TECHNICAL MEMORANDUM (NASA) 66

ANALYSIS AND DESIGN OF A SECOND-ORDER DIGITAL PHASE-LOCKED LOOP

A second-order digital phase-locked loop is analyzed by application of a Markov chain model with alternatives. Steady-state loop error statistics and mean transient time are determined for various loop parameters. In addition a hardware digital phase-locked loop was constructed and tested to demonstrate the applicability of the Markov chain model.

by

Paul R. Blasche

Avionics Engineering Center
Department of Electrical Engineering
Ohio University
Athens, Ohio 45701

March, 1979

Supported by

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

I. CONCLUSIONS

A specific second-order digital phase-locked lopp (DPLL) has been modeled as a first-order Markov chain with alternatives. From the matrix of transition probabilities of the Markov chain, the steady-state phase error of the DPLL was determined. In a similar manner the loop's transient response was calculated for a fading input.

Additionally, a hardware DPLL was constructed and tested to provide a comparison to the results obtained from the Markov chain model. In all cases tested, good agreement was found between the theoretical predictions and the experimental data.

II. INTRODUCTION

The phase-locked loop has long been recognized as a circuit with many important applications, and as such, the description of analog phase-locked loops (APLL's) has become well known as a large volume of material has been published to facilitate their use. In recent years, there has been an increasing use of various types of loops employing discrete elements. Among these have been hybrid PLL schemes that utilize both analog and digital circuitry. Newer loop realizations have been circuits composed entirely of digital element (DPLL's). The importance of these types of configurations lies in the relative ease of design and construction and, of equal importance, the ease in which such circuits can be maintained.

Unfortunately, the very attributes that make DPLL's attractive from a design standpoint also contribute to the difficulties in the theortical analysis of DPLL operation. In this area the available literature is relatively thin. For the area of analysis with a fading input, Weinburg and Lin [1] derive steady-state results for general first- and second-order DPLL's, but the analysis is limited by the assumption of small phase error values. In [2] and [3] steady-state loop performance in the presence of Gaussian noise is discussed for specific digital loop configurations. However, for both of these papers the analysis is limited to first-order DPLL's only.

In a previous Technical Memorandum [4], the analysis of a first-order DPLL was performed by modeling the loop as a first-order Markov chain. In the following, the ideas and methods used in this previous Technical Memorandum will be modified to allow a similar approach to be used for the analysis of second-order DPLL's. It will be shown that a second-order DPLL can be modeled as a first-order Markov chain with alternatives and that these alternatives themselves can be thought of as states in a first-order Markov chain. The steady-state distribution of the Markov chain alternatives can be determined and from this distribution it is possible to find the steady-state phase error of the DPLL. The transient response of the loop is determined in a similar manner.

Also detailed in the following is the design and testing of a hardware DPLL capable of either first- or second-order operation. A primary design objective followed in the construction of the hardware loop was to allow easy alteration of important loop parameters

to provide insight into their effects on loop performance. The general range of these parameters, however, was limited to values thought to be useful for an Omega navigation receiver application. It must be emphasized, however, that neither the theory developed in this TM nor the DPLL design are in any way limited to this particular application.

III. MARKOV CHAIN MODEL OF SECOND-ORDER DPLL

A. Second-Order DPLL Configuration. A block of the second-order DPLL configuration considered here is shown in Figure 1. This configuration differs from the first-order loop presented in [4] only by the addition of the S-bit counter and adder blocks. For this loop, the input is sampled at the positive-going zero crossing of the reference clock and then quantized to a value of ± 1. The quantized signal is applied directly to the divide-by-$2^{(M+N)}$ up/down counter (count up one on plus one and count down one on minus one) giving first-order phase updating. The quantized hard-limiter output is also applied to the input of the divide-by-2^S up/down counter which has the additional property that the counter will saturate at the limit points $\pm 2^{S-1} - 1$. Once the counters have been updated their values are summed and the sum loaded into the (M + N)-bit counter. The phase of the reference clock is then set to a value determined by the contents of the most significant bits of the (M + N) bit counter.

B. Markov Chains with Alternatives. A Markov chain can be characterized by a system containing a number of distinguishable states (finite or infinite) which transition to a new state denoted s_j from any present state s_i depends solely on the present state s_i. For the present, assume that the number of states is N, Then for each state s_i, $i = 1,2,\ldots, N$ a vector of transition probabilities may be written as,

$$P_i = (p_{i1}\ p_{i2}\ \cdots p_{iN}, \quad i = 1,2,\ldots,N \tag{1}$$

where p_{ij} is the probability of transition to state s_j given that the present state is s_i. Note that the N transition probability vectors must satisfy the requirement

$$p_{ij} \geq 0 \quad i,j = 1,2,\ldots,N$$

and
$$\sum_{j=1}^{N} p_{ij} = 1.0 \quad i = 1,2,\ldots,N \tag{2}$$

and are called stochastic vectors. These N vectors may be arranged in a matrix of size N x N to give the matrix of transition probabilities $\underline{P}$.

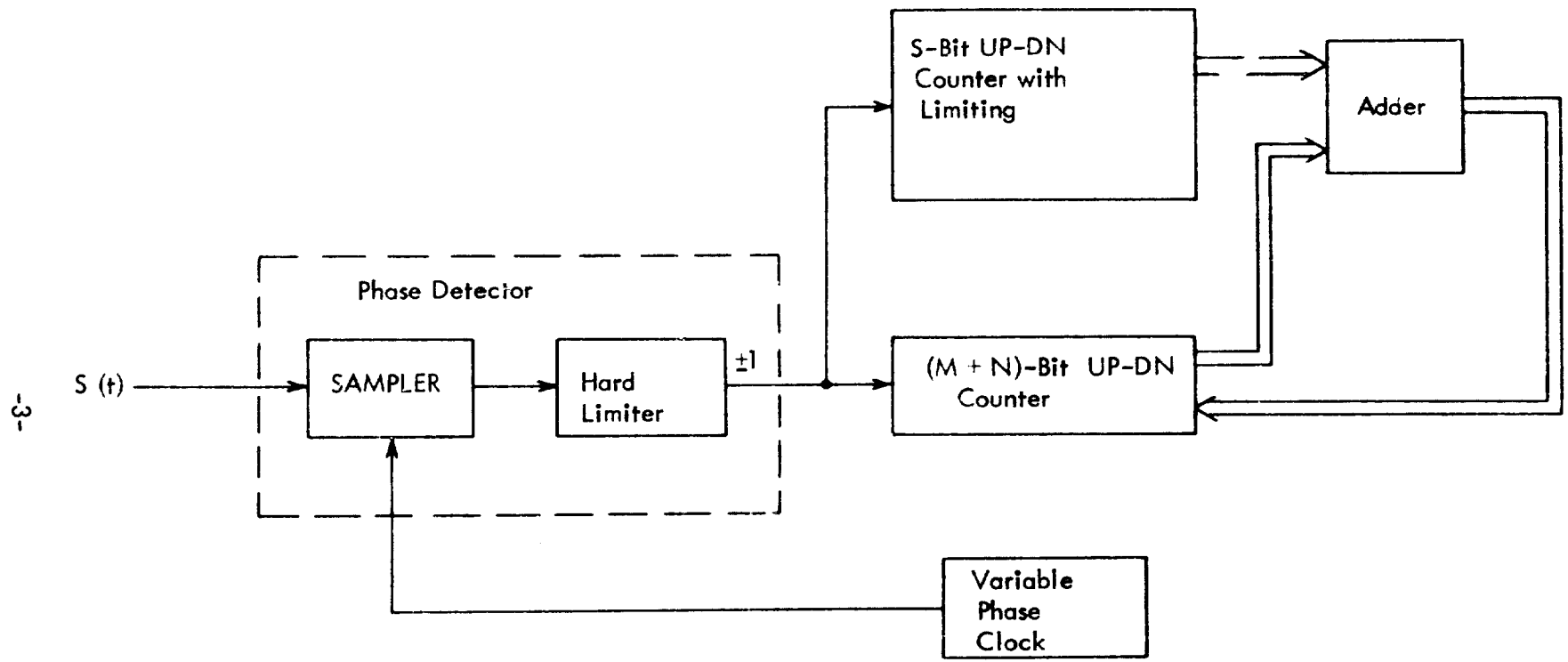

Figure 1. Block Diagram of Second-Order DPLL.

$$\underline{P} = \begin{bmatrix} p_{11} & p_{12} \cdots & p_{1N} \\ p_{21} & p_{22} \cdots & p_{2N} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ p_{N1} & p_{N2} \cdots & p_{NN} \end{bmatrix} \tag{3}$$

Note that this matrix defines all of the state-to-state transition probabilities for the Markov chain with a finite number of states.

A Markov chain can also be specified for which each state has one or more possible vectors of transition probabilities. For each state s_i, the possible vectors are called alternative vectors and transition from the present state to some new state is governed by one and only one of the alternative vectors associated with the present state. In this case, for each state s_i there exists K_i alternative vectors,

$$\begin{matrix} {}^{k}p_{i} = ({}^{k}p_{i1} & {}^{k}p_{i2} & \cdots {}^{k}p_{iN}) \\ & & \end{matrix} \quad \begin{matrix} i = 1,2,\ldots N \\ k = 1,2,\ldots K_i \end{matrix} \tag{4}$$

where ${}^{k}p_{ij}$ is the probability that the system will make a transition to state s_j given that the present state is s_i and k^{th} alternative for s_i is used. As before, each of the alternative vectors of transition probabilities are stochastic vectors and must satisfy the conditions,

$$\begin{aligned} &{}^{k}p_{ij} \geq 0 \quad i,j = 1,2, \ldots, N \text{ and } k = 1,2,\ldots, K_i \\ \text{and} \quad &\sum_{j=1}^{N} {}^{k}p_{ij} = 1.0 \quad i = 1,2,\ldots, N \text{ and } k = 1,2, \ldots, K_i \end{aligned} \tag{5}$$

The vectors may be combined to give the K x N stochastic matrix, $\underline{P}$,

$$\underline{P} = \begin{bmatrix} {}^{1}_{P_{11}} & {}^{1}_{P_{12}} & \cdots & {}^{1}_{P_{1N}} \\ {}^{2}_{P_{11}} & {}^{2}_{P_{12}} & \cdots & {}^{2}_{P_{1N}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ {}^{K_1}_{P_{11}} & {}^{K_1}_{P_{12}} & \cdots & {}^{K_1}_{P_{1N}} \\ {}^{1}_{P_{21}} & {}^{1}_{P_{22}} & \cdots & {}^{1}_{P_{2N}} \\ {}^{2}_{P_{21}} & {}^{2}_{P_{22}} & \cdots & {}^{2}_{P_{2N}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ {}^{K_2}_{P_{21}} & {}^{K_2}_{P_{22}} & \cdots & {}^{K_2}_{P_{2N}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ {}^{K_N}_{P_{N1}} & {}^{K_N}_{P_{N2}} & \cdots & {}^{K_N}_{P_{NN}} \end{bmatrix} \tag{6}$$

C. Markov Chain Model of DPLL. Use of the Markov chain with alternatives in the modeling of second-order DPLL's will be shown with an example. Consider the DPLL of Figure 1 with $M = 1$, $N = 2$, and $S = 3$. Defining the possible states of the $(M + N)$-bit counter as loop states then the loop will have eight loop states while the reference clock is quantized to $2^N = 4$ possible states. These eight loop states form the states of a first-order Markov chain. However, transfer from loop state to loop state is dependent not only upon the present loop state but also on the value of the S-bit counter. For example, if the value of the S-bit counter is zero, then a + 1 phase detector output will cause the $(M + N)$ counter to be incremented by a value of + 2. Similarly, a -1 phase detector when the S-bit counter is zero will cause the $(M + N)$-bit counter to decrement by a value of 2. However, if the value of the S-bit counter is +1, then a +1 phase detector output will cause the $(M + N)$-bit counter to increment by a value of 3 while a -1 phase detector output causes the counter to decrement by a value of 1. Since the possible values of the $(M+N)$-bit counter are associated with the states of a Markov chain, then the possible values of the S-bit counter can be associated with alternative actions for each loop state. Thus for the present case of $S=3$, there are seven alternative actions associated with each loop state.

A state diagram of the Markov chain model with alternatives for the DPLL under consideration is shown in Figure 2. Since there are eight possible loop states with each having seven possible alternative actions, there are a total of 56 alternative actions for the entire loop; or in terms of Markov chain notation, there are 56 alternative vectors of transition probabilities. For each loop state the alternative vector to be used is uniquely defined by the value of the S-bit counter so that the alternative vectors themselves can be thought of as states in a first-order Markov chain. That is, instead of considering the transitions from loop state to loop state, the transitions from loop alternative to loop alternative are considered. Thus in the state diagram, Figure 2, the loop alternatives are successively numbered $s_1, s_2, \ldots s_{56}$ and the possible transition from loop alternative to loop alternative is assigned a probability as indicated by the directed arrows. The values of the indicated probabilities are dependent on the state of the loop reference clock only. The method for determining their value was discussed in [4].

Denoting the steady-state probability of loop occupancy of state s_i as a_i then from Figure 2,

$$a_1 = a_{30} p_3 + a_{29} p_3 \tag{7}$$

It is clear that there will be 56 such equations which can be put in matrix form as

$$[A] \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_{56} \end{bmatrix} = [0] \tag{8}$$

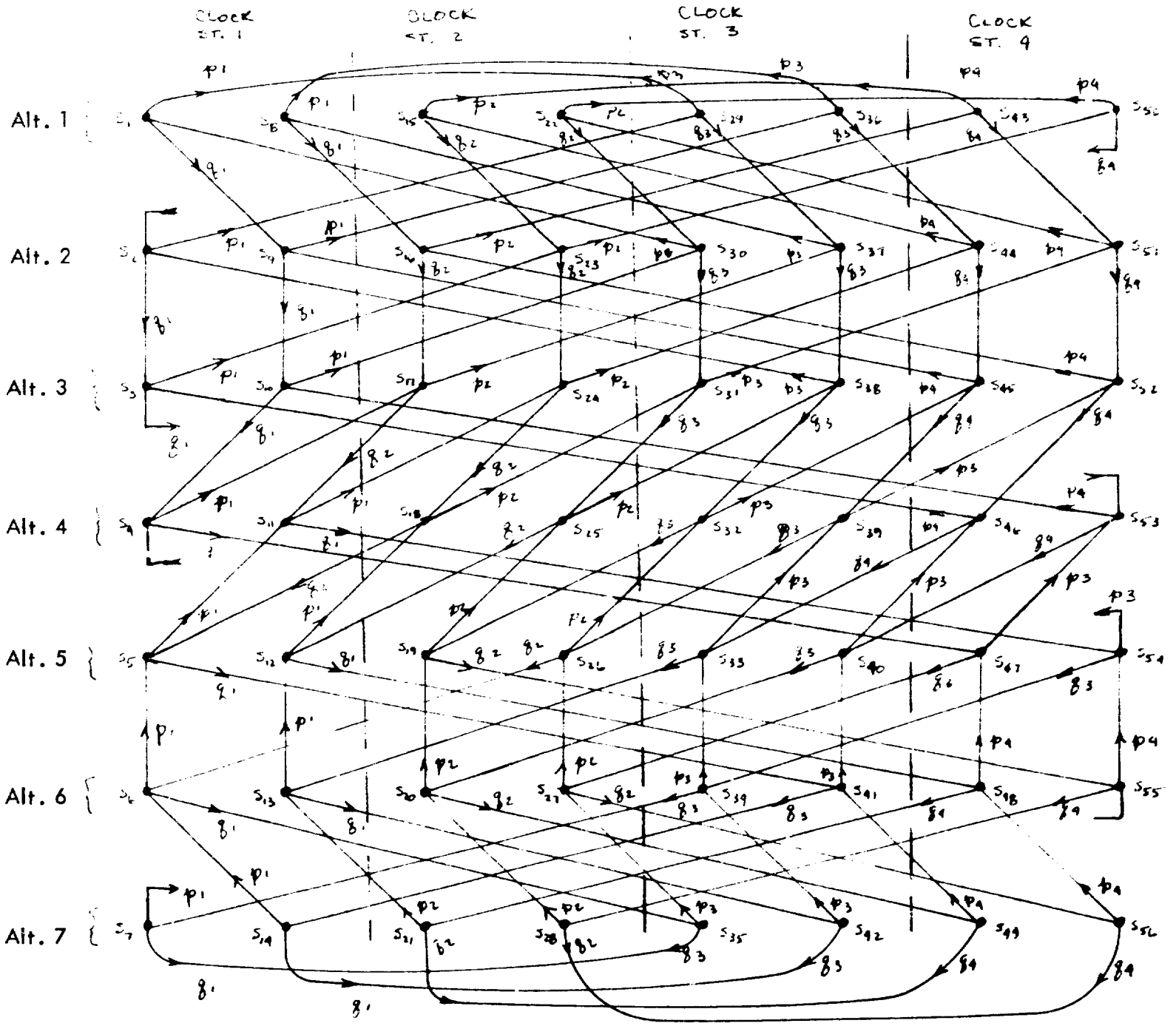

Figure 2. DPLL State Diagram.

for which a unique solution does not exist. However, since a_i represents a steady-state probability then

$$\sum_{i=1}^{56} a_i = 1.0 \tag{9}$$

so that the above homogeneous system in conjunction with this additional constraint can be solved for the steady-state probabilities a_i. If the steady-state probability of the occurrence of the k^{th} clock state is denoted as R_k then

$$R_k \quad \sum_{i=L}^{L+2^{M+S}-2M} a_i \qquad K = 1,2, \ldots, 2^N \tag{10}$$

where

$$L = (k-1) \quad [2^{M+S} - 2M] + 1 \tag{11}$$

The distribution of R_k then establishes the statistics for the phase error of the DPLL.

A similar method is used to find the mean time to loop lock-up for a given initial state. For the loop represented by Figure 2 with a noiseless input signal,

$$p_1 = p_2 = q_3 = q_4 = 1$$

and

$$p_3 = p_4 = q_1 = q_2 = 0$$

and it is easy to see that regardless of initial state the loop will move to a stable condition involving a transition loop of,

$$S_{38} \rightarrow S_{32} \rightarrow S_{19} \rightarrow S_{25}$$

Thus, occurrence of any of these four states can be used as defining the lock-up condition for the DPLL. Denoting T_i as the mean number of loop transitions to a lock-up condition for some initial state s_i, then a set of difference equations can be written in terms of mean time to loop lock as was done for the determination of steady-state loop probabilities. For instance if the loop is initially in state s_7, then

$$T_7 = p_1 \; T_{55} + q_1 \; T_{35} + 1 \tag{12}$$

Clearly, a similar equation can be written for each initial state. Note, for state S_{38}, S_{32}, S_{19}, and S_{25} then

$$T_{38} = T_{32} = T_{19} = T_{25} = 0$$

since the loop is defined as being initially in a lock condition.

D. Application of the Markov Chain Model. The DPLL's steady-state distribution and transient properties are determine by a solution of the systems of equations given by (8) and (9), respectively. A computer program PBSTGEN was written to establish the state-to-state transitions (i.e., the systems of equations) for arbitrary values of M, N, and S. Using the output of PBSTGEN a second program, PBDPL2, solved the system of equations to find the steady-state phase distribution for the DPLL. Similarly, the output of PBSTGEN by another program, PBDPL2T, to solve for the mean time to lock for an abritrary initial phase offset. In both PBDPL2 and DPL2T, Jacobi's iterative algorithm was used to solve the system of equations. These programs may be found in Appendix A. Results obtained from these programs will be presented in a later section of this TM.

IV. DIGITAL PHASE-LOCKED LOOP DESIGN

To demonstrate experimentally the validity of the Markov chain model described in the previous section, a hardware DPLL was designed and constructed. A block diagram of the hardware loop is shown in Figure 3. All register lengths shown in the block diagram indicate the maximum values and during testing the actual register lengths were altered to verify the effects of various parameters upon loop performance.

Referring to Figure 3, the phase detector will sample the incoming binary signal of frequency f_c and then output a count-down signal if the sample is a ZERO or a count-up signal if the input is ONE. The count-up or count-down signal is then applied to both the S-bit saturating counter and the (M + N)-bit counter. Inhibiting logic is included with the S-bit counter so that the counter will saturate at selectable values of $\pm (2^i - 1)$, $i = 1, 2, \ldots, 7$. The sample command also initiates the control logic so that the new value of the (M + N)-bit counter (following the count-up or count-down signal) is loaded into the 12-bit buffer by means of the $\overline{\text{LOAD2}}$ signal. After settling, the output of the 12-bit adder will contain the sum of the S-bit saturating counter and the (M + N)-bit counter. This value is then loaded into the (M + N)-bit counter by means of the $\overline{\text{LOAD1}}$ signal. Note, the value of the N most significant bits represents the phase estimate of the DPLL. To establish variable phase reference clock, this phase estimate is compared to the value of an N-bit counter being clocked at a $2^N * f_c$ rate by an N-bit binary magnitude comparator. Upon coincidence of the two input words the magnitude comparator output takes on a value of ONE. Note that this pulse output occurs at an f_c rate. Notice also that this loop will operate in the first-order mode simply by inhibiting the $\overline{\text{LOAD1}}$ signal to the (M + N) counter. Detailed schematics for the DPLL may be found in Appendix B.

The basic test configuration used for all DPLL experiments is shown in Figure 4. Since it was primarily desired to determine the steady-state phase error and the mean time to lock for an initial phase offset, it was possible to use the same clock source for both the DPLL input and the external reference source. A Sulzer temperature-compensated crystal oscillator was used as the reference source to minimize effects caused by frequency drift of this source. This source was quantized to binary levels and applied to both the DPLL variable phase reference clock and a phase shifting network. The phase shifted clock was then filtered, mixed with noise, and hardlimited to provide a noisy binary input to

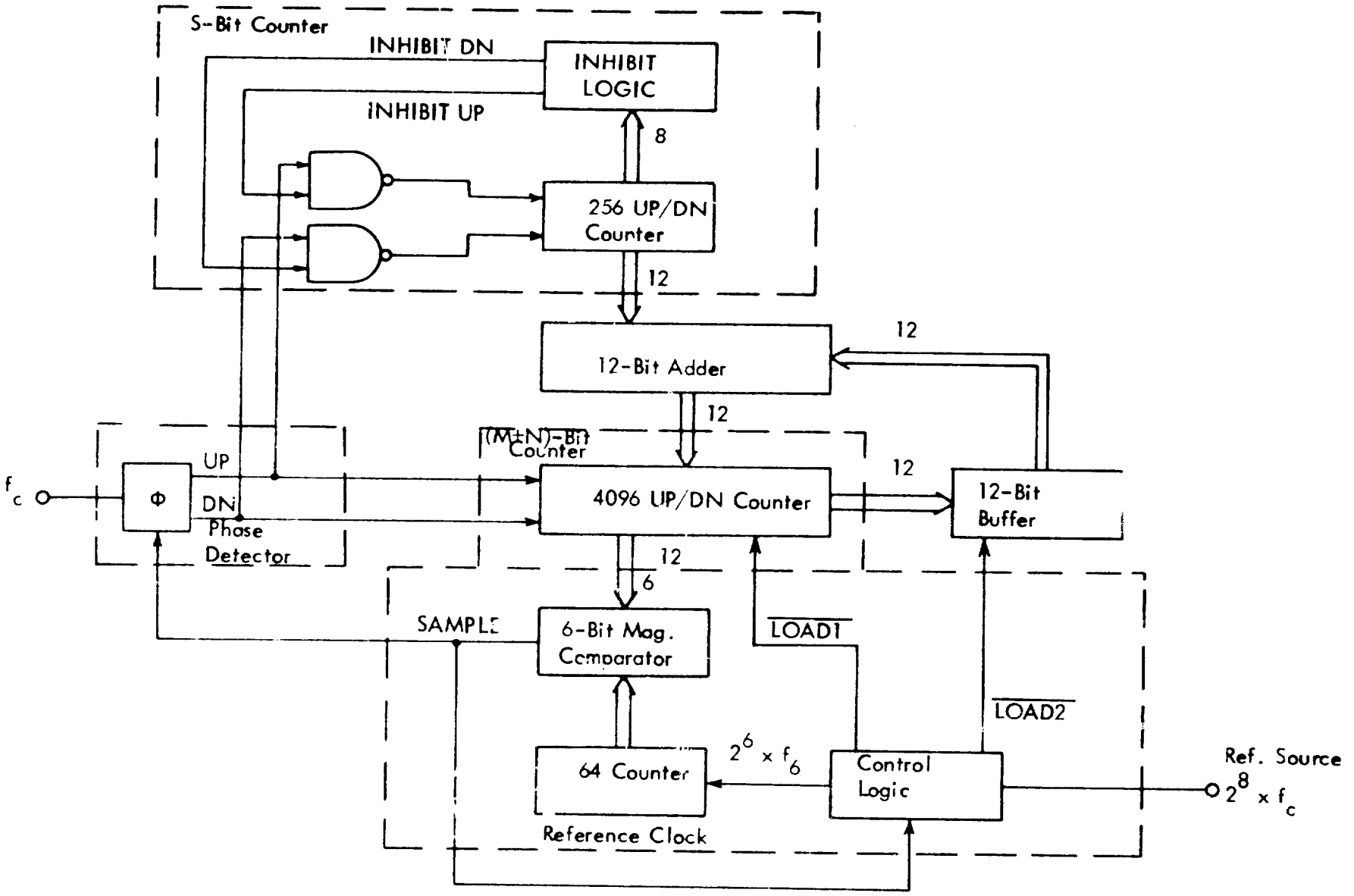

Figure 3. Hardware DPLL Block Diagram.

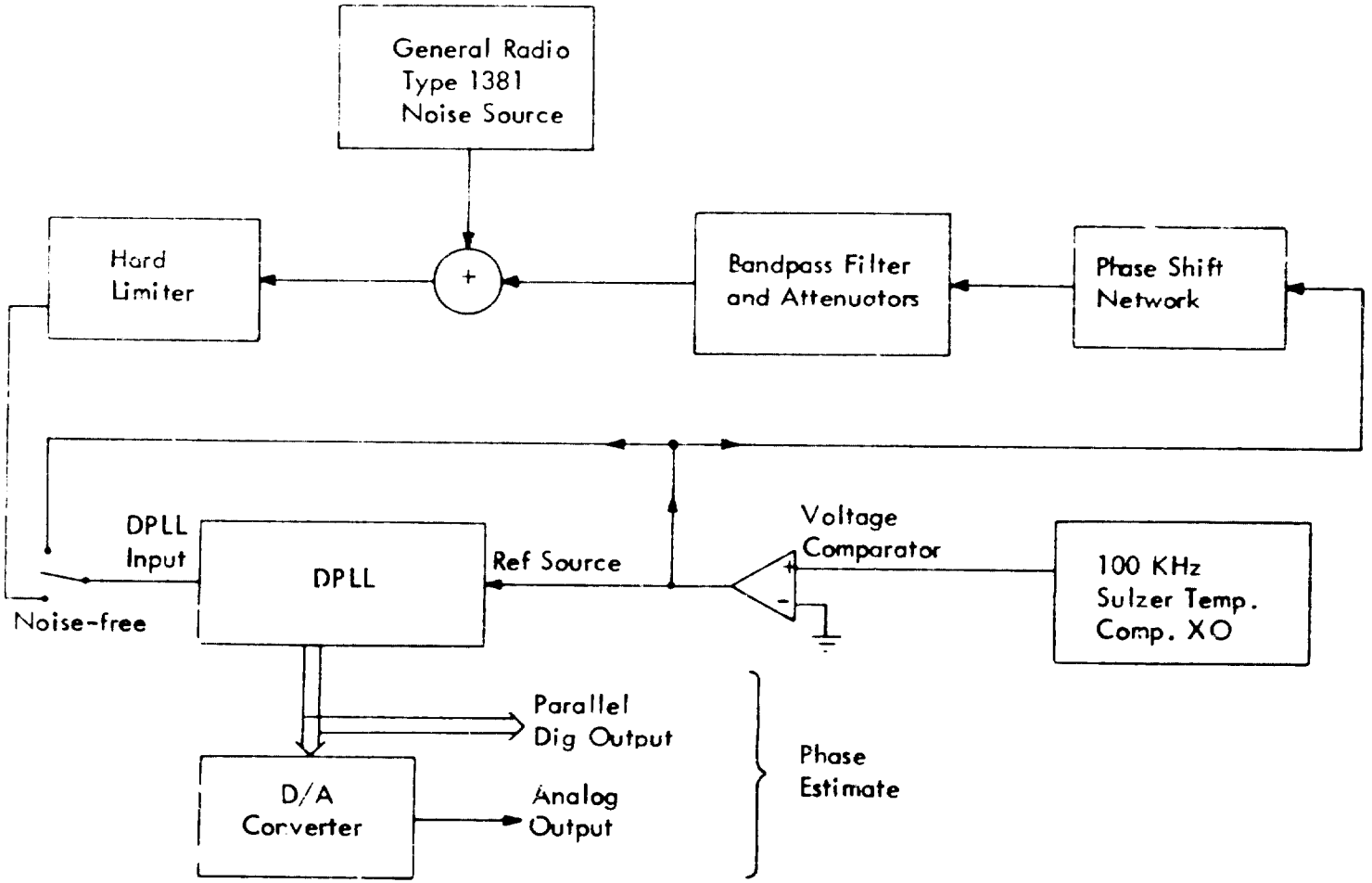

Figure 4. DPLL Test Circuit.

the DPLL. The phase estimate of the loop is continously stored in a buffer register to provide a parallel digital phase output. This value is also applied to a D/A converter to provide an analog phase output for monitoring purposes.

V. ANALYSIS AND HARDWARE TEST RESULTS

Solutions to the systems of equations describing the Markov chain model for the DPLL were performed for various loop parameter values by means of the computer programs found in Appendix A. Of prime concern was the steady-state phase error and the transient response of the DPLL for a fading input. In addition, data was taken from experiments involving the hardware DPLL to determine both of the parameters for comparison to the results obtained from the Markov chain model.

Figures 5,6, and 7 are plots of standard deviation of the phase error for the DPLL as a function of noise-to-signal ratio for various loop parameters. The results obtained from the Markov chain model are overplotted with the experimental results and, as can be seen, there is close agreement between the two. Notice for the case of S=1 the loop is operating in the first-order mode while for S greater than one the loop is operating second-order. Referring to these three figures it is seen that for constant M and N there is a reduction of 5 to 10 dB in loop performance as S increases in unit steps. Similarly, as S and N are held constant, there is a 5 to 10 dB increase in loop performance as M increases in unit steps.

Figures 8-14 are plots of the loop transient response for various loop parameters and signal-to-noise ratios. Again the experimental results are overplotted on the theoretical predictions. Each experimental data point represents the mean of at least 500 trials and once again there is good agreement between experimental and theoretical data. A particularly significant result may be found from a comparison of Figures 8-10. From these figures it is seen that for small values of initial phase offset, the mean time to lock increases with an increasing value of S. That is, for small phase offsets the first-order DPLL will achieve lock in less time than will a second-order DPLL. This is likewise found to be true when comparing Figures 11-13 and can be a significant factor in selection of loop parameters for an application such as an Omega navigation receiver.

VI. REFERENCES

[1] A. Weinberg and B. Liu, "Discrete Time Analysis of Non-uniform Sampling First- and Second-Order Digital Phase Lock Loops," IEEE Trans. on Communications, vol. COM-22, pp. 123-137, February 1974.

[2] J. Holmes, "Performance of a First-Order Transition Sampling Digital Phase-Locked Loop Using Random-Walk Models," IEEE Trans. on Communications, vol. COM-20, pp. 119-131, April 1972.

[3] J. Cessna and D. Levy, "Phase Noise and Transient Times for a Binary Quantized Digital Phase-Locked Loop in White Gaussian Noise," IEEE Trans. on Communications, vol. COM-20, pp. 94-104, April 1972.

[4] P. Blasche, "Analysis of a First-Order Phase-Locked Loop in the Presence of Gaussian Noise," NASA Technical Memorandum Number 46, Avionics Engineering Center, Ohio University, Athens, Ohio, March 1977.

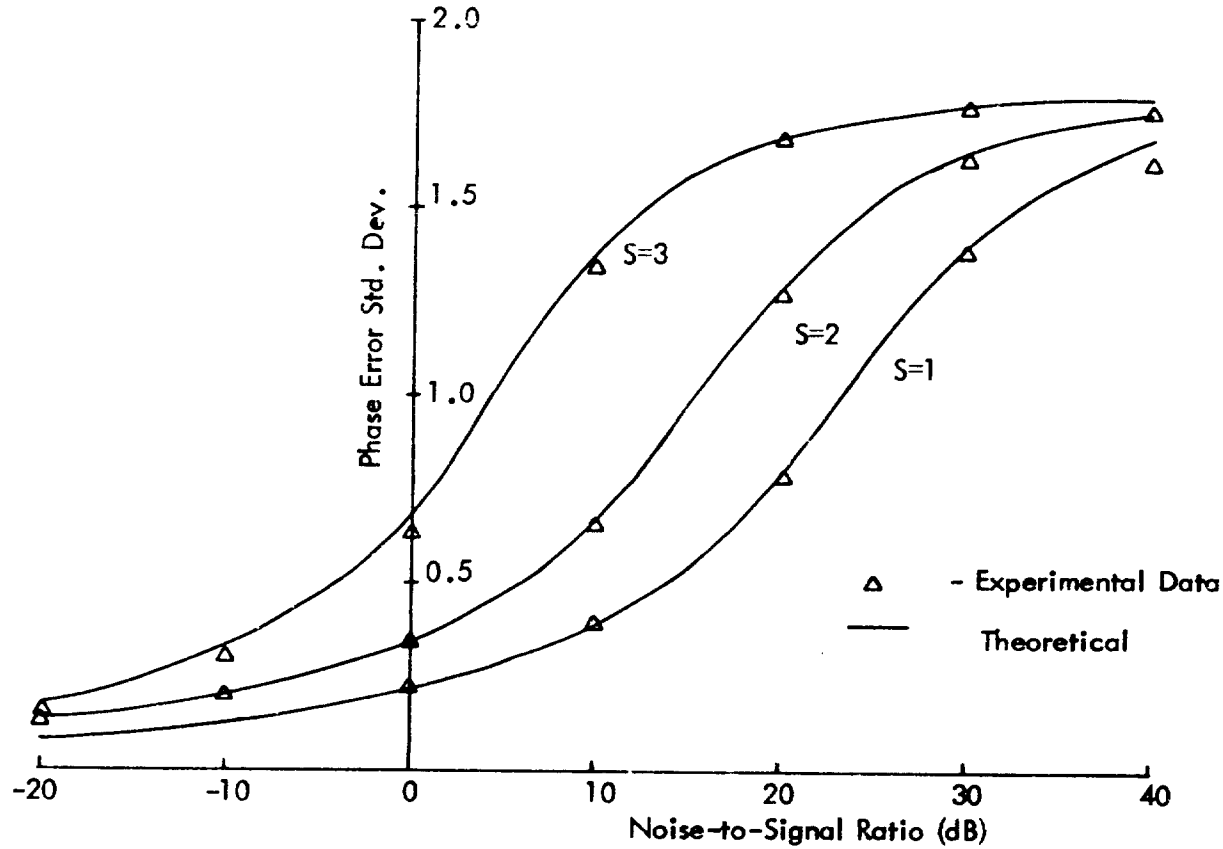

Figure 5. Loop Phase Jitter, M = 0, N = 6.

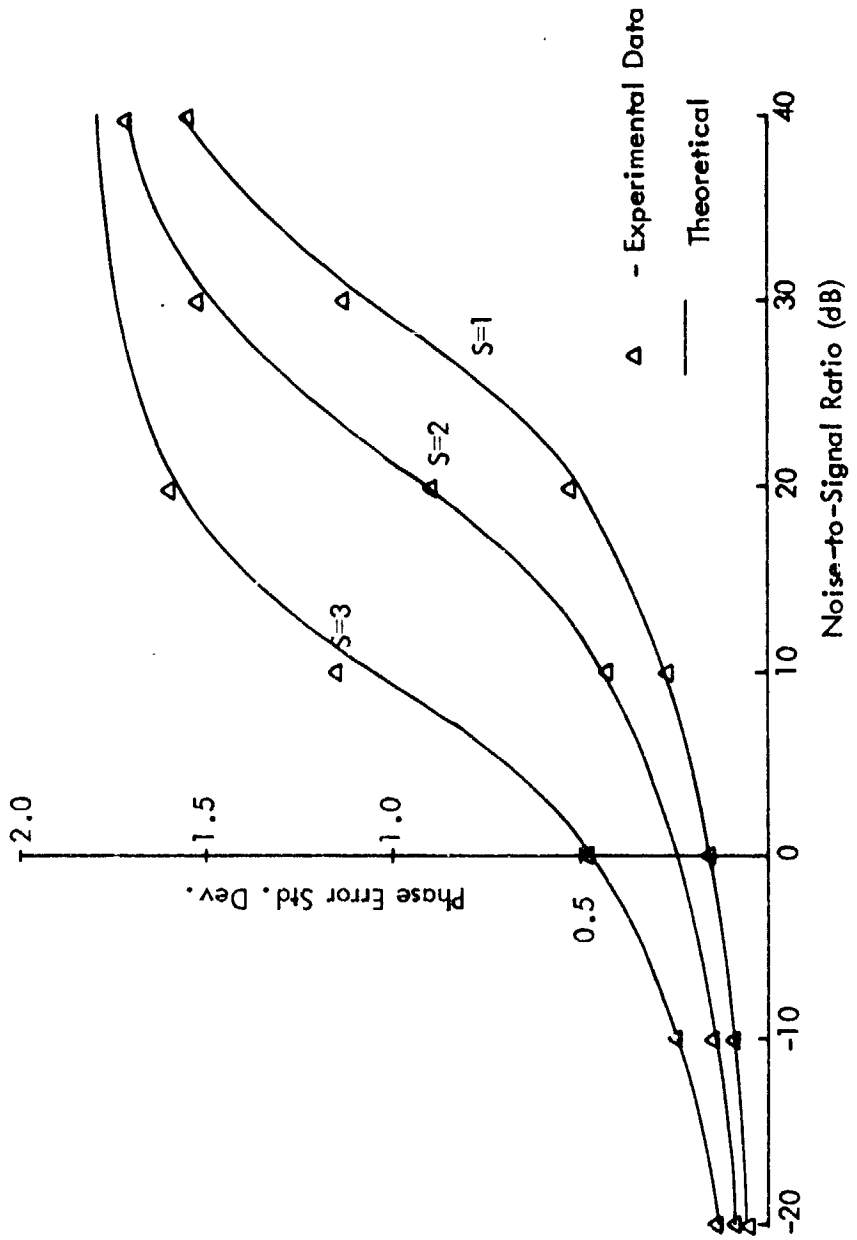

Figure 6. Loop Phase Jitter, M=1, N=6.

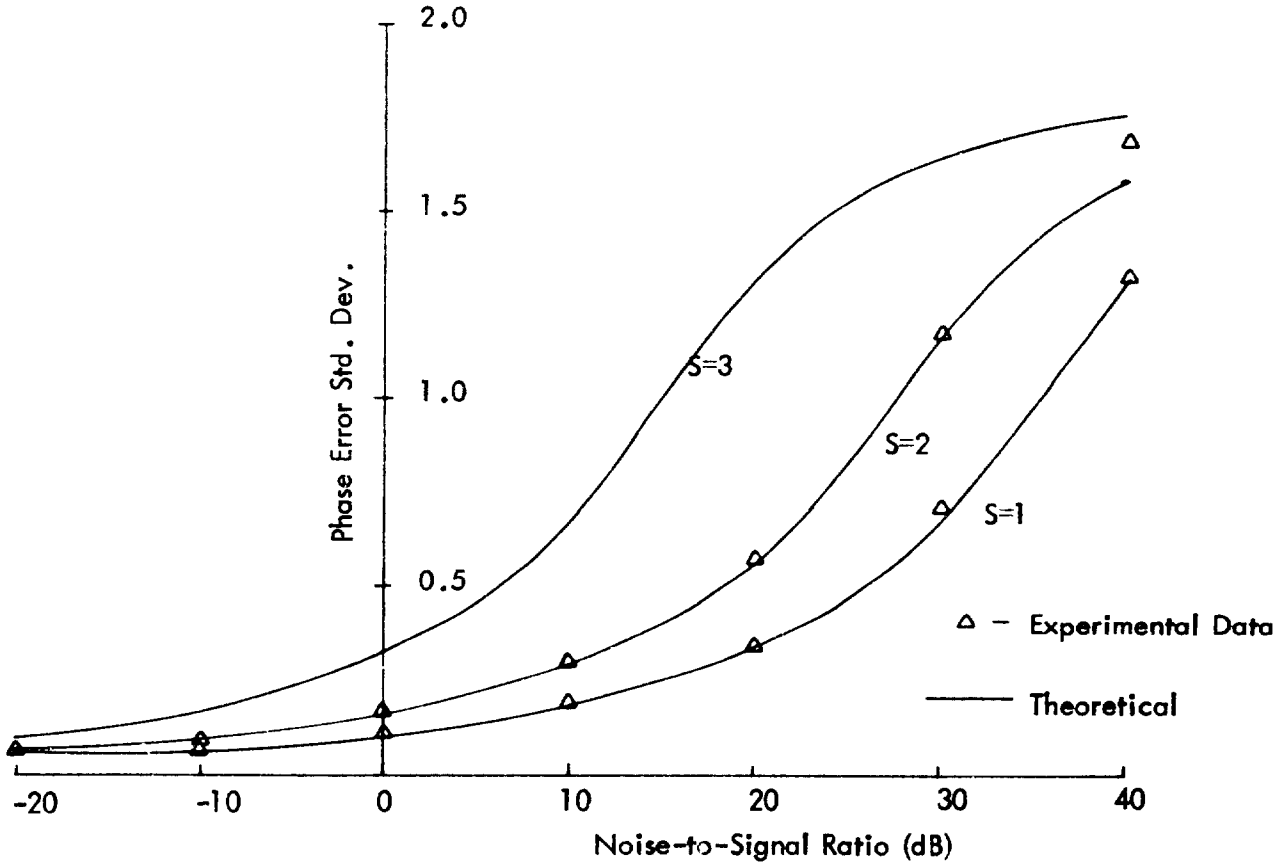

Figure 7. Loop Phase Jitter, M = 2, N = 6.

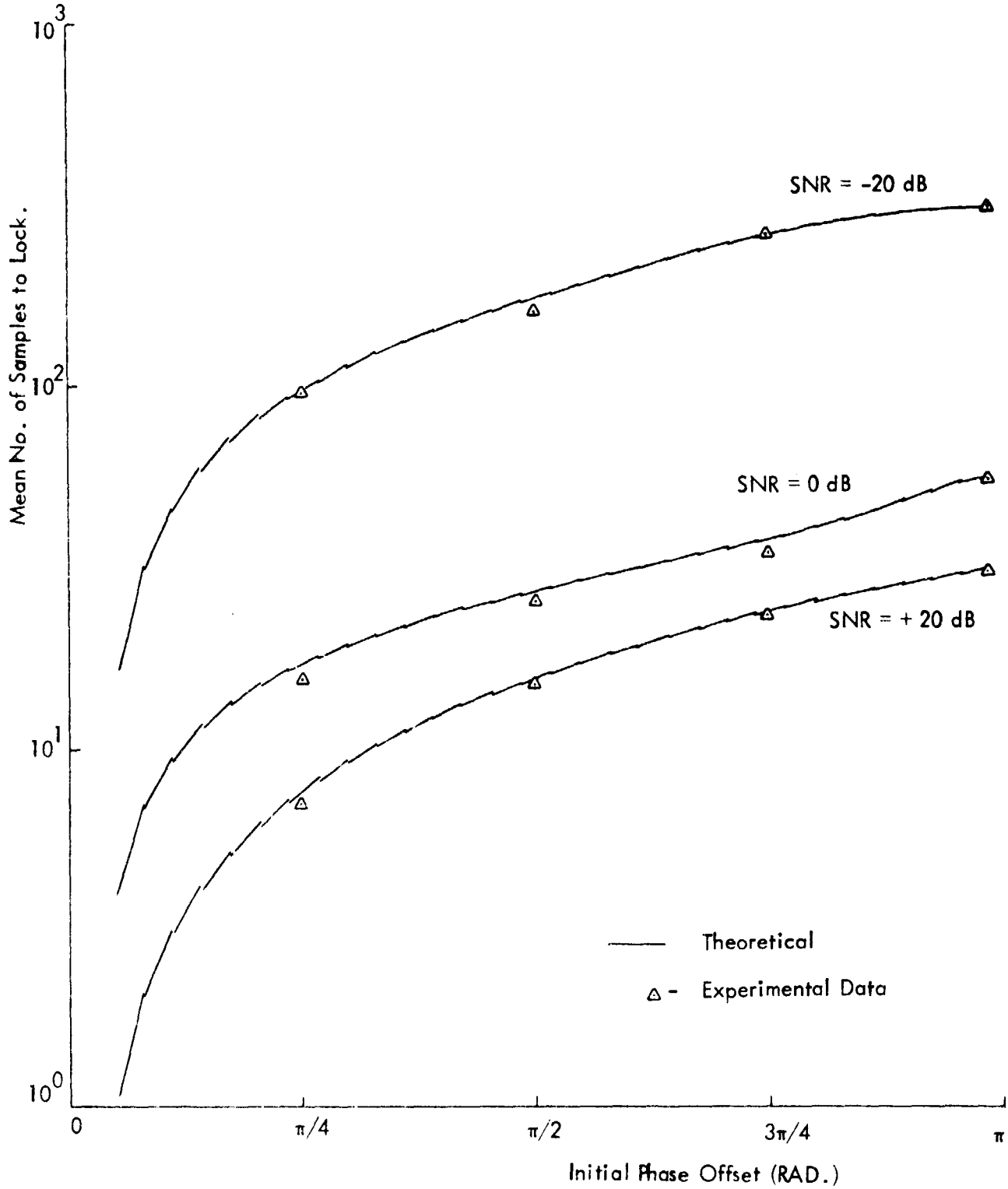

Figure 8. Loop Transient Response, M=0, S=1, N=6

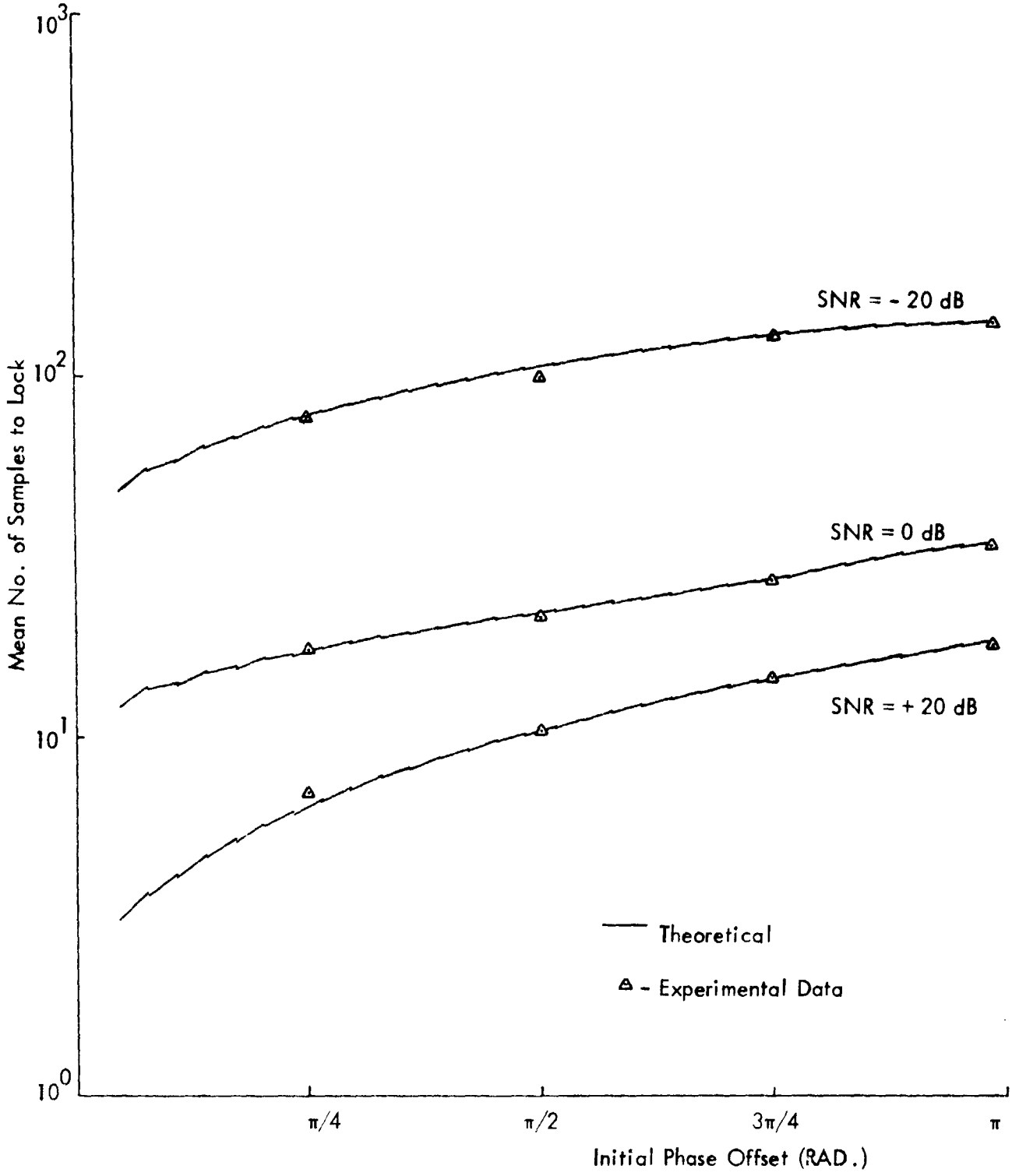

Figure 9. Loop Transient Response, M=0, N=6, S=2.

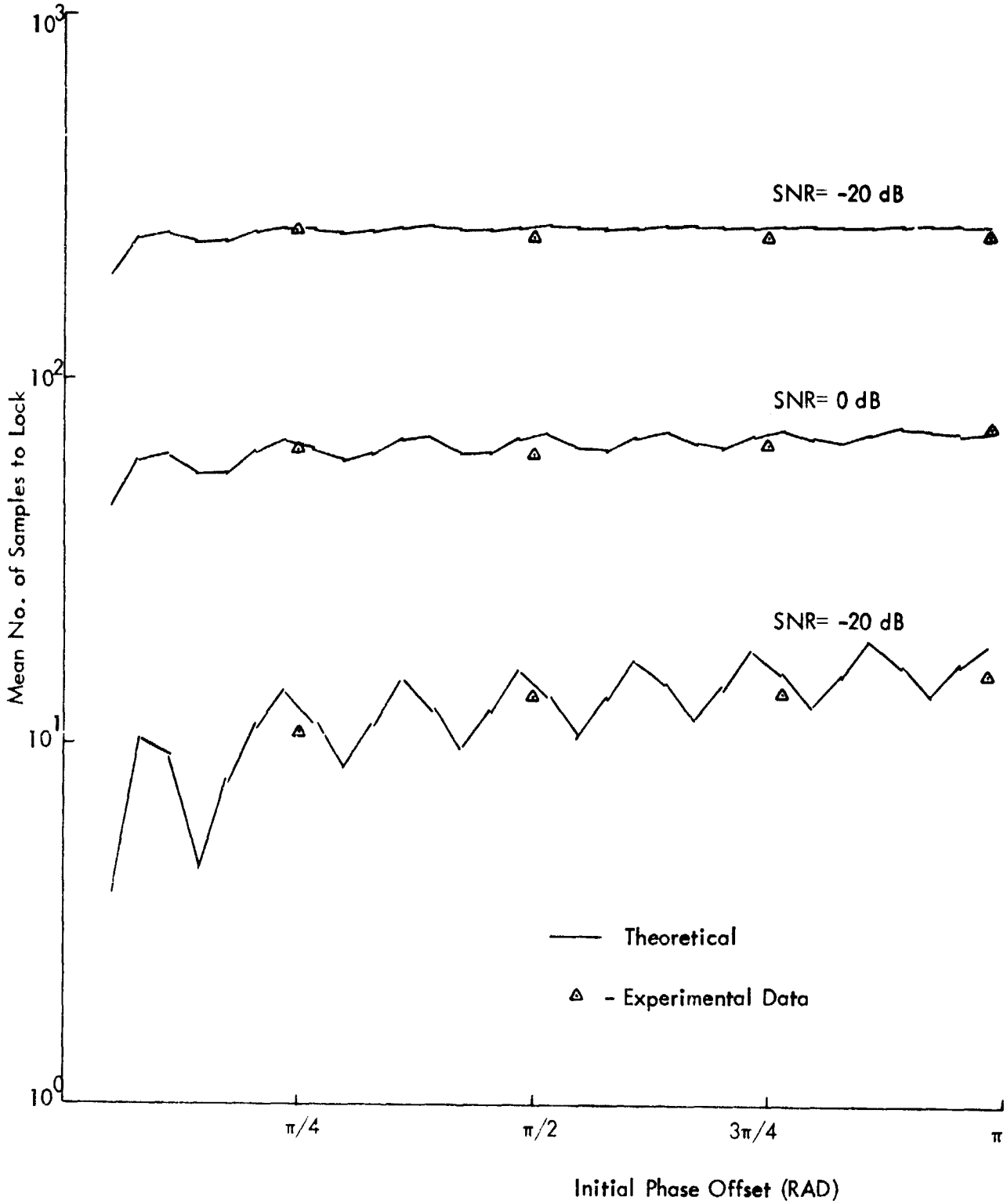

Figure 10. Loop Transient Response, M=0, N=6, S=3.

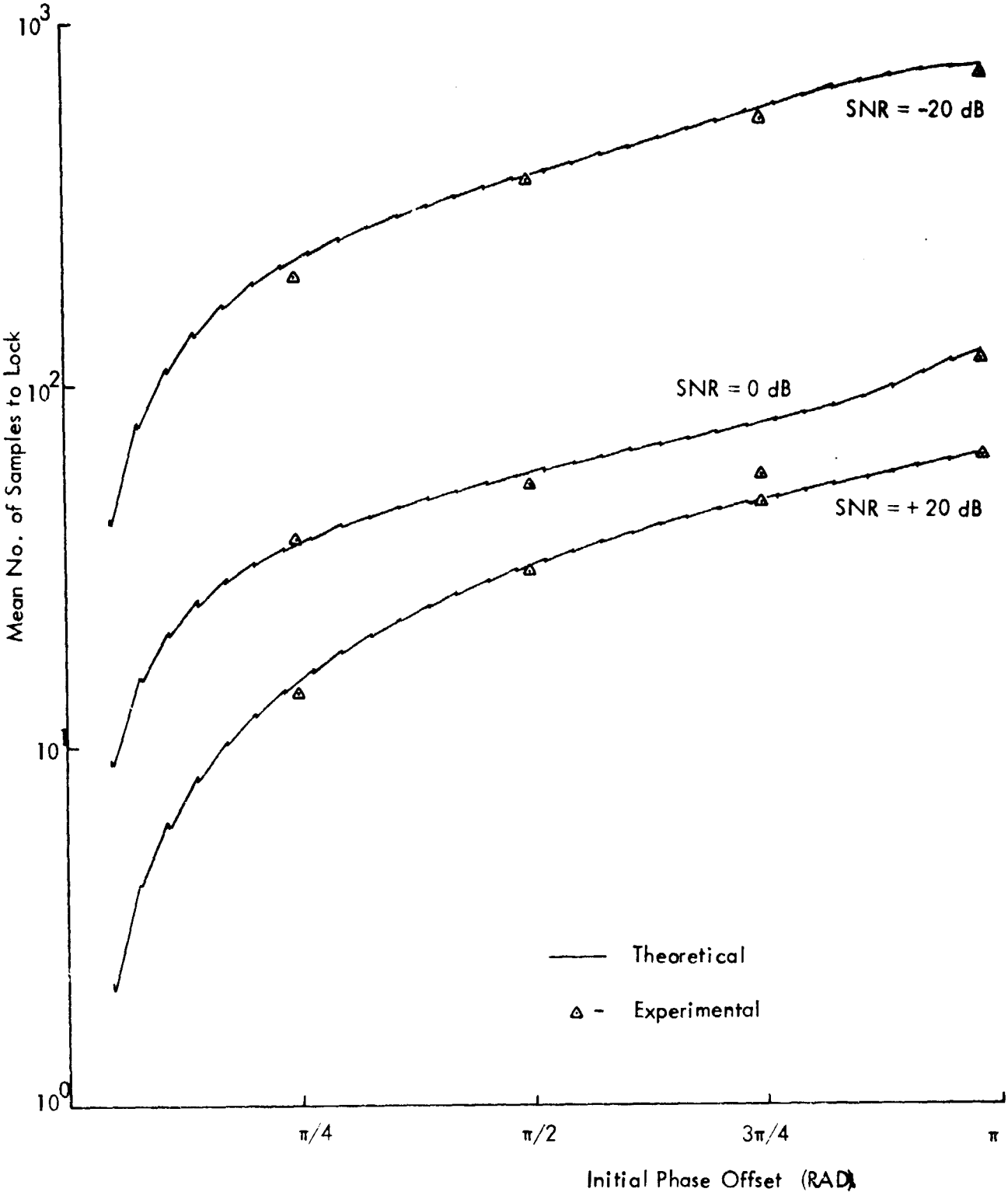

Figure 11. Loop Transient Response, M=1, N=6, S=1.

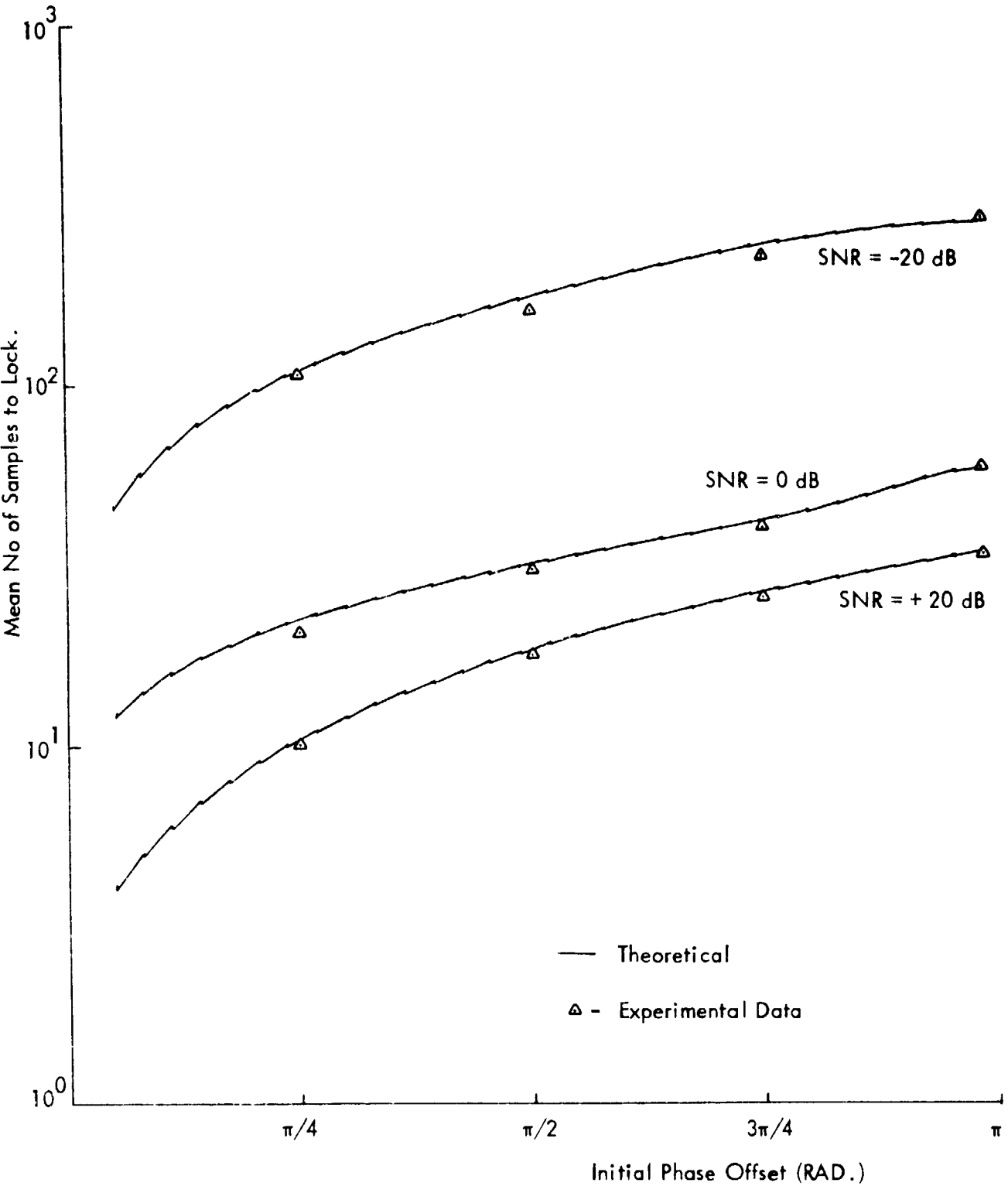

Figure 12. Loop Transient Response, M=1, N=6, S=2.

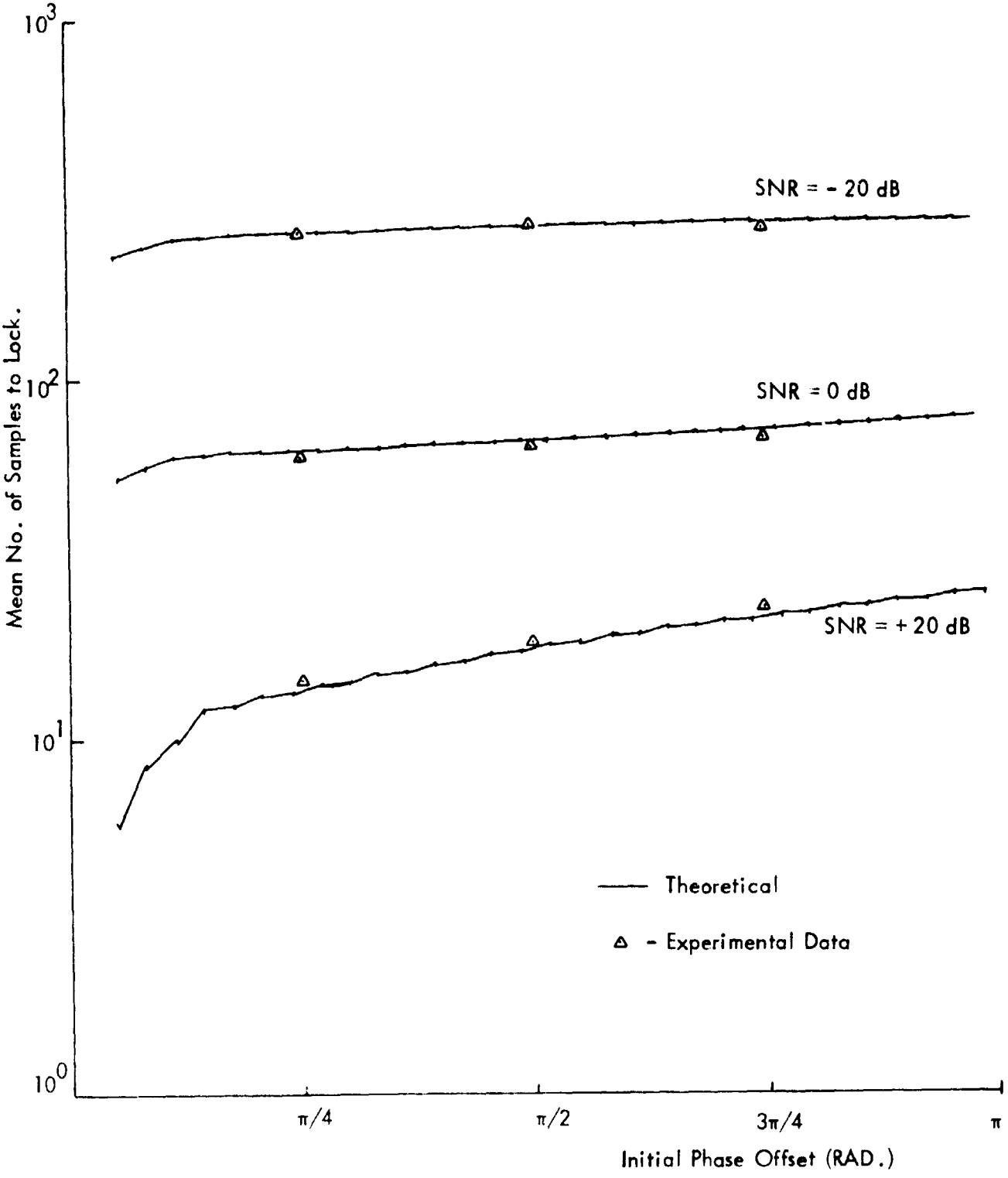

Figure 13. Loop Transient Response, M=1, N=6, S=3.

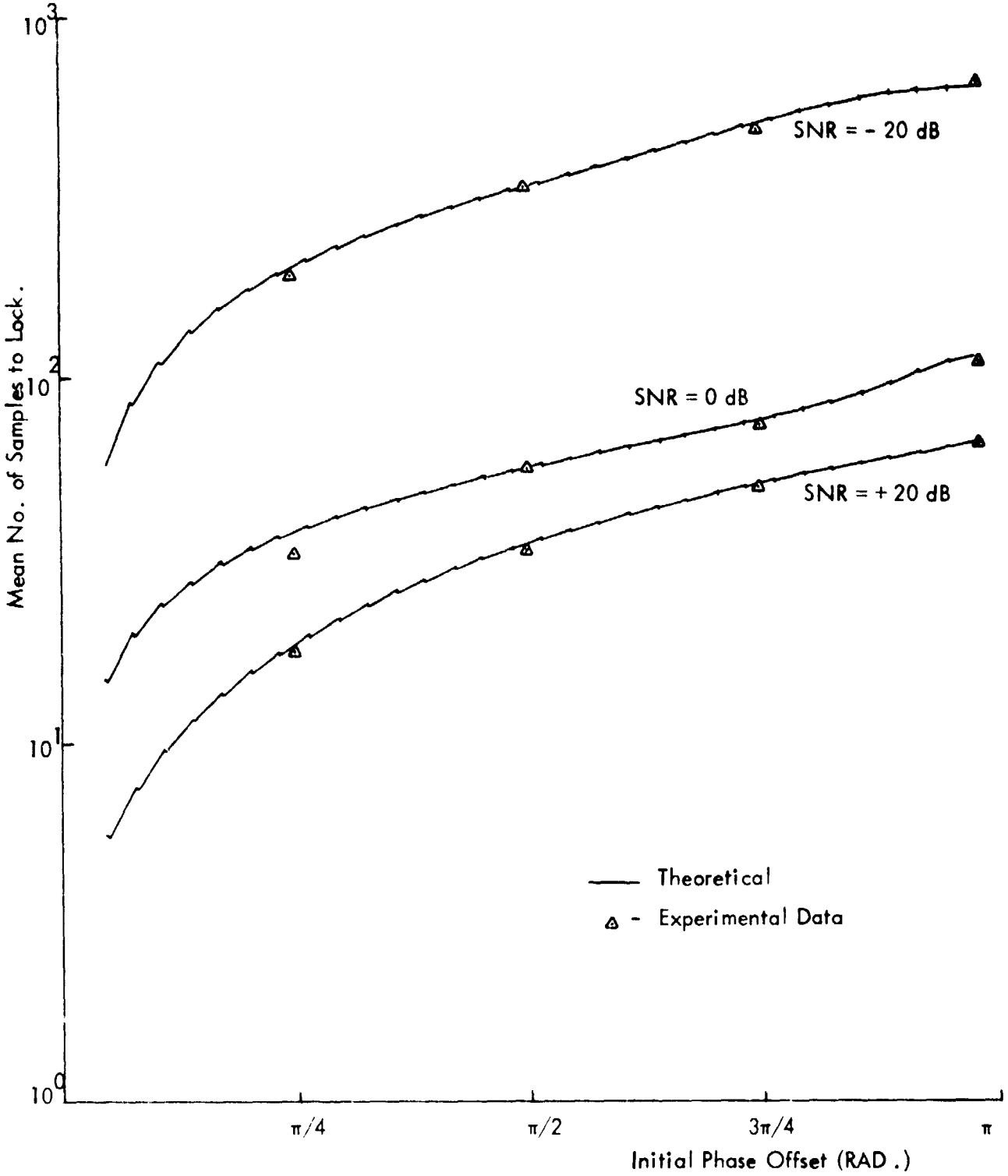

Figure 14. Loop Transient Response, M=2, N=6, S=2.

VII. APPENDICES

A. Fortran Programs Used for Solution of Markov Chain Model. Reference Section III - C and D.

```
C*****************************************************************PBS00010
C                                                                 PBS00020
C    CALCULATION OF STATE TRANSFER VECTORS FOR DPLL               PBS00030
C    MARKOV CHAIN MODELS.  THIS PROGRAM TO BE USED IN             PBS00040
C    CONJUNCTION WITH PBDPL2 AND PBDPL2T.                         PBS00050
C                                                                 PBS00060
C*****************************************************************PBS00070
      INTEGER PTR1(7168),PTR2(7168),IN(7,512),ALT,ALT2            PBS00080
      INTEGER T1(7168),T2(7168)                                   PBS00090
C                                                                 PBS00100
C     SET DPLL PARAMETERS                                         PBS00110
C                                                                 PBS00120
      M=1                                                         PBS00130
      N=4                                                         PBS00140
      ALT=3                                                       PBS00150
      WRITE(8,2)N,M,ALT                                           PBS00160
      WRITE(9,2)N,M,ALT                                           PBS00170
    2 FORMAT(1X,'N=',I3,5X,'M=',I3,5X,'ALT=',I3)                  PBS00180
      MN=M*N                                                      PBS00190
      DO 10 I=1,ALT                                               PBS00200
      K=I                                                         PBS00210
      DO 20 J=1,MN                                                PBS00220
      IN(I,J)=K                                                   PBS00230
   20 K=K+ALT                                                     PBS00240
   10 CONTINUE                                                    PBS00250
      L=1                                                         PBS00260
      RALT=(ALT+1.)/2.                                            PBS00270
      ALT2=ALT/2                                                  PBS00280
      K=1                                                         PBS00290
      DO 40 J=1,MN                                                PBS00300
      DO 30 I=1,ALT                                               PBS00310
      RI=I                                                        PBS00320
      IF(RI-RALT) 50,60,70                                        PBS00330
   50 IF(I.EQ.1) GO TO 51                                         PBS00340
      IK=I-1                                                      PBS00350
      JK=J+ALT2+1                                                 PBS00360
      IK1=I+1                                                     PBS00370
      JK1=J+ALT2-1-I                                              PBS00380
      GO TO 80                                                    PBS00390
   51 IK=I                                                        PBS00400
      JK=J+ALT2+1                                                 PBS00410
      IK1=I+1                                                     PBS00420
      JK1=J+ALT2-2                                                PBS00430
      GO TO 80                                                    PBS00440
   60 IF(ALT.EQ.1) GO TO 61                                       PBS00450
      IK=I-1                                                      PBS00460
      JK=J+2                                                      PBS00470
      IK1=I+1                                                     PBS00480
      JK1=J-2                                                     PBS00490
      GO TO 80                                                    PBS00500
   61 IK=I                                                        PBS00510
      JK=J+1                                                      PBS00520
      IK1=I                                                       PBS00530
      JK1=J-1                                                     PBS00540
      GO TO 80                                                    PBS00550
```

```
C*****PBS00010
C                                                                      PBS00020
C     CALCULATION OF STATE TRANSFER VECTORS FOR DPLL                   PBS00030
C     MARKOV CHAIN MODEL.  THIS PROGRAM TO BE USED IN                  PBS00040
C     CONJUNCTION WITH PBDPL2 AND PBDPL2T.                             PBS00050
C                                                                      PBS00060
C*****PBS00070
      INTEGER PTR1(7168),PTR2(7168),IN(7,512),ALT,ALT2                 PBS00080
      INTEGER T1(7168),T2(7168)                                        PBS00090
C                                                                      PBS00100
C     SET DPLL PARAMETERS                                              PBS00110
C                                                                      PBS00120
      M=1                                                              PBS00130
      N=4                                                              PBS00140
      ALT=3                                                            PBS00150
      WRITE(8,2)N,M,ALT                                                PBS00160
      WRITE(9,2)N,M,ALT                                                PBS00170
    2 FORMAT(1X,'N=',I3,5X,'M=',I3,5X,'ALT=',I3)                       PBS00180
      MN=M*N                                                           PBS00190
      DO 10 I=1,ALT                                                    PBS00200
      K=I                                                              PBS00210
      DO 20 J=1,MN                                                     PBS00220
      IN(I,J)=K                                                        PBS00230
   20 K=K+ALT                                                          PBS00240
   10 CONTINUE                                                         PBS00250
      L=1                                                              PBS00260
      RALT=(ALT+1.)/2.                                                 PBS00270
      ALT2=ALT/2                                                       PBS00280
      K=1                                                              PBS00290
      DO 40 J=1,MN                                                     PBS00300
      DO 30 I=1,ALT                                                    PBS00310
      RI=I                                                             PBS00320
      IF(RI-RALT) 50,60,70                                             PBS00330
   50 IF(I.EQ.1) GO TO 51                                              PBS00340
      IK=I-1                                                           PBS00350
      JK=J+ALT2+1                                                      PBS00360
      IK1=I+1                                                          PBS00370
      JK1=J+ALT2-1-I                                                   PBS00380
      GO TO 80                                                         PBS00390
   51 IK=I                                                             PBS00400
      JK=J+ALT2+1                                                      PBS00410
      IK1=I+1                                                          PBS00420
      JK1=J+ALT2-2                                                     PBS00430
      GO TO 80                                                         PBS00440
   60 IF(ALT.EQ.1) GO TO 61                                            PBS00450
      IK=I-1                                                           PBS00460
      JK=J+2                                                           PBS00470
      IK1=I+1                                                          PBS00480
      JK1=J-2                                                          PBS00490
      GO TO 80                                                         PBS00500
   61 IK=I                                                             PBS00510
      JK=J+1                                                           PBS00520
      IK1=I                                                            PBS00530
      JK1=J-1                                                          PBS00540
      GO TO 80                                                         PBS00550
```

```
 70 IF(I.EQ.ALT) GO TO 71                                          PBS00560
    IK=I-1                                                         PBS00570
    JK=J-I+ALT2+3                                                  PBS00580
    IK1=I+1                                                        PBS00590
    JK1=J-I+ALT2-1                                                 PBS00600
    GO TO 80                                                       PBS00610
 71 IK=I-1                                                         PBS00620
    JK=J-ALT2+2                                                    PBS00630
    IK1=I                                                          PBS00640
    JK1=J-ALT2-1                                                   PBS00650
 80 IF(JK.GT.MN) JK=JK-MN                                          PBS00660
    IF(JK1.GT.MN) JK1=JK1-MN                                       PBS00670
    IF(JK.LT.1) JK=JK+MN                                           PBS00680
    IF(JK1.LT.1) JK1=JK1+MN                                        PBS00690
    PTR1(K)=IM(IK,JK)                                              PBS00700
    PTR1(K+1)=IM(IK1,JK1)                                          PBS00710
    K=K+2                                                          PBS00720
 30 CONTINUE                                                       PBS00730
    IF(MOD(J,M).EQ.0) L=L+1                                        PBS00740
 40 CONTINUE                                                       PBS00750
    NST=K-1                                                        PBS00760
    K=1                                                            PBS00770
    LC=-1                                                          PBS00780
    LC1=0                                                          PBS00790
 93 DO 90 I=1,NST                                                  PBS00800
    IF(MOD(I,2).NE.0) GO TO 94                                     PBS00810
    IF(LC1.NE.1) GO TO 92                                          PBS00820
 94 IF(PTR1(I).NE.(M-1)/2) GO TO 92                                PBS00830
    T1(K)=(I+1)/2                                                  PBS00840
    PTR2(K)=(((T1(K)+M-1)/2)+ALT-1)/ALT                            PBS00850
    K=K+1                                                          PBS00860
    LC=LC+1                                                        PBS00870
    LC1=1                                                          PBS00880
 92 IF(LC.NE.1) GO TO 90                                           PBS00890
    LC=-1                                                          PBS00900
    LC1=0                                                          PBS00910
    GO TO 93                                                       PBS00920
 90 CONTINUE                                                       PBS00930
    IF(K.GT.NST) GO TO 96                                          PBS00940
    LC1=1                                                          PBS00950
    GO TO 93                                                       PBS00960
 96 DO 100 I=2,NST,2                                               PBS00970
100 PTR2(I)=PTR2(I)+N                                              PBS00980
    IF(ALT.EQ.1) GO TO 101                                         PBS00990
    MM=2*ALT                                                       PBS01000
    DO 110 I=2,NST,MM                                              PBS01010
110 PTR2(I)=PTR2(I)-N                                              PBS01020
    MM1=MM-1                                                       PBS01030
    DO 120 I=MM1,NST,MM                                            PBS01040
120 PTR2(I)=PTR2(I)+N                                              PBS01050
101 DO 95 I=1,NST                                                  PBS01060
    T2(I)=PTR1(I)                                                  PBS01070
    PTR1(I)=T1(I)                                                  PBS01080
 95 T1(I)=T2(I)                                                    PBS01090
C                                                                  PBS01100
```

```
C     STADY STATE DISTRIBUTION DATA                                    PBS01110
C      PTR1=STATE TO STATE TRANSITION VECTOR.                          PBS01120
C      PTR2=PROBABILITIES VECTOR FOR STATE TO STATE TRANSITIONS.       PBS01130
C                                                                      PBS01140
      WRITE(8,1)(PTR1(I),I=1,NST)                                      PBS01150
      WRITE(8,1)(PTR2(I),I=1,NST)                                      PBS01160
    1 FORMAT(1X,20I5)                                                  PBS01170
      K=1                                                              PBS01180
      DO 130 I=1,NST,2                                                 PBS01190
      T2(I)=K                                                          PBS01200
      T2(I+1)=K+N                                                      PBS01210
      KK=MOD((I+1)/2,M*ALT)                                            PBS01220
      IF(KK.EQ.0) K=K+1                                                PBS01230
  130 CONTINUE                                                         PBS01240
C                                                                      PBS01250
C     MEAN TRANSIENT RESPONSE DATA                                     PBS01260
C     T1=STATE TO STATE TRANSITION VECTOR                              PBS01270
C     T2=PROBABLITY VECTOR FOR STATE TO STATE TRANSITIONS              PBS01280
C                                                                      PBS01290
      WRITE(9,1)(T1(I),I=1,NST)                                        PBS01300
      WRITE(9,1)(T2(I),I=1,NST)                                        PBS01310
      STOP                                                             PBS01320
      END                                                              PBS01330
```

```
C*****************************************************************PBD00010
C                                                                 PBD00020
C     DETERMINATION OF LOOP PHASE PDF AND STEADY-STATE ERROR      PBD00030
C     FROM MARKOV CHAIN MODEL.                                    PBD00040
C                                                                 PBD00050
C*****************************************************************PBD00060
      IMPLICIT REAL*8(A-H,O-Z)                                    PBD00070
      INTEGER POINT1(7168),POINT2(7168),ALT                       PBD00080
      DIMENSION POLD(3584),PNEW(3584),A0(64),A1(64),A(128),THETA(64)  PBD00090
      DIMENSION PHASE(64),PTEMP(3584)                             PBD00100
C                                                                 PBD00110
C     SET DPLL PARAMETERS                                         PBD00120
C                                                                 PBD00130
      READ(8,7) N,M,ALT                                           PBD00140
    7 FORMAT(3X,I3,7X,I3,9X,I3)                                   PBD00150
      SNR=-40.                                                    PBD00160
      MN=M*N                                                      PBD00170
      IST=ALT*M*N                                                 PBD00180
      NST=2*IST                                                   PBD00190
      DO 20 I=1,IST                                               PBD00200
   20 POLD(I)=1./IST                                              PBD00210
C                                                                 PBD00220
C     READ STATE TRANSFER VECTORS                                 PBD00230
C                                                                 PBD00240
      READ(8,1)(POINT1(I),I=1,NST)                                PBD00250
      READ(8,1)(POINT2(I),I=1,NST)                                PBD00260
    1 FORMAT(1X,20I5)                                             PBD00270
C                                                                 PBD00280
C     CALCULATE STATE TRANSFER PROBABILITIES                      PBD00290
C                                                                 PBD00300
      DO 500 MM=1,13                                              PBD00310
      PI=3.14159265                                               PBD00320
      AC=DSQRT(2.D0)*10.**(SNR/20.)                               PBD00330
      PSI=-PI+PI/N                                                PBD00340
      DO 10 I=1,N                                                 PBD00350
      RMEAN=AC*DSIN(PSI)/DSQRT(2.D0)                              PBD00360
      A1(I)=0.5+0.5*DERF(RMEAN)                                   PBD00370
      A0(I)=1.-A1(I)                                              PBD00380
      THETA(I)=PSI                                                PBD00390
   10 PSI=PSI+2.*PI/N                                             PBD00400
      DO 12 I=1,N                                                 PBD00410
      A(I)=A0(I)                                                  PBD00420
      A(I+N)=A1(I)                                                PBD00430
   12 CONTINUE                                                    PBD00440
      JJ=1                                                        PBD00450
C                                                                 PBD00460
C     CALCULATE STEADY-STATE LOOP STATE PROBABILITIES             PBD00470
C     BY ITERATION.                                               PBD00480
C                                                                 PBD00490
   30 DIFF=0.0                                                    PBD00500
      JJ=JJ+1                                                     PBD00510
      K=1                                                         PBD00520
      J=1                                                         PBD00530
      DO 40 I=1,IST                                               PBD00540
      PNEW(I)=A(POINT2(J))*POLD(POINT1(J))+A(POINT2(J+1))*        PBD00550
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     1POLD(POINT1(J+1))                                                 PBD00560
      IF(PNEW(I).LT.1.D-20) PNEW(I)=0.000                               PBD00570
   41 J=J+2                                                             PBD00580
   40 CONTINUE                                                          PBD00590
      SUM=0.                                                            PBD00600
      DO 50 I=1,IST                                                     PBD00610
   50 SUM=SUM+PNEW(I)                                                   PBD00620
      DO 60 I=1,IST                                                     PBD00630
   60 PNEW(I)=PNEW(I)/SUM                                               PBD00640
      DO 61 I=1,IST                                                     PBD00650
   61 PTEMP(I)=POLD(I)                                                  PBD00660
      DO 70 I=1,IST                                                     PBD00670
      CNG=DABS(PNEW(I)-POLD(I))                                         PBD00680
      IF(CNG.GT.DIFF) DIFF=CNG                                          PBD00690
   70 POLD(I)=PNEW(I)                                                   PBD00700
      IF(JJ.EQ.10000) GO TO 80                                          PBD00710
      IF(DIFF.GT.1.E-6) GO TO 31                                        PBD00720
   80 SUM=0.                                                            PBD00730
C                                                                       PBD00740
C       CALCULATE STEADY-STATE LOOP PHASE PROBABILITIES                 PBD00750
C       AND STANDARD DEVIATION OF LOOP PHASE.                           PBD00760
C                                                                       PBD00770
      STD=0.                                                            PBD00780
      PSI=-PI+PI/N                                                      PBD00790
      K=1                                                               PBD00800
      DO 90 I=1,IST                                                     PBD00810
      SUM=SUM+POLD(I)                                                   PBD00820
      IF(MOD(I,M*ALT).NE.0) GO TO 90                                    PBD00830
      PHASE(K)=SUM                                                      PBD00840
      K=K+1                                                             PBD00850
      STD=STD+SUM*PSI**2                                                PBD00860
      SUM=0.                                                            PBD00870
      PSI=PSI+2.*PI/N                                                   PBD00880
   90 CONTINUE                                                          PBD00890
      STD=DSQRT(STD)                                                    PBD00900
   91 WRITE(6,2)                                                        PBD00910
    2 FORMAT('1','DPLL PARAMETERS')                                     PBD00920
      WRITE(6,8) SNR,N,M,ALT                                            PBD00930
    8 FORMAT(1X,'SNR=',F7.1,5X,'N=',I3,5X,'M=',I3,5X,'ALT=',I3,/)       PBD00940
      WRITE(6,3) STD,JJ,DIFF                                            PBD00950
    3 FORMAT(1X,'STD DEV=',F8.3,5X,'NO. OF ITER=',I6,5X,'DIFF=',E12.4,/)PBD00960
      WRITE(6,4)(THETA(I),PHASE(I),I=1,N)                               PBD00970
    4 FORMAT(1X,'PHASE DIFF=',F12.4,5X,'PDF=',E12.4)                    PBD00980
  500 SNR=SNR+5                                                         PBD00990
  100 STOP                                                              PBD01000
      END                                                               PBD01010
```

```
C**********************************************************************PBD00010
C                                                                      PBD00020
C   DETERMINATION OF MEAN TIME TO LOCK FOR INITIAL PHASE OFFSET        PBD00030
C   FROM MARKOV CHAIN MODEL.                                           PBD00040
C                                                                      PBD00050
C**********************************************************************PBD00060
      IMPLICIT REAL*8(A-H,O-Z)                                         PBD00070
      INTEGER POINT1(7168),POINT2(7168),FLI                            PBD00080
      DIMENSION POLD(3584),PNEW(3584),A0(64),A1(64),A(128),THETA(64)   PBD00090
      DIMENSION PHASE(64),PTEMP(3584)                                  PBD00100
                                                                       PBD00110
C     SET DPLL PARAMETERS                                              PBD00120
                                                                       PBD00130
      READ(8,7) N,M,ALT                                                PBD00140
    7 FORMAT(3X,I3,7X,I3,3X,I3)                                        PBD00150
      SNR=20.                                                          PBD00160
      MN=M*N                                                           PBD00170
      IST=ALT*M*N                                                      PBD00180
      NST=2*IST                                                        PBD00190
      DO 20 I=1,IST                                                    PBD00200
   20 POLD(I)=1.                                                       PBD00210
      JL=IST/2-(ALT/2)                                                 PBD00220
      JL1=JL-(M-1)*ALT                                                 PBD00230
      JLU=JL1+((2*M)-1)*ALT                                            PBD00240
                                                                       PBD00250
C     READ STATE TRANSFER VECTORS                                      PBD00260
                                                                       PBD00270
      READ(8,1)(POINT1(I),I=1,NST)                                     PBD00280
      READ(8,1)(POINT2(I),I=1,NST)                                     PBD00290
    1 FORMAT(1X,20I5)                                                  PBD00300
                                                                       PBD00310
C     CALCULATE STATE TRANSFER PROBABILITIES                           PBD00320
                                                                       PBD00330
      DO 500 MM=1,13                                                   PBD00340
      PI=3.14159265                                                    PBD00350
      AC=DSQRT(2.D0)*10.**(SNR/20.)                                    PBD00360
      PSI=-PI+PI/N                                                     PBD00370
      DO 10 I=1,N                                                      PBD00380
      RMEAN=AC*DSIN(PSI)/DSQRT(2.DO)                                   PBD00390
      A1(I)=0.5+0.5*DERF(RMEAN)                                        PBD00400
      A0(I)=1.-A1(I)                                                   PBD00410
      THETA(I)=PSI                                                     PBD00420
   10 PSI=PSI+2.*PI/N                                                  PBD00430
      DO 12 I=1,N                                                      PBD00440
      A(I)=A0(I)                                                       PBD00450
      A(I+N)=A1(I)                                                     PBD00460
   12 CONTINUE                                                         PBD00470
      JJ=1                                                             PBD00480
                                                                       PBD00490
C     CALCULATE MEAN TIME TO LOCK                                      PBD00500
                                                                       PBD00510
   30 DIFF=0.0                                                         PBD00520
      JJ=JJ+1                                                          PBD00530
      K=1                                                              PBD00540
      J=1                                                              PBD00550
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 40 I=1,IST                                                     PBD00560
      PNEW(I)=A(POINT2(J))*POLD(POINT1(J))+A(POINT2(J+1))*              PBD00570
     1POLD(POINT1(J+1))+1.                                              PBD00580
      IF(PNEW(I).LT.1.D-20) PNEW(I)=0.0D0                               PBD00590
   41 J=J+2                                                             PBD00600
   40 CONTINUE                                                          PBD00610
      DO 71 JJ1=JL1,JLU,ALT                                             PBD00620
   71 PNEW(JJ1)=0.                                                      PBD00630
      DO 70 I=1,IST                                                     PBD00640
      CNG=DABS(PNEW(I)-POLD(I))                                         PBD00650
      IF(CNG.GT.DIFF) DIFF=CNG                                          PBD00660
   70 POLD(I)=PNEW(I)                                                   PBD00670
      IF(JJ.EQ.10000) GO TO 80                                          PBD00680
      IF(DIFF.GT.1.E-5) GO TO 30                                        PBD00690
   80 SUM=0.                                                            PBD00700
      K1=ALT/2+1                                                        PBD00710
      K2=2*ALT                                                          PBD00720
      J=1                                                               PBD00730
      DO 110 I=K1,IST,K2                                                PBD00740
      POLD(J)=PNEW(I)                                                   PBD00750
  110 J=J+1                                                             PBD00760
   91 WRITE(6,2)                                                        PBD00770
    2 FORMAT('1',*DPLL PARAMETERS*)                                     PBD00780
      WRITE(6,8) SNR,N,M,ALT                                            PBD00790
    8 FORMAT(1X,*SNR=*,F7.1,5X,*N=*,I3,5X,*M=*,I3,5X,*ALT=*,I3,/)       PBD00800
      WRITE(6,3) JJ,DIFF                                                PBD00810
    3 FORMAT(1X,*NO. OF ITER=*,I6,5X,*DIFF=*,E12.4,/)                   PBD00820
      WRITE(6,4)(THETA(I),POLD(I),I=1,N)                                PBD00830
    4 FORMAT(1X,*PHASE DIFF=*,E12.4,5X,*MEAN NO. SAMP.=*,E12.4)         PBD00840
  500 SNR=SNR-5                                                         PBD00850
  100 STOP                                                              PBD00860
      END                                                               PBD00870
```

ORIGINAL PAGE IS
OF POOR QUALITY

B. Binary Phase-Locked Loop Design.

The following sections discuss in detail the design of the hardware DPLL used to validate the Markov chain model. The block diagram for the DPLL may be found in Section IV, Figure 3 of the main text.

a. Binary Phase Detector. Referring to the schematic of Figure B-1, the binary phase detector operates by sampling the binary input signal f_c and producing a complemented pulse output on either the count-up or count-down line. This function is produced by three D-type flip-flops as follows. The OPEN SW signal is applied to the clock input of ff B3 while the binary signal f_c is applied to the data input of the same ff. Thus on a positive transition of OPEN SW the value of f_c is latched, giving the sampled value IN at B3's Q output. The two ff's of A6 are initailly set to the ONE state $_x$ so that when SET LAT latches the input values to the ff's (IN and $\overline{\text{IN}}$), one ff goes to the ZERO state while the other remains in the ONE state. The SET LAT signal is followed by the $\overline{\text{CLR LAT}}$ signal which sets both ff's to the ONE state. Thus the count-up or count-down are produced in a mutually exclusive manner when one of the ff's of A6 toggles HIGH LOW HIGH while the other remains HIGH.

b. S-bit Saturating Counter. The saturating up-down S-bit counter is shown in Figure B-2. The counting function is performed by two serially connected 74193 4-bit synchronous up-down counters. The ouput states of the counter is detected by a logic network to produce the $\overline{\text{INHIB UP}}$ and $\overline{\text{INHIB DN}}$ signals that will inhibit the UP and DN clock signals respectively. For example, if switches SW1 through SW6 are closed, then for a counter state of 0000 0001 the $\overline{\text{INHIB UP}}$ signal will be TRUE, thus inhibiting the UP clock signal and saturating the counter at that value. Note, a DN clock signal will still count the counter to the 0000 0000 state. Similarly, for all switches closed and the counter in state 1111 1111 the $\overline{\text{INHIB DN}}$ signal will be TRUE and the counter is saturated at that state. If switch SW1 is opened while all other switches remain closed, then the counter saturation states will be 0000 0011 and 1111 1101 (± 3). Continuing in this manner, the counter saturation states may be selected to be $\pm 2^i - 1$, $i = 1, 2, \ldots 7$ by opening switches SW1 through SW (i-1) while all other switches remain closed.

c. (M + N) -Bit Counter. The schematic for the (M + N)-bit counter along with the buffer register and adder are shown in Figure B-3. The (M + N)-bit counter consists of three serially connected 74193 4-bit up-down counters. Following an UP or DN clock signal the value of the (M + N)-bit counter is loaded into the 12-bit buffer register composed of two 74174 hex D-type flip-flops by the $\overline{\text{LOAD2}}$ signal. The values of the 12-bit buffer and the S-bit saturating counter compose the inputs to the adder circuit consisting of three 7483 4-bit binary adders. The ouput of the 12-bit adder is applied to the preset terminals of the (M + N) -bit counter. If the type select switch is set for first-order operation, the adder output will not be loaded into the (M + N)-bit counter. If a second-order loop is selected, the $\overline{\text{LOAD1}}$ signal will be applied to the load inputs of the 74193's thus presetting the (M + N)-bit counter to the value of the 12-bit adder output.

d. Variable Phase reference Clock. Referring to Figure B-4, six selectable contiguous bits from the (M + N)-bit counter are applied to one input side of a binary magnitude comparator formed by two 7485 4-bit magnitude comparators. The other input to the comparator is obtained from the lower N bits of the two series-connected 74193 counters that are being clocked at a rate of 2^N time the input frequency. Thus the EQUALS output pin 6 of A31, of the magnitude comparator will be a ONE when the two inputs are equal and this will occur at a rate equal the input frequency f_c. The circuit formed by A33 and A34 prevents the sample output from occurring at a rate greater than f_c. For example, if pin 8 of A34 is HIGH then when the magnitude comparator detects the equal condition, pin 5 of A34 is set HIGH causing the phase detector to sample the input signal. The SAMPLE signal is reset LOW by the $\overline{\text{LOADT}}$ signal at the end of the loop phase update. However, since the (M + N)-bit counter may have counted up during the last phase update and the counters of the reference clock have also counted up, it is possible for the magnitude comparator to detect equal states occurring at a rate of $2^N * f_c$. Thus it is necessary for the $\overline{\text{LOADT}}$ signal to reset pin 8 of A33 to a ZERO so that another sample cannot occur unit the reference clock counter loads a ZERO into that flip-flop.

e. Control Logic. A timing diagram for the necessary control waveforms and the logic schematic used in their generation are given in Figures B-5 and B-6, respectively. Use of the control waveforms has been described in the previous sections.

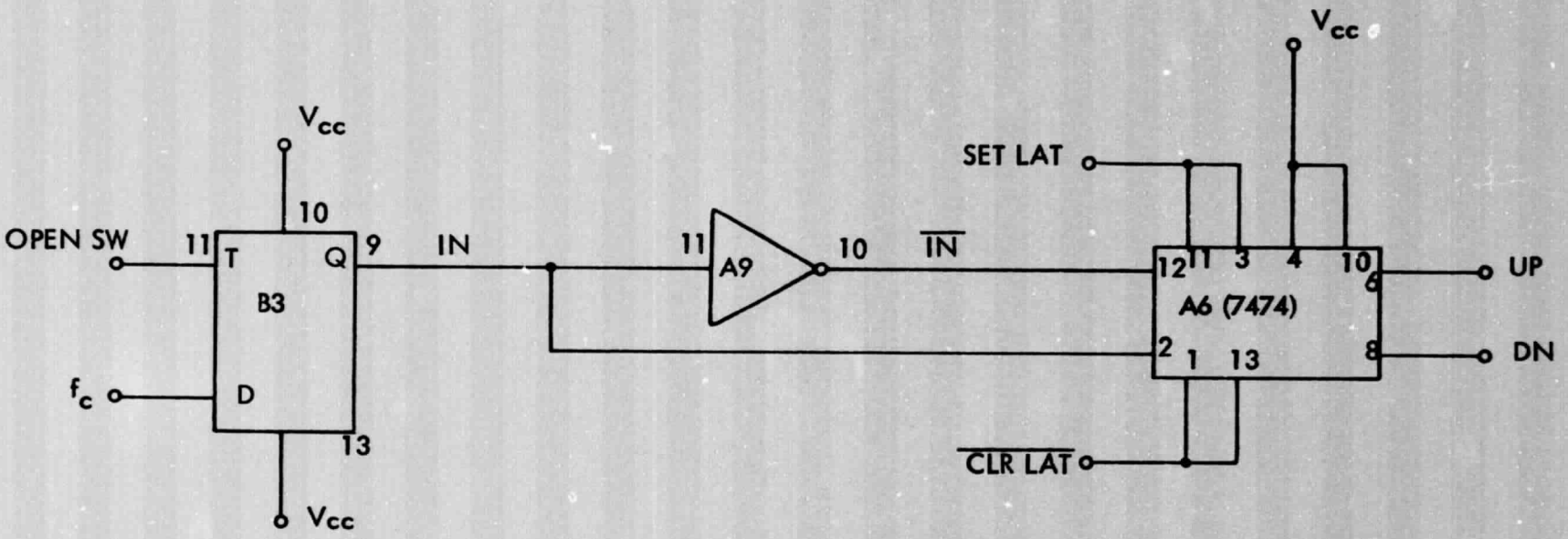

Figure B-1. DPLL Binary Phase Detector.

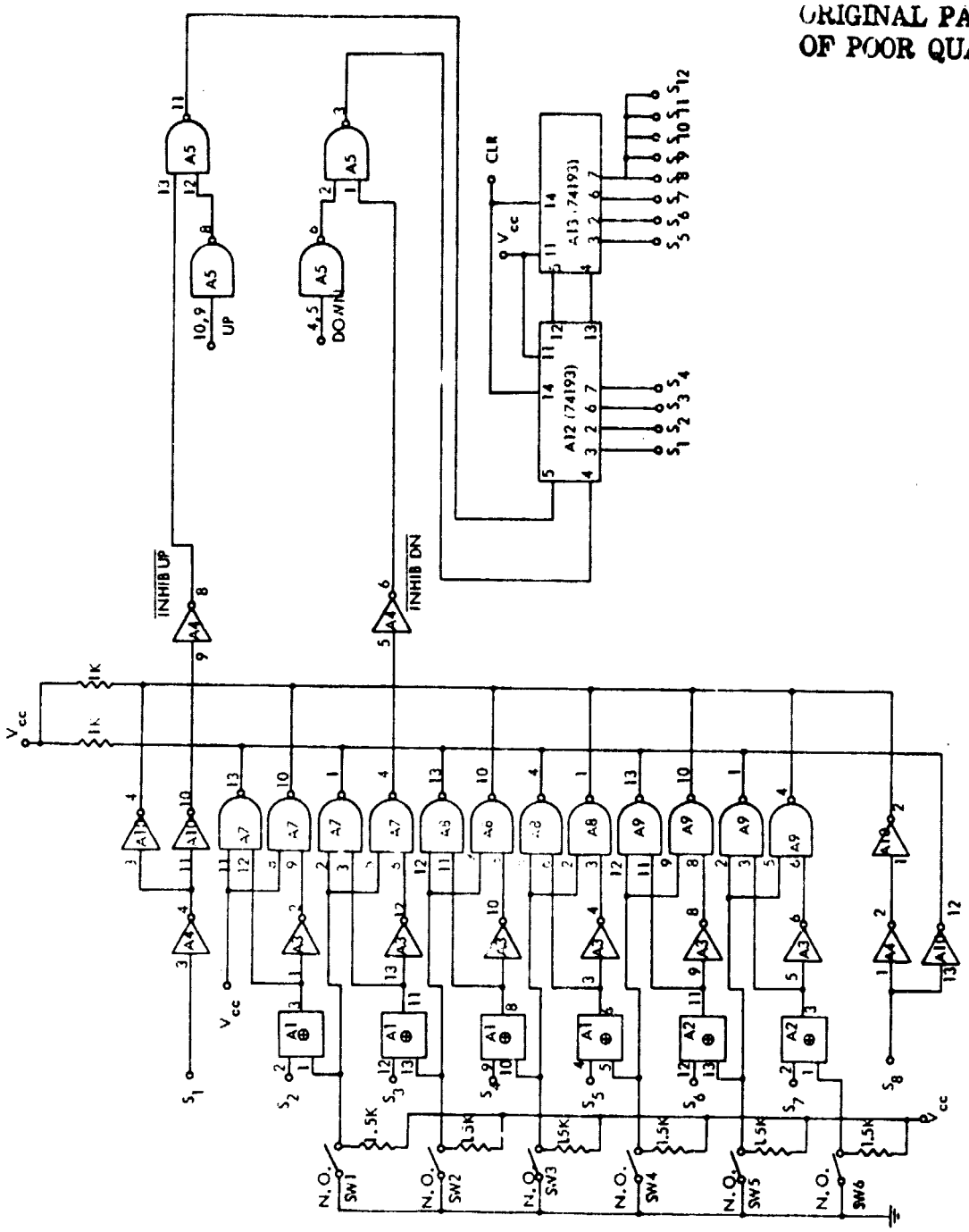

Figure B-2. 5-Bit Saturating Counter.

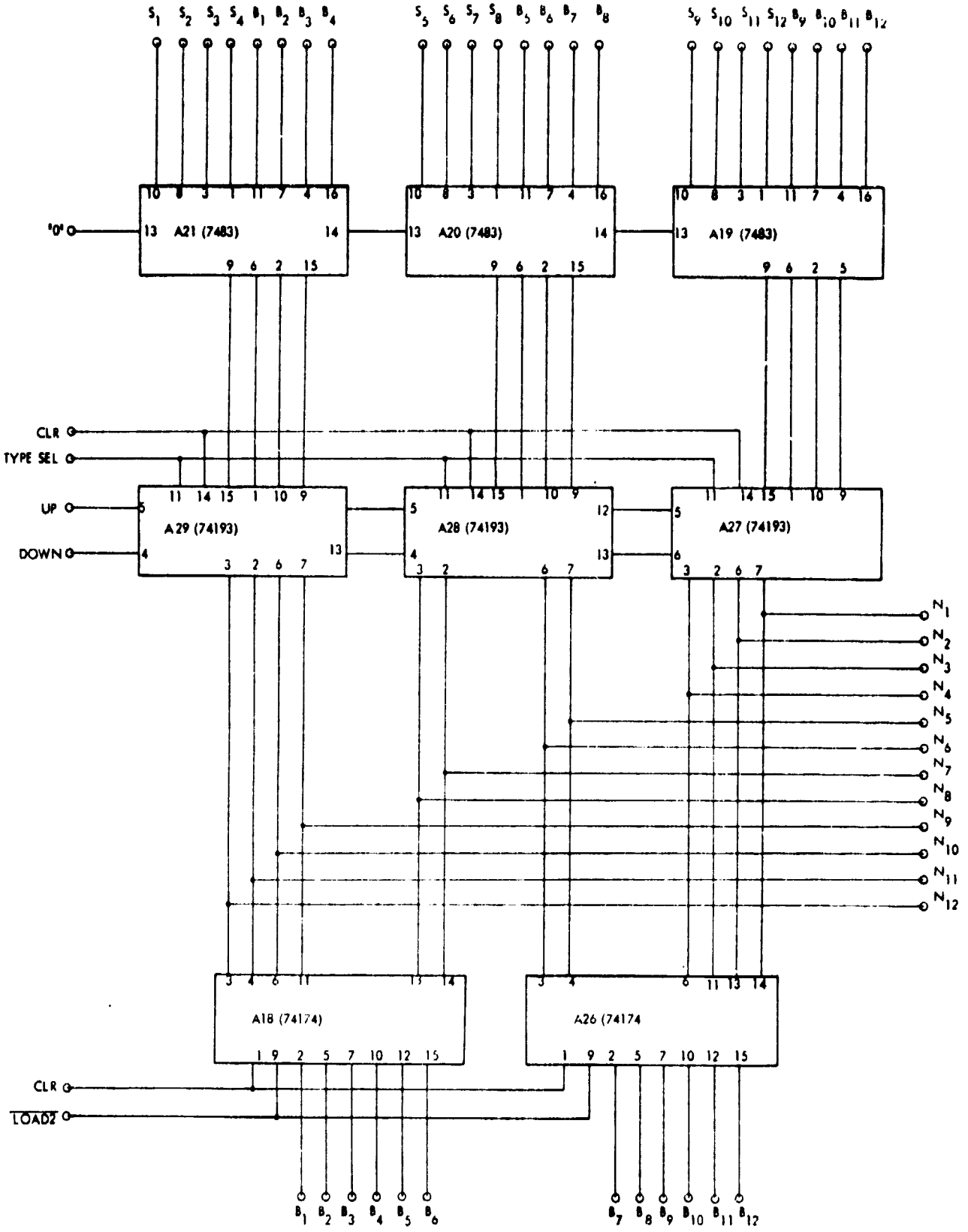

Figure B-3. (M+N) Bit Counter with Adder Function.

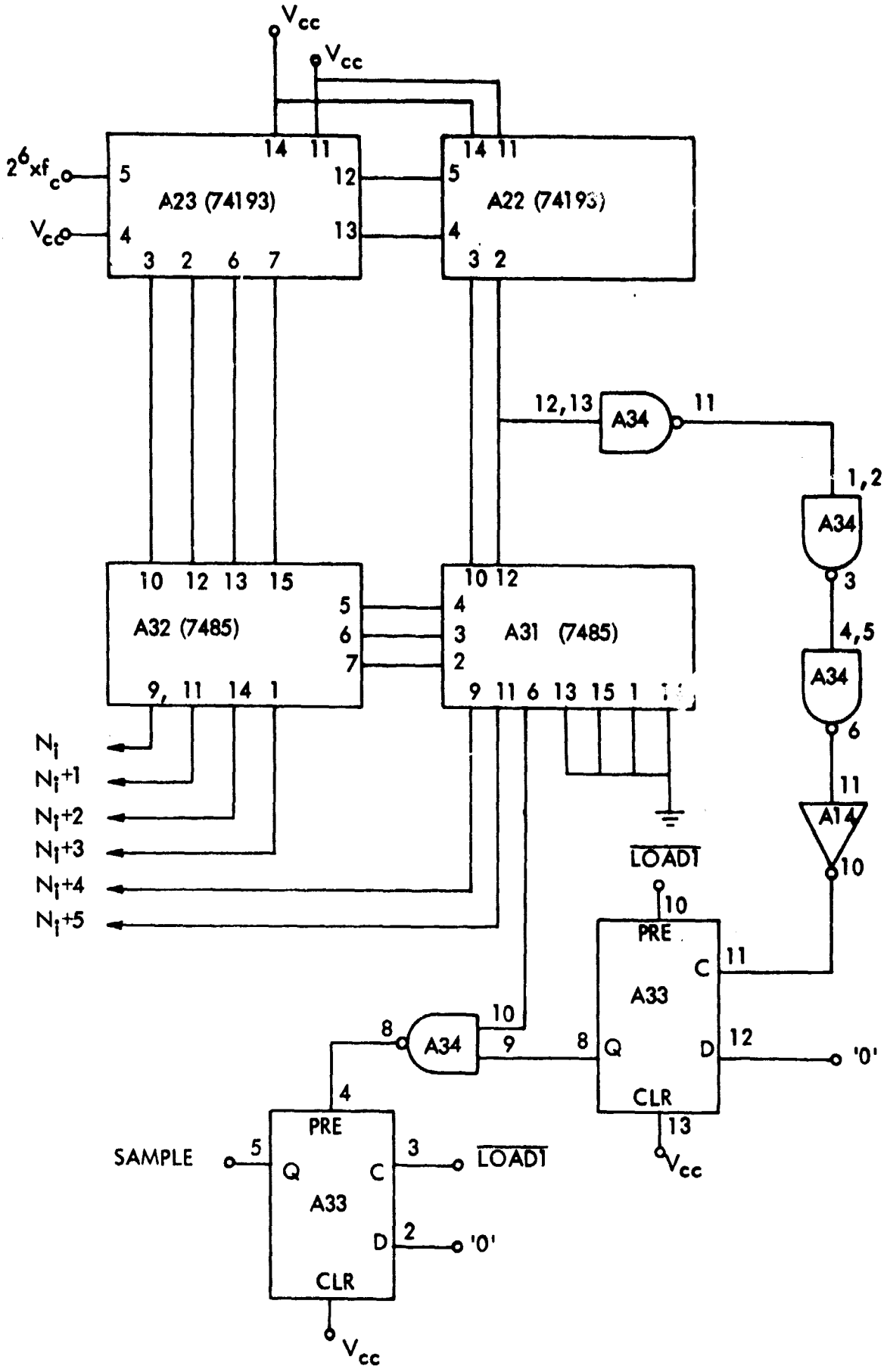

Figure B-4. DPLL Variable Phase Reference Clock.

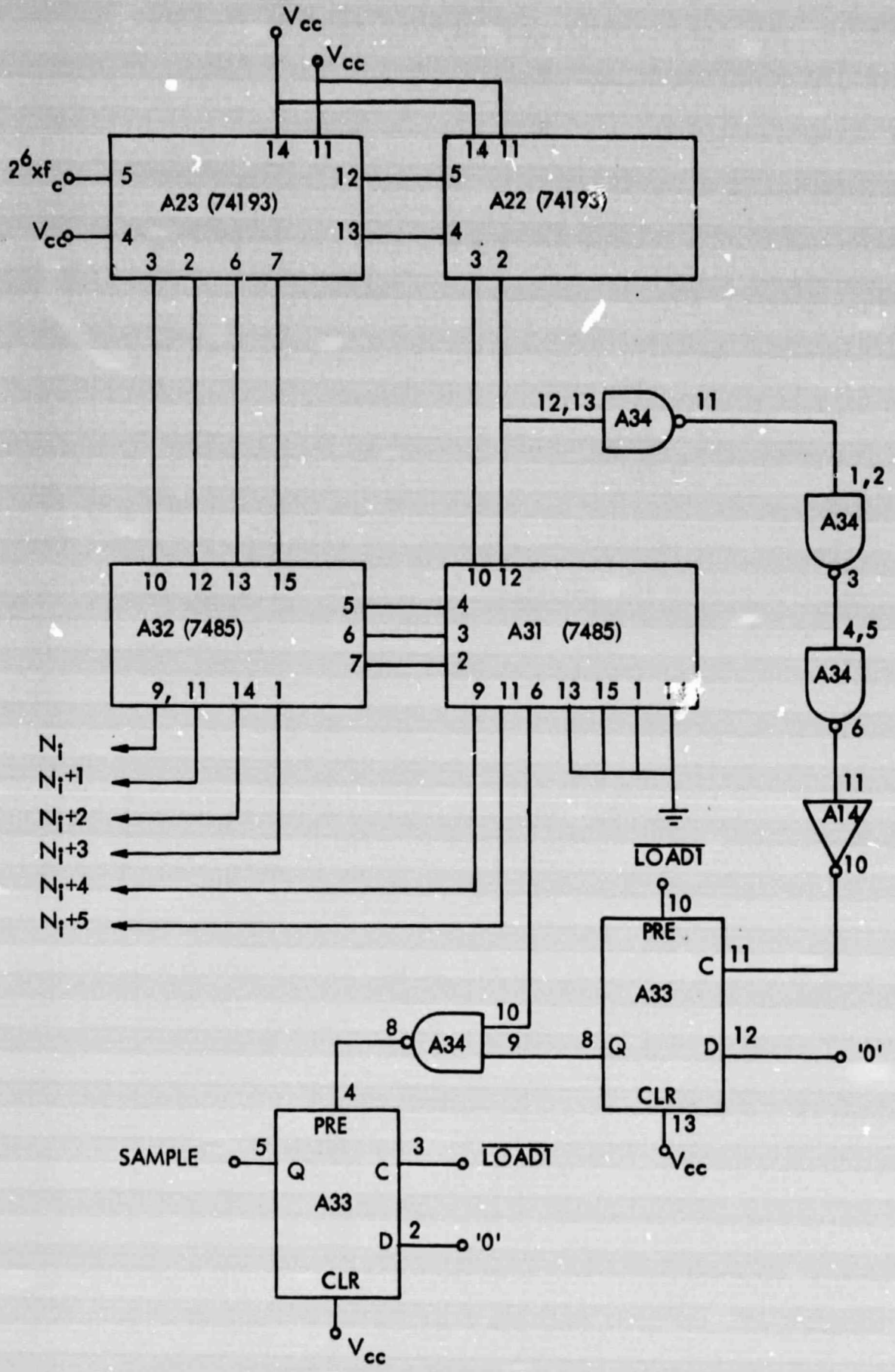

Figure B-4. DPLL Variable Phase Reference Clock.

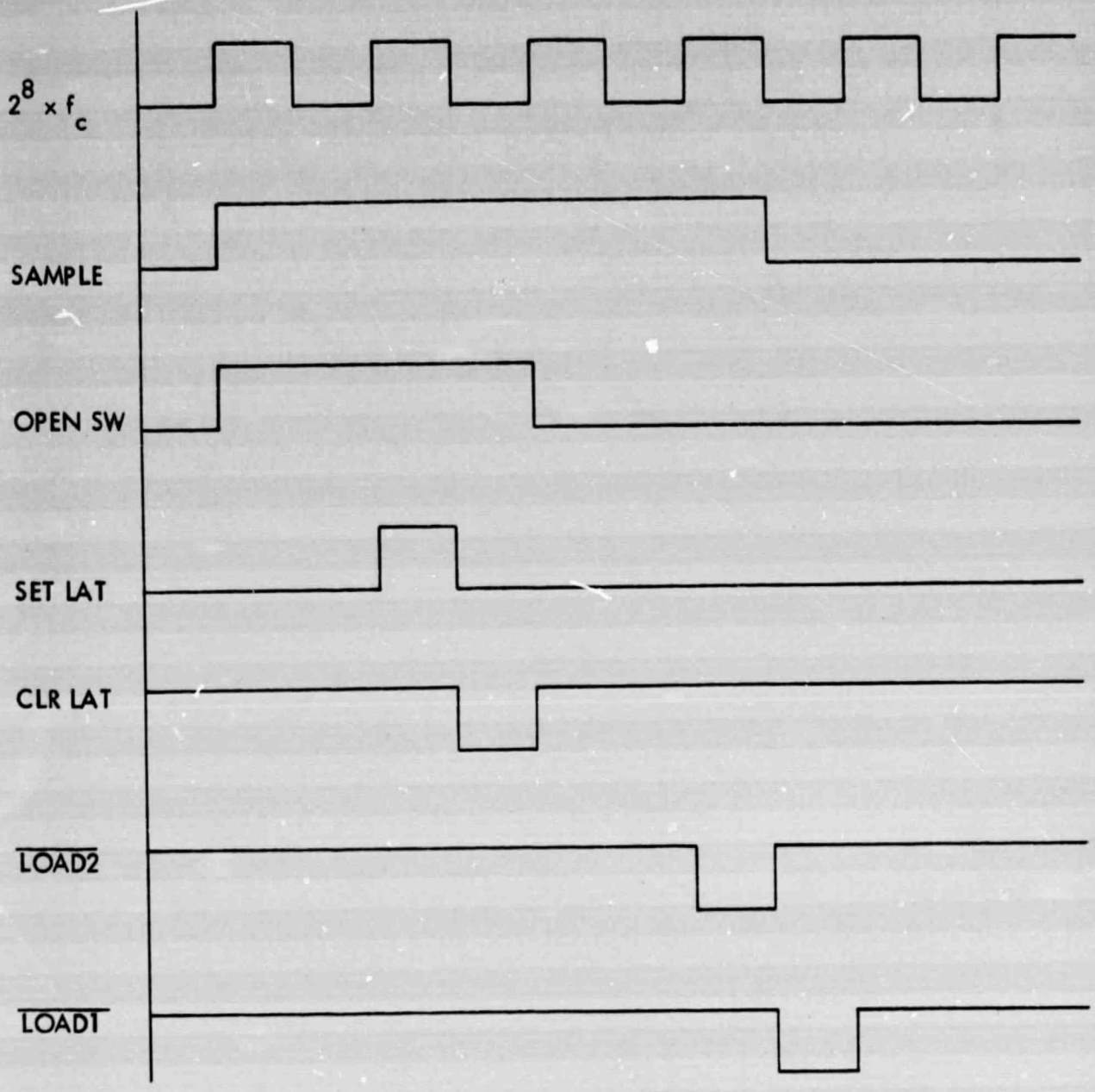

Figure B-5. Control Waveforms.

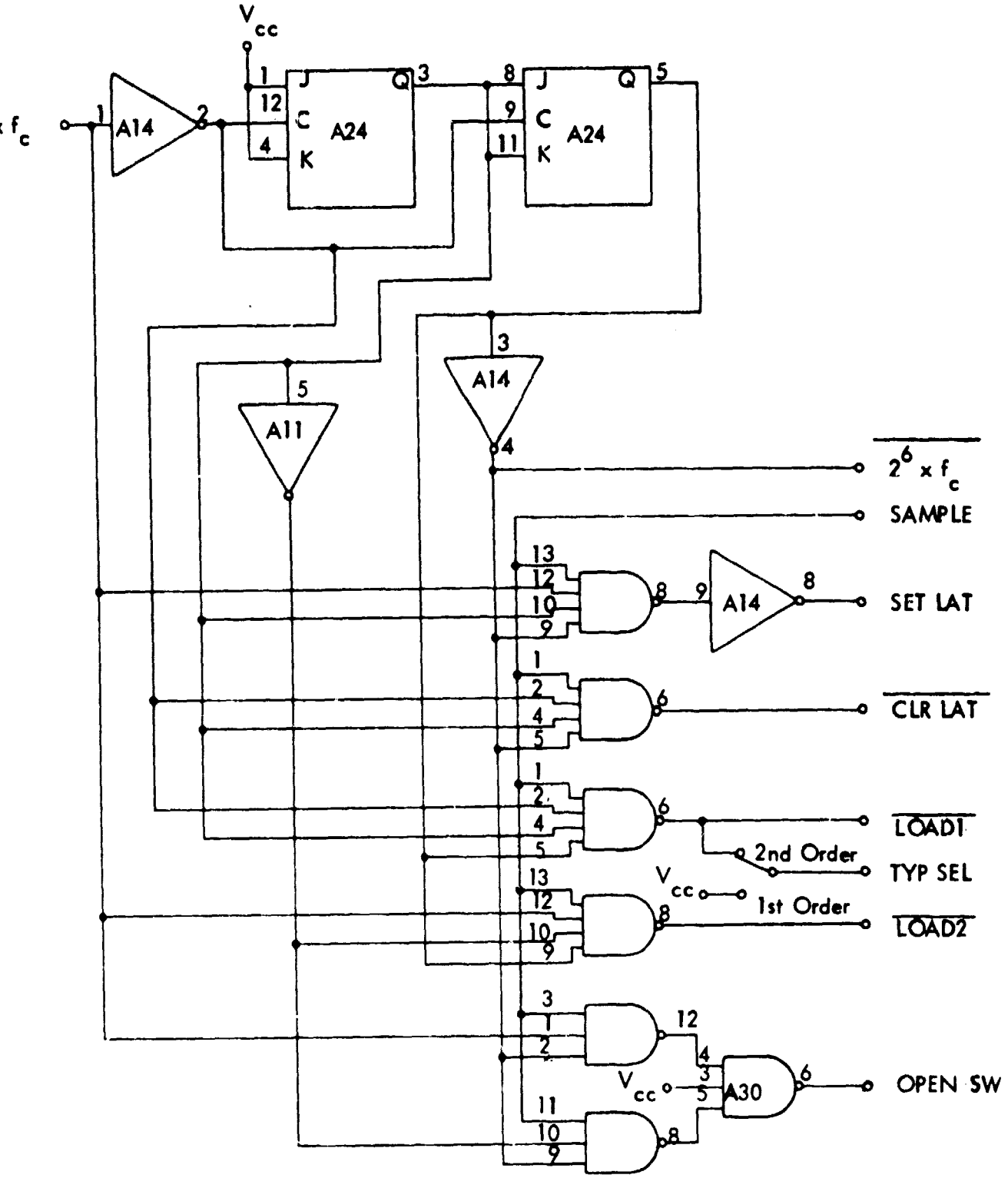

Figure B-6. Control Logic.

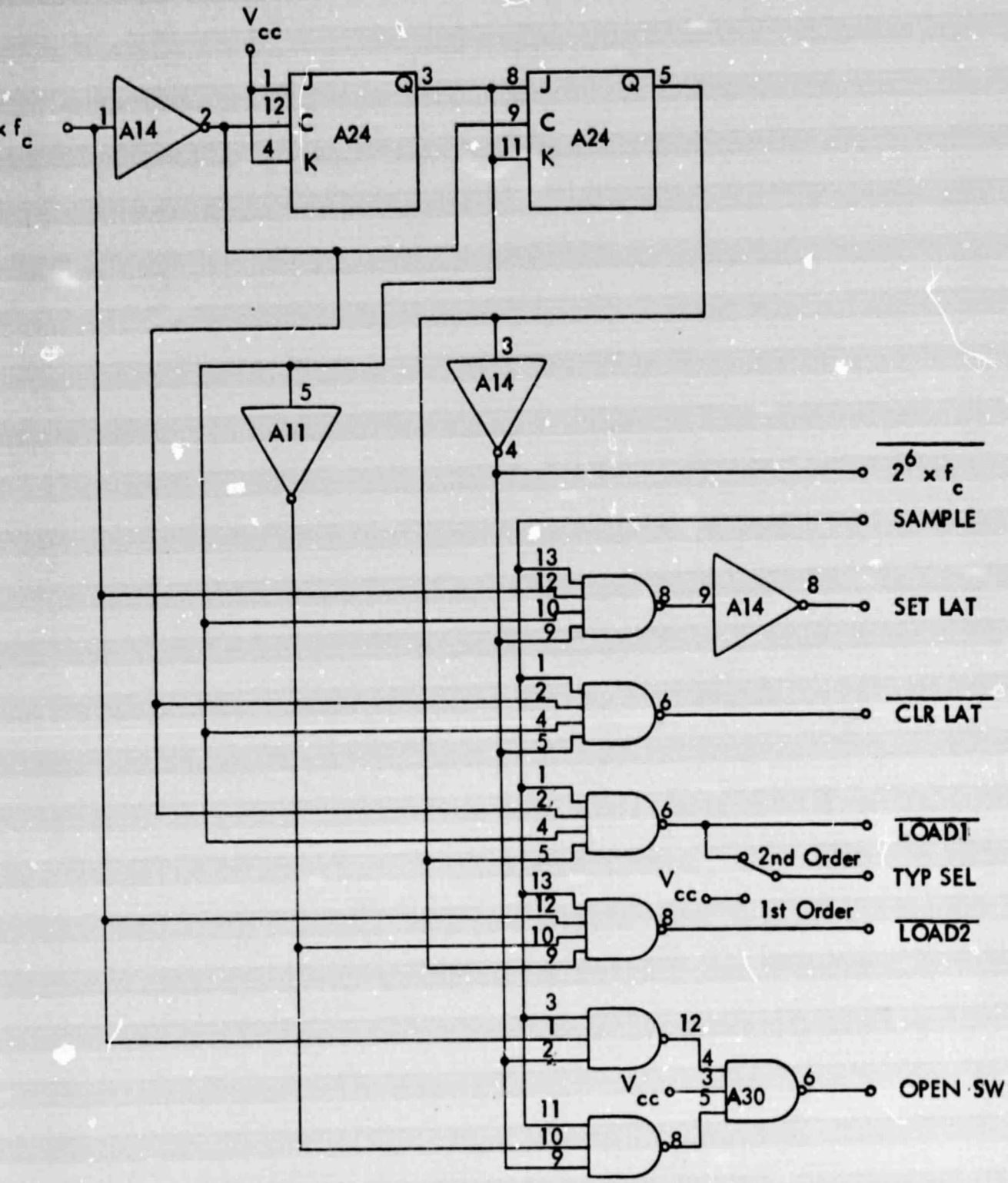

Figure B-6. Control Logic.