TECHNICAL MEMORANDUM (NASA) 15

A SIMULATION ANALYSIS OF PHASE PROCESSING CIRCUITRY IN THE OHIO UNIVERSITY OMEGA RECEIVER PROTOTYPE

A first-order digital phase-lock loop is modelled on the computer. Loop response to signal phase in noise is evaluated. Optimum integration time is determined. Phase jitter in a frequency synthesizer used as the local oscillator is quantified, and design is optimized. Design rules for use of synchronous rate multipliers are presented. Overall system response is discussed.

by

R. A. Palkovic
Avionics Engineering Center
Department of Electrical Engineering
Ohio University
Athens, Ohio 45701

June, 1975

Supported By

National Aeronautics and Space Administration
Langley Research Center
Langley Field, Virginia
Grant NGR 36-009-017

FOREWORD

This paper was submitted to and approved by the Department of Electrical Engineering and the Graduate College of Ohio University on May 29, 1975 as a master's thesis.

TABLE OF CONTENTS

I. INTRODUCTION

This paper describes a Fortran IV simulation study of the all-digital phase-processing circuitry designed by the NASA Omega staff of the Ohio University Avionics Engineering Center. A digital phase-lock loop (DPLL) forms the heart of the Omega navigation receiver prototype. Through the DPLL, the phase of the 10.2 KHz Omega signal is estimated when the true signal phase is contaminated with noise. The study has provided a convenient means of evaluating loop performance in a variety of noise environments, and has proved to be a useful tool for evaluating design changes.

The DPLL uses a frequency synthesizer as the reference oscillator. The synthesizer is composed of synchronous rate multipliers (SRM's) driven by a temperature-compensated crystal oscillator (TCXO). Use of the SRM's in this application introduces phase jitter which degrades system performance. Simulation of the frequency synthesizer is discussed in Part III of this paper.

The goals of the simulation were fourfold:

A. To analyze the circuits on a bit-by-bit level in order to evaluate the overall design;

B. To see easily the effects of proposed design changes prior to actual breadboarding;

C. To determine the optimum integration time for the DPLL in an environment typical of general aviation conditions; and

D. To quantify the phase error introduced by the SRM synthesizer and examine its effect on the system.

II THE DIGITAL PHASE-LOCK LOOP

A. Introduction. DPLL's have been the subject of intense study in the past few years (4,6,7). Application of phase-lock techniques to digital systems has led to loop filters of first, second, and higher orders. The DPLL described in this section is a first-order loop which stores information about the relative phase difference between the local oscillator and the input signal.

B. Description of Circuit. The DPLL follows closely the ideas of J.M. Clark(1), first presented in 1968. The DPLL presently in use at Ohio University is essentially a highly adapted version of his digital phase tracking filter. A detailed discussion of the loop circuit operation is given by Chamberlin(2).

The circuit employs a 6-bit counter which cycles once every 64 pulses of the 652.8 KHz clock (see Figure 1). This counter therefore cycles at a rate of 10.2 KHz (=652.8 KHz ÷ 64). The number in the counter is compared continuously to the six most significant bits in the up-down counter, which is in turn fed by the phase detector. During the brief period when the numbers in the two counters are bit-wise identical, the output of the comparator goes to the logical 1 state. Although the duration of this pulse may vary, the uni-directional counter generally advances at a much faster rate, indicating a pulse width of 1/652.8 KHz, or approximately 1.5 μsec. This pulse is used to trigger a monostable, creating a window whenever the contents of the two registers are identical.

When viewed through this window, the zero crossings of the incoming signal are seen to be either leading or lagging in relative phase. When a lag is indicated, the up-down counter receives a down-count command, and the next window is created earlier in time. In this way the edge of the window gating function is brought more and more closely into coincidence with the zero crossings of the true Omega signal. Similarly, when a phase lead is indicated, count-up commands cause the window to be created later in time.

When lock is obtained, the six most-significant bits in the bi-directional counter represent the relative phase difference between the input frequency and the local oscillator (LO). The LO can be considered as a 10.2 KHz clock, whose zero crossings coincide in time with the zero-count of the 6-bit counter.

C. The DPLL as a Cross-Correlation Device. In a cross-correlation type receiver scheme, two signals are combined to give a d.c. level by first delaying one signal in time, multiplying their resulting signals, and integrating or lowpass filtering (see Figure 2). To see how the DPLL performs this same function in an all-digital sense, it is first necessary to view the circuit operation under no-noise conditions. Under these conditions there is no phase shift due to noise perturbations, nor is there a Doppler shift. The only phase error is then due to initial phase difference at turn-on.

Assume that the initial phase difference is such that a count-up command is given to the up-down counter. If N bits are being integrated, there will be a delay of 2^N up-count commands before the comparator logic "sees" a difference in time between successive occurrences of identical numbers in the two counters. The time delay in such a case is $(2^N/10.2 \times 10^3)$ seconds. This integration performs the low-pass filtering operation of h(t) in Figure 2.

The output of the comparator, $\hat{\Theta}$, is the estimate of the Omega signal phase, Θ_{Ω}. When lock is obtained, $\hat{\Theta} = \Theta_{\Omega}$, and the binary number contained

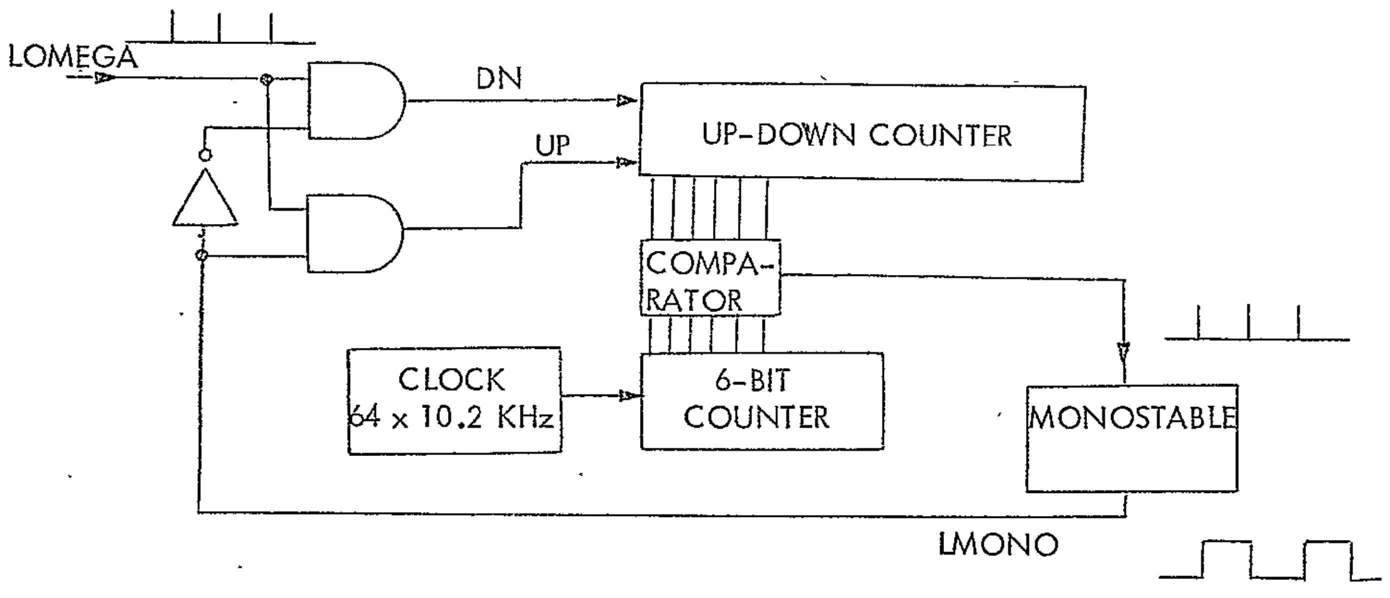


Figure 1. The DPLL Circuit.

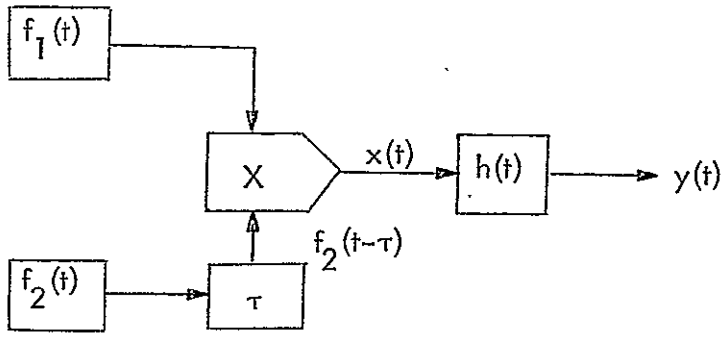


Figure 2. A Generalized Cross-Correlation Device.

in the six most significant bits of the up-down counter is the phase difference, analogous to the d.c.level obtained in an analog cross-correlation device.

D. Description of the Simulation. The digital nature of the circuit indicated the use of the digital computer as the most convenient simulation tool. Since the circuit was already built and was, in fact, yielding phase information which compared favorably with the Tracor series 599R receiver used for control-group data collection purposes, the circuit model was wholly deterministic.

The generation of zero-crossing information for the Omega signal in noise, however, was necessarily stochastic in nature. A complete description of the signal-generating procedures is given in Part II E.

CORDET is based on the technique of periodic scanning; that is, each portion of the DPLL is observed and updated with each clock pulse, significant information about the state of the system is recorded, and the process is repeated. In the case of CORDET, the significant information includes the states of all logic circuit elements, including phase difference information referenced to a local oscillator. The phase information is, of course, of primary importance in an Omega navigation receiver. The ability to compare the DPLL output with the true phase of the simulated Omega signal, and to provide graphical output, are advantages of a computer simulation over a laboratory hardware experiment. A user's technical description is provided in Appendix A.

E. Signal Input Simulation. Since the receiver front end provides a 15-Hz bandwidth narrow-band filter, the derived phase information can be considered to result from a noise-plus-signal input to a narrow-band system.

The composite input signal can be expressed as

$$g(t) = f(t) + A \cos(\omega_m t + \psi) \tag{1}$$

where ω_m is the center frequency of the filter, and f(t) is the narrow-band noise process

$$f(t) = E(t) \cos(\omega_m t + \phi(t)). \tag{2}$$

It can be shown(3) that the amplitude process E(t) is Rayleigh distributed, while the phase process $\phi(t)$ is uniformly distrubuted on the interval $(0, 2\pi)$.

Now, f(t) can be written in the form defining its quadrature components,

$$f(t) = N_1(t) \cos \omega_m t + N_2(t) \sin \omega_m t \tag{3}$$

where $N_1(t)$ and $N_2(t)$ are independent Gaussian processes(3).

From equation 1, g(t) may be expressed as

$$g(t) = f(t) + A \cos\psi \cos \omega_m t + A \sin\psi \sin \omega_m t . \tag{4}$$

Substituting for f(t), and using

$$\cos(a + b) = \cos a \cos b - \sin a \sin b,$$

we have

$$g(t) = (N_1(t) + A \cos\psi) \cos\omega_m t + (N_2(t) + A \sin\psi) \sin\omega_m t. \tag{5}$$

Equation 5 may be put in the form

$$g(t) = R(t) \cos(\omega_m t + \Theta(t)) \tag{6}$$

where

$$R(t) = \left\{ \left[N_1(t) + A \cos\psi\right]^2 + \left[N_2(t) + A \sin\psi\right]^2 \right\}^{\frac{1}{2}} \tag{7}$$

and

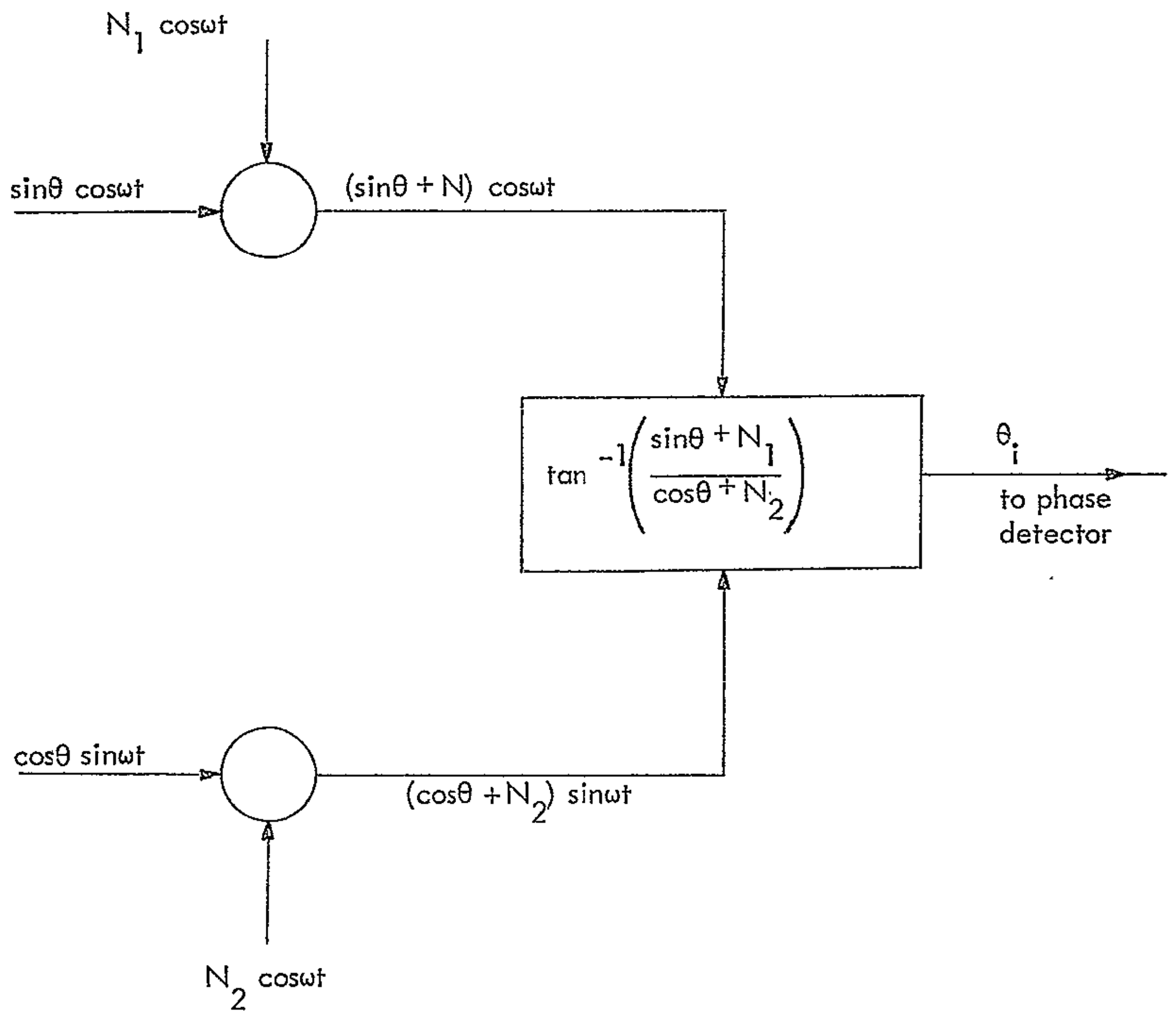
$$\Theta(t) = \arctan \frac{N_2(t) + A \sin\psi}{N_1(t) + A \cos\psi} \tag{8}$$

Equation 8 is therefore the input phase representation to the phase detector, where $N_1(t)$ and $N_2(t)$ are statistically-independent Gaussian white noise processes with zero mean and unit variance, and A is the normalized signal amplitude. In general, ψ can be a function of time to represent the Doppler shift in the dynamic case. At speeds less than Mach I, however, the Doppler shift is well below the 5.06° (1/64 of a 10.2 KHz cycle) quantizing interval of the DPLL, and so ψ can be considered constant over the sampling interval.

A block diagram of the noise input phase simulation is given in Figure 3.

F. <u>Conclusions</u>. In determining the optimal integration time of the first order loop, consider the case of a small aircraft in a noisy environment. As the time-multiplexed Omega transmission sequence progresses, the aircraft will change position between successive transmitting time intervals of the 10.2 KHz frequency. For an aircraft flying a course coincident with a station pair base-line (worst case), the change in position will amount io 12° of 10.2 KHz phase at a speed of 200 mph, conservative for private aircraft. For this reason, step phase inputs of 12° were taken both in a noise environment of 10 decibels SNR and in the no-noise case.

To determine optimal integration time, phase error at the end of the 625 msec. sampling interval is plotted vs. integration time in Figures 4 and 5. As could be expected, the best integration time in noise is one which is as long as possible without leaving a residual phase error at the end of a sampling interval in the no-noise case. For aircraft at 200 mph, 0.1 sec., or 10 bits of integration proves to be optimal.


$N_1 \cos\omega t$
$\sin\theta \cos\omega t$
$(\sin\theta + N) \cos\omega t$
$\tan^{-1}\left(\frac{\sin\theta + N_1}{\cos\theta + N_2}\right)$
θ_i
to phase detector
$\cos\theta \sin\omega t$
$(\cos\theta + N_2) \sin\omega t$
$N_2 \cos\omega t$

N_1 and N_2 are statistically-independent Gaussian white noise processes with zero means and unit variances.

Figure 3. Block Diagram Simulation of Narrow-Band Noise.

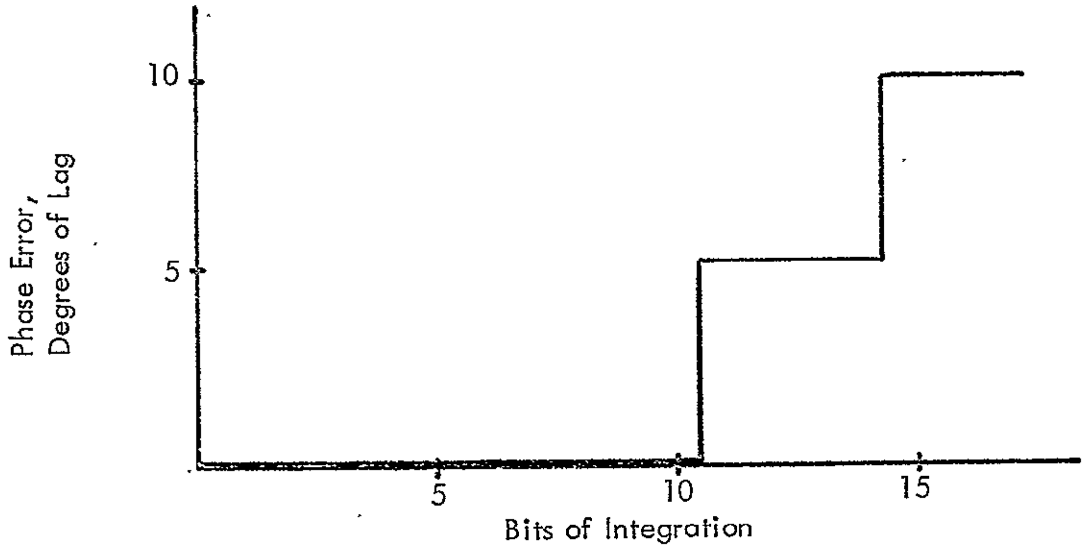


Figure 4. Phase Error at End of Sampling Interval vs. Bits of Integration, No-Noise Case.

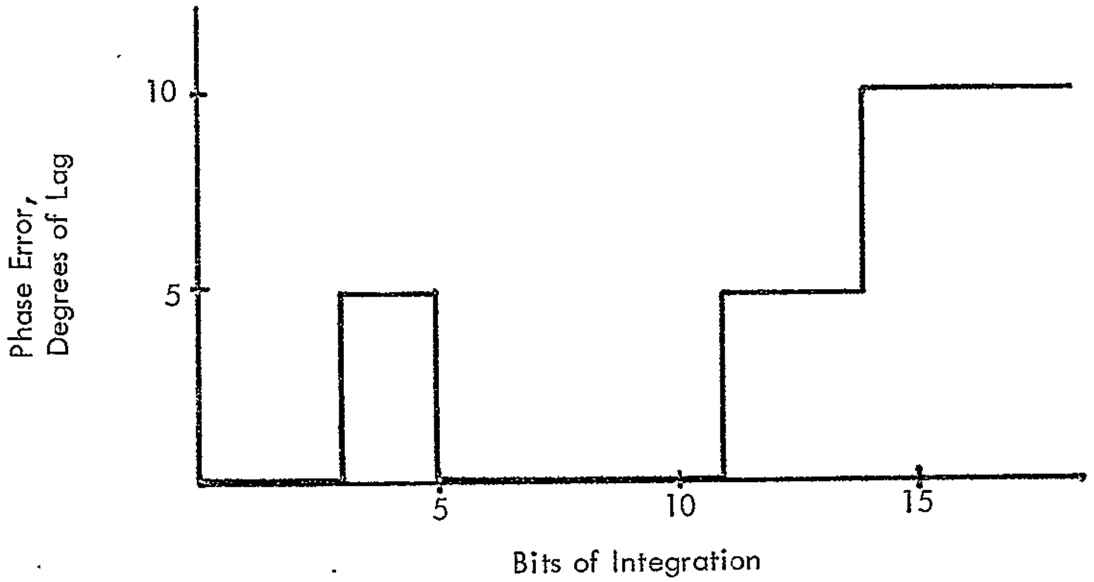


Figure 5. Typical Plot of Phase Error at End of Sampling Interval vs. Bits of Integration, 10 Decibel SNR.

III SRM FREQUENCY SYNTHESIZER

A. Introduction. Many applications in communications and navigation require synchronous signal detection using a local oscillator with high stability, and at non-standard frequencies. For example, most Omega navigation receivers operating in hyperbolic mode require that a local oscillator (LO) at some integral multiple frequency of 40.8 KHz be used for detection. The 40.8 KHz frequency is the least common multiple of 10.2 KHz and 13.6 KHz (i.e., 4 x 10.2 and 3 x 13.6). These are two of the three frequencies broadcast by the Omega system, the other being 11.333 KHz.

The Omega receiver prototype under development at Ohio University uses a LO frequency of 2.6112 MHz (64 x 40.8 KHz) (8). It is required that the LO have a long-term stability of at least one part in 10^7. To meet this requirement the LO must be of the temperature-compensated crystal oscillator type (TCXO). Since 2.6112 MHz is not a standard-frequency crystal, and special cutting of the crystal would be expensive, it was decided to synthesize the required frequency from a 5 MHz TCXO standard. The synthesized 2.6112 MHz frequency is then divided by 4 and the resulting 652.8 KHz wave (64 x 10.2 KHz) is used as the reference in the digital phase-lock loop described in Part II A.

Use of the 5 MHz TCXO reduces the problem of long-term drift in the LO, but a new short-term phase jitter is introduced by the frequency synthesizer. This phase jitter, seen as additive noise when referred to the input signal off the air, must first be quantified before system degradation and design improvements can be evaluated.

B. The SRM Synthesizer. Rate multipliers have been in use in discrete form for several years [9], and recently (1972) programmable versions have become available in integrated-circuit form[13]. Rate multipliers are essentially counters which count modulo-N. When supplied with appropriate logic circuitry, the counter will gate clock pulses in a predetermined sequence, effectively multiplying the input clock rate by a factor of M/N, where M is determined by the programming circuitry.

Literature concerning synchronous rate multipliers (SRM's) appears to be scarce, although the design employed by the Avionics Research Center staff at Ohio was inspired by references 9, 10, and 11. The synthesizer utilizes the SN 74167 Decimal Rate Multiplier (DRM) manufactured by Texas Instruments. Each DRM multiplies the input frequency by M/10, where M can be programmed to any integer from 0 through 10.

A block diagram of the frequency synthesizer is shown in Figure 6. It consists of a 5 MHz TCXO followed by a chain of DRM's. The DRM chain is programmed in binary to multiply the input rate by .52224. That is, whenever 10^5 clock pulses enter the chain, 52,224 pulses are produced at the OR-gate output. To be able to calculate phase jitter in the output, it is clear that the sequence of pulses produced by each DRM must first be known. A FORTRAN computer program was written to determine the DRM outputs.

C. DRM Simulation. Each DRM is composed of four T-type flip-flops and accompanying control logic gates (see Figure 7). Using the truth table associated with each of the logic elements, a program was written for the IBM System 360 computer using FORTRAN logical variables. The program, which

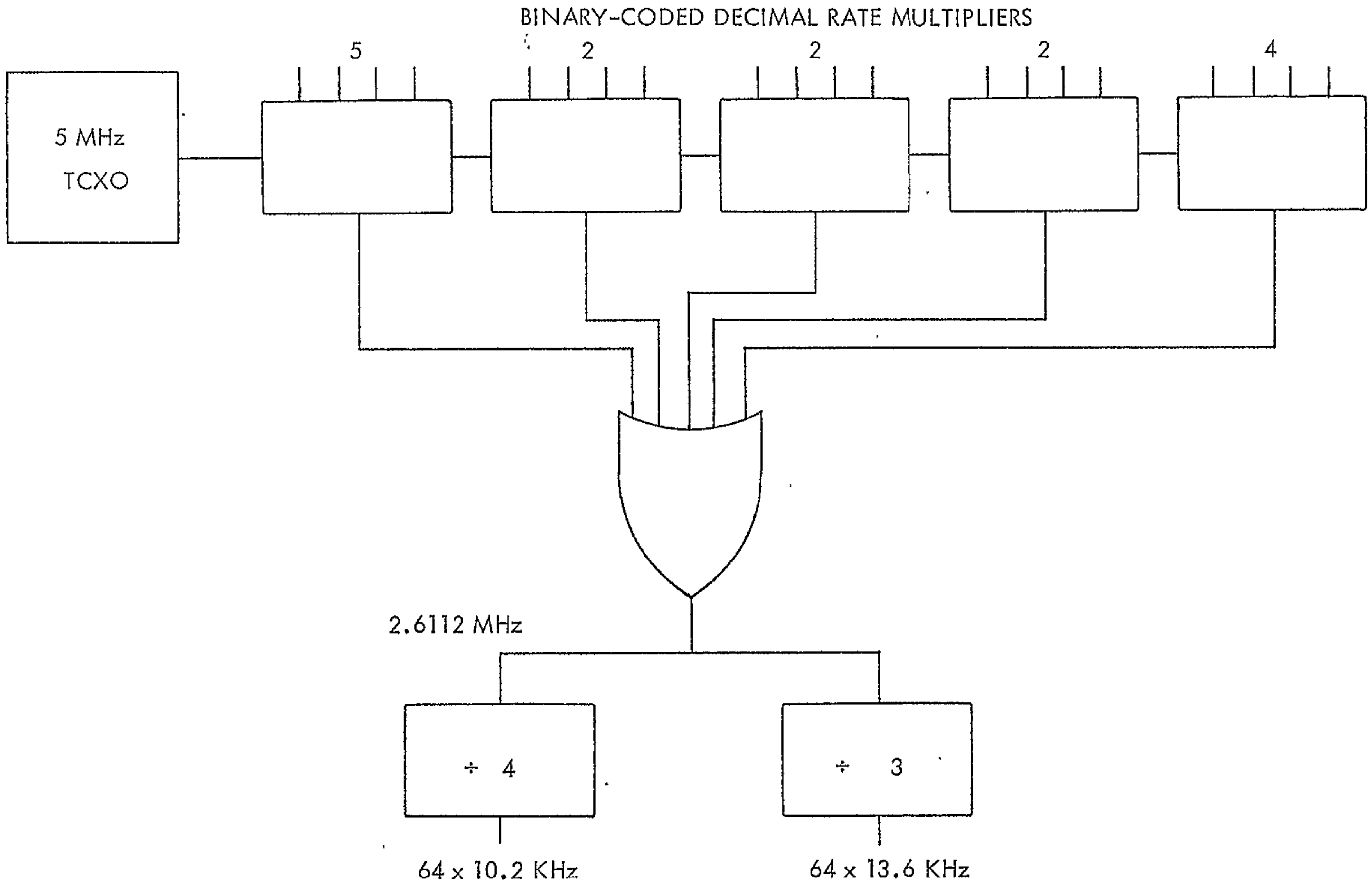


Figure 6. DRM Frequency Synthesizer.

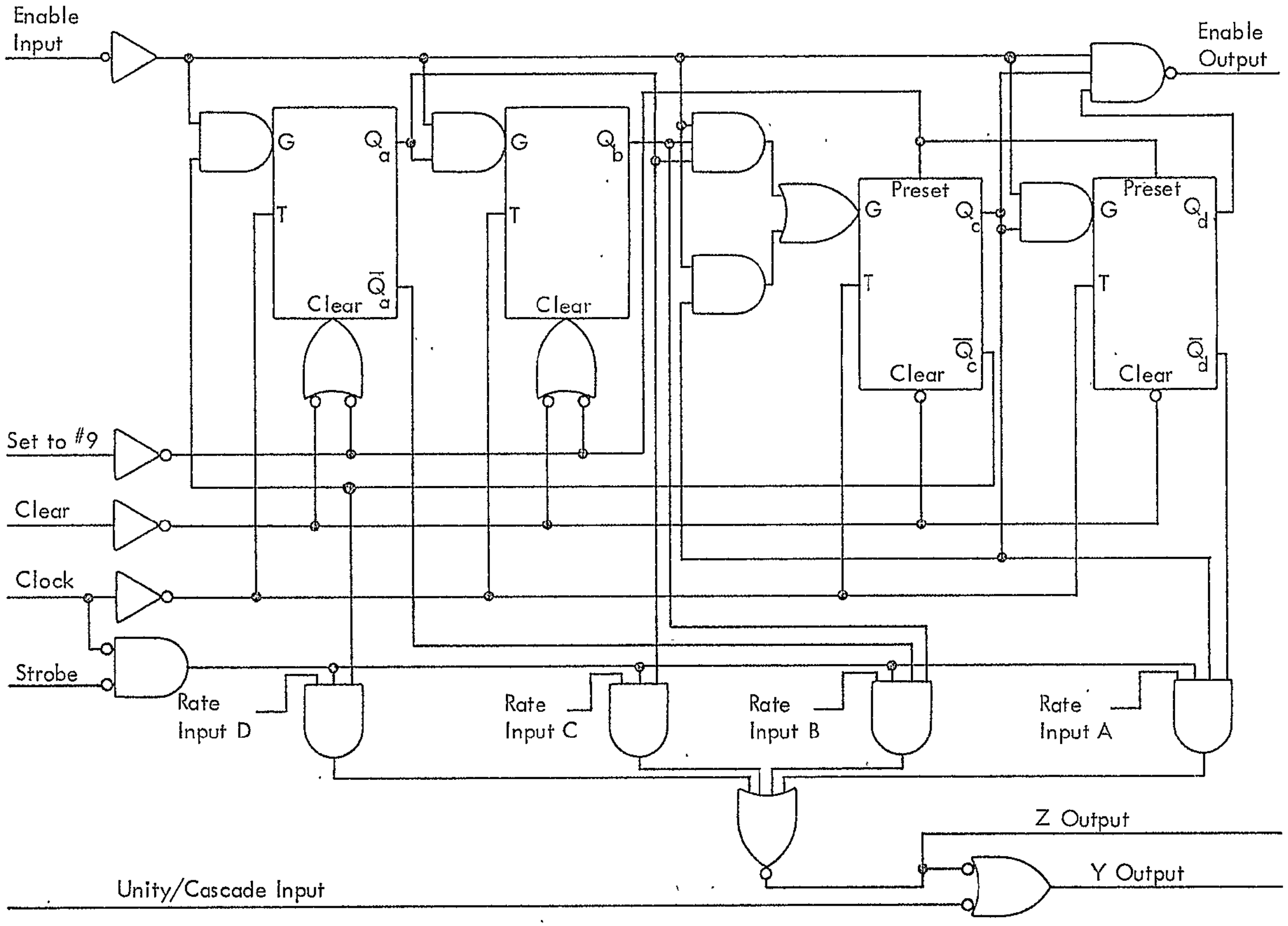


Figure 7. Functional Block Diagram of SN 54167/SN 74167 Synchronous Decade Rate Multiplier.

takes 10 seconds to execute, is listed in Appendix B.

The results of the computer model are summarized in Figure 8. From the timing diagrams it is evident that programming the DRM inputs serves only to delete certain of the clock pulses. When the DRM's are cascaded in order to multiply by a longer decimal fraction, pulses are added whenever the ENABLE line to lower-order stages goes LOW. As seen from the ENABLE output in Figure 8, this can occur only for a period of one clock pulse in every ten.

Note that the outputs determined by the simulation are identical to those produced by the circuit in the laboratory. Programmings of 5, 2, and 4 are shown in Figure 9. Results for all programmed inputs of Figure 8 were in agreement with laboratory observations.

It is also evident that the first stage in the DRM chain will determine the worst case for phase jitter. For example, when the DRM is programmed to multiply by .5, the output will be the train of pulses shown in line 5 of Figure 8. The leading edge of the second pulse in this train will be two clock periods from the corresponding edge of the first pulse, the third pulse will be only one period from the second, the fourth again two periods from the third, and so on. Note that the time difference between the last of the five pulses and the first of the next train will be three clock periods. During this part of the pulse train, the ENABLE gate to succeeding stages goes LOW, and a pulse may be inserted by one of the following DRM's.

D. <u>Cascaded DRM Simulation.</u> The computer was programmed to keep track of the number of oscillator pulses which occur between output edges of a series of cascaded DRM's. Summing the number of occurrences of each gives

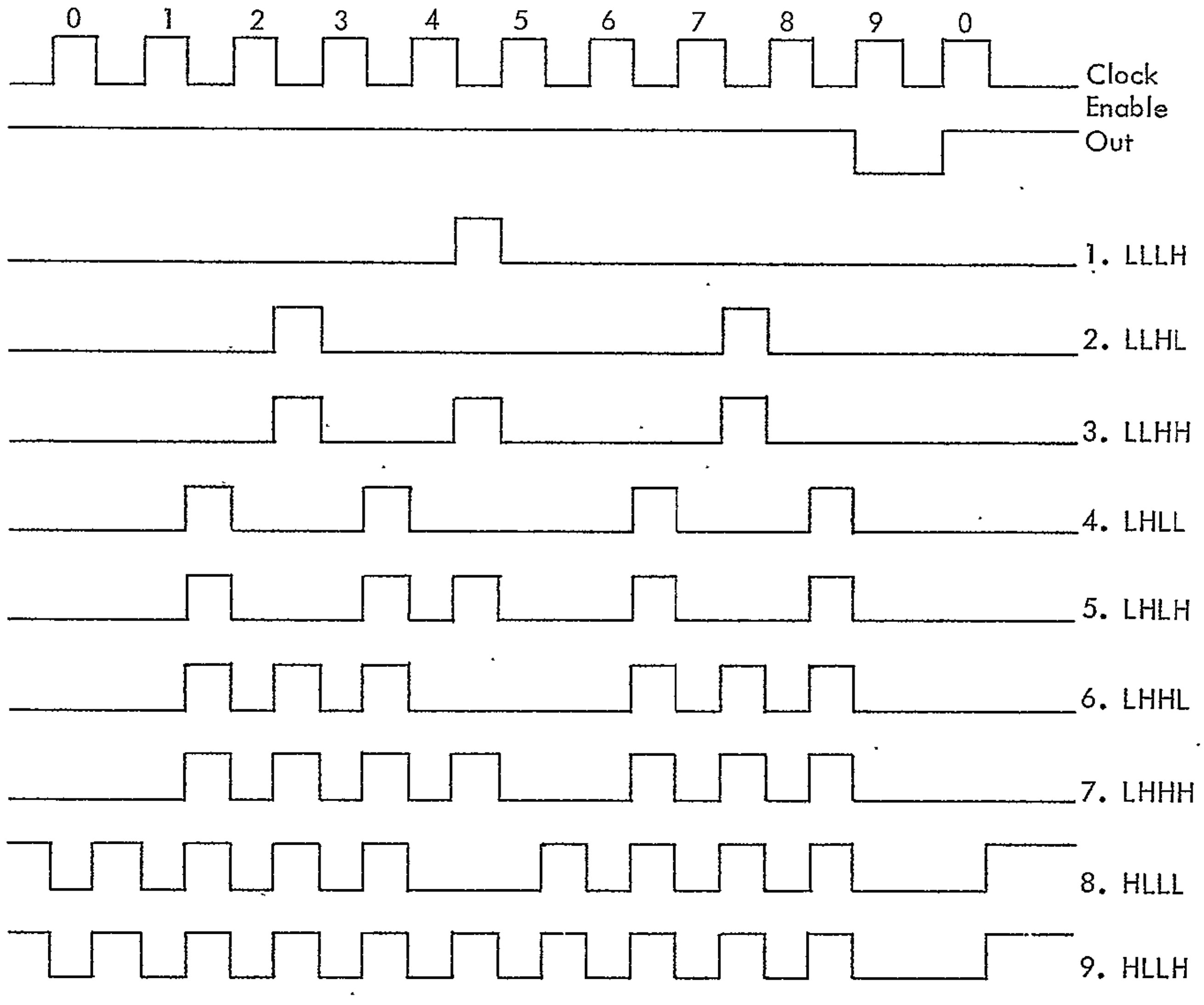
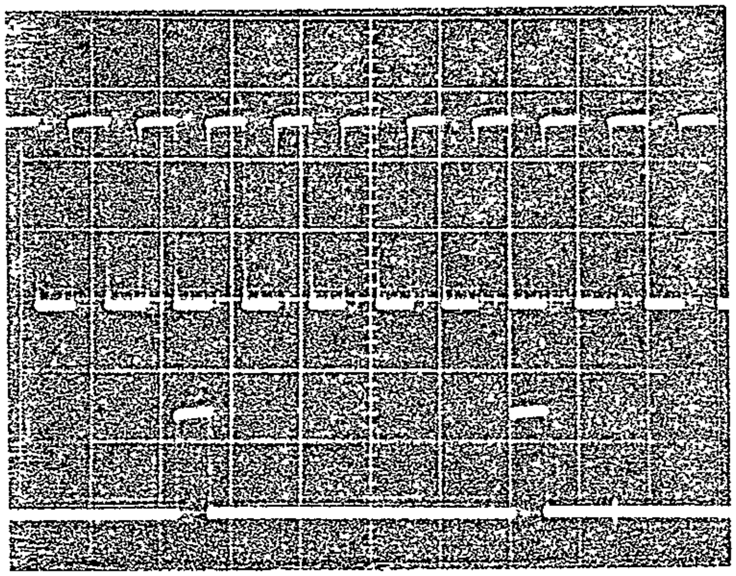
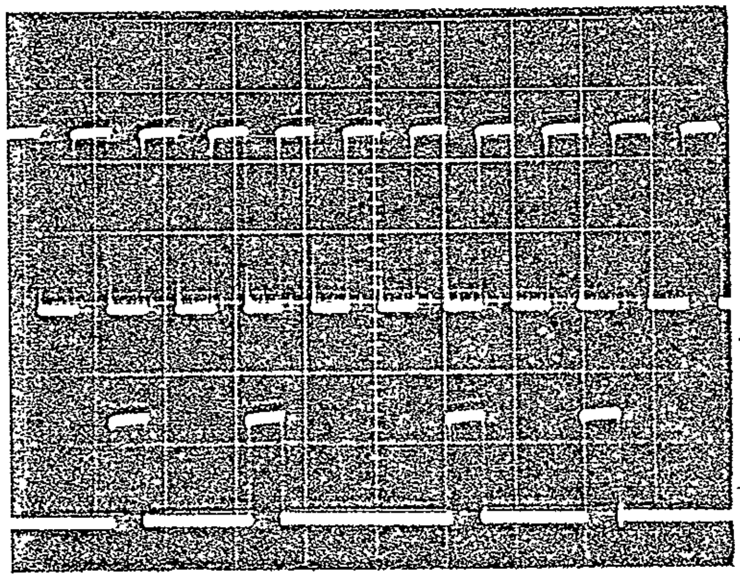


Figure 8. Output Waveforms from the SN 74167 DRM. Program inputs are shown on the right.





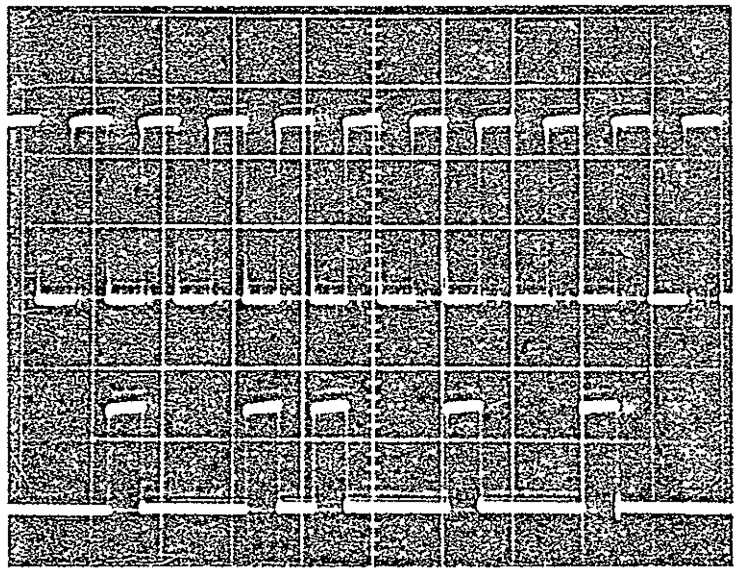


Figure 9. Observed Outputs of the SN 74167 DRM. Compare with lines 2, 4, and 5 of Figure 3. The clock input is shown at the top of each photograph.

information which can be translated into a mean-squared-error figure for phase jitter. The program SYN, which simulates the DRM chain, when programmed to multiply by .52224, is given in Appendix C.

Since the first DRM will determine worst-case output in terms of phase jitter, it is not surprising that in this case there are three possible time differences between output pulses. In terms of the oscillator period, the time between outputs is one, two or three periods. These results are summarized in Figure 10.

E. Realization of Omega Frequencies.

1. Post-division by 4. In order to obtain a frequency of 652.8 KHz from a 2.6112 MHz source, division by 4 is required. This division can easily be performed by a pair of flip-flops. Using the SYN program with post-division by four (see Appendix C), it was determined that four distinct time differences between output pulse edges are produced, as seen in Figure 11. Calculating the weighted mean of these pulse occurrences in time gives:

$$\bar{x} = \frac{\sum \text{(output pulses)} \times \text{(number of clock pulses between output edges)}}{\sum \text{output pulses}}$$

$$= \frac{(4448 \times 6) + (16672 \times 7) + (23328 \times 8) + (7776 \times 9)}{52224}$$

$$= 7.65931 \text{ clock pulses.}$$

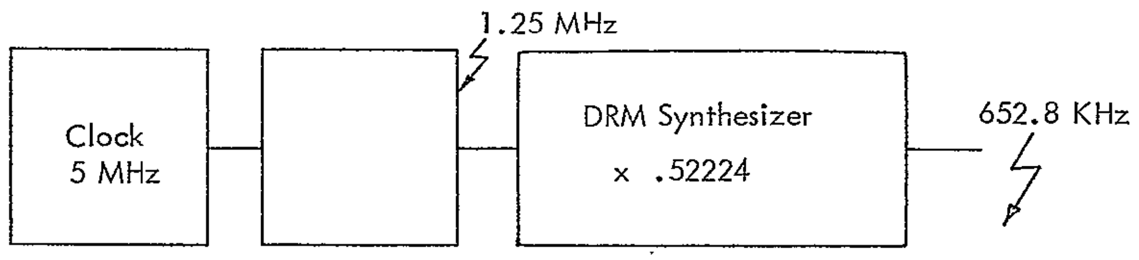
As a check,

$$\frac{f_{in} \text{ (clock pulses/second)}}{\bar{x} \text{ (clock pulses)}} = f_{out}$$

or,

$$\frac{5 \times 10^6}{7.65931} = 652.8 \text{ KHz} \quad (=10.2 \text{ KHz} \times 64)$$

as expected.



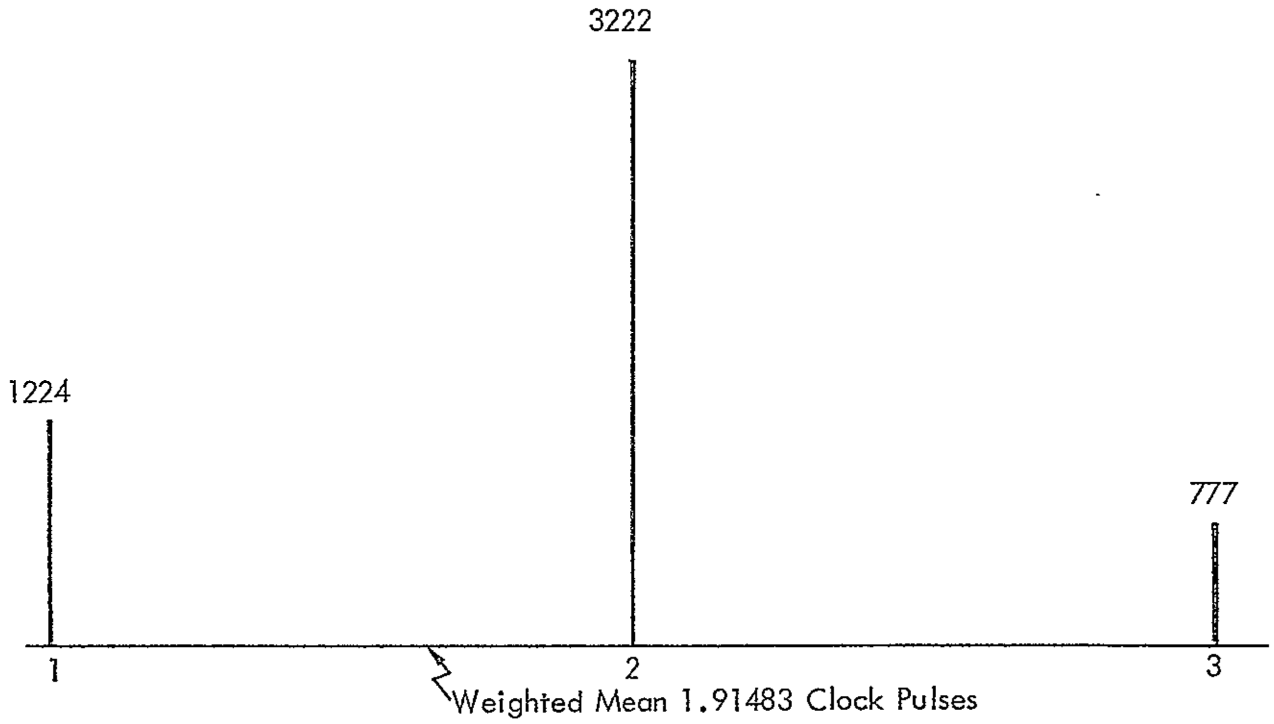


Figure 10. Output of the DRM Synthesizer with No Post-Division Circuitry, In Terms of Input Clock Pulses. Baseline numbers are number of input clock pulses between output pulse edges. The lines represent frequency of occurrence of output pulses.

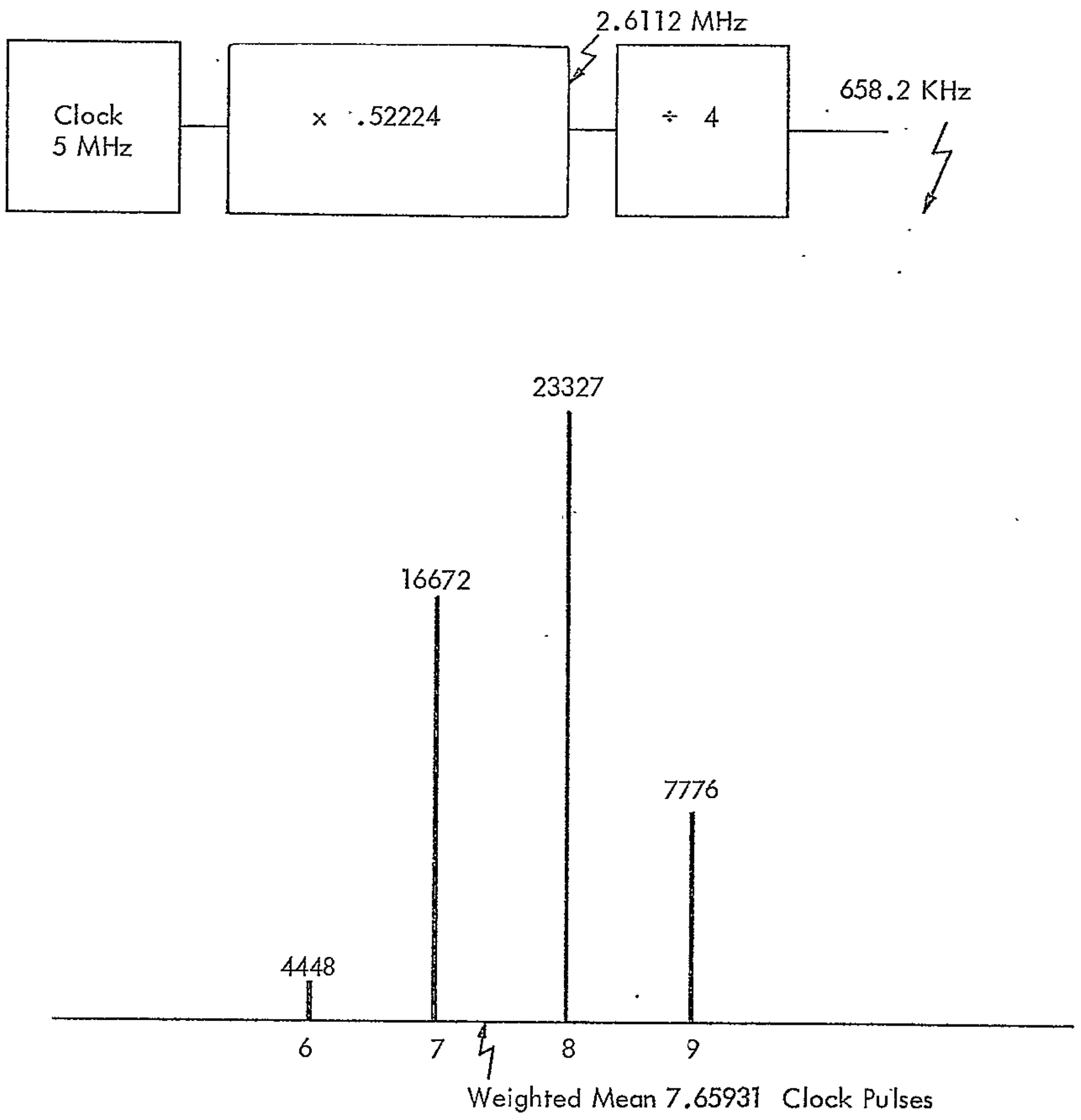


Figure 11. Output of Synthesizer with Post-Division by 4.

The weighted mean can now be used to calculate mean-squared error.

$$\bar{\epsilon}^2 = \frac{\sum \begin{pmatrix}\text{time difference between pulse} \\ \text{occurrences and weighted mean}\end{pmatrix}^2 \times \text{(number of pulses)}}{\sum \text{output pulses}}$$

or,

$$\begin{aligned}\bar{\epsilon}^2 &= \Big\{[(7.65931 - 7.00000)\tau]^2 \times 16672 \\ &\quad + [(7.65931 - 6.00000)\tau]^2 \times 4448 \\ &\quad + [(8.00000 - 7.65931)\tau]^2 \times 23328 \\ &\quad + [(9.00000 - 7.65931)\tau]^2 \times 7776 \Big\} \div 52224 \\ &= .69277\tau^2\end{aligned}$$

where:

$$\tau = \frac{1}{f_{in}} \; :$$

For a 5 MHz clock,

$$\bar{\epsilon}^2 = .69277\left[\frac{1}{5 \times 10^6}\right]^2 = 2.77 \times 10^{-14} \text{ sec.}^2$$

2. <u>Post-division by 8.</u> Since the mean-squared error figure is a function of the clock period, it is a reasonable hypothesis to suppose that reducing the period will result in reducing the mean-squared error. To test this hypothesis, the SYN program was modified to perform post-division by 8 (see Appendix C), and a 10 MHz clock was assumed. The resulting output, illustrated in Figure 12, shows five possible positions for the resulting output pulse in terms of the input clock period. The weighted mean of the output is 15.31857 clock pulses, which yields an average frequency of

$$\frac{10 \text{ MHz}}{15.31857} = 652.8 \text{ KHz,}$$

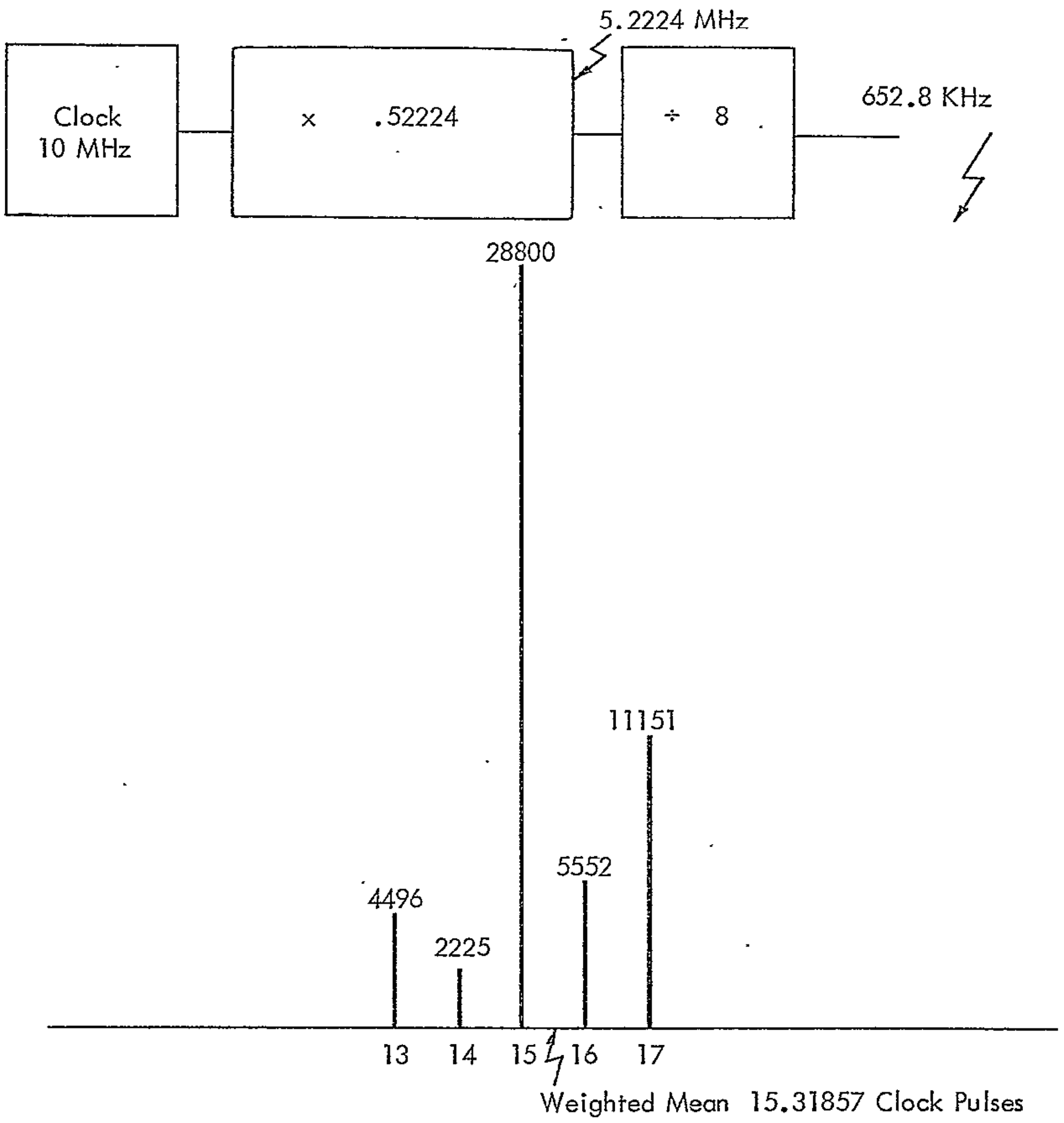


Figure 12. Output of DRM Synthesizer with Post-Division by 8.

as expected. Mean-squared error in this case is 3.12×10^{-15} sec.^{2}, an improvement of an order of magnitude over the divide-by-four case.

3. Pre-division by 4. As a point of reference, it is instructive to look at the output produced by the DRM chain when driven by a 1.25 MHz clock, or a 5 MHz TCXO divided by 4 (See Figure 10). The weighted mean in this case is 1.91483 clock pulses, resulting in a mean-squared error of 2.41×10^{-13} sec.^{2}

F. Conclusions. In the case of the DRM synthesizer used in the Omega receiver prototype at Ohio University, this study has shown that significant reduction in phase jitter, and a corresponding reduction in the error introduced into the system, is possible by doubling the TCXO frequency at the input and dividing by 2 at the output. These results are summarized in Table 1.

	f_{in}	Rate Multiplication	$\bar{\epsilon}^2$	Percent Improvement
I	5 MHz	$\div 4 \times .52224$	2.41×10^{-13} sec^2	------
II	5 MHz	$\times .52224 \div 4$	2.77×10^{-14} sec^2	I-II 88.5%
III	10 MHz	$\times .52224 \div 8$	3.12×10^{-15} sec^2	I-III 98.7% II-III 88.7%

$$\text{(Percent Improvement} = \frac{\bar{\epsilon}_2^{\,2} - \bar{\epsilon}_1^{\,2}}{\bar{\epsilon}_1^{\,2}} \times 100\%)$$

Table 1. Phase Jitter Produced by Three DRM Synthesizer Configurations.

Dividing the frequency of the DRM synthesizer output by powers of 2 results in successively larger numbers of possible positions for the output pulse edges in terms of TCXO clock periods. It is difficult, therefore, to prove a general theorem relating circuit configuration to resulting phase jitter. However, for a given system, the following rules of thumb for programming the DRM's can be observed:

1. Remember that the first stage determines worst-case phase jitter. Therefore, pick as high a programming number as possible for the first stage.
2. Successive doubling of the input frequency and corresponding division by two at the output of the DRM chain has been observed to reduce phase jitter.

In an application where reduction of phase jitter is critical, simulation of the system is not difficult using the techniques described in this paper.

G. Recommendations. Frequency synthesizers utilizing rate multipliers hold great promise for use in digital systems where the inherent phase jitter can be tolerated. Applications in digital phase- and frequency-lock loops (16) appear to be especially useful. When programmable versions like the SN 74167 are employed, added versatility is possible. For example, if it is desired to navigate with the 13.6 KHz frequency in the Omega system, a simple programming change in the synthesizer will yield the proper clock rate for driving the DPLL. Noting that the ratio of 10.2 KHz to 13.6 KHz is 0.75, one could program the rate multipliers to 52224/0.75 = 69632 and arrive at the desired 64 x 13.6 KHz rate. (The 11.333 KHz frequency, however, is not so easily accessable.)

Care must be taken when using the synthesized frequencies derived from SRM chains in analog applications. For example, since the synthesizer described here is periodic in 10^5 input clock pulses (or 10 msec. for a 10 MHz clock), only harmonics of 100 Hz will be present in the output. Changing the phase jitter will alter the relative power contained in these harmonics.

IV GENERAL CONCLUSIONS

A. Phase Jitter as Noise. The graphic illustrations of phase jitter shown in Figures 10, 11, and 12 can be interpreted as probability density functions (pdf's) when the height of each line, i.e., the number of pulse edges occurring at each position, is divided by the total number of pulses. The probability that a pulse edge will occur within the limits of the discrete envelope then will be unity. Although the phase process is not truly random at the output of the synthesizer, such a density function will describe the position of the output pulses when observed at random instants of time.

In the Omega application described in this paper, this density function for phase noise can be referred to input since the phase detector used in the DPLL sees only relative phase difference between the received signal and the reference oscillator. Theoretically it then would be possible to obtain a signal-to-noise figure with phase jitter from the frequency synthesizer treated as added noise on the incoming Omega signal. Resulting system degradation could then be translated into a lower signal-to-noise ratio (SNR) in the signal off-the-air. This approach toward analysis has several shortcomings, as discussed below.

Since the signal off-the-air is band-pass filtered, we can again apply the theory of noise in narrowband systems with added sinusoids to describe the probability density function for phase. In particular, when the noise input is white, it can be shown(12) that:

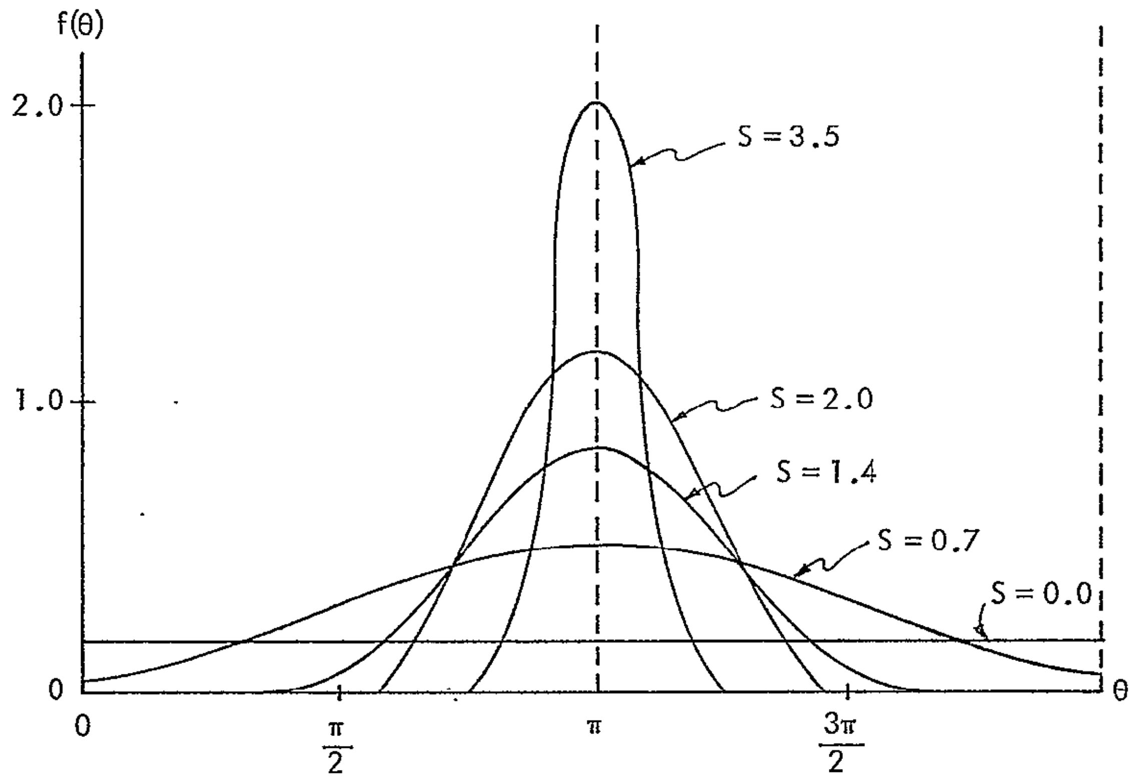


Figure 13. Phase pdf for sinusoid-plus-white noise in a narrowband system.

$$q(\theta) = \frac{e^{-s^2}}{2\pi} + \frac{1}{2}\sqrt{\frac{s^2}{\pi}} \cos\theta \; e^{-s^2 \sin^2\theta} \left[1 + \text{erf}(s \cos\theta)\right] \tag{9}$$

Where: $\text{erf}\, x = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2}\, dy$

s^2 = power signal-to-noise ratio (SNR).

This density function is sketched in Figure 13 for $\Theta = \pi$ and increasing values of s. Note that when the signal vanishes, the phase is uniformly distributed over the interval 0 to 2π. As the noise level approaches zero, $q(\theta)$ approaches the delta function at $\Theta = \pi$.

If one wants to speak of the contamination caused by synthesizer phase jitter in terms of a lowered apparent SNR, he must fit the empirically-obtained pdf to the pdf of equation (9). Several immediate difficulties preclude this approach.

First, the empirical phase pdf is discrete, not continuous. In the general case, even the contour of its envelope need not match any curve described by equation (9). Therefore, since equation (9) was derived for band-limited white noise (the usual case), we would assume that a phase pdf of the discrete form had been produced by some other noise source.

It is apparent that any signal-to-noise ratio which would result from relating equation (9) to one of the empirical pdf's would have a meaning different from the one familiar to the design engineer. Such a SNR would describe a synthetic pseudo-noise source with no counterpart in atmospheric channel noise. For this reason it would be more useful to formulate a figure-of-merit for a synthesizer with given empirical phase pdf. One would then attempt, through careful design, to optimize this figure of merit.

In digital applications of rate multipliers, it is desired that the empirically-determined pdf be distributed over as narrow a phase interval as possible. The MSE value of deviation from the mean, when used as a figure of merit, has the advantage of penalizing large phase excursions more than small ones. When there is no phase jitter, the MSE is zero and the density function becomes the delta function at the mean. Adopting the MSE as the gauge with which to measure phase jitter, the designer should attempt to minimize MSE.

B. Phase Jitter Relative to Omega Applications. How great an effect the MSE has in degrading the performance of a given system depends on the particular application. In the case of the digital phase lock loop application for which the synthesizer design was developed, it is clear that phase jitter will have no effect on the system while the Omega signal phase lies outside the interval embraced by the pdf of the synthesizer. Once the pdf overlaps the Omega signal zero crossings, however, the phase detector will begin to give erroneous outputs to the tracking filter. The effect of this phenomenon will be to increase the lockup time of the DPLL, making it desirable to minimize MSE in the synthesizer.

After the loop has attained lockup, the phase jitter will drive the loop out of lock if the pdf is not symmetrically centered about the lock point, or mean. For example, after lock is attained in the pdf of Figure 12, the position of Omega phase will coincide with the weighted mean. If we consider the pdf to be the result of a random process, then 68% of the time the clock pulses will arrive too soon at the phase detector, and 32% of the time they will arrive too late. On the average, then, the loop will be driven out of lock twice as often in one direction as in the other.

To completely determine the parameters of this process would require a dynamic mating of the SYN program and the CORDET simulation of Part II. However, the present study reveals that the error will be no more than the least significant bit in the bi-directional count register. This is true since the interval spanned by the synthesizer phase pdf is less than 1.53 μsec, which is the minimum time interval the phase can be advanced or retarded at the comparator output. The net effect of the phase jitter in the present system therefore, is to introduce a possible one-bit uncertainty.

This least significant bit represents 1/64 of a 10.2 KHz lane. Examination of Omega lane geometry in North America reveals that in no case is it necessary to use station pairs which produce lanes of more than 30 nm separation (15). An error of 1/64 lane is the worst-case equivalent of less than 0.5 nm. When the major emphasis is on producing a low-cost receiver, this error can be considered acceptable until such time as TCXO's are commonly available in frequencies compatable with Omega.

V ACKNOWLEDGEMENTS

The author gratefully acknowledges the assistance of NASA which funds research on Omega at Ohio University; Dr. Richard H. McFarland, Project Director at the Avionics Engineering Center; Mr. Ralph Burhans, Project Engineer; and Dr. Robert W. Lilley, Staff Consultant. Additional assistance was provided by Don Seyler, Student Intern; K. A. Chamberlin and J. E. Grover, Graduate Assistants; D. G. Herold, Staff Engineer; and Dr. J. E. Essman, Assistant Chairman, Department of Electrical Engineering at Ohio University.

VI REFERENCES

(1) Clark, John M., "Aircraft Navigation Using Omega", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-5, No. 5, September,1969.

(2) Chamberlin, Kent A., "The Memory-Aided Phase-Locked Loop," paper presented at the Second Omega Symposium, ION, Washington,D.C., November 7, 1974.

(3) Thomas, John B.,Statistical CommunicationTheory, New York, John Wiley & Sons, Inc., 1969.

(4) Weinberg, A. and B. Liu, "Discrete time Analyses of Nonuniform Sampling First- and Second-Order Digital Phase Lock Loops," IEEE Transactions on Communications, Vol. COM-27, No. 2, February, 1974.

(5) Garrett, P. H., "Optimum Adaptive Phase Estimation Receiver for One-Way Ranging Aircraft Navigation," Technical Report ECOM-0384-S, United States Army Electronics Command, Fort Monmouth, N.J., October, 1972.

(6) Garodnick, J., et al., "Response of an All-Digital Phase Locked Loop," IEEE Transactions on Communications, Vol. COM-22, No. 6, June, 1974.

(7) Gupta, S.C., "Phase-Locked Loops," Proceedings of the IEEE, Vol. 63, No. 2, February, 1975.

(8) Lilley, R.W., "Binary Processing Concepts for Omega Receivers," Proceedings of the Second Omega Symposium, ION, pp. 160-167, Washington, D.C., November, 1974.

(9) Micro Instrument Company, "Instruction Manual Model 1107 Omega Navigation Receiver," 12901 Crenshaw Boulevard, Hawthorne, CA 90250.

(10) Small, G.W., "Synthesis of the Musical Scale Using Noninteger Frequency Division," Journal of the Audio Engineering Society, Vol. 21, No. 4, May,1973.

(11) Small, G.W., "Rate-feedback Binary Counters in Musical Scale Generation," Journal of the Audio Engineering Society, Vol. 21, No.9, November,1973.

(12) Schwartz, Mischa, Information Transmission, Modulation and Noise, McGraw-Hill, 1959, New York, N.Y.

(13) The TTL Data Book, Texas Instruments, Inc., Market Communications Department, Dallas, TX 75222.

(14) Palkovic, R.A., "Computer Program CORDET", Technical Memorandum NASA 14, Avionics Engineering Center, Ohio University, Athens, Ohio 45701, November, 1974.

(15) Sectional Aeronautical Chart, National Ocean Survey, Distribution Division (C44), Riverdale, MD 20840.

(16) "MSI/TTL Integrated Circuits from TI", bulletin #CB-12, Texas Instruments, Inc., Dallas, TX 75222, 1972.

VII APPENDIX

A. COMPUTER PROGRAM CORDET.

1. General Description.

CORDET is a logic simulation of the all-digital phase-lock loop (DPLL) presently employed in the Omega receiver prototype.

Output provides graphic display of the timing diagrams of the circuit, plots of phase difference vs. time, and a histogram of phase difference between the DPLL output and the object-lock frequency.

Provisions have been made to vary the number of bits of integration, the signal-to-noise ratio of the input signal, and the initial phase difference between the DPLL output and the object-lock frequency.

2. Capabilities and Limitations.

a. The program assumes the presence of a local oscillator in the circuit with a frequency of 64 times the object-lock frequency.

b. Compile and load time is 15 seconds. Execute time is approximately 0.15 seconds per cycle of object-lock frequency specified (when only phase difference and histogram are plotted).

c. Approximately 21×10^3 bytes of storage are used.

d. Subroutines HISTG and RANDNR are called from the Ohio University computer center request-call library. RANDNR uses the multiplicative congruential method of pseudo-random number generation. HISTG is a plotting subroutine.

3. Technical Description.

Language:	FORTRAN IV
Number of arguments:	11
1. Data card 1:	NCYC, MNUMB, NNUMB, DCYC, DELPHI, NOISE, A

	Mode:	INTEGER (1,2,3,6,), REAL (4,5,7)
2.	Data card 2:	TD, PHA, HISTOG
	Mode:	INTEGER
3.	Data card 3:	TITLE
	Mode:	ALPHANUMERIC

Argument definitions:

NCYC	- is the number of object-frequency cycles to be considered.
MNUMB	- is the number of bits in the up-counter.
NNUMB	- is the total number of bits in the up-down counter.
DCYC	- is the duty cycle of the monostable.
DELPHI	- is the initial phase difference in degrees of lag (DPLL output lags input).
NOISE	- is <u>zero</u> for no-noise case, <u>1</u> otherwise.
A	- is voltage signal-to-noise ratio in rational number form (<u>not</u> db).
TD	- is <u>1</u> if timing diagram are desired, <u>0</u> otherwise.
PHA	- is <u>1</u> if phase difference vs. time plots are desired, <u>0</u> otherwise.
HISTOG	- is <u>1</u> if histogram is desired, <u>0</u> otherwise.
TITLE	- is title for histogram plot.

Figure A-1 represents the positive-logic implementation of the DPLL. CORDET performs one pass through a DO-loop for each pulse of the 64 x 10.2 KHz clock. New values for the logical variables indicated in the diagram are assigned on each pass. Design changes in the circuit will, of course, necessitate changes in the logical statements in CORDET's main program.

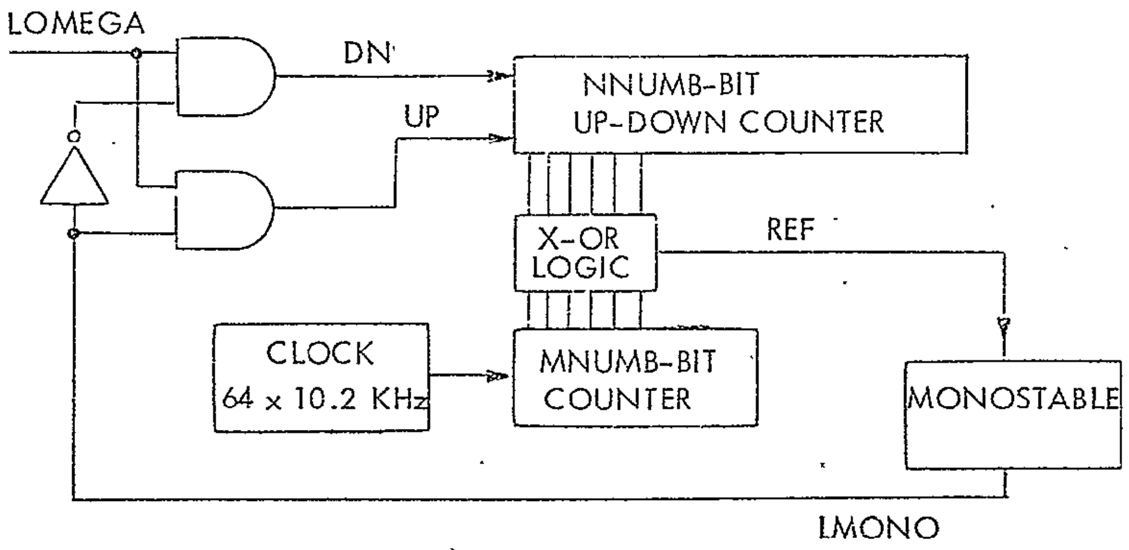
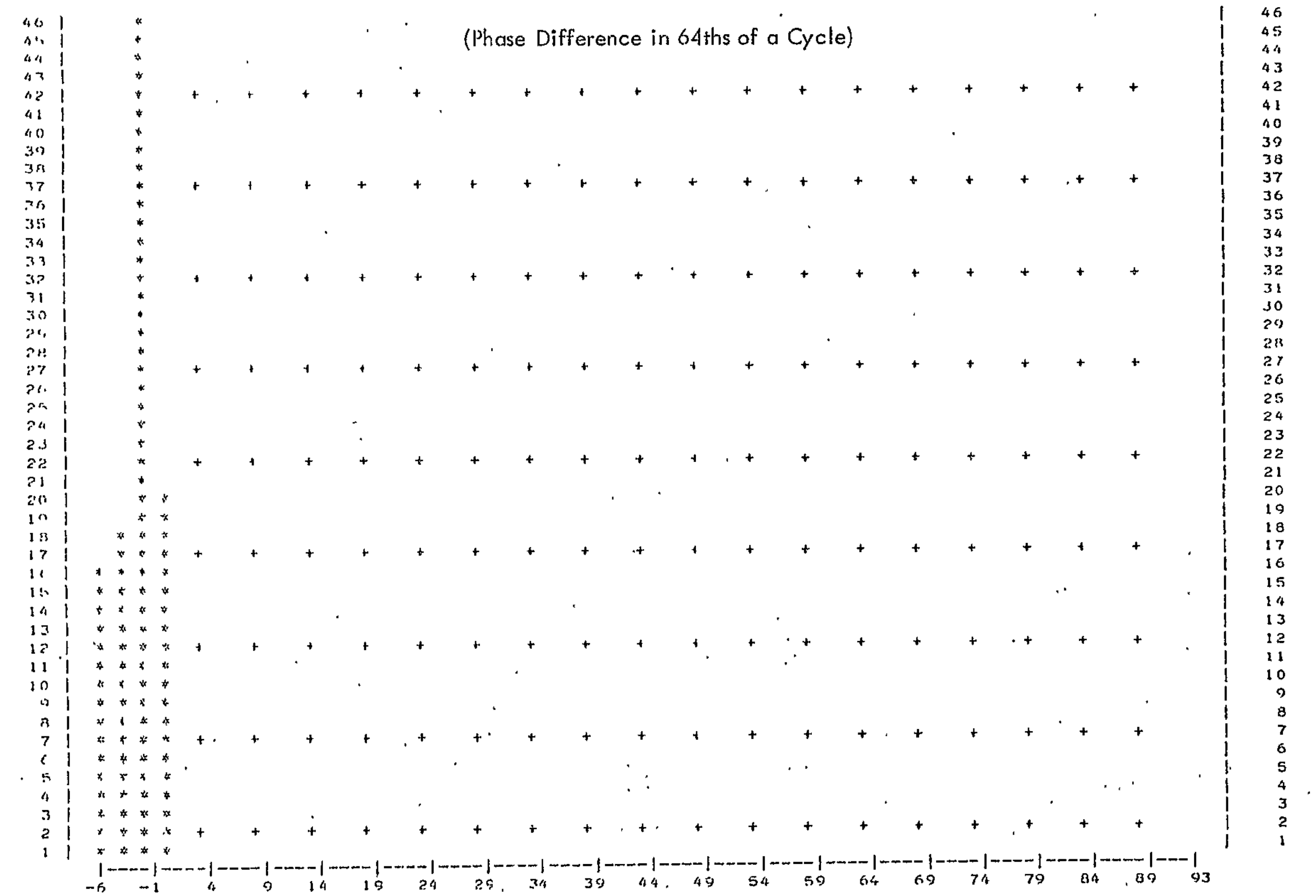


Figure A-1. Positive-Logic Implementation of the DPLL.

4. Program Listing and Sample Outputs.

```
                                                LABEL MAP
 LABEL      LOCATION         LABEL      LOCATION      LABEL      LOCATION       LABEL      LOCATION       LABEL      LOCATION
     1      0003AE               2      0003F4            3      000418             4      000434             5      00045C
     6      00046C               7      000488            8      0004A4             9      0004C0            10      0004DC
    11      0004F8              12      00053C
   TOTAL MEMORY REQUIREMENTS 000594 BYTES          PLOT1
      /DATA
MODULE     ADDR  LENGTH                 ENTRY POINTS (E), EXTERNAL (X) AND COMMON (C) REFERENCES
MAIN44#    0F000          E-MAIN44      0F000         X-IBCOM#               X-TRUN                   X-MONOS
                 00604    X-OMEGA1                    X-PLOT1                X-FIXPI#
TRUN#      0F608 00388    E-TRUN        0F608         X-FIXPI#
OMEGA1#    0F9C0 001E4    E-OMEGA1      0F9C0
MONOS#     0FBA8 00274    E-MONOS       0FBA8
PLOT1#     0FE20 00594    E-PLOT1       0FE20         X-IBCOM#

   MODULES LOADED FROM AUTO - CALL LIBRARY

BOAFIXPI 10388 000A8    X-IBCOM#                      E-FIXPI#    103C0
  MODULE SUCCESSFULLY LOADED --- 2F858 BYTES OF STORAGE REMAINING / EXECUTION BEGUN AT 0F000

   40     3    5       0.50      90.00       0.13


   I   N  NTRUN   M     REF   LOMEGA  POMEGA    UP      ON     LMONO   LMONO2  INDLOC   EX


   1   0     0    1     I      I       I       I       I       I       I       I       I
   2  31     0    2     I        I     I       I         I     I       I       I       I
   3  31     7    3     I      I       I       I       I       I       I       I       I
   4  31     7    4     I      I       I       I       I       I       I       I       I
   5  31     7    5     I      I       I       I       I       I       I       I       I
   6  31     7    6     I      I       I       I       I       I       I       I       I
   7  31     7    7       I    I       I       I       I         I       I     I       I
   8  31     7    0     I      I       I       I       I         I     I       I       I
   9  31     7    1     I      I       I       I       I         I     I       I       I
  10  32     7    2     I        I     I         I     I         I     I       I       I
  11  32     0    3     I      I       I       I       I       I       I       I       I
  12  32     0    4     I      I       I       I       I       I       I       I       I
  13  32     0    5     I      I       I       I       I       I       I       I       I
  14  32     0    6     I      I       I       I       I       I       I       I       I
  15  32     0    7     I      I       I       I       I       I       I       I       I
  16  32     0    0       I    I       I       I       I         I       I     I       I
  17  32     0    1     I      I       I       I       I         I     I       I       I
  18  33     0    2     I        I     I         I     I         I     I       I       I
  19  33     0    3     I      I       I       I       I         I     I       I       I
  20  33     0    4     I      I       I       I       I       I       I       I       I
  21  33     0    5     I      I       I       I       I       I       I       I       I
  22  33     0    6     I      I       I       I       I       I       I       I       I
  23  33     0    7     I      I       I       I       I       I       I       I       I
  24  33     0    0       I    I       I       I       I         I       I     I       I
  25  33     0    1     I      I       I       I       I         I     I       I       I
  26  34     0    2     I        I     I         I     I         I     I       I       I
  27  34     0    3     I      I       I       I       I         I     I       I       I
  28  34     0    4     I      I       I       I       I       I       I       I       I
  29  34     0    5     I      I       I       I       I       I       I       I       I
  30  34     0    6     I      I       I       I       I       I       I       I       I
  31  34     0    7     I      I       I       I       I       I       I       I       I
  32  34     0    0       I    I       I       I       I         I       I     I       I
  33  34     0    1     I      I       I       I       I         I     I       I       I
  34  35     0    2     I        I     I         I     I         I     I       I       I
  35  35     0    3     I      I       I       I       I         I     I       I       I
  36  35     0    4     I      I       I       I       I       I       I       I       I
  37  35     0    5     I      I       I       I       I       I       I       I       I
  38  35     0    6     I      I       I       I       I       I       I       I       I
  39  35     0    7     I      I       I       I       I       I       I       I       I
  40  35     0    0       I    I       I       I       I         I       I     I       I
  41  35     0    1     I      I       I       I       I         I     I       I       I
  42  36     0    2     I        I     I         I     I         I     I       I       I
  43  36     1    3     I      I       I       I       I         I     I       I       I
  44  36     1    4     I      I       I       I       I       I       I       I       I
  45  36     1    5     I      I       I       I       I       I       I       I       I
  46  36     1    6     I      I       I       I       I       I       I       I       I
  47  36     1    7     I      I       I       I       I       I       I       I       I
  48  36     1    0     I      I       I       I       I       I       I       I       I
  49  36     1    1       I    I       I       I       I         I       I     I       I
  50  37     1    2     I        I     I         I     I         I     I       I       I
  51  37     1    3     I      I       I       I       I         I     I       I       I
  52  37     1    4     I      I       I       I       I         I     I       I       I
  53  37     1    5     I      I       I       I       I       I       I       I       I
  54  37     1    6     I      I       I       I       I       I       I       I       I
  55  37     1    7     I      I       I       I       I       I       I       I       I
  56  37     1    0     I      I       I       I       I       I       I       I       I
  57  37     1    1       I    I       I       I       I         I       I     I       I
  58  38     1    2     I        I     I         I     I         I     I       I       I
  59  38     1    3     I      I       I       I       I         I     I       I       I
  60  38     1    4     I      I       I       I       I         I     I       I       I
  61  38     1    5     I      I       I       I       I       I       I       I       I
```

ORIGINAL PAGE IS
OF POOR QUALITY


(Phase Difference in 64ths of a Cycle)

```
NUMBER OF POINTS REPRESENTED IS    100                  NUMBER OF POINTS TO REPRESENT ONE   * =    1

ACTUAL PLOTTED DATA LIMITS ARE FROM    -6 TO      0     INTERVAL FOR THE GRAPH IS      1

VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ  VAL.  FREQ

  -6  16 I    -4  18 I    -2  46 I     0  20 I
```

```
C-      ---------INITIATE AND DIMENSION----------
         INTEGER*2  REF,B(24),INDI,NS,TD,PHA,HISTOG
        INTEGER PHADIF,PH(650)
        LOGICAL LMONO,UP,INDLOC,UPPE,DN,LOMEGA,IND,LMONO2,EX,POMEGA
C-      ------OM AND AM MUST BE DIMENSIONED TO 2*(2**MNUMB)+1
        LOGICAL*1 OM(129),AM(129)
        DIMENSION ARRAY(650)
        EX = .FALSE.
        LMONO2 = .FALSE.
        INDLOC = .FALSE.
        POMEGA = .FALSE.
        READ (1,13) NCYC,MNUMB,NNUMB,DCYC,DELPHI,NOISE,A
13      FORMAT (3I5,2F10.2,I2,F5.2)
        READ (1,14) TD,PHA,HISTOG
14      FORMAT (3I5)
C-      -------THETA IS INITIAL PHASE SHIFT,=2*PI*DELPHI/360.
        THETA = DELPHI*0.01745329
        SINE = SIN(THETA)
        COSINE = COS (THETA)
        DO 57 I=1,129
        OM(I) =.FALSE.
        AM(I)=.FALSE.
57      CONTINUE
        JJ = 0
        JJJ = 0
        II = 0
        MONO = 0
        M = 0
        N = 0
        MREGM = 2**MNUMB
        MREGN = 2**NNUMB
        HALF = FLOAT(MREGM)/2.
        MHALF = HALF
        K = MREGM
        NCYCP = NCYC*MREGM
        NCYCT = NCYC*.1
        X = (DELPHI/360.)*MREGM
        IX = X
        III = 0
        JII = 64
        INDCH = 0
C-      ---------DO-LOOP BEGINS WITH OCCURRENCE OF FIRST CLOCK PULSE-
        DO 12 I=1,NCYCP
C-      -----M IS NUMBER STORED IN UP-COUNTER-
        M = M+1
        IF (M-MREGM) 10,11,11
11      M = 0
10      CONTINUE
C-      ----BINARY NUMBER IN UP-DN COUNTER IS TRUNCATED DUE TO BIT
C-      ------INTEGRATION.   NTRUN IS RETURNED---------------
        CALL TRUN (NNUMB,MNUMB,N,NTRUN,B)
C-      ----IF M=NTRUN, REF GOES HIGH
        IF (M-NTRUN) 1,2,1
1       REF = 0
        GO TO 3
2       REF = 1
3       CONTINUE
C-      --------MONOS PROVIDES OUTPUT OF MONOSTABLE.  K IS TIMER FOR MONO
C-      -------LMONO IS LOGICAL OUTPUT OF MONOSTABLE.
        K = K+1
        CALL MONOS (K,REF,DCYC,MREGM,LMONO)
C-      ----LOMEGA IS OMEGA SIGNAL ZERO CROSSINGS.
C-      ------A IS VOLTAGE SNR.  POMEGA IS TRUE PHASE OF OMEGA SIGNAL.
        IF (IX.EQ.M) GO TO 30
        POMEGA = .FALSE.
        GO TO 31
30      POMEGA = .TRUE.
31      CONTINUE
        IF (NOISE.EQ.0) GO TO 32
        IF (POMEGA) GO TO 34
        GO TO 51
34      INDCH = -1*INDCH+1
        IF (INDCH.EQ.1) GO TO 37
        DO 38 IJK = 1,129
        AM(IJK) =.FALSE.
38      CONTINUE
C-      ----NOISE IS ADDED TO QUADRATURE COMPONENTS OF OMEGA SIGNAL.
39      CALL ANORM (AN)
        ANUM = A*SINE+AN
        CALL ANORM (AN)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DENOM = A*COSINE+AN
      THETAI = ATAN2 (ANUM,DENOM)
      PCYC = (THETAI/6.28318)*MREGM
      INDEX = PCYC
      IF (INDCH.EQ.1) GO TO 50
      AM (MREGM+INDEX) =.TRUE.
      GO TO 51
50    OM (MREGM+INDEX) =.TRUE.
      GO TO 51
37    DO 52 IJK=1,129
      OM(IJK) =.FALSE.
52    CONTINUE
      GO TO 39
51    CONTINUE
      JII = JII+1
      III = III+1
      IF (III.EQ.129) III = 1
      IF(JII.EQ.129) JII =1
      LOMEGA = OM(III).OR.AM(JII)
      IF ((I.EQ.M).AND.(POMEGA)) LOMEGA = POMEGA
32    IF (NOISE.EQ.0) LOMEGA = POMEGA
C--   ----LMONO AND LOMEGA ENTER PHASE DETECTOR
      UP = LMONO.AND.LOMEGA
      DN =(.NOT. LMONO).AND.LOMEGA
C--   ------PHASE DETECTOR OUTPUT FEEDS UP-DN COUNTER
      IF (UP) N = N+1
      IF (DN) N = N-1
C--   -------COUNT REGISTER MUST CONTAIN POSITIVE NUMBER
      IF(N) 8,9,9
8     N = MREGN+N
9     CONTINUE
C--   --------UP-DN COUNTER COUNTS MODULO-MREGN-
      IF (MREGN-N) 23,24,24
23    N=0
24    CONTINUE
      IF (TD.EQ.0) GO TO 501
      IF (I-1) 40,40,70
40    CONTINUE
C--   -----TIMING DIAGRAM IS PLOTTED--
      PRINT 500
500   FORMAT('1',2X,'I',3X,'N',3X,'NTRUN',3X,'M',4X,'REF',3X,'LOMEGA',2X
     *,'POMEGA',4X,'UP',6X,'DN',4X,'LMONO',3X,'LMONO2',2X,'INDLOC',4X,'E
     *X'//)
70    CONTINUE
      CALL PLOT1 (I,N,NTRUN,M,REF,LOMEGA,LMONO,LMONO2,DN,UP,INDLOC,EX,PO
     *MEGA)
501   CONTINUE
      IF ((PHA.EQ.0).AND.(HISTOG.EQ.0)) GO TO 12
      IF (.NOT.POMEGA) GO TO 65
      JJ = JJ+1
      IF (JJ.EQ.1) GO TO 28
      AJ = JJ/10.
      IJ = AJ
      IF (IJ.NE.AJ) GO TO 65
28    CONTINUE
      IF (M.LT.MHALF) GO TO 20
      IF (NTRUN.LT.MHALF) GO TO 21
22    PHADIF = 2*(NTRUN-M)
      GO TO 45
21    PHADIF = (NTRUN + MREGM-M)*2
      GO TO 45
20    IF (NTRUN.LT.MHALF) GO TO 22
      PHADIF = 2*(NTRUN-MREGM-M)
45    CONTINUE
C--   ----------ARRAY IS LOADED FOR PLOT SUBROUTINE-
      JJJ = JJJ+1
      PH(JJJ) = PHADIF
65    CONTINUE
12    CONTINUE
      IF (PHA.EQ.0) GO TO 502
      CALL PLOT (PH,JJJ)
502   CONTINUE
      IF (HISTOG.EQ.0) GO TO 503
      ------ARRAY IS LOADED FOR HISTG SUBROUTINE
      DO 27 I = 1,JJJ
      ARRAY (I) =  FLOAT(PH(I))
27    CONTINUE
      DIMENSION TITLE(20)
      READ (1,146) TITLE
146   FORMAT (20A4)
```

```
      DATA AST/'*'/
C------HISTOGRAM IS PLOTTED---
      CALL HISTG   (JJJ,ARRAY,TITLE,1,AST)
 503  CONTINUE
      STOP
      END

      SUBROUTINE TRUN (NNUMB,MNUMB,N,NTRUN,B)
C        THIS SUBROUTINE CONVERTS DECIMAL INTEGERS TO BINARY FORM
C          AND TRUNCATES, SHIFTS DOWN REMAINING BITS AND RETURNS
      INTEGER*2 B(NNUMB)
      DO 9 I = 1,NNUMB
      B(I) = 0
 9    CONTINUE
      I=0
      DUM = N
 1    DUM=DUM/2.
      I = I+1
      IDUM=DUM
      IF (DUM-IDUM) 2,3,2
 2    B(I) = 1
      DUM=DUM-.5
      IF (DUM) 4,4,1
 3    B(I) = 0
      IF (DUM) 4,4,1
 4    CONTINUE
C     THE BINARY NUMBER IS TRUNCATED
      DO 11 I = 1,MNUMB
      B(I) = B(I+(NNUMB-MNUMB))
 11   CONTINUE
C     THE TRUNCATED BINARY NUMBER IS CONVERTED BACK TO DECIMAL
      NTRUN = 0
      DO 10 I=1,MNUMB
      K=B(I)*(2**(I-1))
      NTRUN = NTRUN + K
  10  CONTINUE
      RETURN
      END

      SUBROUTINE MONOS (K,REF,DCYC,MREGM,LMONO)
      LOGICAL LMONO
      INTEGER*2 REF
      X = DCYC*MREGM - 1.
      IX = X
      IF (REF) 10,10,12
 10   IF (K-IX) 14,14,11
 11   MONO = 0
      GO TO 18
 12   IF (K-IX) 14,14,19
 19   MONO = 1
      K=0
      GO TO 18
  14  MONO = 1
  18  IF (MONO)15,15,16
  15  LMONO = .FALSE.
      GO TO 17
  16  LMONO = .TRUE.
  17  RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
       SUBROUTINE PLOT (PH,JJJ)
       REAL LINE(129)
       INTEGER PHADIF,PH(JJJ)
       DATA DOT/'.'/,PLUS/'+'/,BLK/' '/,VLINE/'|'/,AST/'*'/
       DO 12 JJ = 1,JJJ
       IF (JJ.NE.1) GO TO 5
       PRINT 8
8      FORMAT  ('1',45X,'PHASE DIFFERENCE IN DEGREES VS. TIME IN MSEC.')
4      PRINT 2
2      FORMAT   (3X,'-180',28X,'-90',30X,'0',31X,'90',28X,'180')
6      PRINT 7
7      FORMAT (4X,'+',16('-------+'))
5.     CONTINUE
       NSP = PH(JJ) + 65
       IF (NSP.GT.129) GO TO 15
       IF (NSP.LT.1) GO TO 15
       GO TO 3
15     CONTINUE
       PRINT 23,PH(JJ)
23     FORMAT (100X,'PH= ',I10)
       RETURN
3      CONTINUE
C         CLEAR LINE
       DO 1 J = 1,129
       LINE(J) = BLK
1      CONTINUE
       LINE(1) = VLINE
       LINE(129) = VLINE
       AI = JJ/10.
       II = AI
       IF(AI.NE.II) GO TO 9
       LINE(17) = DOT
       LINE(33) = PLUS
       LINE(49) = DOT
       LINE (65) = PLUS
       LINE (81) = DOT
       LINE (97) = PLUS
       LINE (113) = DOT
9      CONTINUE
10     LINE(NSP) = AST
       IF (AI.EQ.II) GO TO 20
       WRITE(3,11) LINE
11     FORMAT (4X,129A1)
       GO TO 12
20     TIME =(FLOAT(JJ)*.0098039216)
       WRITE (3,21) LINE
21     FORMAT (4X,129A1)
       WRITE (3,22) TIME
22     FORMAT ('+',4X,E8.2)
12     CONTINUE
       RETURN
       END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
       SUBROUTINE PLOT1(I,N,NTRUN,M,REF,LOMEGA,LMONO,LMONO2,DN,UP,INDLOC,
      *EX,POMEGA)
       INTEGER*2 REF
       LOGICAL LMONO,UP,INDLOC,DN,LOMEGA,LMONO2,EX,POMEGA
       INTEGER LINE(66)
       INTEGER BLK/' '/,VLINE/'|'/
       DO 1 J = 1,66
       LINE(J) = BLK
 1     CONTINUE
       LINE(1) = VLINE
       DO 2 K = 1,8
       LINE (8*K) = VLINE
 2     CONTINUE
       IF (REF.NE.1) GO TO 3
       LINE(3) = VLINE
       LINE (1) = BLK
 3     IF (.NOT.LOMEGA) GO TO 4
       LINE (10) = VLINE
       LINE (8) = BLK
 4     IF (.NOT.POMEGA) GO TO 5
       LINE(18)= VLINE
       LINE (16)= BLK
 5     IF(.NOT.UP) GO TO 6
       LINE(26)= VLINE
       LINE (24)= BLK
 6     IF(.NOT.DN) GO TO 7
       LINE(34)= VLINE
       LINE (32)= BLK
 7     IF (.NOT.LMONO) GO TO 8
       LINE(42)= VLINE
       LINE (40)= BLK
 8     IF (.NOT.LMONO2) GO TO 9
       LINE(50)= VLINE
       LINE (48)= BLK
 9     IF (.NOT.INDLOC) GO TO 10
       LINE(58)= VLINE
       LINE (56)= BLK
 10    IF (.NOT.EX) GO TO 11
       LINE(66)= VLINE
       LINE (64)= BLK
 11    WRITE(3,12) I,N,NTRUN,M,LINE
 12    FORMAT (1X,2I4,I6,I5,5X,66A1)
       RETURN
       END
   100    6    8     0.50          0.0 1   3.60
     0    1    1
 PHASE DIFFERENCE IN 64THS OF A CYCLE
```

ORIGINAL PAGE IS
OF POOR QUALITY

B. DRM SIMULATION

```
VERSITY  O U I J I  SYSTEM                                           16:17:56    DECEMBER 9, 1
RAMMER          ACCOUNT   IDENT    TIME    RECORDS                              COMP  LIST  EXEC  (
VIC    R         N4513              60       500                                  F     Y     F

          C-----------DRM MODEL-----------
                FQUIVALENCE (CK,TA,TB,TC,TD),(NENABL,L1,K1,I1,J1,G1,M1),
               *(CL,F1,F1,CC,CD),(ST9,F2,E2,PRC,PRD),(NCK,D2,C2,B2,A2),(RID,D1),
               *(RIC,C1),(RIB,B1),(RIA,A1),(D4,N1),(C4,N2),(B5,N3),(A5,N4),
               *(N5,Z,O1),(UNICA,O2),(O3,Y),(L2,D3,OCBAR),(OA,K2,I2,C3),
               *(OB,I3,B3),(OC,G2,M2,A3,J2),(I4,H1),(J3,H2),(H3,GC),(G3,GD),
               *(OD,M3),(ODBAR,A4),(F3,CA),(E3,CB),(B4,OABAR),(M4,ENCUT)
               *,(K3,GB),(L3,GA)
                LOGICAL*1 PRA,PRB,OBBAR,NCK,ENABLE,CK,TA,TB,TC,TD,NENABL,L1,K1,I1
               *,J1,G1,M1,CL,F1,D2,E1,CC,CD,ST9,F2,E2,PRC,PRD,NCK,C2,B2,A2,RID,D1,
               *RIC,C1,RIB,B1,RIA,A1,D4,N1,C4,N2,B5,N3,A5,N4,N5,Z,O1,UNICA,O2,O3,
               *Y,L2,D3,OCBAR,OA,D2,I2,C3,OB,I3,B3,OC,G2,M2,A3,J2,I4,H1,J3,H2,H3,
               *GC,G3,GD,OD,M3,ODBAR,A4,F3,CA,E3,CB,B4,OABAR,M4,ENCUT,K2,L3,K3,GA
               *,G3
                INTEGER LINE(66)
                UNICA = .TRUE.
                OA = .FALSE.
                OB = .FALSE.
                OC = .FALSE.
                OD = .FALSE.
                OABAR = .TRUE.
                OCBAR = .TRUE.
                ODBAR = .TRUE.
                PRA = .FALSE.
                PRB = .FALSE.
                ICK = 1
                READ (1,1) ENABLE,ST9,CL
      1         FORMAT (3(4XL1))
          C----------DETERMINE FF INPUTS
                NENABL = .NOT.ENABLE
                DO 6 J=1,10
                READ (1,7) RID,RIC,RIB,RIA
      7         FORMAT (4(4XL1))
                DO 4 I = 1,40
                F3 = F1.OR.F2
                E3 = F1.OR.E2
                L3 = L1.AND.L2
                K3 = K1.AND.K2
                I4 = I1.AND.I2.AND.I3
                J3 = J1.AND.J2
                H3 = H1.OR.H2
          C-----GENERATE CLOCK PULSE-----
                ICK = -1*ICK+1
                IF (ICK.EQ.1) GO TO 2
                CK = .FALSE.
                GO TO 3
      2         CK = .TRUE.
      3         CONTINUE
          C---------LEADING EDGE OF CK TRIGGERS FFLOPS-----
                CALL TFF (CA,PRA,GA,TA,OA,OABAR)
                CALL TFF (CB,PRB,GB,TB,OB,OBBAR)
                CALL TFF (CC,PRC,GC,TC,OC,OCBAR)
                CALL TFF (CD,PRD,GD,TD,OD,ODBAR)
          C-----------DETERMINE OUTPUT---------
                NCK = .NOT.CK
                M4 = .NOT.(M1.AND.M2.AND.M3)
                D4 = D1.AND.D2.AND.D3
                C4 = C1.AND.C2.AND.C3
                B5 = B1.AND.B2.AND.B3.AND.B4
                A5 = A1.AND.A2.AND.A3.AND.A4
                N5 = .NOT.(N1.OR.N2.OR.N3.OR.N4)
                O3 = (.NOT.(O1.AND.O2))
                IF (I.EQ.1) PRINT7,RID,RIC,RIB,RIA
                IF (I.EQ.1) PRINT 5
      5         FORMAT      (23X,'CK',7X,'Y',8X,'OA',6X,'OB',6X,'OC',6X,'OD',6X,
               *'EN',3X,'ENCUT',4X,'ODBAR')
                CALL PLOT1 (I,J   ,ICK,ICK,CK,Y,OA,OB,OC,OD,ENABLE,ENCUT,ODBAR)
      4         CONTINUE
      6         CONTINUE
                STOP
                END
                                   EQUIVALENCE DATA MAP

    LOCATION       SYMBOL    LOCATION       SYMBOL    LOCATION       SYMBOL    LOCATION
    0000BC         TA        0000BC         TB        0000BC         TC        0000BC
    0000BD         L1        0000BD         K1        0000BD         I1        0000BD
```

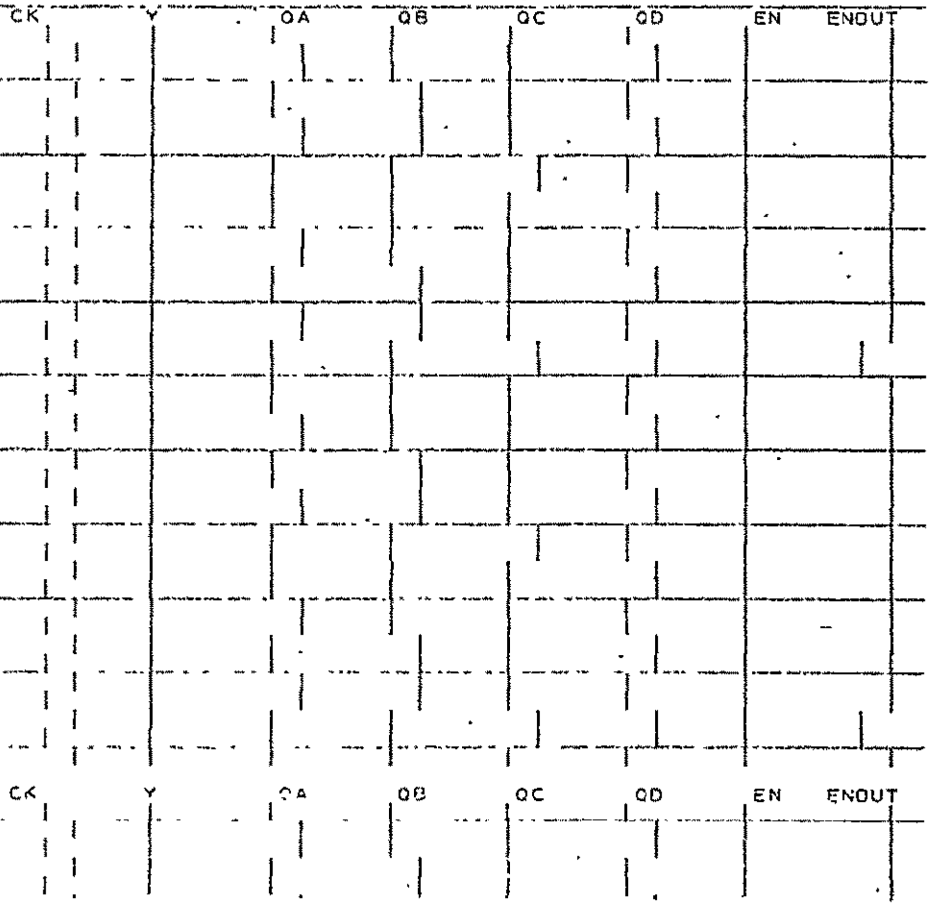
ORIGINAL PAGE IS
OF POOR QUALITY

```
                   SUBROUTINE PLOT1 (II,JJ,KK,MM,L1,L2,L3,L4,L5,L6,L7,L8,L9)
                   LOGICAL*1 L1,L2,L3,L4,L5,L6,L7,L8,L9
                   INTEGER LINE(65)
                   INTEGER BLK/' '/,VLINE/'|'/
                   DO 1 J = 1,66
                   LINE(J) = BLK
         1         CONTINUE
                   LINE(1) = VLINE
                   DO 2 K = 1,8
                   LINE (8*K) = VLINE
         2         CONTINUE
                   IF (.NOT.L1) GO TO 3
                   LINE(3) = VLINE
                   LINE (1) = BLK
         3         IF (.NOT.L2)  GO TO 4
                   LINE (10) = VLINE
                   LINE (8) = BLK
         4         IF (.NOT.L3) GO TO 5
                   LINE(18)= VLINE
                   LINE (16)= BLK
         5         IF (.NOT.L4) GO TO 6
                   LINE(26)= VLINE
                   LINE (24)= BLK
         6         IF (.NOT.L5) GO TO 7
                   LINE(34)= VLINE
                   LINE (32)= BLK
         7         IF (.NOT.L6) GO TO 8
                   LINE (42)= VLINE
                   LINE (40)= BLK
         8         IF (.NOT.L7) GO TO 9
                   LINE(50)= VLINE
                   LINE (48)= BLK
         9         IF (.NOT.L8) GO TO 10.
                   LINE(58)= VLINE
                   LINE (56)= BLK
        10         IF (.NOT.L9) GO TO 11
                   LINE(66)= VLINE
                   LINE (64)= BLK
        11         WRITE (3,12) II,JJ,KK,MM,LINE
        12         FORMAT (1X,2(4)I6,I5,5X,66A1)
                   RETURN
                   END
                                    SCALAR MAP

   LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   0000E4         VLINE      0000E8        J          0000EC        K          0000F0
   0000F8         KK         0000FC        MM         000100        L1         000104
   000106         L4         000107        L5         000108        L6         000109
   00010B         L9         00010C
                                    ARRAY MAP

   LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   000110
                                    SUBPROGRAMS CALLED

   LOCATION       SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   000218
                                    LABEL MAP

   LOCATION       LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION
   0003BC             2      0003F2            3      000426            4      000444
   000480             7      00049F            8      0004BC            9      0004DA
   000516            12      00055C
 MEMORY REQUIREMENTS 000584 BYTES          PLOT1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
OHIO UNIVERSITY  D U I J I  SYSTEM  FORTRAN IV  G-LEVEL COMPILER                     D
    0001                SUBROUTINE TFF (C,PR,G,T,C,QBAR)
    0002                LOGICAL*1 C,PR,T,C,QBAR,G
    0003                IF (.NOT.(PR.OR.C)) GO TO 1
    0004                IF (PR) GO TO 2
    0005                Q=.FALSE.
    0006                GO TO 3
    0007          2     CONTINUE
    0008                Q=.TRUE.
    0009          3     QBAR=.NOT.Q
    0010                IF (.NOT.(PR.AND.C)) RETURN
    0011                Q=.TRUE.
    0012                QBAR=.TRUE.
    0013                RETURN
    0014          1     IF (.NOT.G) RETURN
    0015                IF (T) Q=.NOT.Q
    0016                QBAR=.NOT.Q
    0017                RETURN
    0018                END
                                            SCALAR MAP

  SYMBOL     LOCATION       SYMBOL    LOCATION       SYMBOL    LOCATION       SYMBOL    L
  PR         0000B0         C         0000B1         Q         0000B2         QBAR      0
  T          0000B5
                                            LABEL MAP

  LABEL      LOCATION       LABEL     LOCATION       LABEL     LOCATION       LABEL     L
      2      000188             3     0001C0             1     000208
    TOTAL MEMORY REQUIREMENTS 000284 BYTES       TFF
      /DATA
 MODULE    ADDR  LENGTH                   ENTRY POINTS (E), EXTERNAL (X) AND COMMON (C) REFE
 MAIN44#   0F030 00644    E-MAIN44        0F000             X-IBCOM#                X-TFF
 PLCT1#    0F6A8 005B4    E-PLCT1         0F6A9             X-IBCOM#
 TFF#      0FC6C 00284    E-TFF           0FC60
   MODULE SUCCESSFULLY LOADED --- 30000 BYTES OF STORAGE REMAINING / EXECUTION BEGUN
```

```
    F    F    F    F
    1    1    0    0
    2    1    1    1
    3    1    0    0
    4    1    1    1
    5    1    0    0
    6    1    1    1
    7    1    0    0
    8    1    1    1
    9    1    0    0
   10    1    1    1
   11    1    0    0
   12    1    1    1
   13    1    0    0
   14    1    1    1
   15    1    0    0
   16    1    1    1
   17    1    0    0
   18    1    1    1
   19    1    0    0
   20    1    1    1
   21    1    0    0
   22    1    1    1
   23    1    0    0
   24    1    1    1
   25    1    0    0
   26    1    1    1
   27    1    0    0
   28    1    1    1
   29    1    0    0
   30    1    1    1
   31    1    0    0
   32    1    1    1
   33    1    0    0
   34    1    1    1
   35    1    0    0
   36    1    1    1
   37    1    0    0
   38    1    1    1
   39    1    0    0
   40    1    1    1
    F    F    F    T
    1    2    0    0
    2    2    1    1
    3    2    0    0
    4    2    1    1
    5    2    0    0
```



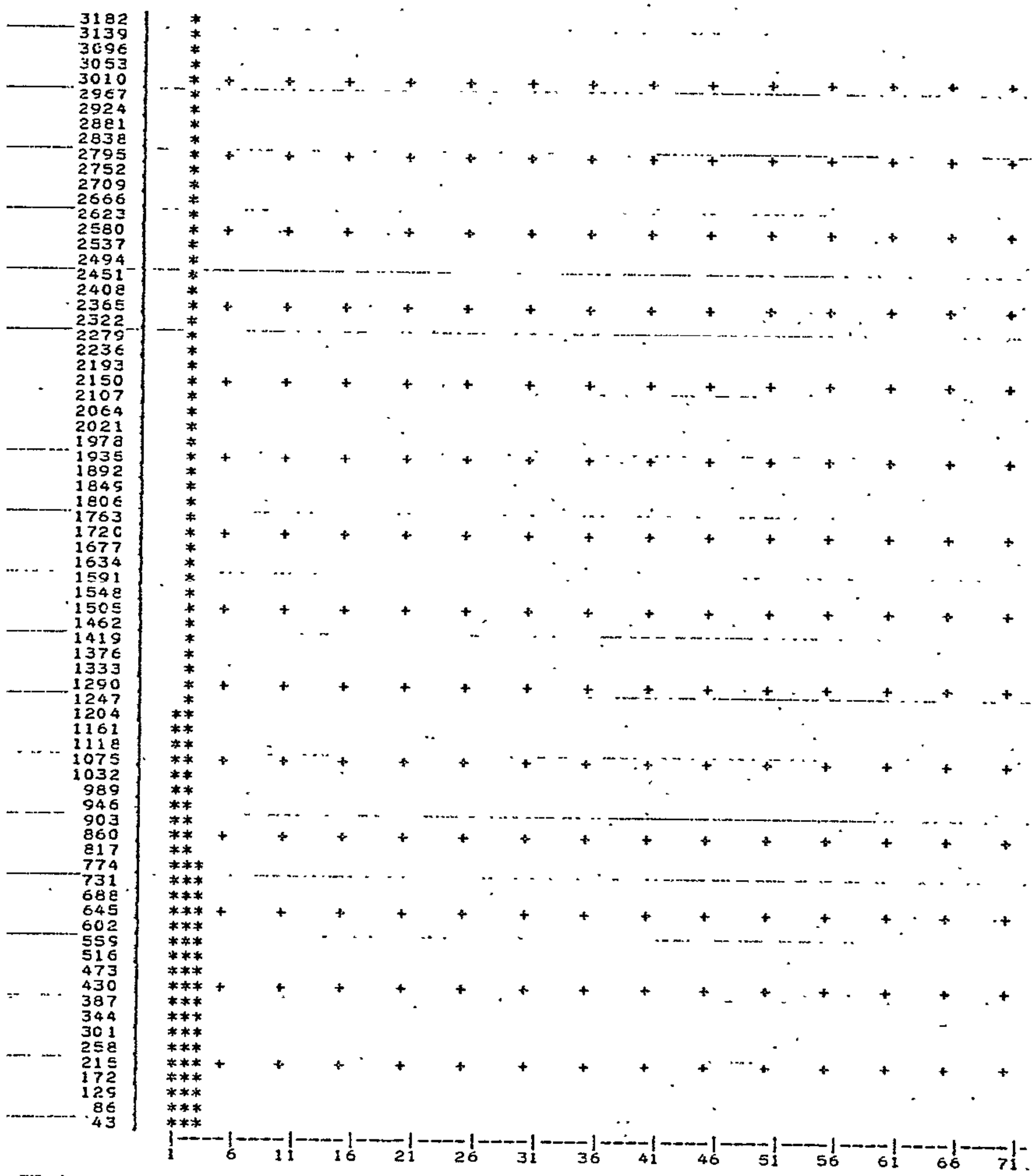
ORIGINAL PAGE IS
OF POOR QUALITY

C. COMPUTER PROGRAM SYN

```
:VERSITY . O U I J I   SYSTEM                                          10:15:27    FEBRUARY 14, 1 .

;RAMMER          ACCOUNT   IDENT   TIME   RECORDS                          COMP  LIST  EXEC  O
}ERLIN K          N2546            100     1500                             F     Y     F

'#                 HISTG
                LOGICAL*1 SYN1,SYN2,SYN3,SYN4,SYN5,SYN,ENABLE
                INTEGER PREV
                DIMENSION TITLE (20),DIF(8000),IDIF(8000)
                PREV = 0
                M = 0
                J = 0
                K = 0
                KK = 0
                DO 9 II = 1,10
                N = 0
                IF (II.NE.1) PREV = PREV - 10000
                DO 8 ICK = 1,10000
      C---------FIRST DRM IS PROGRAMMED TO 5--------------
      C---------SUCCEEDING STAGES ARE ENABLED WHEN ENABLE = .FALSE.---------
                I = MOD (ICK,10)
                SYN1 = ((I.EQ.2).OR.(I.EQ.4).OR.(I.EQ.5).OR.(I.EQ.7).OR.(I.EQ.9))
                ENABLE =(I.NE.0)
      C---------SECOND DRM IS PROGRAMMED TO 2----
                IF (ENABLE) GO TO 4
                M = M+1
                I = MOD(M,10)
                SYN2 = ((I.EQ.3).OR.(I.EQ.8))
                ENABLE =(I.NE.0)
      C-----------THIRD DRM IS PROGRAMMED TO 2------
      4         IF (ENABLE) GO TO 5
                J = J + 1
                I = MOD(J,10)
                SYN 3 = ((I.EQ.3).OR.(I.EQ.8))
                ENABLE =(I.NE.0)
      C---------FOURTH DRM IS PROGRAMMED TO 2------
      5         IF (ENABLE) GO TO 6
                K = K+1
                I = MOD (K,10)
                SYN 4 = ((I.EQ.3).OR.(I.EQ.8))
                ENABLE =(I.NE.0)
      C--------FIFTH DRM IS PROGRAMMED TO 4------
      6         IF (ENABLE) GO TO 7
                KK = KK +1
                I = MOD(KK,10)
                SYN5 = ((I.EQ.2).OR.(I.EQ.4).OR.(I.EQ.7).OR.(I.EQ.9))
      7         CONTINUE
      C------CLOCK OUTPUT IS SYNTHESIZED WITH OR-GATE------------
                SYN = SYN1.OR.SYN2.OR.SYN3.OR.SYN4.OR.SYN5
      C------NORMALIZED PHASE DIFFERENCE IS CALCULATED--------
                IF (.NOT.SYN) GO TO 8
                N = N+ 1
                IDIF(N) = ICK-PREV
                PREV = ICK
                SYN1 = .FALSE.
                SYN2 = .FALSE.
                SYN3 = .FALSE.
                SYN4 = .FALSE.
                SYN5 = .FALSE.
      8         CONTINUE
                DO 10 I = 1,N
                DIF(I) = FLOAT(IDIF(I))
      10        CONTINUE
                DATA AST /'*'/
                READ (1,2) TITLE
      2         FORMAT (20A4)
                CALL HISTG (N,DIF,TITLE,1,AST)
      9         CONTINUE
                STOP
                END
                                     SCALAR MAP

  LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
  0000D4        M         0000D8        J         0000DC        K         0000E0
  0000E8        N         0000EC        ICK       0000F0        I         0000F4
  000CFC        ENABLE    0000FD        SYN2      0000FE        SYN3      0000FF
  000101        SYN       000102
                                     ARRAY MAP

  LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
  000104        DIF       000154        IDIF      007E54
```

ORIGINAL PAGE IS OF POOR QUALITY

NORMALIZED FREQUENCY SPECTRUM   PLOT 1



NUMBER OF POINTS REPRESENTED IS   5223   NUMBER OF P(

ACTUAL PLOTTED DATA LIMITS ARE FROM   1 TO   3   INTERVAL FOR

```
VERSITY  O U I J I  SYSTEM                                          11:35:18    MARCH 31, 1975

RAMMER          ACCOUNT    IDENT    TIME    RECORDS                        COMP  LIST  EXEC  O
IERLIM K         L2546              500      500                            E     Y     E

      C---------DRM SYNTHESIZER MODEL WITH POST-DIVISION BY 4-------------
            LOGICAL    SYN1,SYN2,SYN3,SYN4,SYN5,SYN,ENABLE,OUT
            INTEGER PREV
            INTEGER*2 IA
            DIMENSION ICNT(50)
            DO 10 I = 1,50
            ICNT(I) = 0
      10    CONTINUE
            IDIF = 1
            PREV = 0
            M = 0
            J = 0
            K = 0
            KK = 0
            NN = 0
            DO 8 ICK = 1,400000
      C---------FIRST DRM IS PROGRAMMED TO 5-------------
      C---------SUCCEEDING STAGES ARE ENABLED WHEN ENABLE = .FALSE.----------
            I = MOD (ICK,10)
            SYN1 = ((I.EQ.2).OR.(I.EQ.4).OR.(I.EQ.5).OR.(I.EQ.7).OR.(I.EQ.9))
            ENABLE =(I.NE.0)
      C-------SECOND DRM IS PROGRAMMED TO 2----
            IF (ENABLE) GO TO 4
            M = M+1
            I = MOD(M,10)
            SYN2 = ((I.EQ.3).OR.(I.EQ.8))
            ENABLE =(I.NE.0)
      C-----------THIRD DRM IS PROGRAMMED TO 2------
      4     IF (ENABLE) GO TO 5
            J = J + 1
            I = MOD(J,10)
            SYN 3 = ((I.EQ.3).OR.(I.EQ.8))
            ENABLE =(I.NE.0)
      C---------FOURTH DRM IS PROGRAMMED TO 2-----
      5     IF (ENABLE) GO TO 6
            K = K+1
            I = MOD (K,10)
            SYN 4 = ((I.EQ.3).OR.(I.EQ.8))
            ENABLE =(I.NE.0)
      C---------FIFTH DRM IS PROGRAMMED TO 4-----
      6     IF (ENABLE) GO TO 7
            KK = KK +1
            I = MOD(KK,10)
            SYN5 = ((I.EQ.2).OR.(I.EQ.4).OR.(I.EQ.7).OR.(I.EQ.9))
      7     CONTINUE
      C-----CLOCK OUTPUT IS SYNTHESIZED WITH OR-GATE-----------
            SYN = SYN1.OR.SYN2.OR.SYN3.OR.SYN4.OR.SYN5
      C-------OUTPUT IS DIVIDED BY 4-------------
            IF (.NOT.SYN) GO TO 11
            NN = NN+ 1
            MN = MOD (NN,4)
            OUT = (MN.EQ.0)
      C-------NORMALIZED PHASE DIFFERENCE IS CALCULATED--------
            IF (OUT) IDIF = ICK - PREV
            IF (OUT) ICNT (IDIF) = ICNT (IDIF) + 1
            IF (OUT)  PREV = ICK
            SYN1 = .FALSE.
            SYN2 = .FALSE.
            SYN3 = .FALSE.
            SYN4 = .FALSE.
            SYN5 = .FALSE.
      11    CONTINUE
      8     CONTINUE
            DO 9 I = 1,50
            PRINT 500,I,ICNT(I)
      9     CONTINUE
      500   FORMAT (1X,2I7)
            STOP
            END
                                  SCALAR MAP

   LOCATION        SYMBOL     LOCATION         SYMBOL     LOCATION         SYMBOL     LOCATION
   0000D4          IDIF       0000D8           PREV       0000DC           M          0000E0
   0000E8          KK         0000EC           NN         0000F0           ICK        0000F4
   0000FC          SYN2       000100           SYN3       000104           SYN4       000108
   000110          MN         000114           OUT        000118
```

```
  OHIO UNIVERSITY  O U I J I  SYSTEM   FORTRAN IV  G-LEVEL COMPILER

                                              ARRAY MAP
    SYMBOL      LOCATION        SYMBOL     LOCATION       SYMBOL     LOCATION        SYMBOL
    ICNT        00011C
                                              SUBPROGRAMS CALLED

    SYMBOL      LOCATION        SYMBOL     LOCATION       SYMBOL     LOCATION        SYMBOL
    IBCOM#      0001E4
                                              LABEL MAP

    LABEL       LOCATION        LABEL      LOCATION       LABEL      LOCATION        LABEL
        10      00023E              4      00039E             5      000410              6
        11      0005DC              8      00050C             9      00062C            500
       TOTAL MEMORY REQUIREMENTS 000684 BYTES        MAIN44
            /DATA
  MODULE    ADDR  LENGTH                 ENTRY POINTS (E), EXTERNAL (X) AND COMMON (C) RI
  MAIN44#   0F000 000684    E-MAIN44     0F000        X-IBCOM#
    MODULE SUCCESSFULLY LOADED --- 30900 BYTES OF STORAGE REMAINING / EXECUTION BEGI

         1        0
         2        0
         3        0
         4        0
         5        1
         6     4446
         7    15672
         8    23327
         9     7776
        10        0
        11        0
        12        0
        13        0
        14        0
        15        0
        16        0
        17        0
        18        0
        19        0
        20        0
        21        0
        22        0
        23        0
        24        0
        25        0
        26        0
        27        0
        28        0
        29        0
        30        0
        31        0
        32        0
        33        0
        34        0
        35        0
        36        0
        37        0
        38        0
        39        0
        40        0
        41        0
        42        0
        43        0
        44        0
        45        0
        46        0
        47        0
        48        0
        49        0
        50        0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
VERSITY  W A T F I V  SYSTEM                                        09:45:05     MARCH 31, 1975

RAMMER          ACCOUNT    IDENT    TIME     RECORDS                           COMP   LIST   EXEC   OF
EBLIM K          L2546                500       500                             F      Y      F

       C----------DRM SYNTHESIZER MODEL WITH POST-DIVISION BY 8-------------
                LOGICAL     SYN1,SYN2,SYN3,SYN4,SYN5,SYN,ENABLE,OUT
                INTEGER PREV
                INTEGER*2 IA
                DIMENSION ICNT(50)
                DO 10 I = 1,50
                ICNT(I) = 0
       10       CONTINUE
                IDIF = 1
                PREV = 0
                M = 0
                J = 0
                K = 0
                KK = 0
                NN = 0
                DO 8 ICK = 1,800000
       C---------FIRST DRM IS PROGRAMMED TO 5-------------
       C---------SUCCEEDING STAGES ARE ENABLED WHEN ENABLE = .FALSE.---------
                I = MOD (ICK,10)
                SYN1 = ((I.EQ.2).OR.(I.EQ.4).OR.(I.EQ.5).OR.(I.EQ.7).OR.(I.EQ.9))
                ENABLE =(I.NE.0)
       C--------SECOND DRM IS PROGRAMMED TO 2----
                IF (ENABLE) GO TO 4
                M = M+1
                I = MOD(M,10)
                SYN2 = ((I.EQ.3).OR.(I.EQ.8))
                ENABLE =(I.NE.0)
       C-----------THIRD DRM IS PROGRAMMED TO 2------
       4        IF (ENABLE) GO TO 5
                J = J + 1
                I = MOD(J,10)
                SYN 3 = ((I.EQ.3).OR.(I.EQ.8))
                ENABLE =(I.NE.0)
       C---------FOURTH DRM IS PROGRAMMED TO 2-----
       5        IF (ENABLE) GO TO 6
                K = K+1
                I = MOD (K,10)
                SYN 4 = ((I.EQ.3).OR.(I.EQ.8))
                ENABLE =(I.NE.0)
       C--------FIFTH DRM IS PROGRAMMED TO 4-----
       6        IF (ENABLE) GO TO 7
                KK = KK +1
                I = MOD(KK,10)
                SYN5 = ((I.EQ.2).OR.(I.EQ.4).OR.(I.EQ.7).OR.(I.EQ.9))
       7        CONTINUE
       C-----CLOCK OUTPUT IS SYNTHESIZED WITH OR-GATE------------
                SYN = SYN1.OR.SYN2.OR.SYN3.OR.SYN4.OR.SYN5
       C-------OUTPUT IS DIVIDED BY 4------------
                IF (.NOT.SYN) GO TO 11
                NN = NN+ 1
                MN = MOD (NN,8)
                OUT = (MN.EQ.0)
       C------NORMALIZED PHASE DIFFERENCE IS CALCULATED--------
                IF (OUT) IDIF = ICK - PREV
                IF (OUT) ICNT (IDIF) = ICNT (IDIF) + 1
                IF (OUT)  PREV = ICK
                SYN1 = .FALSE.
                SYN2 = .FALSE.
                SYN3 = .FALSE.
                SYN4 = .FALSE.
                SYN5 = .FALSE.
       11       CONTINUE
       8        CONTINUE
                DO 9 I = 1,50
                PRINT 500,I,ICNT(I)
       9        CONTINUE
       500      FORMAT (1X,2I7)
                STOP
                END
                                          SCALAR MAP

  LOCATION        SYMBOL     LOCATION       SYMBOL     LOCATION        SYMBOL      LOCATION
  0000D4          IDIF       0000D8         PREV       0000DC          M           0000E0
  0000E8          KK         0000EC         NN         0000F0          ICK         0000F4
  0000FC          SYN2       000100         SYN3       000104          SYN4        000108
  000110          MN         000114         OUT        000118
  OHIO UNIVERSITY  O U I J I  SYSTEM   FORTRAN IV  G-LEVEL COMPILER

                                              ARRAY MAP
  SYMBOL      LOCATION        SYMBOL      LOCATION      SYMBOL      LOCATION        SYMBOL
  ICNT        00011C
                                              SUBPROGRAMS CALLED
  SYMBOL      LOCATION        SYMBOL      LOCATION      SYMBOL      LOCATION        SYMBOL
  IBCOM#      0001E4
                                              LABEL MAP
  LABEL       LOCATION        LABEL       LOCATION      LABEL       LOCATION        LABEL
      10      00023E              4       00039E
      11      0005DC              3       0005DC            5       000410              6
    TOTAL MEMORY REQUIREMENTS 000684 BYTES                  9       00062C            500
          /DATA                                     MAIN44
  MODULE    ADDR  LENGTH                    ENTRY POINTS (E), EXTERNAL (X) AND COMMON (C) R
  MAIN44#   0F000 00684    E-MAIN44     0F0C0          X-IBCOM#
   MODULE SUCCESSFULLY LOADED --- 30900 BYTES OF STORAGE REMAINING / EXECUTION BEG
         1       0
         2       0
         3       0
         4       0
         5       0
         6       0
         7       0
         8       0
         9       0
        10       0
        11       0
        12       0
        13    4436
        14    2225
        15   28800
        16    5552
        17   11151
        18       0
        19       0
        20       0
        21       0
        22       0
        23       0
        24       0
        25       0
        26       0
        27       0
        28       0
        29       0
        30       0
        31       0
        32       0
        33       0
        34       0
        35       0
        36       0
        37       0
        38       0
        39       0
        40       0
        41       0
        42       0
        43       0
        44       0
        45       0
        46       0
        47       0
        48       0
        49       0
        50       0
```

ORIGINAL PAGE IS
OF POOR QUALITY