

NATIONAL RADIO ASTRONOMY OBSERVATORY  
Charlottesville, Virginia

(860506)

To: Correlator Memo Series  
FROM: J. H. Greenberg  
SUBJECT: Station Based Lobe Rotation in the Correlator

I. Summary

This memo presents how the wavefront phase for a lobe rotator can be approximated as a Taylor Series. A scheme for implementing the Taylor Series in hardware is presented. This could form the lobe rotator for a central correlator. A numerical analysis quantifies the system size. It is shown how the system size could be reduced by having the Taylor Series done in two stages. One stage would be done by a central controller. The other stage would be done by 160 channel based elements.

II. Introduction

The wavefront phase for VLBA antennae can be referenced to the center of the Earth. The wavefront phase varies as:

$$F = A \cos HA + C$$

Where: F = Wavefront Phase  
HA = Hour Angle.

A and C are constants.

$$A = \frac{\text{Earth Radius} * \cos \text{Latitude} * \cos \text{Declination}}{\text{Wavelength}}$$

$$C = \frac{\text{Earth Radius} * \sin \text{Latitude} * \sin \text{Declination}}{\text{Wavelength}}$$

A lobe rotator needs to generate this function. The function can be approximated by the Taylor Series:

$$F(HA) = F_0 + (HA - HA_0) * F_0' + (HA - HA_0)^2 F_0'' / 2! + (HA - HA_0)^3 F_0''' / 3! \dots$$

For an initial HA = HA<sub>0</sub>, the initial F and its derivatives are calculated for use in the series as follows:

$$\begin{aligned} F_0 &= A \cos HA_0 + C \\ F_0' &= -A \sin HA_0 \\ F_0'' &= -A \cos HA_0 \\ F_0''' &= A \sin HA_0 \end{aligned}$$

Substituting:  $F(HA) = A (\cos HA_0 - (HA - HA_0) \sin HA_0$   
 $(HA - HA_0)^2 \cos HA_0 / 2! + (HA - HA_0)^3 \sin HA_0 / 3! + \dots)$

### III. A Hardware Representation of the Taylor Series

To represent the Taylor Series by digital logic, discrete values of HA are used, rather than a continuous function. Represent the hour angle as:

$$HA = HA_0 + kB$$

Where: B = The incremental value of HA. This is a constant equal to the rotational rate of the Earth times the clock period.

k = the integral number of B's since  $HA_0$

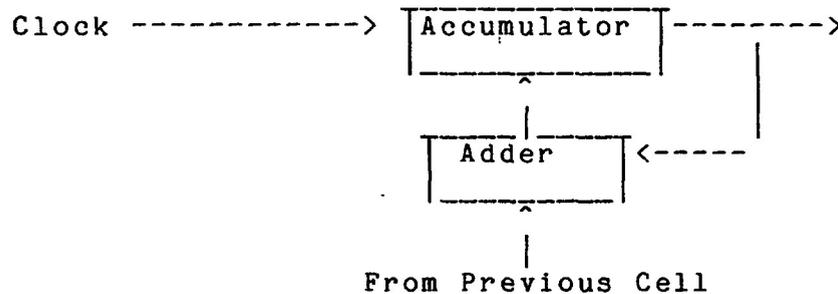
Hence:  $HA - HA_0 = kB$

Substituting:

$$F(k) = A (\cos HA_0 - kB \sin HA_0 -$$
  
 $(kB)^2 \cos HA_0 / 2! + (kB)^3 \sin HA_0 / 3! + \dots)$

Note that k is the only changing variable. To represent this equation in hardware, the following cell will be used.

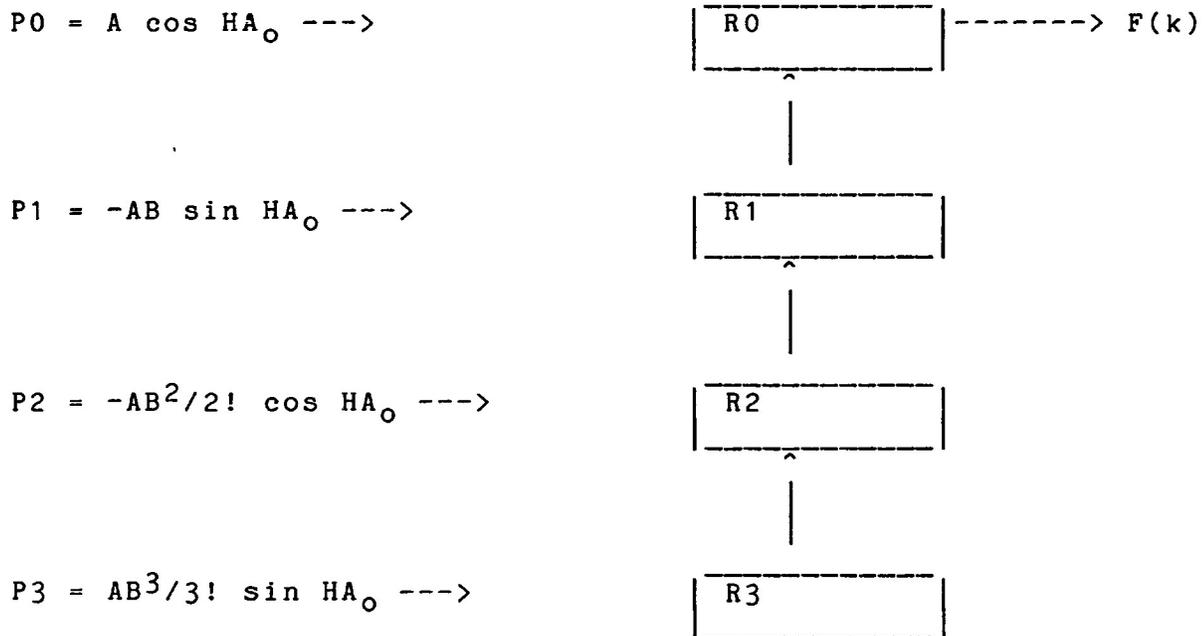
Figure 1



When triggered by the clock, the value from the previous cell is added to the contents of the accumulator and stored in the accumulator. On subsequent figures, this cell will be shown as a block, and called a register.

Now, combining these registers, the above equation can be represented in hardware. This is shown in Figure 2.

Figure 2



The values on the left are loaded in the registers when  $HA_0$  is defined. For each value of  $k$ , the lowest register is added to the one above it, which is then added to the one above it, until it ripples through to the top.

The net result, after  $k$  cycles is:

$$\begin{aligned}
 F(k) &= P_0 + k(P_1 + k(P_2 + kP_3)) \\
 &= P_0 + kP_1 + k^2P_2 + k^3P_3 \\
 &= A (\cos HA_0 - kB \sin HA_0 - \\
 &\quad (kB)^2 \cos HA_0 / 2! + (kB)^3 \sin HA_0 / 3! )
 \end{aligned}$$

Thus a hardware representation of the Taylor Series is provided.

The main computer could allow for corrections to the wavefront phase function by varying the coefficients supplied.

#### IV. Numerical Analysis of the Hardware Taylor Series

In the hardware, it is desirable to know how big the registers have to be and how often they have to be initialized. This is accomplished by substituting values for the variables. For a 32 MHz clock, with a unity playback factor, the incremental HA is found to be:

$$B = 2 * 3.1416 \text{ radians} / (86164 \text{ sec.} * 32E6 \text{ increments/sec.}) \\ = 2.28E-12 \text{ radians/increment}$$

A value for the constant A is found as follows:

$$A = \text{Earth Radius} * \cos \text{Latitude} * \cos \text{Declination} \\ / \text{Wavelength}$$

$$\text{Earth Radius} = 6.378E6 \text{ meters}$$

For maximum wavefront phase rate, set Latitude and Declination to zero.

The shortest observing wavelength can be taken to be 3.26 mm. This corresponds to the high edge of the 92 GHz radio astronomy band. This gives  $A = 6.378E6 \text{ m} / 3.26E-3 \text{ m/cycle} = 1.95E9 \text{ cycles}$ . Rounding to  $2.00E9 \text{ cycles}$  gives the value of A used.

For the latitude or declination equal to zero, the constant C from page 1 equals zero. C only affects the initial phase value, since it disappears on differentiation.

Table I

Pa Term #	b Term Value	c k until 0.0001 cycle Error	d Time until 0.0001 cycle Error	e Ratio
P0	2.00E9 cos HA <sub>0</sub>			
P1	-4.56E-3 sin HA <sub>0</sub>	.022	0.69 ns.	6.32E6
P2	-5.19E-15 cos HA <sub>0</sub>	1.39E5	4.34 ms.	212
P3	3.94E-27 sin HA <sub>0</sub>	2.94E7	0.92 sec.	15.6
P4	2.25E-39 cos HA <sub>0</sub>	4.59E8	14.4 sec.	5.4
P5	-1.02E-51 sin HA <sub>0</sub>	2.50E9	78.1 sec.	3.2
P6	-3.89E-64 cos HA <sub>0</sub>	7.97E9	249 sec.	2.3
P7	1.27E-76 sin HA <sub>0</sub>	1.87E10	583 sec.	1.9
P8	3.61E-89 cos HA <sub>0</sub>	3.59E10	1120 sec.	1.7
P9	-9.13E-102 sin HA <sub>0</sub>	6.06E10	1890 sec.	

Table I substitutes B and A to obtain the terms for the Taylor Series. The letters a through e above the column represents a variable with the value of the column.

Also shown is how often it is necessary to reinitialize the registers. The Taylor Series error is due to using a finite number of terms. The error can be approximated as the maximum value of the next term. The maximum allowable phase error is specified as 0.0001 cycle. Thus, the Taylor Series term is set equal to 0.0001 and solved for k. The sine or cosine function is set equal to one. Hence:

$$bk^a = 0.0001$$

$$k \text{ until } 0.0001 \text{ cycle Error} =$$

$$c = (0.0001 / \text{absolute value } b) ** (1/a) \text{ increments}$$

The time required for 0.0001 cycle error to accumulate is the number of increments until 0.0001 cycle error, divided by the number of increments per second, i.e. the clock rate. Hence:

$$\text{Time until } 0.0001 \text{ cycle error} = d = c / 32E6 \text{ seconds.}$$

The error for the Taylor Series is approximately the next term beyond what is used. The time,  $d$  from Table I, for that error to accumulate is the time for a system using all the registers, up to, but not including the register for which the time is shown.

Example: For a 5 register hardware implementation, the time until error would be 78.1 seconds. The registers would be initialized with P0 through P4.

Column e shows the ratio between the value of  $d$  in the row below the current row and the value of  $d$  in the row above. This ratio shows the improvement in update time obtained by adding another register. Note that diminishing returns set in as the number of registers increases.

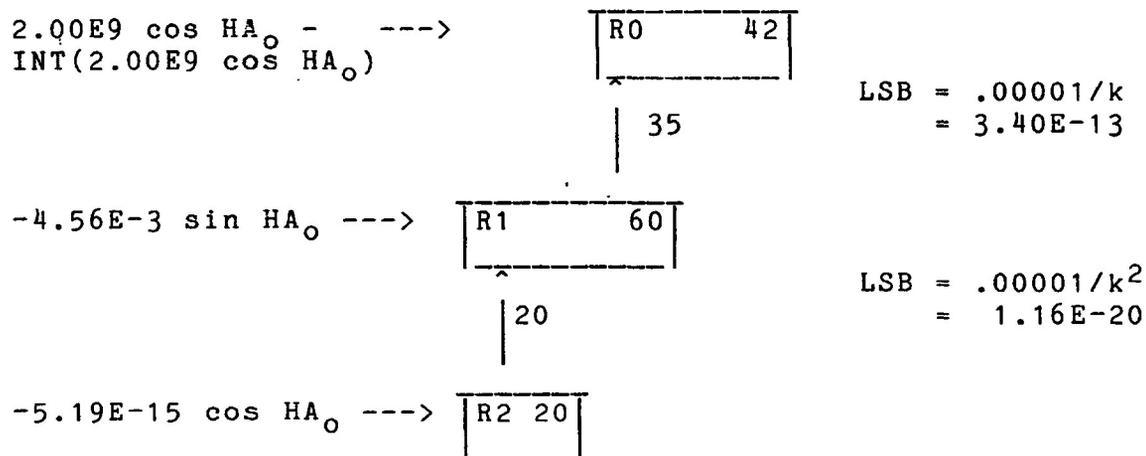
These and subsequent calculations were performed by the VAX program UMA3:[JHG]TAYLOR.PLI which is available for review.

## V. Resolutions

The number of bits in the registers and on the data busses between them must be specified.

Consider the three register system pictured below. Each register initially receives the shown Taylor Series coefficients. These coefficients are functions of the current hour angle.

Figure 3



The registers will need to be reinitialized at least every  $k = 2.94E7$  increments or 0.92 seconds, per Table I.

To calculate the resolution of the registers, it is necessary to know the value of the least significant bit (LSB). The sum of the additions of the LSB should not add up to one tenth the

allowable error of 0.0001 cycle. For R0, there are k additions. For R1, there are  $k^2$  additions. Since nothing is added to R2, it has the same LSB value as R1. The LSB values are shown on Figure 3.

The resolution of a register equals the maximum value to be stored, divided by the value of the LSB. For R0,  $1/3.40E-13$  equals  $2^{41.6}$ , thus 42 bits are required. The value of phase in R0 will always be positive, since the telescope must always be on the side of the Earth illuminated by the source.

The largest value to be in R1 is  $4.56E-3$ .  $4.56E-3/1.16E-20$  requires 59 bits, plus one bit for sign. For R2,  $-5.19E-15/1.16E-20$  requires 20 bits.

The resolution for the data bus R0/R1 is obtained by dividing the maximum value of R1 by the LSB of R0. Thus,  $4.56E-3/3.40E-13$  requires 34 bits plus 1 bit for sign gives 35 bits. R1/R2 requires 20 bits.

Table II, which follows, shows the required resolutions for the registers and data busses, for systems using different numbers of registers. Figure 3 is represented by the column for 3 registers.

Table II

Number of Registers	2	3	4	5
Update Interval in seconds	0.00434	0.92	14.4	78.1
Resolutions in bits				
R0	34	42	46	48
R0/R1	27	35	39	42
R1	27	60	68	73
R1/R2		20	28	33
R2		20	57	64
R2/R3			17	24
R3			17	55
R3/R4				15
R4				15
Chip Counts	16	32	50	64
Times 160 for System Chip Count	2560	5120	8000	10240

In the correlator, 160 lobe rotators will be required. (20 stations times 8 channels). The update period trades off against chip count.

#### V. An Intermediate Controller

A technique to reduce the chip count is to have an intermediate controller between the main computer and the 160 lobe rotators. See Figure 4. The 160 lobe rotators could each be implemented using 2 registers. This performs a linear interpolation between updates. The controller must update each lobe rotator every 0.00434 seconds or sooner.

On Figure 4, the variables associated with the channels are denoted with a '. R0 can be transferred directly to R0'. However R1 is too large, by a factor of the 32 MHz clock rate times T', the update period of the channel. That is:

$$R1' = R1 / (32E6 * T')$$

If  $32E6 * T'$  is a factor of 2, the divide can be implemented by a hard-wired shift in the transfer of data from R1 to R1'.

Example: For updating the two register series, the update period must be less than 0.00434 seconds, per Table II.

$$32E6 * .00434 = 2^{*17.1}$$

$$2^{*17} / 32E6 = 4.096 \text{ ms.}$$

Hence, a 4.096 ms update rate will require a 17 bit shift in going from R1 to R1'.

The register and bus resolution requirements calculate to be the same for the 4.096 ms. update rate as for the 4.34 ms. rate.

The controller is also a hardware Taylor Series. Only now the clock rate is  $1/0.004096\text{sec} = 244.14 \text{ Hz}$  instead of 32 MHz. The controller can be time shared between the 160 lobe rotators, with intermediate results stored in RAM.

The initial register values are calculated as in Figure 2. The variable A is the same value as before. B has changed to reflect the new clock rate.

$$\begin{aligned} B &= 2 * 3.1416 \text{ radians} / (86164 \text{ sec.} * 244.14 \text{ increments/sec.}) \\ &= 2.99E-7 \text{ radians/increment} \end{aligned}$$

The initial register values are shown in Figure 4. Only the fractional values of R0 are required, since the phase repeats every cycle. The resolution of R0 is given a lower bounds of the resolution of R0'. This maintains the precision of R0'. The integral portion of R1 must be retained, since it will become a fraction when shifted on the way to R1'. Only the fractional portion of R1 will be transferred to R0. The time interval before updating the registers is the same as calculated in Table 1, for a given number of registers. The resolutions are calculated as before. The width in bits is the sum of the register widths, minus the overlap of the data bus widths. The 84 bit, width in bits is pictured below.

Figure 4

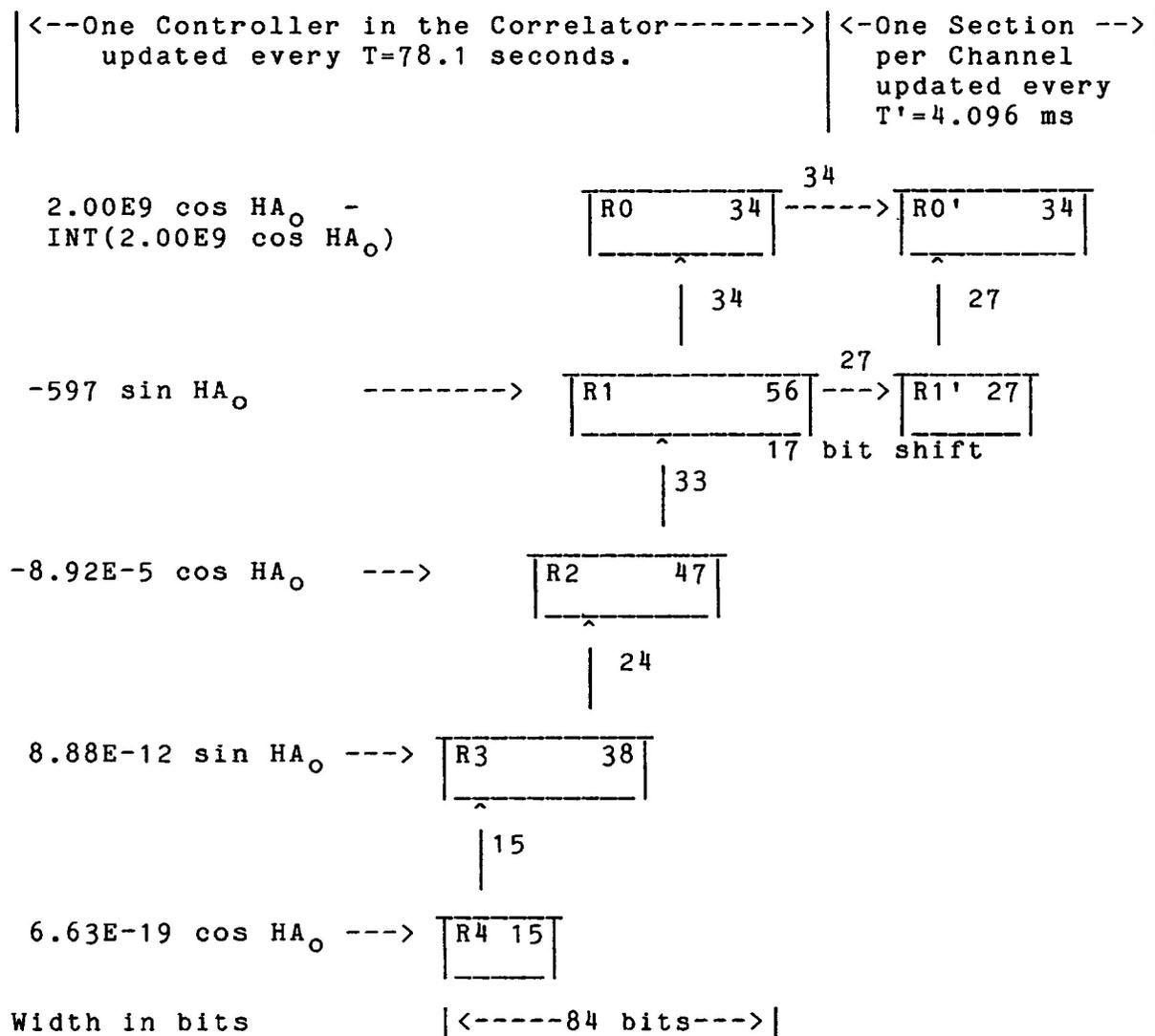


Table III following shows the statistics for systems with various numbers of controller registers. Two lobe rotator registers are used in all cases, in each of the 160 channel based sections. Figure 4 is represented in the 5 register column.

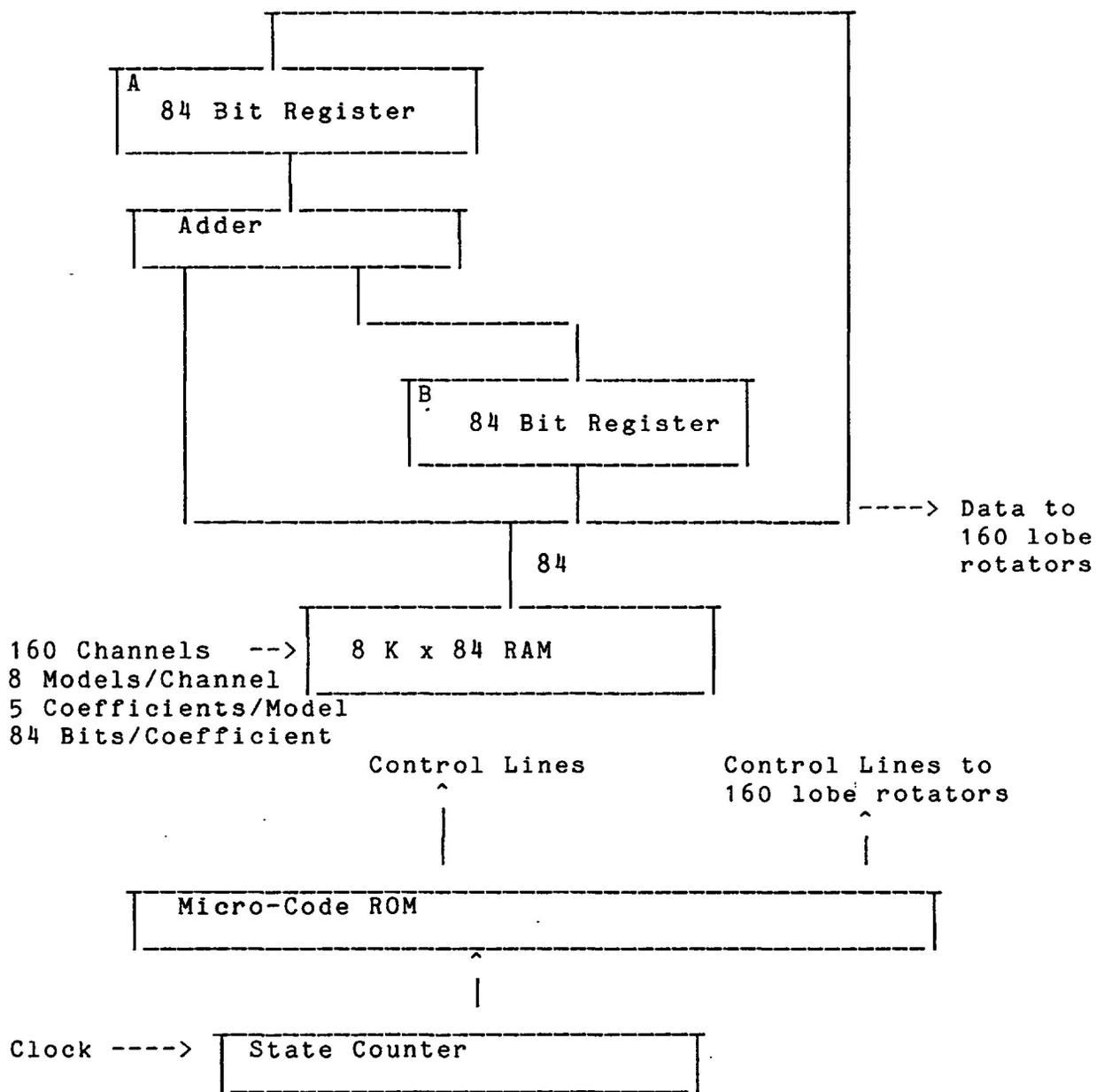
Table III

Number of Registers	5	6	7	8
Update Period in seconds	78	249	583	1120
Resolutions in bits				
R0	34	34	34	35
R0/R1	34	34	34	35
R1	56	59	62	63
R1/R2	33	36	39	41
R2	47	52	56	59
R2/R3	24	29	33	36
R3	38	45	50	54
R3/R4	15	21	26	30
R4	15	37	43	48
R4/R5		13	19	24
R5		13	36	42
R5/R6			12	18
R6			12	36
R6/R7				11
R7				11
Width in Bits	84	107	130	153

## VI. Controller Implementation

The controller could be implemented using a micro-code ROM, an adder, and RAM memory. See Figure 5.

Figure 5



The state counter would increment the micro-instructions in the ROM. These would control the register operations by means of register control lines.

The amount of storage in the RAM is derived as follows. The 20 stations, with 8 channels each, give 160 channels. The 8 models allow the updating of coefficients for several sources. This allows rapid switching between sources, without needing the main computer to reinitialize the coefficients. 4 models could be

waiting to be updated by the main computer, while the other 4 could be tracking. They would switch on a 50% duty cycle. This allows the system to be asynchronous with the main computer.

A typical ROM instruction sequence for updating a 5 register system could be:

Table IV

Instruction	A Register Contents	B Register Contents
Load B with R4		R4
Add R3 to B	R3 - new	R4
Store R3 - new	R3 - new	R4
Load B with R2	R3 - new	R2 - old
Add B to A	R2 - new	R2 - old
Store R2 - new	R2 - new	R2 - old
Load B with R1	R2 - new	R1 - old
Add B to A	R1 - new	R1 - old
Store R1 - new	R1 - new	R1 - old
Output R1 - new to lobe rotator being updated.		
Load B with R0	R1 - new	R0 - old
Add B to A	R0 - new	R0 - old
Store R0 - new	R0 - new	R0 - old
Output R0 - new to lobe rotator being updated.		

Thus 14 instructions were used to update the lobe rotator. since there are 4 models of 160 lobe rotators being updated every 0.004096 second, that leads to 2,187,500 instructions per second. That is a 0.457 micro-second clock rate.

The controller could be implemented with about 100 chips. Referring to Table II, this allows a total system chip count of 2660. The chip count for 160, 5 register lobe rotators was 10,240. Thus the controller allows a substantial savings, in addition to allowing for different models.

The controller could be expanded to the larger register counts of Table III. An eight register controller is attractive for addressing purposes. This would require the main computer to update the controller every 1120 seconds. The width in bits would be 153. Rather than have the RAM be 153 bits wide, it could be narrower and shifted on loading.

## VII. More on Resolutions

This system shows R0, the phase register, having the required number of bits to maintain 0.0001 cycle resolution. For the two register system, this was 34 bits. This number of bits is necessary maintain the accuracy of the large number of additions of small increments to the phase. The correlator will input the phase using much less resolution. Eight bits is a possible implementation.

VLBA Correlator Memo VC 050, Revised Phase/Delay Specification by Martin Ewing, specifies a closure phase error of less than 0.01 degree averaged over 0.1 second. 0.01 degree corresponds to 0.0000278 cycle, requiring 16 bits.

The eight bit resolution could have a much higher average resolution over the 0.1 second. The R0 accuracy is needed to maintain the average accuracy of the more significant bits. Exactly how many bits of R0 are required can be a subject of further study.

#### VIII. Conclusions

A hardware Taylor Series system can provide channel based lobe rotation in the correlator. Resolution and update rate are determined by the number of registers used. Having an intermediate controller provides a substantial reduction in chip count.

