ms. Of course 16K x 4bit or 8K x 8bit chips might be available shortly and provide an economical alternative. We could well be looking at \$50-100K for the recirculator input buffers alone.

On the correlator output side, a recirculation factor of 64 yields a maximum of 8K channels times 60 separate products of (say) 32 bits giving a total of 2Mbyte. This must be at least double and perhaps triple buffered to allow subsequent van Vleck correction, lag to frequency transformation and transfer to disk (assuming that we bypass the on-line VAX-750).

Now suppose that all 2Mbytes must be read into at a 5ms rate and accumulated according to the pattern of subproducts and XCELL readout order. In fact, for each 32 bit summed lag channel, some 32 separate 24 bit XCELL subproducts must be summed (assuming a basic 10MHz XCELL clock rate). Any one set of 256 lags must therefore be updated at a rate of 0.6 μ s. Given that each 24 bit XCELL readout costs perhaps 10-20 μ s, this suggests that each 256 correlator block will require 15-30 accumulator units.

It would obviously be preferable to have one accumulator per 64, 128 or 256 lag channels. The maximum readout rate would then for 256 channels be 80-160ms.

3. Basic XCELL array operation

As a starting point, I shall relate the current XCELL philosophy as told me by Andrew Hunt. The XCELL chip is an array of 8x8 1-bit correlators (let's stick with 1-bit for the moment) shown in figure 1.

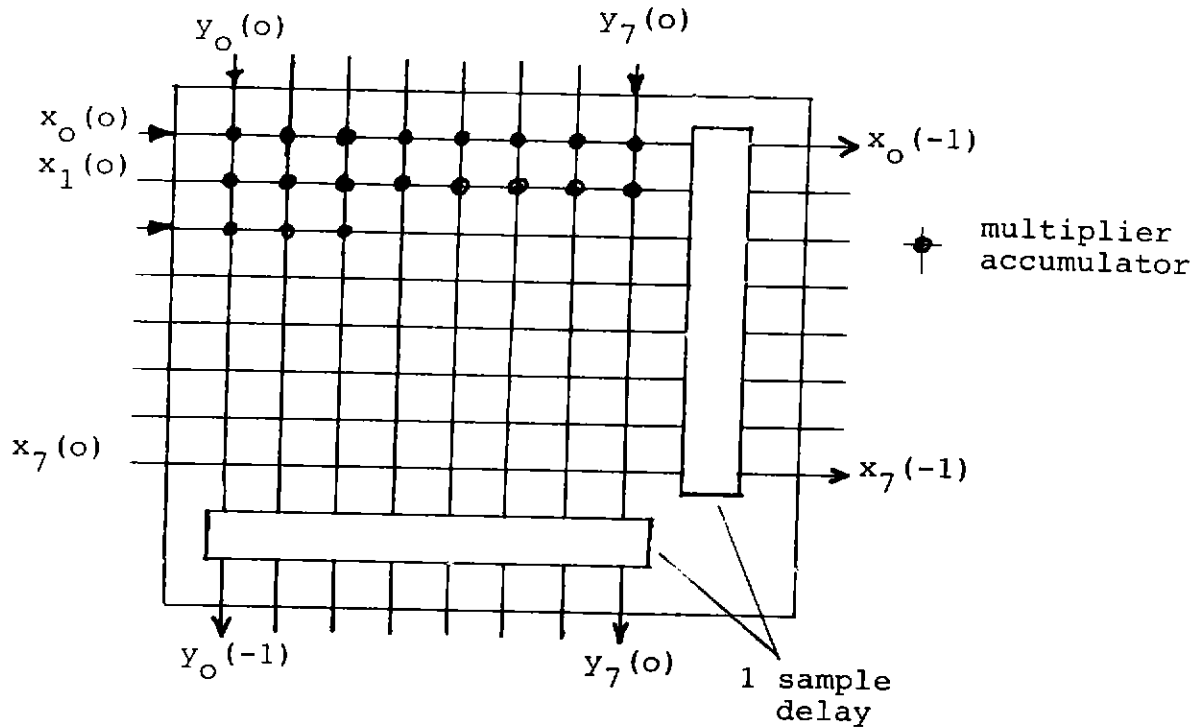


Figure 1: A schematic representation of an 8x8 XCELL.

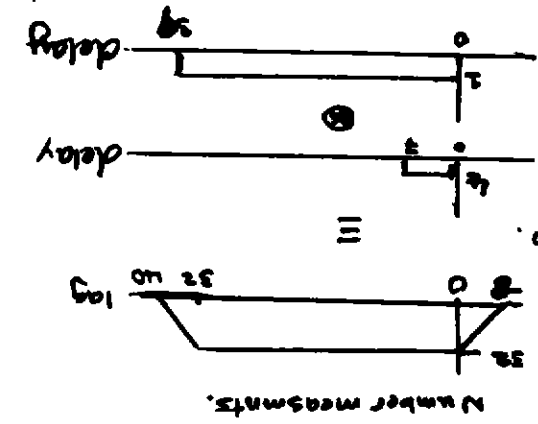
Each intersection is correlated i.e., a full set of products $\sum_I x(I) y_j(I)$ are performed. The outputs are delayed by one sample.

The elegant solution proposed by Ables uses arrays of XCELL arrays cascaded as shown in figure 2.

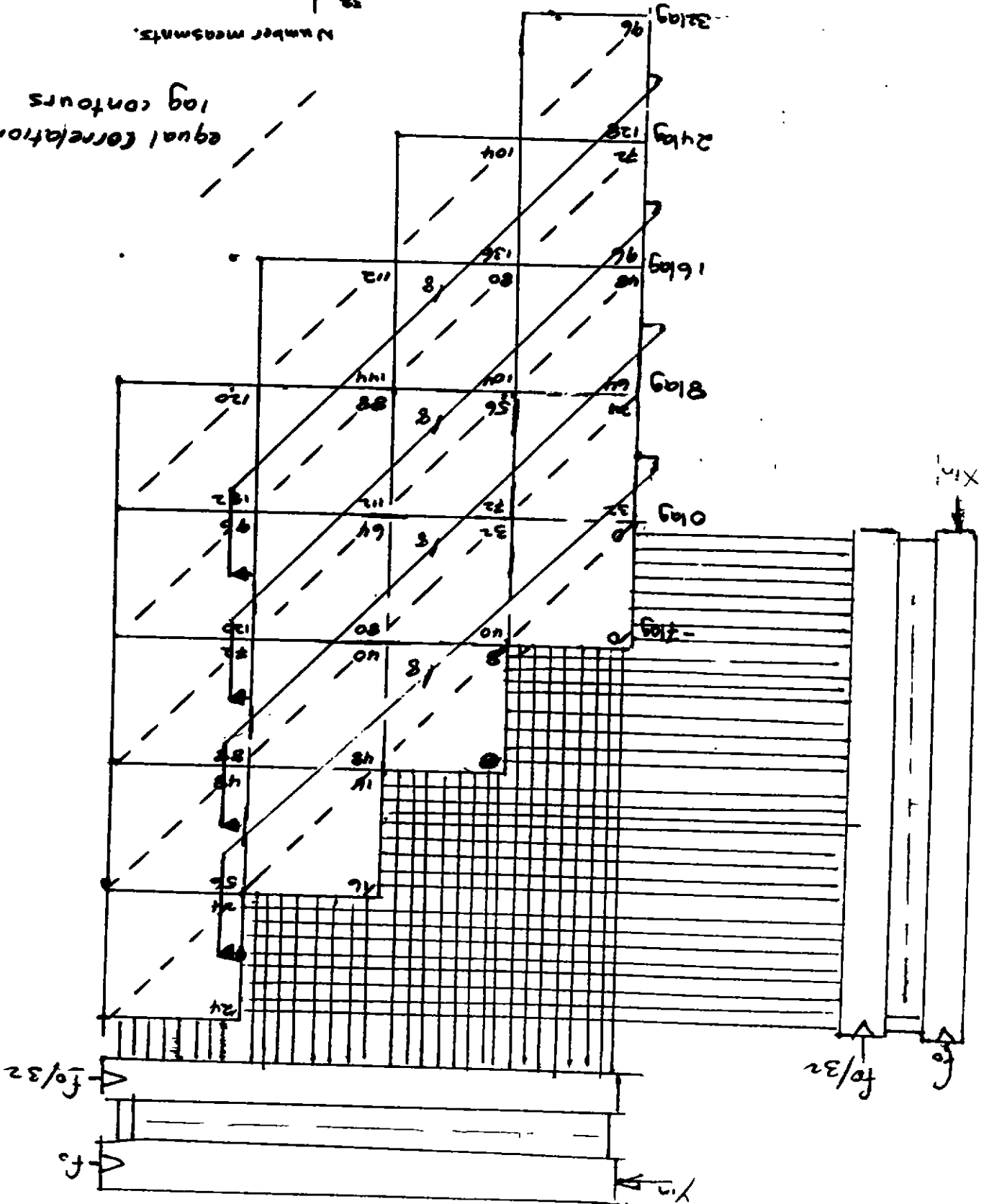
Groups of 32 (in this example) samples are converted to parallel inputs with delays 0 to 31 and passed to the XCELL array. We can assign delays by following each signal through the array and when crossing an XCELL boundary adding 32. This is determined by the XCELL clock rate $f_0/32$ in this example. The figure in the lower left hand corner of each cell is the first x input signal delay to that XCELL, and the figure in the upper left corresponds to first y input signal.

The correlator lag channels are then relatively easily followed. Some 32 separate low speed correlations are required to estimate a single lag at the full f_0 rate. The equal lag contours are shown dotted. A check that any pair of correlations are distinct can be made by examining the signal input delays modulo 32.

Figure 2. Full bandwidth operation.



equal correlation
log contours



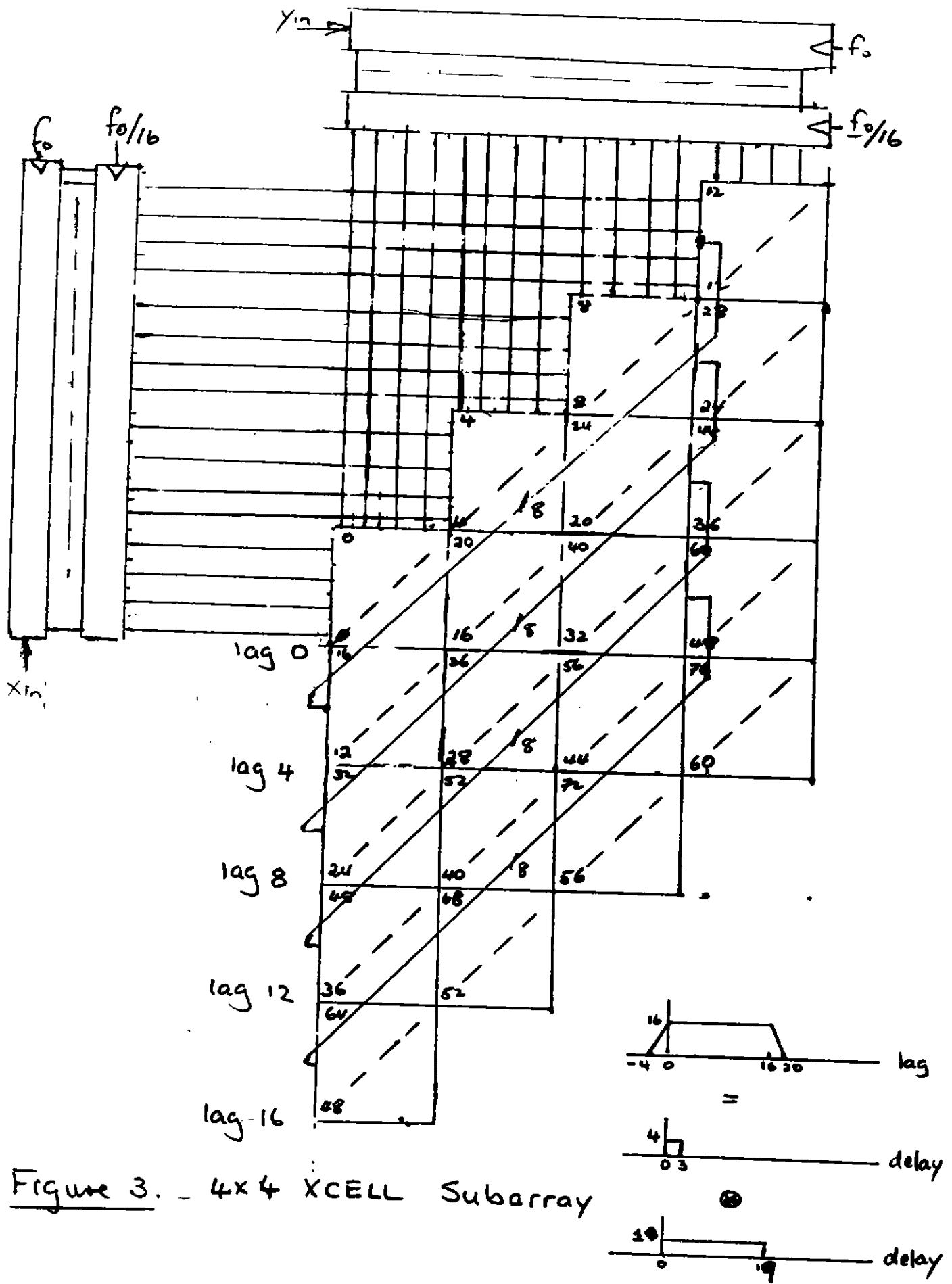


Figure 3. - 4x4 XCELL Subarray

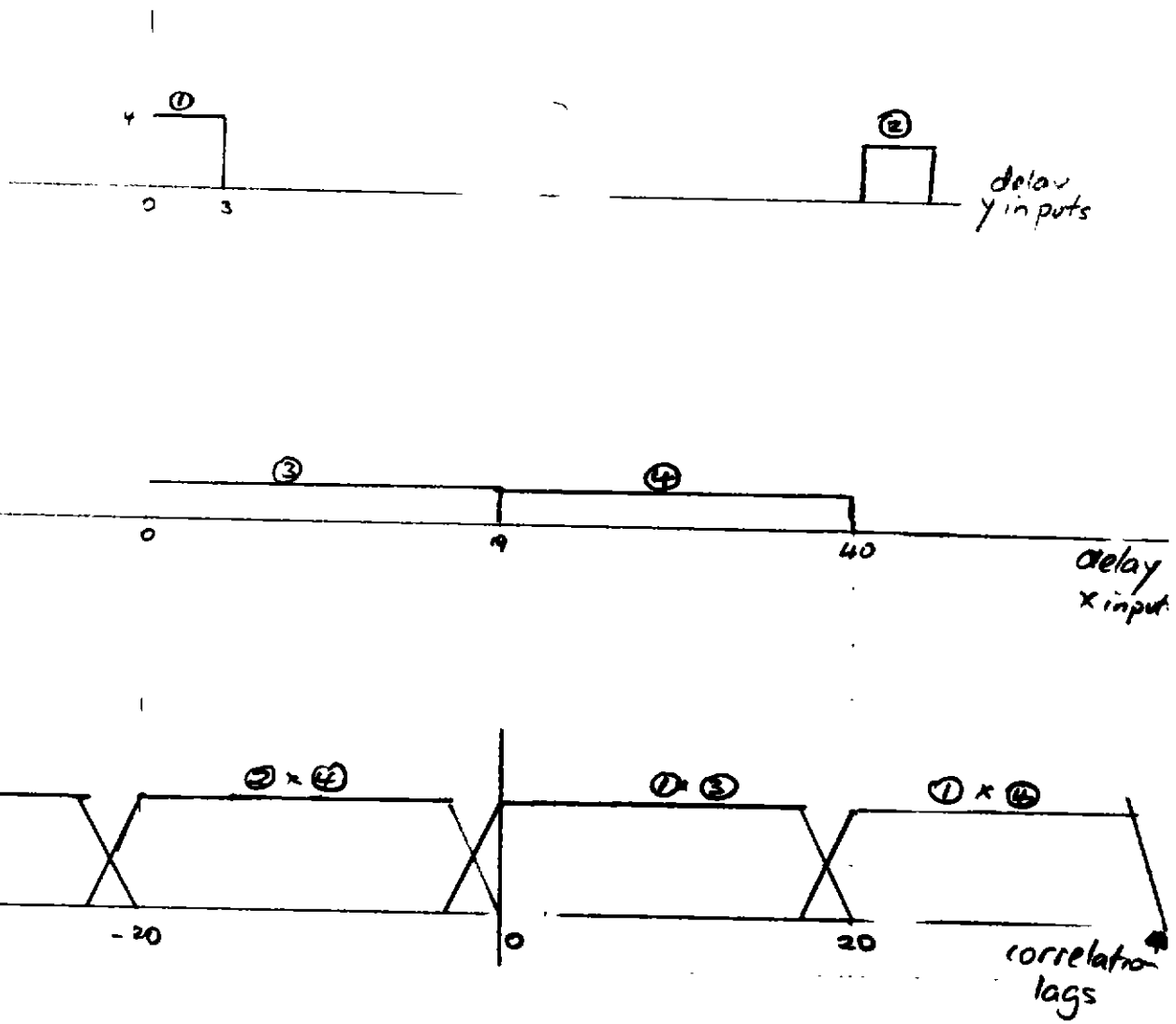


Figure 4 Possible solution to measure all lags with 4 4x4 XCELL subarrays.

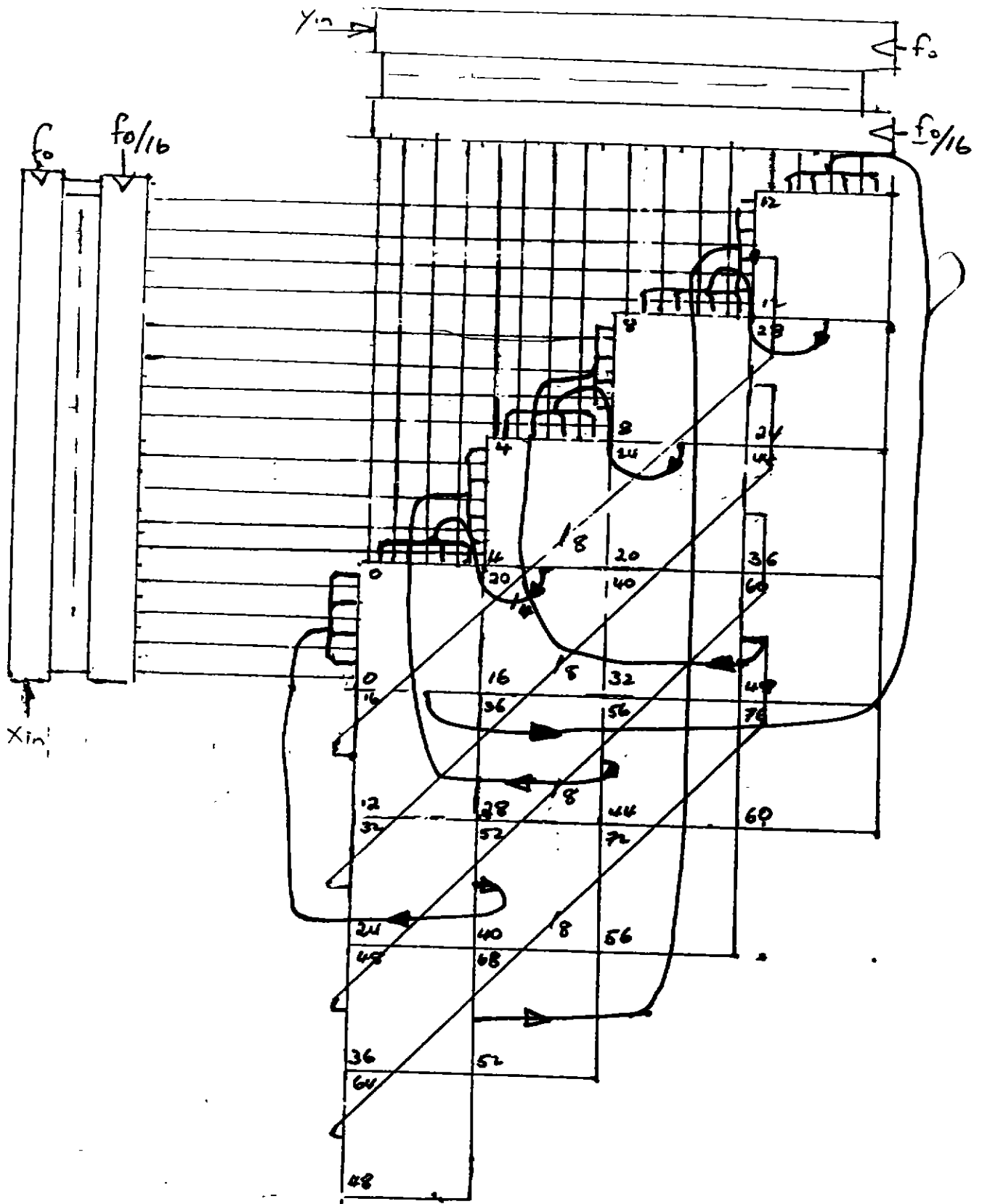


Figure 5 — An implementation of f_0 to produce lags as in figure 4.

Figure 2 shows that the lag values measured are complete from 0 to 32 and tail off linearly to zero at lags -8 and + 40. The systolic array of XCELL's has effectively convolved a set of 8 y input samples with a set of 32 x input samples.

4. Half bandwidth operation

Operation at half bandwidth in effect requires that for an input frequency f_0 , a division factor $f_0/16$ is required in order to maintain the XCELL clock rate at full capacity. Obviously, we could draw a new XCELL array appropriate to 16 bits wide and twice as long but such a solution would require extensive switching. We accordingly look for a solution where only the inputs are altered.

The XCELL array can be made appropriate to 16 bits wide by utilising every other input for a different function. The result is 4 separate 4x4 XCELL arrays providing a total width of 16 bits in this example. Figure 3 shows the 4x4 XCELL array lags for one such subarray. Now only lags -3 to 19 are measured.

There are 4 such subarrays corresponding to the 2 sets of x inputs and 2 sets of y inputs.

Delays must be inserted in one or more input sets to arrange that each subarray measures a distinct set of lags. Various solutions are feasible and figure 4 shows one such solution.

The delays required can be provided externally or obtained from existing delayed signals within the XCELL array. The latter solution has a certain simplicity (not apparent from my drawing) and is shown in figure 5. The exact "feedback" points will depend on the overall cascade length of the array.

Note that a delay lag range double that of the full bandwidth case has been obtained but now almost symmetrically arranged around zero lag. For a single interferometer system the small offset could easily be compensated in a small delay system per telescope. For the 15 interferometer case each input must be split into delayed and undelayed versions. This method would appear to alter the symmetry of each stage so some form of delay must be provided.

5. Further bandwidth halving

The procedure can be repeated a further two times, at which time the XCELLS are split from one 8x8, four 4x4, sixteen 2x2 arrays and 64 1x1 arrays.

Furthermore, by simply arranging a selector for each second, fourth and eighth inputs with similar feedback connections for bandwidth doubling we can arrange to select each of the four bandwidth reduction possibilities. The following table summarises the connections required.

TABLE 5.1

XCELL Input	Max band	$\frac{1}{2}$ band	$\frac{1}{4}$ band	$\frac{1}{8}$ band
0	S0	S0	S0	S0
1	S1	F0	F0	F0
2	S2	S1 (S2)	F1	F1
3	S3	F2	F2	F2
4	S4	S3 (S4)	S1 (S4)	F3
5	S5	F4	F4	F4
6	S6	S4 (S6)	F5	F5
7	S7	F6	F6	F6

S_i are shift register bits.

F_i are feedback bits. Feedback placement depends on array size.

Note: The diagrams are drawn assuming f_0 is lowered at each step. In that case consecutive shift register bits are required as shown in the table. An alternative would be to remain sampling at full bandwidth but not use all samples. The shift register bits in brackets are relevant to that case.

Conclusions

There is little point in pursuing this case in detail any further. The ultimate array to produce 32 bidirectional lags at 320 MHz sample rate should be examined. It seems that provided satisfactory answers to the asymmetry problem can be found, reconnection to 160 MHz, 80 MHz and 40 MHz should be possible (perhaps further also).

Larger numbers of channels could be provided by recirculation beyond this stage (perhaps added later). The lower speed recirculation should ease input and output buffer problems.