ANALYSIS AND DESIGN OF MOBILE TERMINAL TRANSMITTERS FOR MIXED VOICE AND DATA PACKETS IN THE 800 MHz FREQUENCY BAND

Submitted according to the terms
of Contract No. OSU80-00131

(RN 36100-0-9517)
(FC 4113-16350-2202)

Prepared by: Dr. S.A. Mahmoud
(Principal Investigator)

Dr. D. Falconer
Mr. M. El-Tanany
Dr. J. Wight

Carleton University

March 1981

ABSTRACT

The implementation of an integrated voice/data communications system over mobile radio channels require the availability of two essential mobile terminal components: (1) a spectrum efficient modem with fast carrier and clock recovery circuits, and (2) an efficient low cost speech detector to suppress the RF carrier when no speech signal is generated at the audio input of the terminal. This report presents the design, implementation and testing results of fast clock and carrier recovery circuits to be used with a Tamed Frequency Modulator (TFM). The report also presents an approach for implementing a speech detector. The detector is simulated first using software algorithms run in real time with recorded mobile speech. The detector will then be implemented using the recently introduced signal processors on LSI chips (e.g. Intel 2920). An experimental RF subsystem is set up as a test bed for measuring the BER performance of the TFM system.

SECTION 1

INTRODUCTION

1.1 General

It is predicted that the demand for data transmission over mobile radio channels will be steadily increasing over the current decade. Example of the applications demanding such a service include law enforcement agencies, taxi and other dispatch systems, automatic vehicle location systems and control of channel assignment in cellular systems. In addition, it is anticipated that a large fraction of the 350,000 radio users which constitute the existing Canadian mobile radio community will add to their system a data transmission capability in the next ten years.

The increasing demand for voice and data communications over mobile radio channels, coupled with the scarce nature of the frequency spectrum resource, have motivated the search for new techniques to achieve better utilization of that resource. One of these techniques relies on the integration of speech and data within a single land mobile radio channel. The feasibility of the technique has been the subject of a number of recent investigations (see list of references in section 1.6). The study reported here represents an on-going research activity which is aimed at demonstrating the practical feasibility of constructing an all digital voice/data mobile radio system. The new system makes use of the advances made in the modem and speech detection technology. The essential concepts underlying our approach have been discussed in a previous report [5]. A summary of these concepts is repeated below in order to facilitate reading of this report.

In existing commercial mobile communications systems, data messages and speech are transmitted on separate dedicated channels. Voice communications (mobile telephone) is carried out using frequency modulation techniques. This led to the standardization of the transmitter/receiver units in which FM channels have 30 KHz bandwidth. The need to transmit data messages over mobile communications sytems was accomodated by using an IF modulation technique (e.g. DPSK) for the input data messages. The output of the IF stage is then inserted in the audio input section of the FM transmitter unit. Figures 1.1 and 1.2 illustrate the components of mobile data terminals and a fixed base station. Data buffering and transmission is controlled by a microcomputer unit which interfaces serially with the modem.

The above technique places severe limitations on the speed of data transmission since the spectrum for the IF signal has to be limited to the bandwidth of the audio input of the transmitter. Increasing the speed of the input signal will lead in this case to higher bit error rates. In random access techniques, the transmission of messages at low rate increases the probability of errors and message retransmissions, which leads to rapid deterioration of the throughput performacne of the channel. As well, the scarcity of the available spectrum resources makes it difficult to satisfy the increasing demand for mobile data communications.

Recent research in the speech communications field indicates the existence of gaps between talk spurts of average durations ranging from 0.6 to 1.2 m.sec, depending on the applications. The channel becomes idle during these gaps. Transmission of voice and data simultaneously over the same channel can obviously be made possible by transmitting data packets during the silent intervals of the channel which is used primarily for voice communications.

375 BYTES

Assuming data packet lengths ranging from 1000 to 2000 bits (as is the case in most applications), it is possible to fit these packets into the silent intervals of the channel using appropriate transmission rates. This is feasible at transmission rates of 8 k.bits/sec or higher.

The realization of the above scheme in mobile radio applications is feasible only under the following conditions:

(1) The availability of low cost modem sets that can transmit data at rates higher than 8 k bits/sec over 30 KHz channels with low BER and extremely fast carrier and clock recovery circuits.

(2) The availability of accurate and low cost speech detector in voice terminals so that the carrier is suppressed when no input speech is detected. The ability of the data terminals to sense the voice carrier on the channel to determine if the channel is busy (talk spurt) or free (gap). The data terminal will transmit its packet only when a gap is detected.

Integration of voice and data transmissions in existing mobile communications systems will require the use of different modulation technique in RF band for the data signals. This is needed since data transmissions wil be an 'added on' service to an existing FM mobile communications system.

The ultimate objective of the on-going research reported here is to construct an all digital voice/data mobile communications sytems for the 800 MHz frequency band. Thus the same modulation technique is used for both voice and data signals. To explain the general concept of such a system, we consider the general structure of the base station and each mobile unit.

The Base Station:

Figure 1.3 illustrates the basic components of the base station. When a speech signal is transmitted over a channel the station adds to it a narrow band tone and a clock signal. When a data signal is transmitted, only the clock signal is added to it (in the frequency domain).

The Mobile Station:

Figure 1.4 illustrates the basic components of each mobile unit. The mobile transmits speech signals after the channel is assigned to it by the base station. The mobile unit transmits data packets only after it senses the carrier on the channel and determines, by examining the busy tone, if the channel is busy or free. The speech detector is used to suppress the mobile carrier when it is not transmitting in the speech mode,

Figure 1.5 illustrates the structure of the mobile terminals proposed for future research. The voice encoder/decoder circuit will be implemented using the CVSD technique and voice will be digitized at 16 k bits/sec. The modem set to be used will be based on the TFM technique, which is discussed in section 1.3.

20/2K
NYQUIST RATE
8X

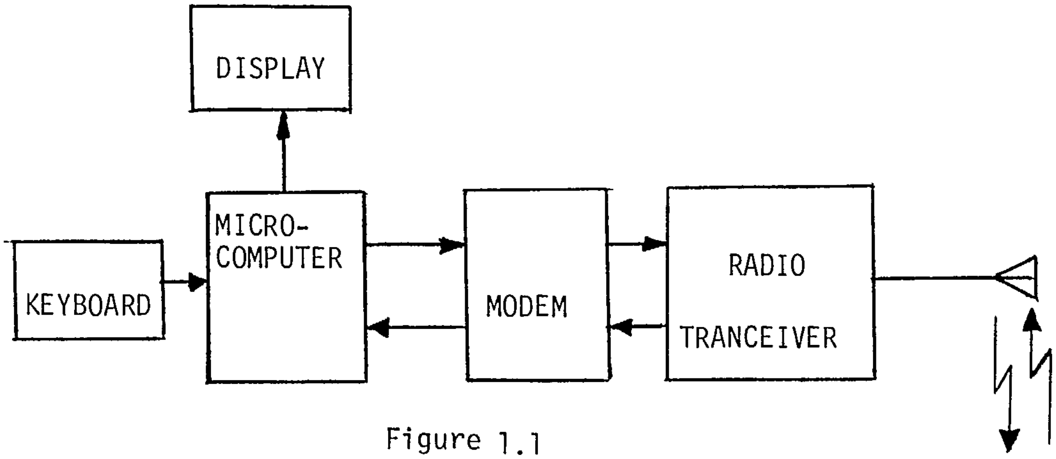


Figure 1.1

REMOTE TERMINAL

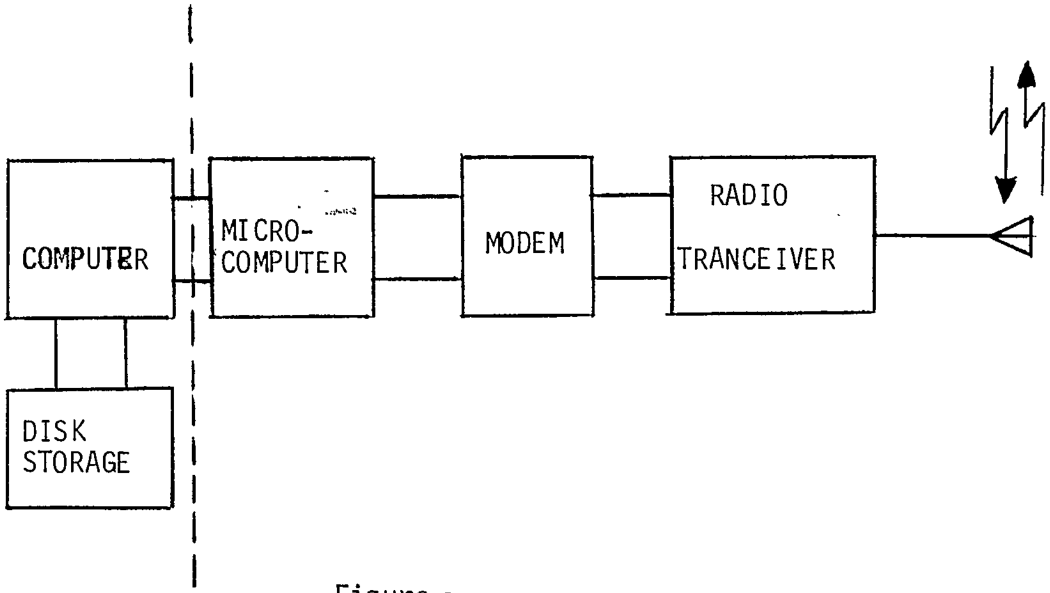


Figure 1.2

BASE STATION

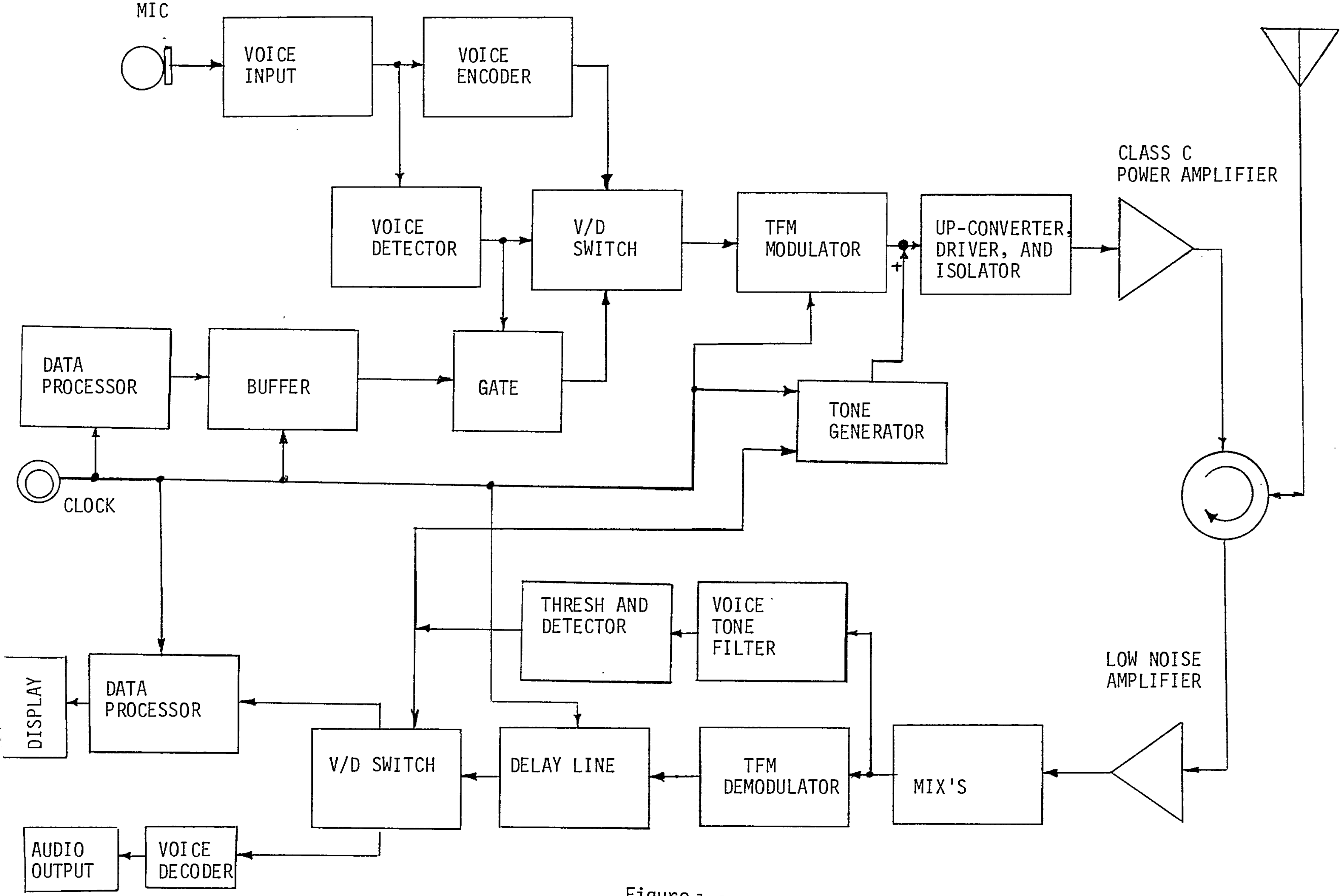


Figure 1.3
BASE STATION COMPONENTS

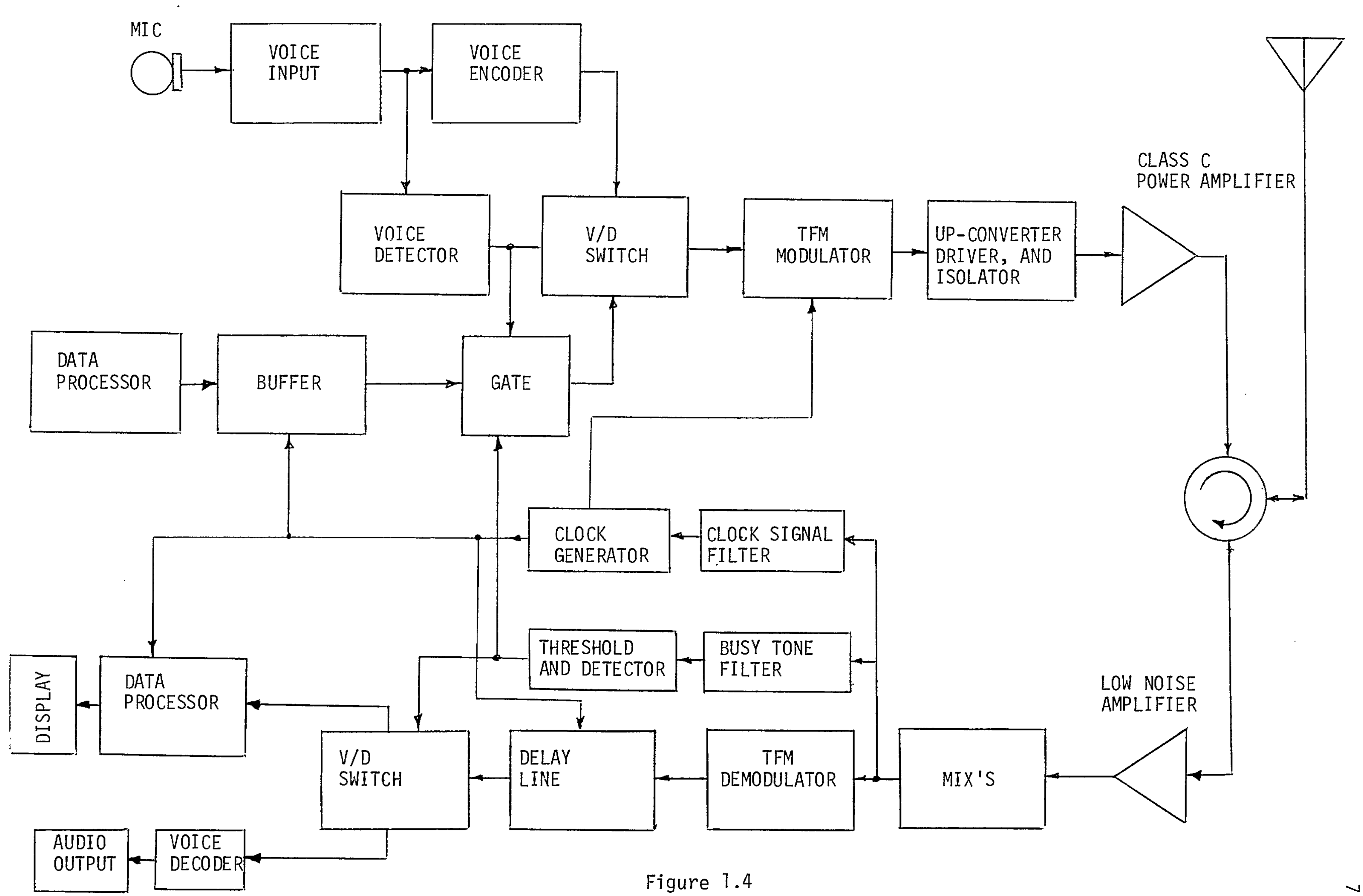


Figure 1.4

MIXED DATA AND VOICE TERMINAL COMPONENTS

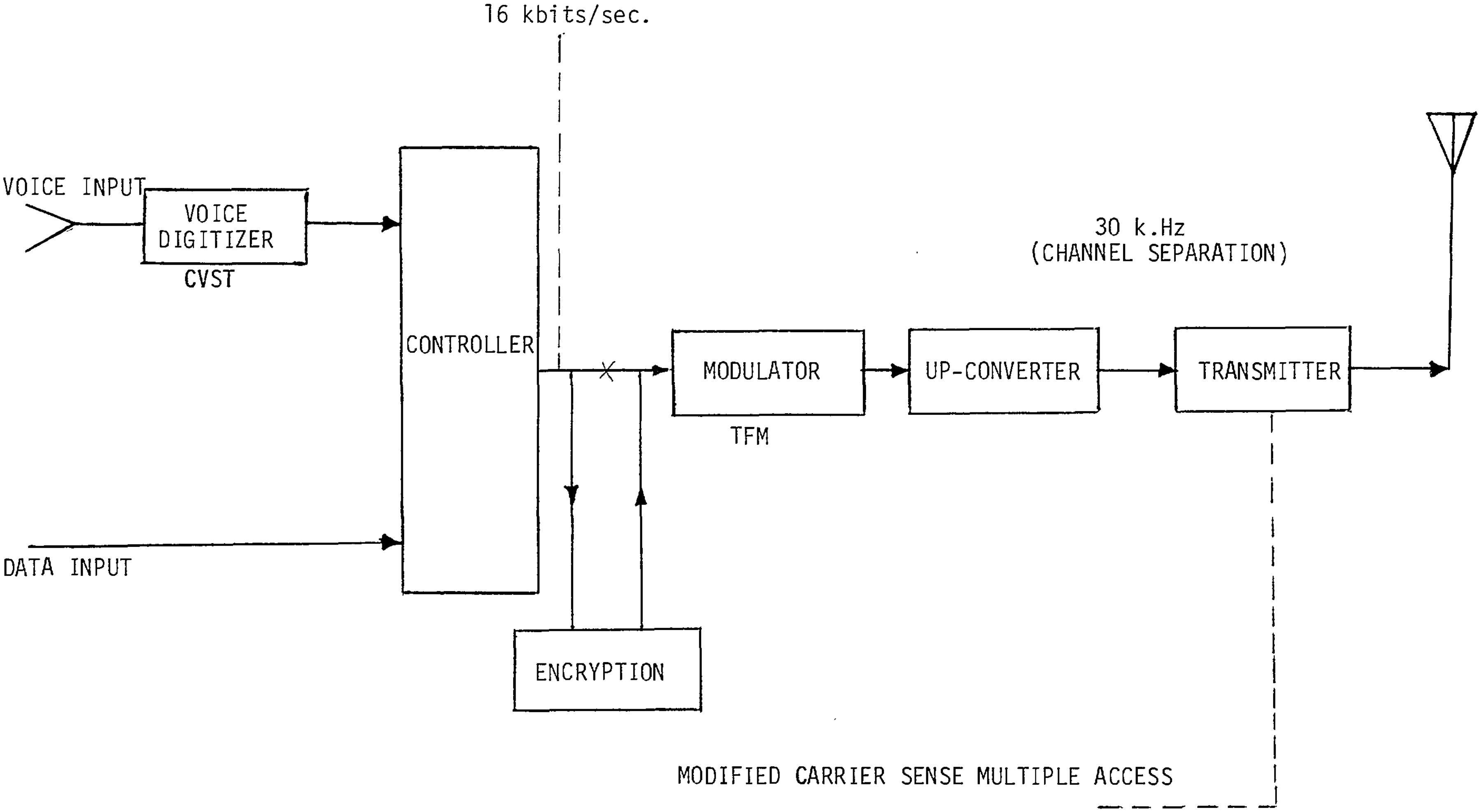


Figure 1.5

PROPOSED EXPERIMENTAL MOBILE TERMINAL

1.2 Scope

The ultimate objective of the research reported here is to demonstrate, both theoretically and practically, the feasibility of integrating speech and data transmissions over mobile radio channels. Such as integration will lead to better utilization of the available frequency spectrum and provide flexibility in channel use for applications involving voice and data communications. The research reported here can be considered as a step in the progress towards achieving the final objective. Specifically, two research aspects have been addressed:

1. The design and implementation of fast carrier and clock recovery circuit. Since data communications take place in the packet mode, it is essential that the receiver be able to lock onto the incoming signal carrier and clock despite the existence of frequency and phase offset and in the presence of noise. At a data rate of 16 k. bit/sec, an acqusition time of 5 to 10 m.sec would correspondend to 80 to 160 synchronization bits to be added to each packet. Our objective is to achieve such figures for the mobile radio channel. The clock and carrier recovery circuits will be part of the demodulation section of the Tame Frequency Modulator which has been selected due to its spectrum efficiency and small out of band radiation. Sections 2 and 3 of this report will present in detail the design and implementation of the carrier and clock recovery circuits.

2. The design implementation and testing of a speech detection circuit which is capable of detecting speech in the presence of variable signal-to-noise ratio. The design of such a circuit is a delicate compromise between accuracy and simplicity (low cost). The circuit will be employed in the transmitter section of each mobile. Its main function will be to turn the RF section off (i.e. suppress the carrier) when no speech is generated at the input of the audio section. Section 4 of this report will present the approach followed in designing this circuit.

1.3 Tamed Frequency Modulation Systems

Tamed Frequency Modulation (TFM) is a form of fast frequency shift keying, in which the abrupt phase changes are smoothed out. It features outstanding spectrum utilization while its noise immunity compares very well with other digital modulation methods. To evaluate the applicability of TFM to mobile radio channels, we summarize first the most important aspects required of a digital modulation method in the mobile radio environment [6].

1. Efficient bandwidth utilization: for a channel spacing of 25 KHz, a bit rate of 16 kb/s will be desirable, so that digitized DCDM encoded speech can be transmitted. The out-of-band radiation should not exceed the 70 dB selectivity values usual in analogue FM transmissions.

2. Good S/N ratio versus bit error rate (BER) properties, resulting in low transmitter power and good channel re-use utilization.

3. The spectrum should not be impaired by the use of a non-linear power amplifier (e.g.) class C amplifier) with its low power consumption. This feature is important for portable mobile terminals.

The first and the second requirements are in conflict since increasing the number of levels in digital transmission means reduction in the bandwidth, which is accompanied at the same time by a degradation in the BER performance. However, the step from two to four levels in phase modulation is the only one that can be made without such loss in performance. This property, taken with the third requirement, leads us to conclude that some kind of constant envelope modulation with four phase position is a promising solution.

Table 1.1 compares a number of modulation schemes based on the four-phase principle. The four-phase PSK is given as a reference. Fast Frequency Shift Keying (FFSK), referred to in many instances as MSK, is given in the table for the filtered version. The data in the forth column are a measure for the interference radiated into the adjacent channel with the IF filters assumed to be wide enough to allow the wanted signals to pass on to the point where the spectrum is 20 dB down. The table shows the value of TFM to be by far the best for this criterion.

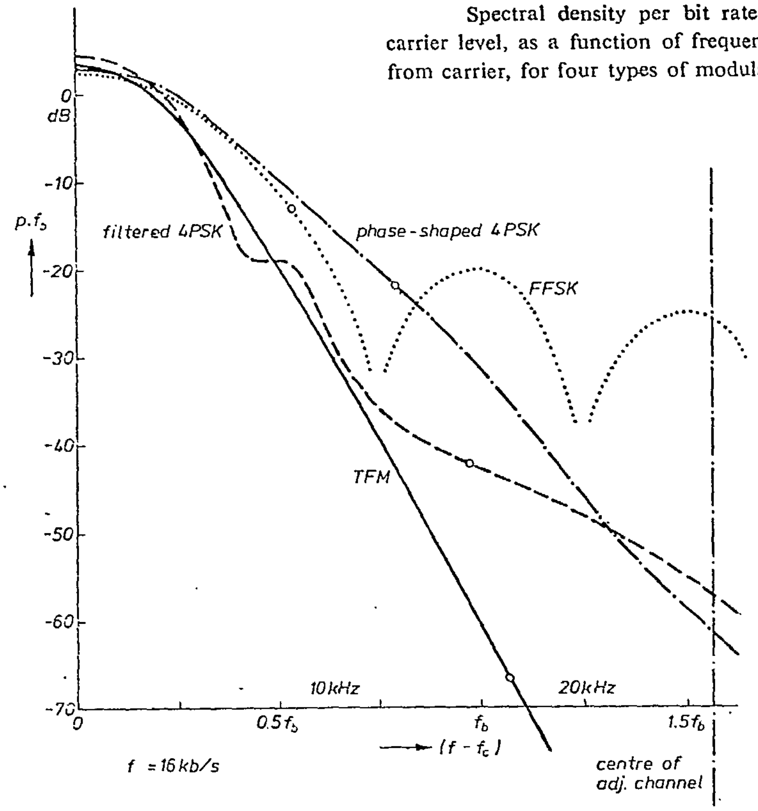
Figure 1.6 shows the bandwidth properties, represented by spectral density curves for the modulation systems of interest. It is observed that TFM has a steep and continuous decrease in power density which is already 60 dB down at bit rate distance from the carrier. At the edge of the adjacent channel the spectrum is 67 dB, down, much better than any of the other

Table 1.1

Comparison of some modulation methods

Type	C/N_0 at $P_e = 10^{-2}$ ref. to 4-PSK	Constant envelope?	Power density per bit rate (16 kb/s) at edge of adjacent channel	Remarks
4-PSK	0 dB (ref.)	Yes	—	excessive bandwidth
Phase-shaped 4-PSK	−1 dB	Yes	−22 dB	Nyquist pulse
Filtered 4-PSK	−1 dB	No	−42 dB	power ampl. class AB
FFSK (MSK)	0 dB	Yes	−14 dB	
Shaped FFSK	−1 dB	Yes	− 5 dB	sinusoidal
TFM	−1 dB	Yes	−67 dB	

Figure ~~6.1~~ 6.1



Spectral density per bit rate relative to carrier level, as a function of frequency distance from carrier, for four types of modulation methods. Obviously TFM is the only method that can be compared based on the bandwidth criterion with analogue FM.

The price paid for obtaining the spectrum efficiency of the TFM technique is a slight degradation in the bit error rate performance compared with FFSK or PSK. In [6], an approximation is obtained for the optimum demodulation filter which shows that the BER versus S/N for TFM is roughly 1 dB worse than that for PSK.

The above discussion explains the rational for selecting TFM as the basis for the all digital integrated voice/data system investigated.

1.4 Speech Detection

An algorithm for detecting the presence or absence of speech at the voice terminal input has been proposed and studied by simulation on a PDP11/55 computer. The simulated algorithm processes digitized samples of speech which were originally recorded in a mobile radio environment. The processed speech can be converted back to analog and compared with the original speech to assess its quality. The simulation facility also allows a detailed examination of algorithm behaviour and measurement of the activity factor of the speech (percentage of the time active speech is detected). Memory limitations of the PDP11/55 computer presently allow the processing of short speech segments only. Therefore a software effort was needed to use the available memory more efficiently. thereby allowing the processing of longer speech segments. At the same time the graphics capability of the computer has been extended to facilitate the monitoring of the algorithm. These improvements in the computer facility have temporarily interrupted the progress of the simulation effort, but were essential to the goal of modifying the speech detection algorithm and refining its parameters to optimize its performance while keeping its complexity moderate.

The algorithm that has been initially proposed and simulated is based on the idea of measuring the power level (which is assumed to be constant or very slowly varying) of the background noise, and setting the speech detection threshold just above this level. When speech is present along with background noise the power level of the speech terminal input signal will be more rapidly varying, (on the order of tens of milliseconds rather than seconds, and of course it will be higher than the power level observed when only noise is present. To determine the noise power level, the algorithm identifies and measures the power level of signal segments which are "almost surely noise". An "almost surely noise" (ASN) segment is identified when the signal's long term and short term averaged power levels are nearly equal, and are no longer than the current noise level estimate. This noise level estimate is allowed to increase slowly during non-ASN during non-ASN periods to permit adaptation to slowly changing noise levels. Once an ASN period is identified, the current noise level estimate is reset (decreased) to current measured signal power level. The current speech detector threshold is also reset at this time to3dB above the measured power level.

Whenever the magnitude of a signal sample exceeds the current speech detector threshold, active speech is declared for at least the next H signal samples , where H is a hangover period which allows continuity during possible low-level periods in active speech. The primary aim of the computer simulation phase of the research is to determine appropriate parameter values such as time constants for measuring power levels, and also to pinpoint weaknesses and possible improvements to the algorithm.

The hardware speech detector unit will operate with its own A/D converter, instead of operating directly on the digitized output of the CVSD coder. With this arrangement any reduced bit rate speech encoder can be substituted for the CVSD coder without affecting the speech detector.

The recently introduced signal processing chips and processor boards make it more convenient to implement the speech detection unit in software, rather than constructing it using IC components.The benefits gained from this approach are substantial in terms of ease of development, flexibility in changing parameter values and the possibility of experimenting with several algorithm variations and options.

1.5 Report Structure

The remaining sections of this report are organized as follows:

- section 2 presents the design and implementation of the carrier recovery circuit . Experimental results as well as computer simulation results are also presented.

- Section 3 presents the design and implementation of the clock recovery circuit, together with experimental results concerning the performance of the circuit.

- Section 4 introduces the main approach followed in the design of a simple speech detector for mobile radio terminals. Computer simulation is discussed along with the application of new signal microprocessors to the implementation of the speech detector.

- Section 5 presents the components and structure of the RF subsystem which will be used in testing the TFM modulation system in the 800 MHz band.

- Section 6 contains some concluding remarks and recommendation for future studies.

1.6 References

[1] E. Arthurs and B.W. Stuck, "A Theoretical Traffic Performance Analysis of an Integrated Voice-Data Virtual Circuit Packet Switch", IEEE Transactions on Communications, 27(7), pp.1104-1111, July 1979.

[2] C.J. Weinstein, M.L. Malpass and M.J. Fischer, "Data Traffic Performance of an Integrated Circuit and Packet Switched Multiplex Structure", International Conference on Communications, pp. 24. 3-1 to 24.3-5, Boston, Mass., 1979.

[3] M.J. Ross, A. Tabbot, J.W. Waite, "Design Approaches and Performance Criteria for integrated Voice/Data Switching", Proc. IEEE, 65(9) pp. 1283-1295, September 1977.

[4] A. Pan, "Integrating Voice and Data Traffic in a Broadcast Network Using Random Access Scheme", International Conference on Computer Communications, pp. 551-556, Kyotot, Japan, September 1978.

[5] S.A. Mahmoud, "Analysis and Design of Land Mobile Communications Systems Based on Digital Techniques," Report submitted to DOC (Contract # OSU79-00060), Department of Systems and Computer Engineering, March 1980.

[6] D. Muilwijk, "TFM - a bandwidth - saving digital modulation method, suited for mobile radio," Philips Telecommunications Review, Vol. 37, No. 1, March 1979, pp. 35-48.

SECTION 2

CARRIER RECOVERY CIRCUIT

We examine in this section the design, analysis and implementation of the carrier recovery circuit which will be part of the demodulation system. The circuit is designed to achieve a short acquisition time for carrier frquencies in the 800 MHz range.

The main building block of the carrier recovery circuit consists of a Costas loop synchronizer with two bandwidths:

(1) a narrow bandwidth for the tracking mode to minimize the carrier phase jitter.

(2) a large bandwidth for the acquisition made.

In addition, the loop is combined with a digital frequency comparator to acquire phase lock from a relatively large frequency error.

The acquisition problem of Costas loop has been studied by Cahn [1] for the case of biphase (BPSK) modulated signals. Using a technique which was first suggested by Richman [2], Cahn demmonstrated that a rapid pull in time from a large initial frequency error can be achieved by combining the Costas loop with an Automatic Frequency Control (AFC) circuit. Dekker [3] applied the same technique for carrier tracking of TFM signals. Messerschmitt [4] also used the same concept for carrier tracking in microwave radio systems.

The basic Costas loop structure reported in this section is similar to those reported in [1] and [3]. However, some changes in the loop configuration and hardware design have been introduced. The modified design proved to be both simple and efficient in the sense that it satisfies the requirement of a short acquisition time in the presence of a large frequency error.

The general loop structure is given in section 2.1. Analysis of the loop acquisition behaviour in the presence of an initial frequency error is presented in section 2.2. This is followed by a description of some experimental results in section 2.3. Implementation details of each of the components of the circuit are included in section 2.4. Finally, section 2.5 lists the set of references for the entire section 2.

2.1 Loop Structure

A block diagram of the loop under investigation is shown in (Fig.2.1). The VCO control is the sum of two error signals; one from the AFC loop filter and the other from the Costas loop filter. When the frequency error is many times the loop natural frequency, the dc output from the phase detectors is essentially zero, so is the Costas loop error signal. At this initial stage the loop behaviour will be controlled by the AFC circuit. The frequency detector compares the IF received signal with the two quadrature components of the VCO, and generates a voltage proportional to the frequency error. This error signal will drive the VCO frequency closer to the received frequency. Once the frequency difference is within the loop bandwidth the phase detectors take over completing the loop acquisitition.

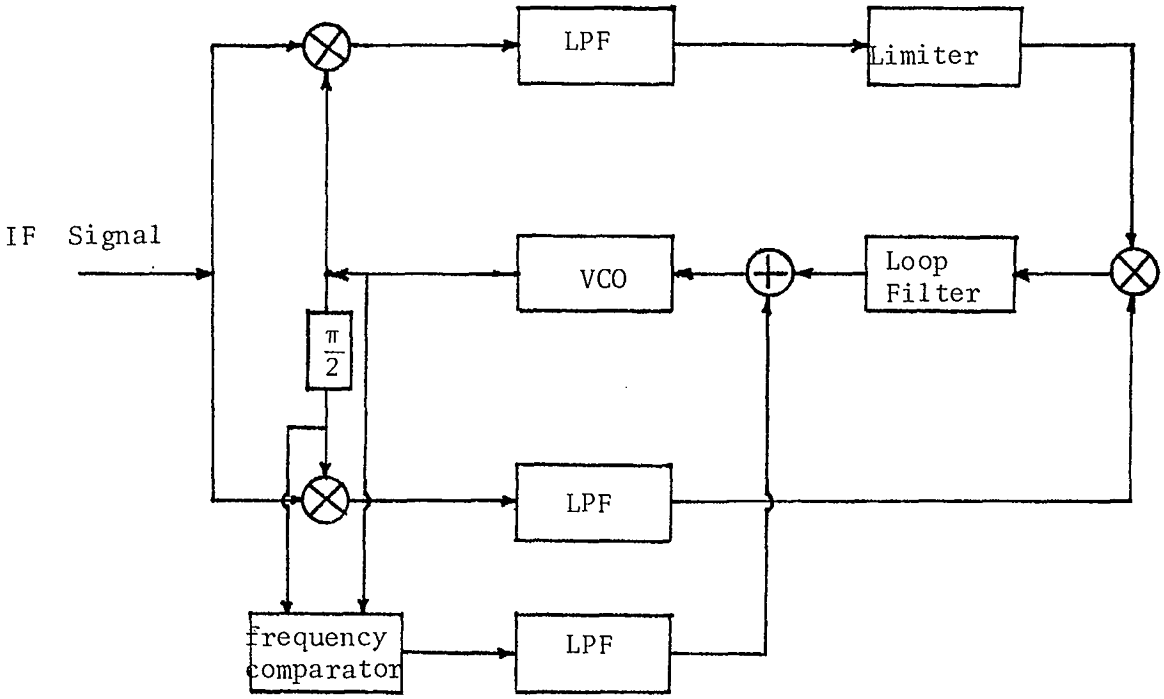


Fig. 2.1 Block Diagram of Costas Loop Combined with a Frequency Comparator

The phase detectors utilized in the loop show a triangular transfer characteristic - i.e. the dc component in the phase detectors output versus the VCO phase offset from the input signal is triangular as depicted in (Fig 2.2). By hard limiting the In-phase channel (I), and multiplying it by the quadrature channel it is obvious that the Costas loop detector transfer characteristic will be a periodic Saw-tooth with period π as shown in (Fig.2.3). Note that this characteristic is independent of the input signal amplitude.

The loop filter is designed to have a high gain at dc and its transfer function has a proportional and integral terms - i.e. the Costas loop filter transfer function is given by;

$$F(S) = \alpha + \frac{\beta}{S} \qquad (2.1)$$

The loop will reach a locking condition when the dc output of Costas loop detector is zero - i.e. when the VCO phase offset is 0, $\pm\ \pi/2$ or π. However, the lock condition at $\pm\ \pi/2$ is unstable (Fig. 2.3) - This suggests that the loop will lock when the phase offset is either zero or π.

The loop frequency comparator is the so called "Rotational frequency detector" [4], since its principle of operation was found to be suitable for CPFSK signals. Its output is a linear function of the loop frequency error if the latter is less than 25% of the carrier frequency.

The rotational frequency comparator is implemented using digital circuitry and is designed to operate as follows:

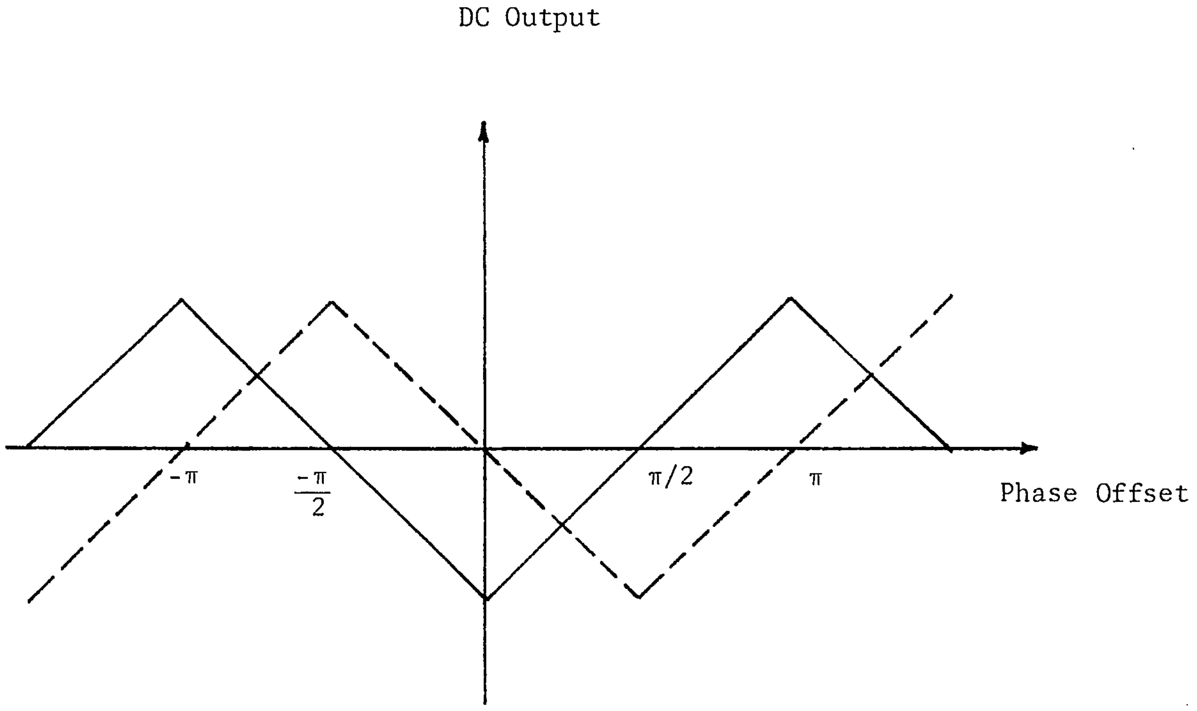


Fig. 2.2 - Input/output Characteristic of the I & Q Channels Phase Detectors

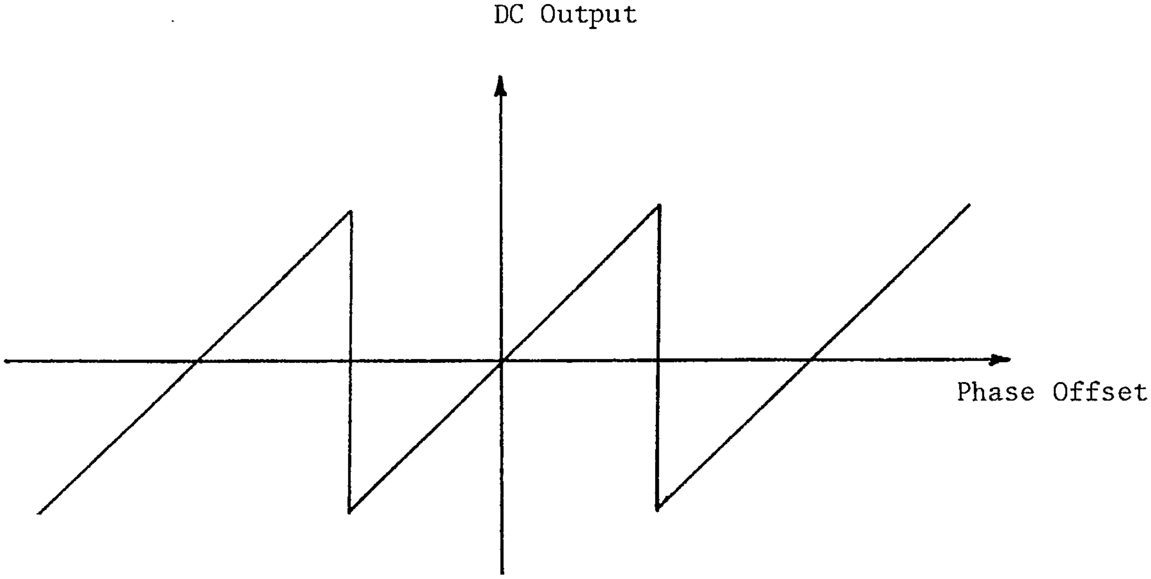


Fig. 2.3 - Input/Output Characteristic of the
Costas Loop Error Detector

(1) If the VCO frequency (f_{VCO}) is equal to the received frequency (f_{IF}), the output is zero

(2) When $f_{IF} > f_{VCO}$, the output is a train of positive pulses whose repetitive rate is proportional to the frequency error

and

(3) When $f_{IF} < f_{VCO}$, the output is a train of negative pulses as in (2).

Then, starting with a large frequency error, the signal coming out of the frequency comparator is a train of high rate pulses. The pulses are integrated in the AFC loop filter and fed back to the VCO control port. As the f_{VCO} gets closer to f_{IF}, the pulse rate will be reduced until it diminishes upon achieving a zero frequency error.

The AFC loop filter is a digital integrator (16-bit., presettable UP/DOWN counter followed by a DAC) having the transfer function;

$$H(S) = \frac{Y}{S} \tag{2.2}$$

The frequency comparator effect on the loop can be modelled as a slight increase in the Costas loop damping coefficient, while the loop natural frequency remains unchanged.

To avoid a false locking to data sidebands, the frequency control integrator is designed to be automatically reset when its output reaches any of two predetermined thresholds.

2.2 Analysis of Loop Acquisition Behaviour

Let the IF input signal be expressed as;

$$v_i(t) = \cos\ (\omega t + \lambda_i(t)) \qquad (2.3)$$

where

$$\lambda_i(t) = \Delta\omega t + \theta_i \qquad (2.4)$$

$\Delta\omega$ is the carrier frequency offset (rad/s) from the VSO free-running frequency, and θ_i is the carrier initial phase.

The VCO output signal can be expressed as;

$$v_o(t) = \cos\ (\omega t + \lambda_o(t)) \qquad (2.5)$$

where $\lambda_o(t)$ is the VCO instantaneous phase.

The dc component of the Costas detector output is given by (Fig. 2.3);

$$v_p = k_p\ [(\lambda_1(t) - \lambda_o(t).\ \ \text{modulo}\ \pi\ ] \qquad (2.6)$$

where k_p is the sensitivity of Costas detector (V/rad). The dc output of the frequency comparator is given by;

$$v_p = k_f \left(\frac{d\lambda_i(t)}{dt} - \frac{d\lambda_o(t)}{dt}\right) \qquad (2.7)$$

where k_f is the frequency comparator sensitivity (V/rad) - Using Eqns. 2.1 - 2.7 the loop can be modelled as shown in (Fig. 2.4). The integral equation governing the loop acquisition performance is given by:

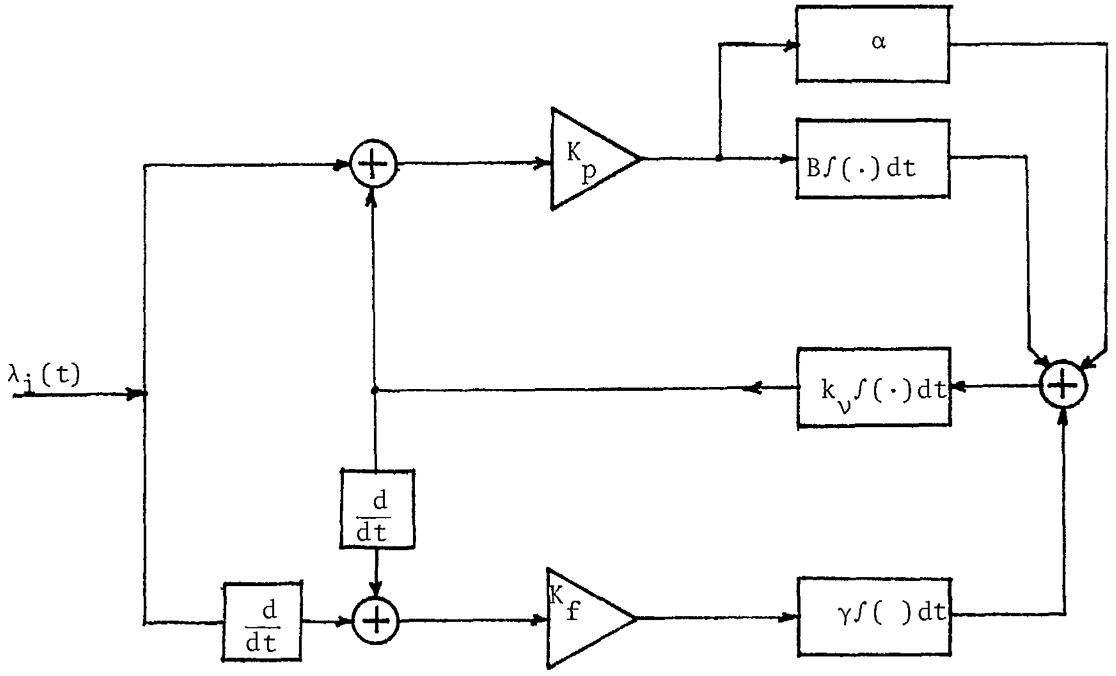


Fig. 2.4 - Analytical Model for the modified Costas Loop

$$\begin{aligned}\lambda_o(t) = {} & k_v \int \Big[k_p \alpha(\lambda_i(t) - \lambda_o(t)) \\ & + k_p \beta \int (\lambda_i(t) - \lambda_o(t))dt \Big] \, dt \\ & + k_v \int \gamma k_f \left(\frac{d\lambda_i(t)}{dt} - \frac{d\lambda_o(t)}{dt} \right) dt \end{aligned} \qquad (2.8)$$

where k_v is the VCO sensitivity (rad/s/V).

Taking the Laplace transform of Eqn. 2.7 and rearranging the terms, the loop transfer function can be expressed as;

$$\frac{\lambda_o(S)}{\lambda_i(S)} = \frac{2\xi_e \omega_n S + \omega_n^2}{S^2 + 2\xi_e \omega_n S + \omega_n^2} \qquad (2.9)$$

where

$$\left.\begin{aligned} \omega_n^2 &= \beta k_v k_p \\ \xi_e &= \xi + \frac{\omega_n}{2}\left(\frac{\gamma k_f}{\beta k_p}\right) \\ \xi &= \frac{\alpha\, k_v\, k_p}{2\omega_n} \end{aligned}\right\} \qquad (2.10)$$

ω_n and ξ are the Costas loop natural frequency and damping coefficient respectively, while ξ_e is the combined loop damping coefficient.

The differential equation governing the loop acquisition can be written directly from Eqn. 2.9 as:

$$\frac{d^2\lambda_i(t)}{dt^2} = \frac{d^2\lambda(t)}{dt^2} + 2\xi_e\, \omega_n \frac{d\lambda(t)}{dt} + \omega_n^2\, \lambda(t) \qquad (2.11)$$

where $$\lambda(t) = \lambda_i(t) - \lambda_o(t) \qquad (2.12)$$

$\lambda(t)$ is the "instantaneous" phase offset of the input signal relative to the VCO output.

LOOP RESPONSE TO A STEP IN FREQUENCY AND PHASE

Let the input signal phase be written as:

$$\lambda_i(t) = \Delta\omega t + \theta_o \qquad (2.13)$$

while the VCO initial phase is equal to zero.

Solving (2.11) subject to (2.13), yields the following solutions:

$\underline{\xi_e > 1}$

$$\lambda(t) = \frac{1}{2a}\left\{\left[\frac{\Delta\omega}{\omega_n} + (\xi + a\ \Theta_o\right] e^{-\omega_n t(\xi_e - a)}\right.$$

$$\left. - \left[\frac{\Delta\omega}{\omega_n} + (\xi_e - a\ \Theta_o\right] e^{-\omega_n(\xi_e + a)t}\right\} \text{ modulo } \pi$$

$$\frac{\dot{\lambda}(t)}{\omega_n} = \frac{1}{2a}\left\{ - (\xi_e - a)\left[\frac{\Delta\omega}{\omega} + (\xi_e + a)\theta_o\right] e^{-\omega_n(\xi_e - a)t}\right.$$

$$+ (\xi_e + a)\left[\frac{\Delta\omega}{\omega} + (\xi_e - a)\theta_o\right] e^{-\omega_n(\xi_e + a)t} \qquad (2.14)$$

$\underline{\xi_e < 1}$

$$\lambda(t) = A \cos (\omega_n \sqrt{1 - \xi_e^2}\, t - \phi) e^{-\omega_n \xi_e t} \qquad \text{modulo } \pi$$

$$\frac{\dot{\lambda}(t)}{\omega_n} = A \cos (\omega_n \sqrt{1 - \xi_e^2}\, t + \psi) e^{-\omega_n \xi_e t}$$

where

$$a = \sqrt{\xi_e^2 - 1}$$

$$A = \left[\theta_o^2 + \frac{\Delta\omega/\omega_n + \xi_e \theta_0}{\sqrt{1 - \xi_e^2}}\right]^{1/2} \qquad (2.15)$$

$$\phi = \text{arct.}\left[\frac{\Delta\omega/\omega_n + \theta_o \xi_e}{\theta_o \sqrt{1 - \xi_e^2}}\right]$$

$$\psi = \text{arct.}\left[\frac{\xi_e \Delta\omega/\omega_n + \theta_o}{\frac{\Delta\omega}{\omega_n} \sqrt{1 - \xi_e^2}}\right]$$

It is worth noting that, during the acquisition mode, $|\lambda\ (t)|$ is a monotonically increasing function of time. Therefore, whenever $|\lambda(t)|$ reaches$\pi/2$, a singular point of the Costas detector characteristic is crossed (see Fig. 2.3), leading to a step change in the phase control voltage. The magnitude of this step is equal to $k_p\pi$, and its polarity is such that it will lead to increasing the frequency offset. The Costas loop filter is a proportional plus integral (Eqn. 1.1), therefore, the step change will reach the VCO Control input via the proportionional part of the loop filter only. The corresponding change in the VCO tuning voltage is $\alpha\ k_p \pi$ and the corresponding change in the VCO output frequency is given by

$$\frac{\Delta\dot{\lambda}}{\omega_n} = 2\xi\pi \qquad (2.16)$$

Eqns. (2.15) and (2.16) have been programmed on to generate the phase trajectories governing the Combined loop acquisition behaviour. The phase plane plots shown in (Fig. 2.5a) are for Costas loop only subject to an initial frequency offset of 10 ω_n and initial phase offset equal to $-\pi/2$. The corresponding phase trajectories for the modified loop assuming that $\xi_e - \xi = 0.15$ is shown in (Fig. 2.5b). It can be noticed that the number of cycles skipped until lock condition is achieved is 37 for Costas loop compared to 4 for the modified loop. The variation of the acquisition time versus the initial frequency detuning is shown in (Fig. 2.6) for both loops for the sake of comparison. It can be seen that the improvement in the loop acquisition time increases for larger frequency offsets. However, there is no significant improvement for frequency error within the loop pull-in range [5]. Therefore switching the loop bandwidth in conjunction with the AFC aid may be useful in applications requiring exceptionally fast and reliable sychronization

The work reported in this section was motivated by the requirement for a fast sychronizer with a wide capture range for use with digital mobile radio systems operating in the 800 MHz frequency band. This system will be used for packetized-data transmission at a rate of 16 kb/s, with a receiver IF frequency of 455 KHz. The loop is fixed at 75 Hz in the tracking mode and 400 Hz in the acquisition mode. The Costas loop

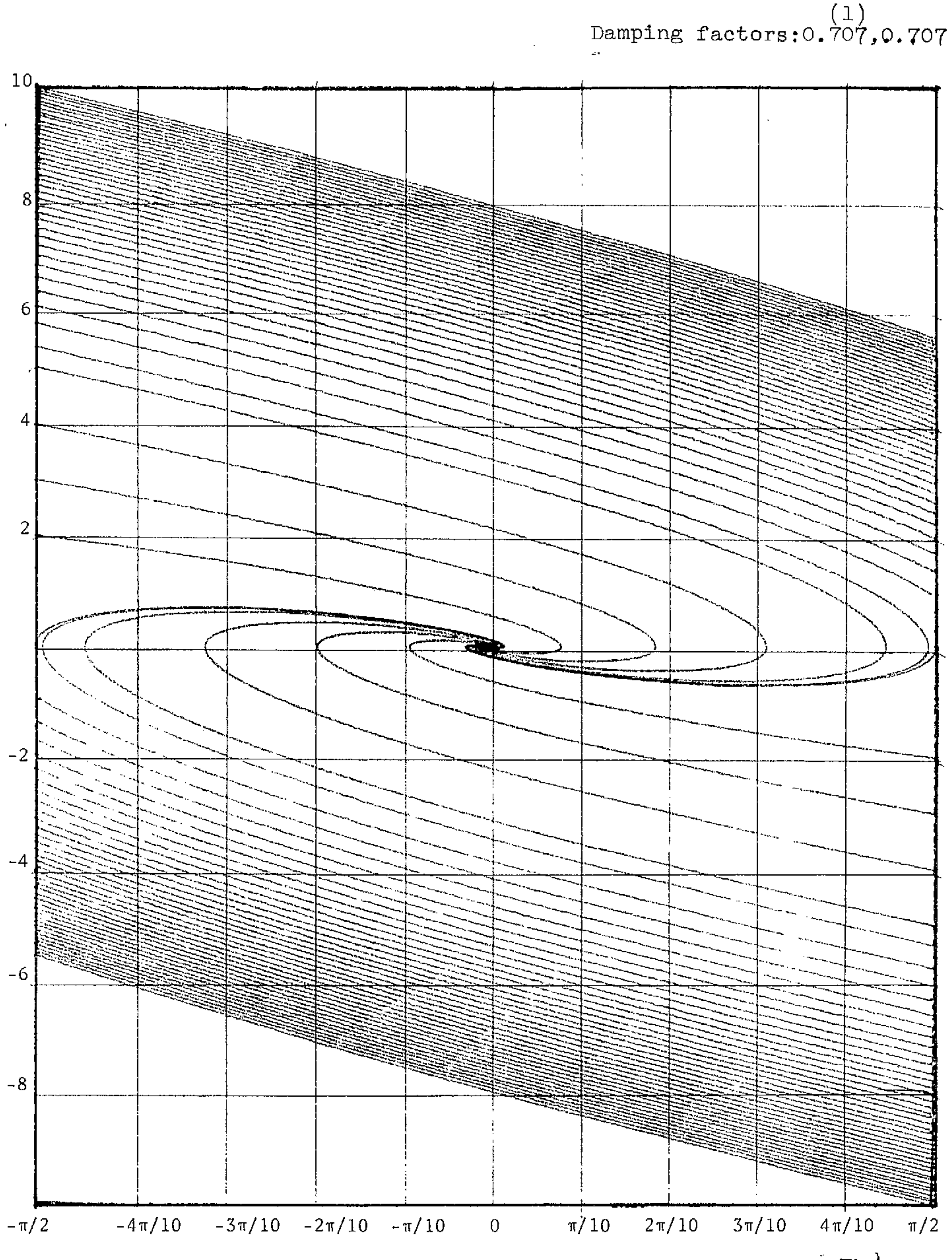


Fig. 2.5a - Phase plane Trajectories for a Costas Loop with dampinj factor = 0.707

(4)
Damping factors:0.857,0.707

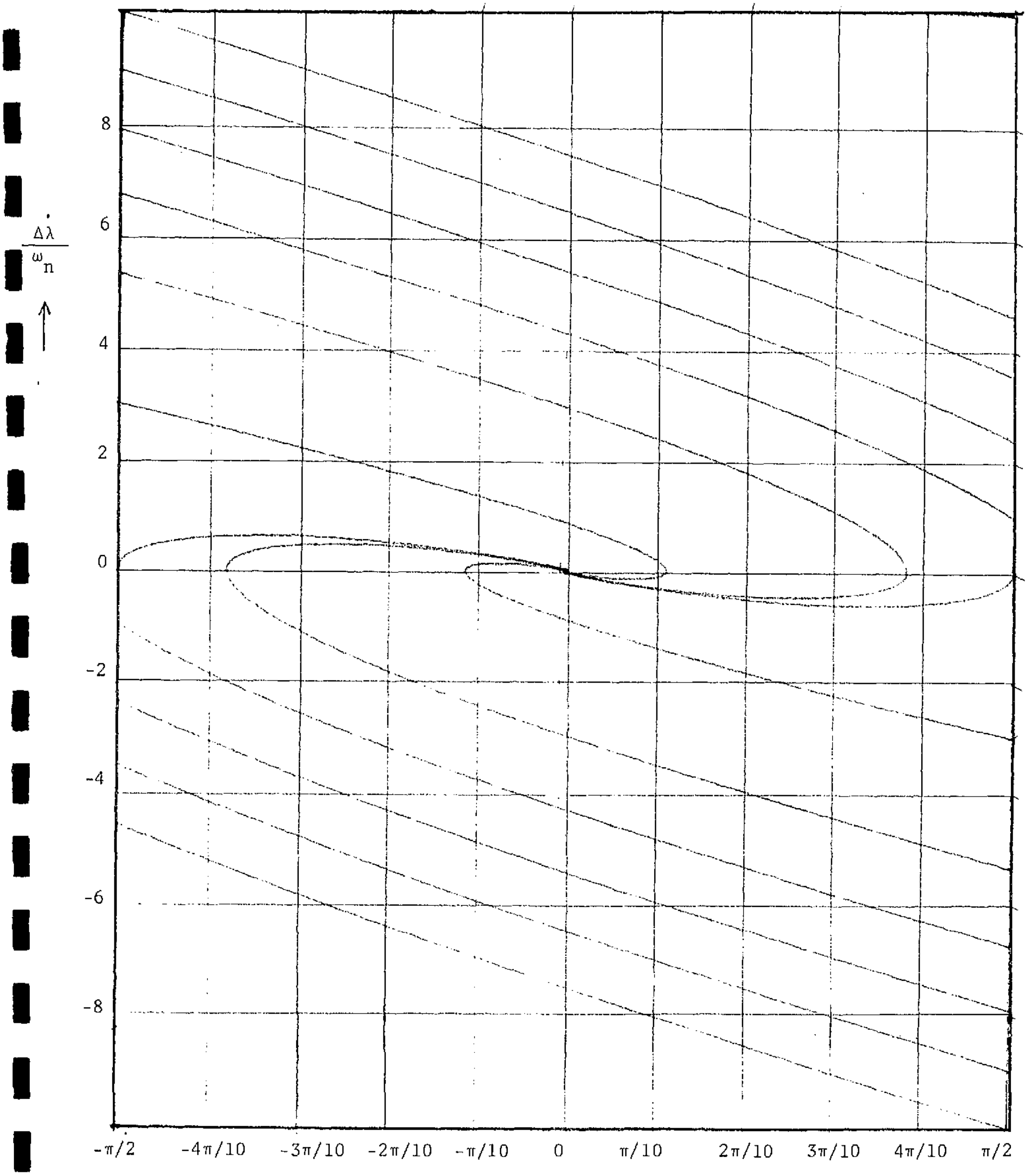


Fig. 2.5b - phase plane Trajectories for a Costas Loop combined with AFC Loop $\xi_a-\xi= 0.15$ $\xi= 0.707$

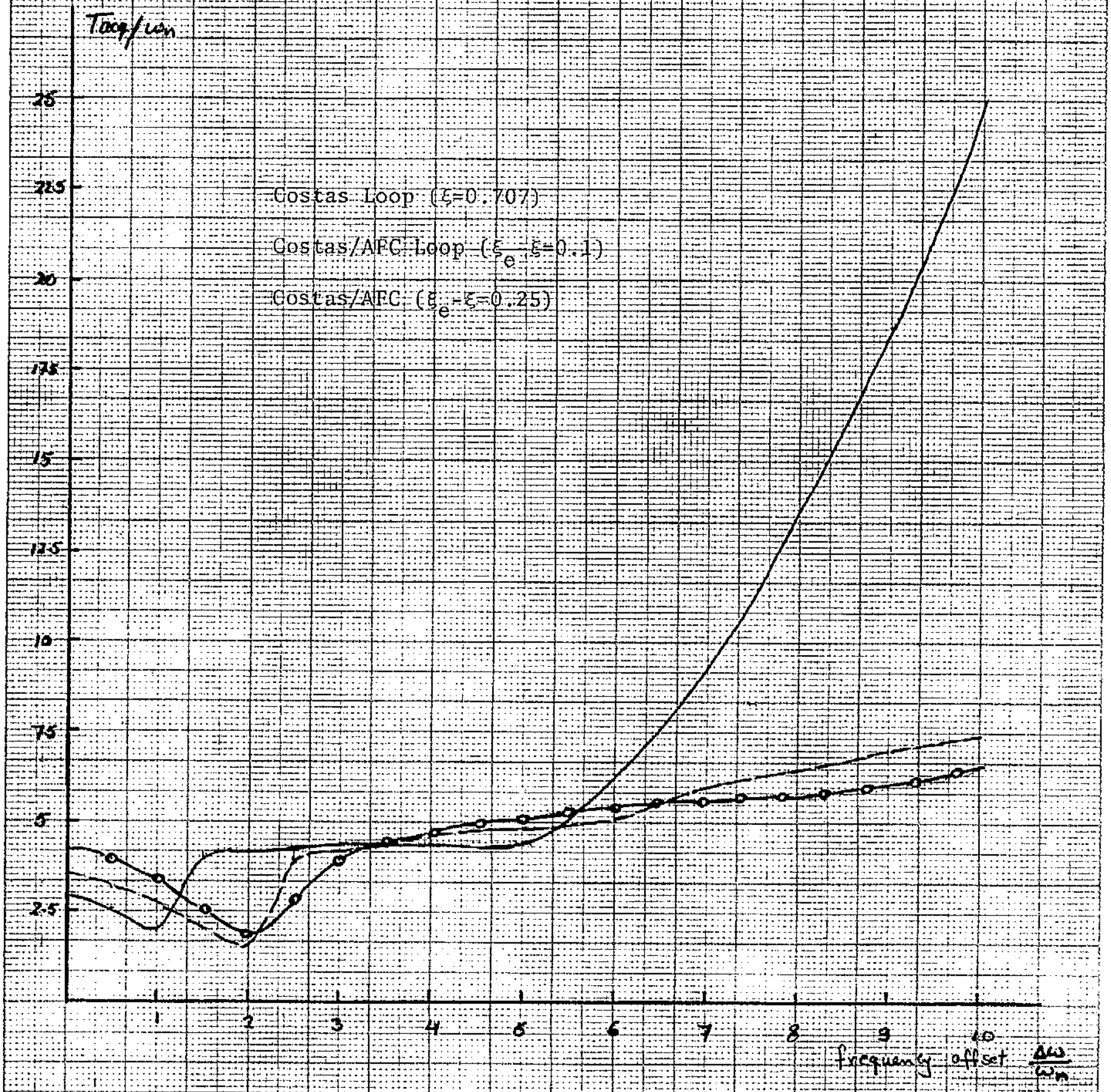


Fig. 2.6 - Loop acquisition time versus frequency offset

factor is 0.707 while the Combined loop has a damping coefficient 0.957. The transient response of the loop to a step-change in frequency was obtained by switching the IF carrier between 445 KHz and 465 KHz at a rate of 100 Hz while the VCO control voltage was monitored on an oascilloscope triggered by the 100 Hz reference. (Fig. 2.7a) shows the VCO control voltage for a sinusoidal input - the acquisition time is approximately 4 m.s. (Fig. 2.7b) is for a sinusoidal input frequency modulated by Gaussian noise such that the carrier to noise ratio was 10 dB. It can be seen that the acquisition time increased to 8 m.s. for the same frequency offset. (Fig. 2.7c) is for TFM input. The increase in acquisition time in (Fig. 2.7b and 2.7c) is mainly due to:

a) As the input SNR decreases, the phase comparator's characteristics cease to be linear and tend to take on a "sinusoidal" shape [6]. This leads to a reduction in the Costas detector sensitivity and, hence a reduction in the loop bandwidth.

b) The frequency comparator characteristic deviates from its linear form [4] as the input SNR decreases.This, in effect, reduces the frequency comparator sensitivity.

It has been noticed that the degradation in the loop SNR increases as the input SNR decreases below 15 dB due to the frequency comparator spurious outputs. However, in these cases the degradation can be eleminated by switching the frequency comparator after the loop achieves lock.

2.4 Loop Components

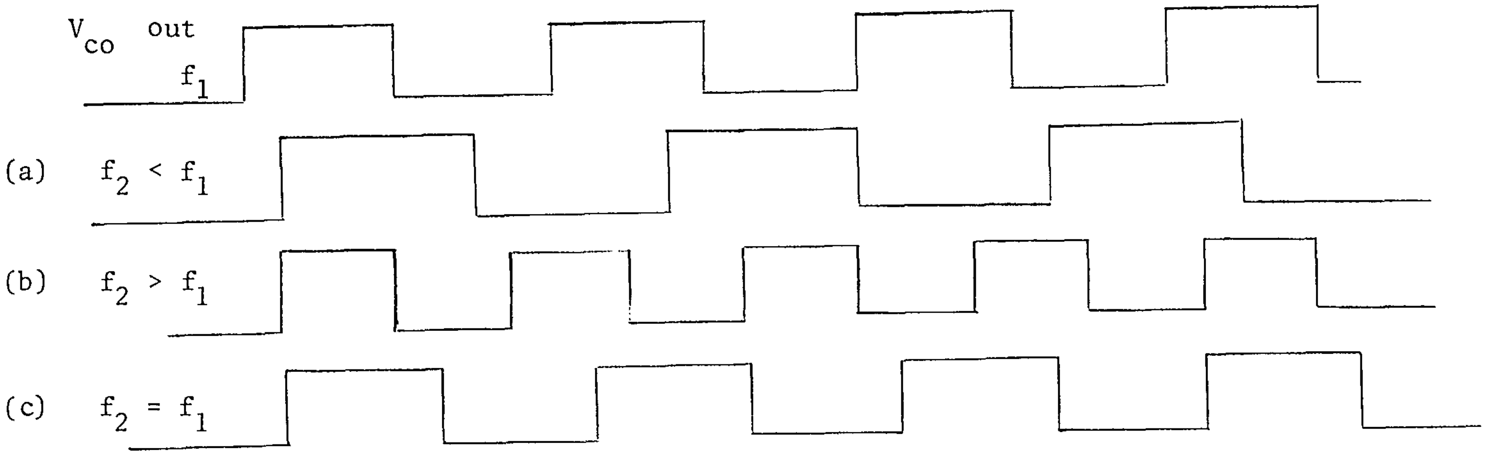
In this Section, a detailed description of the principles of operation of the different loop components utilized in the hardware implementation will be presented.

2.4.1. Rotational Frequency Comparator

Here we assume that the input IF signal is passed through a zero crossing detector before hitting the front end of Costas loop. We also assume that the V_{CO} output is a square Wave at frequency f_1. The different situations to be distinguished by the frequency detector are shown in Figure 2.8.

To view the situation, draw a phasar diagram as shown in Figure 2.9. One cycle of f_2 is shown, and the two phasors represent two relative transitions of f_1. The angle of rotation is readily shown to be $2\pi[\frac{f2}{f1} - 1]$, which is counterclockwise if $f_1 < f_2$ and clockwise if $f_1, > f_2$. Hence detecting the sign of the frequency difference is equivalent to determining the direction of rotation while the magnitude of the frequency difference is related to the angle of rotation. One way for illustrating the frequency detector is to generate the carrier and its quadrature component as shown in Figure 2.10, and designing the hardware such that if a transition (change from 0 to 1) occurs in "B" followed by a transition in C, the detector generates a positive pulse and vice versa for transition in C followed by a transition in B. Each of the four quadrants A,B,C and D is uniquely defined by the carrier

Figure 2.8
Different cases encountered by
Frequency Detector



(a), (b) and (c) are IF input

level and the quadrature carrier level - e.g. B ← 1.1, C ← 0,1. Therefore by sampling the carrier and its quadrature version at the rising edges of the IF signal we can decide what quadrant the IF signal transition occured in. By comparing two consequetive pairs of samples, the direction of rotation can be correclty detected, by using some combinational logic that generates a positive pulse upon detecting a transition from B to C and a negative pulse upon detecting a transition from C to B.

The frequency comparator output will be given by

$$\mu_{fD} = P_r \text{ (positive pusle) } - P_r \text{ (negative pulse)}$$

if the angle of rotion is $\phi = \frac{\Delta\omega}{f1} < \frac{\pi}{2}$, the frequency detector will generate a positive pulse only when the first phasor is within any angle ϕ of the π axis (and located in quadrant B).

$$\therefore \mu_{fD} = P_r\text{(positive pulse)} - 0 = \frac{\phi}{2\pi} - 0 = \frac{\phi}{2\pi}$$

By following the same procedure, a plot for the frequency detector output versus frequency offset can be generated as shown in Figure 2.11.

Effect of phase modulation On the frequency comparator output

Let us consider the case of TFM input signals. In this case the input IF signal phase is a function of time. Therefore the angle of rotation can be written as :

$$\frac{\Delta\omega}{f_1} + \theta_k - \theta_{k+1}$$

Where $\Delta\omega$ is the loop frequency offset, θ_k and θ_{k+1} are the IF signal phase at the k^{th} and $(k+1)^{st}$ positive going zero crossings respectively. Assuming that the frequency offset is less than $\frac{1}{4}\, f_1$, in the absolute value sense, the frequency comparator output will be given by;

$$\mu_{FD} = \frac{\Delta\omega}{f_1} + \theta_k - \theta_{k+1}$$

The average value of the frequency comparator output is given by:

$$\overline{\mu_{FD}} = E\left\{\frac{\Delta\omega}{f_1} + \theta_k - \theta_{k+1}\right\} = \frac{\Delta\omega}{f_1} + E\{\theta_k\} - E\{\theta_{k+1}\} \qquad (2.17)$$

θ_k and θ_{k+1} are not independent. The probability density function of θ_{k+1} conditioned on θ_k is shown in Figure 2.12.

$$\therefore \quad E\{\theta_{k+1}\}|_{\theta_k} = \frac{1}{4}[\theta_k + (\theta_k - \frac{\Delta}{4}) + (\theta_k + \frac{\Delta}{4})] + \frac{1}{2}[(\theta_k - \frac{\Delta}{2}) + (\theta_k + \frac{\Delta}{2})]$$

$$= \frac{1}{4}[3\theta_k] + \frac{1}{2}\,(2\theta_k) = \theta_k$$

$$\therefore \quad E\{\theta_{k+1}\} = \int P(\theta_k)\, E\{\theta_{k+1}\}\,\Big|\,\theta_k\, d\theta_k = \int \theta_k\, P(\theta_k)\, d\theta_k = E\{\theta_k\}$$

$$\therefore \quad \overline{\mu_{fD}} = \frac{\Delta\omega}{f_1} + E\{\theta_k\} - E\{\theta_k\} = \frac{\Delta\omega}{f_1} \qquad (2.18)$$

$$\overline{\mu_{f}{}^{2}{}_{D}} = E\left\{\left(\frac{\Delta\omega}{f_1} + \theta_k - \theta_{k+1} - \frac{\Delta\omega}{f_1}\right)^2\right\} = E\left\{(\theta_k - \theta_{k+1})^2\right\} \qquad (2.19)$$

$$= E(\theta_k 2) + E(\theta^2_{k+1}) - 2\,E\{\theta_k\theta_{k+1}\}$$

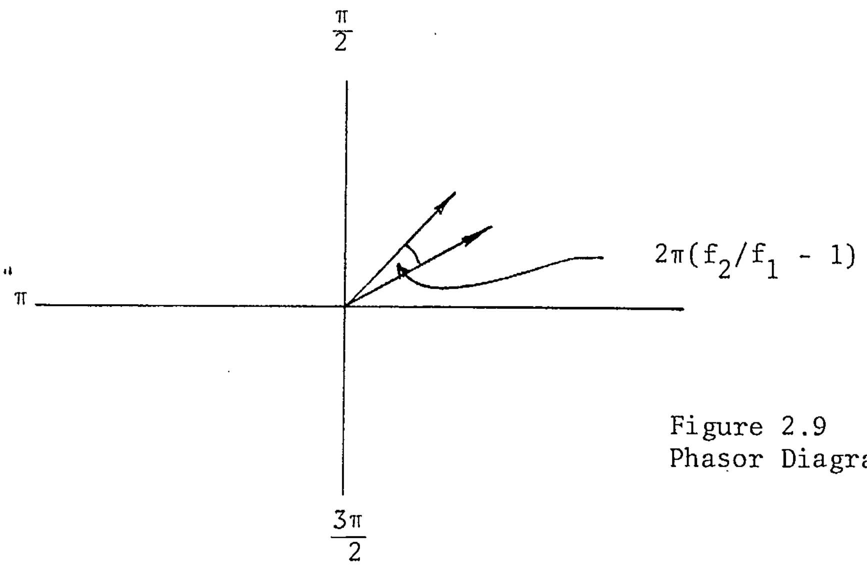


Figure 2.9
Phasor Diagram

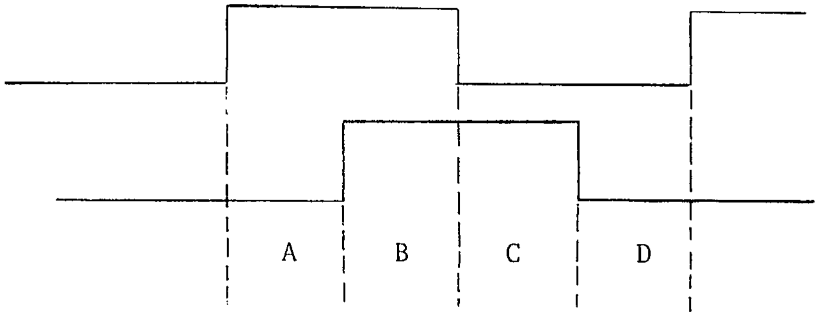


Figure 2.10

The Carrier and Its Quadrature Component

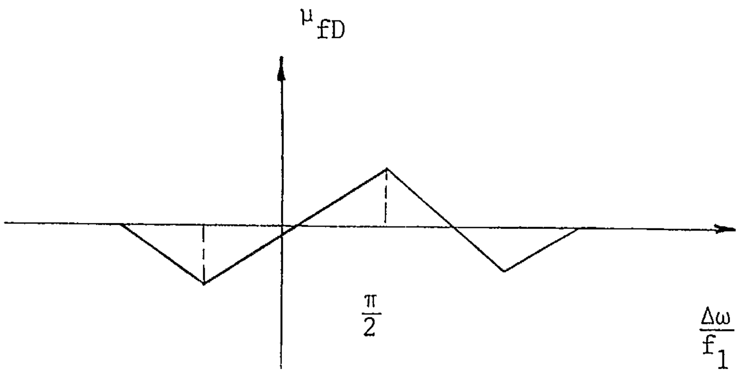


Figure 2.11

Frequency Detector Output versus Frequency Offset using the pdf shwon in Figure 2.12

$$E\left\{\theta^2_{k+1}\right\}\Big|_{\theta_k} = \frac{1}{4}\left[\theta^2{}_k + (\theta_k - \frac{\Delta}{4})^2 + (\theta_k + \frac{\Delta}{4})^2\right] + \frac{1}{2}\left[(\theta_k - \frac{\Delta}{4})^2 + (\theta_k + \frac{\Delta}{2})^2\right]$$

$$= \frac{1}{2}\left[\, 6\ \theta^2{}_k + \frac{\Delta^2}{4} + 2\theta^2_k + \frac{\Delta 2}{2}\,\right] = \theta^2_k + \frac{3\Delta^2}{32}$$

$$\therefore\ E\left\{\theta^2_{k+1}\right\} = \int (\theta^2_k + \frac{3\Delta^2}{32})\ P\ (\theta_k)d\theta_k = \frac{3\Delta 2}{32} + E\left\{\theta^2_k\right\} \qquad (2.20)$$

$$E\left\{\theta_k\theta_{k+1}\right\}\Big|_{\theta_k} = \theta_k\ E\left\{\theta_{k+1}\right\}\Big|_{\theta_k} = \theta_k\ \cdot\ \theta_k = \theta^2_k$$

$$\therefore\ E\left\{\theta_k\theta_{k+1}\right\} = \int \theta^2{}_k\ P\ (\theta_k)d\theta_k = E(\theta^2{}_k) \qquad (2.21)$$

using (4) & (5) in (3) yields

$$\overline{\mu^2{}_{fD}} = E\ (\theta^2{}_k) + E(\theta^2{}_k) + \frac{3\Delta^2}{32} - 2E(\theta^2{}_k)$$

$$= \frac{3\Delta^2}{32} \qquad (2.22)$$

Note that no assumptions have been made concerning the actual probability distribution of θ_k.

In eq. 2.22, $\Delta/_2$ is the max phase change in one IF cucle due to modulation, and is equal to $\pi/2$ divided by 455/16 for TFM

$$\therefore\ \overline{\mu^2}_{FD} = \frac{3\pi^2}{32}\ \text{x}\ (\frac{16}{455})^2 = 0.001144 \qquad (2.23)$$

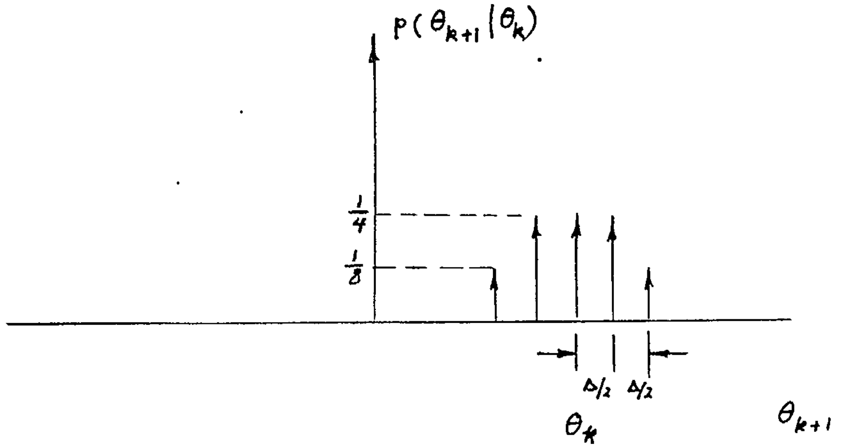


Figure 2.12
probability density function of
θ_{k+1} conditional on θ_k

Frequency comparator implementation

A possible implementation for the frequency comparator is depicted in Figure 2.13. The circuit works as follows:

1. at the k^{th} positive going edge of the IF signal, the carrier and its quadrature component are sampled and the samples stored in 2 D-type Flip Flops (1 & 2).

2. With the $(k+1)^{st}$ positive going edge at the IF signal a new sample is taken and stored in Flip Flop (1 & 2) while the k^{th} sample is shifted to Flip Flops 3 & 4.

3. The 4-input AND gate (5) will generate a short pulse of fixed width only if the k^{th} sample were 0,1 while the $(k+1)^{st}$ sample is 1,1. (positive frequency offset).

4. Gate 6 will generate a short pulse if the k^{th} sample were 1,1 while the $(k+1)^{st}$ is 0,1 (negative frequency offset).

2. 4. 2. AFC Loop Filter

The AFC filter is an integrator. Recall that the frequency comparator has two output parts, one of them is carrying a positive pulse train with density proportional to the frequency offset, while the other is low. Therefore a possible implementation for the AFC filter is to accumulate the count of the FD output pulses - this count will be proportional to the integral of the d.c. component in the FD output. The FD implementation follows the

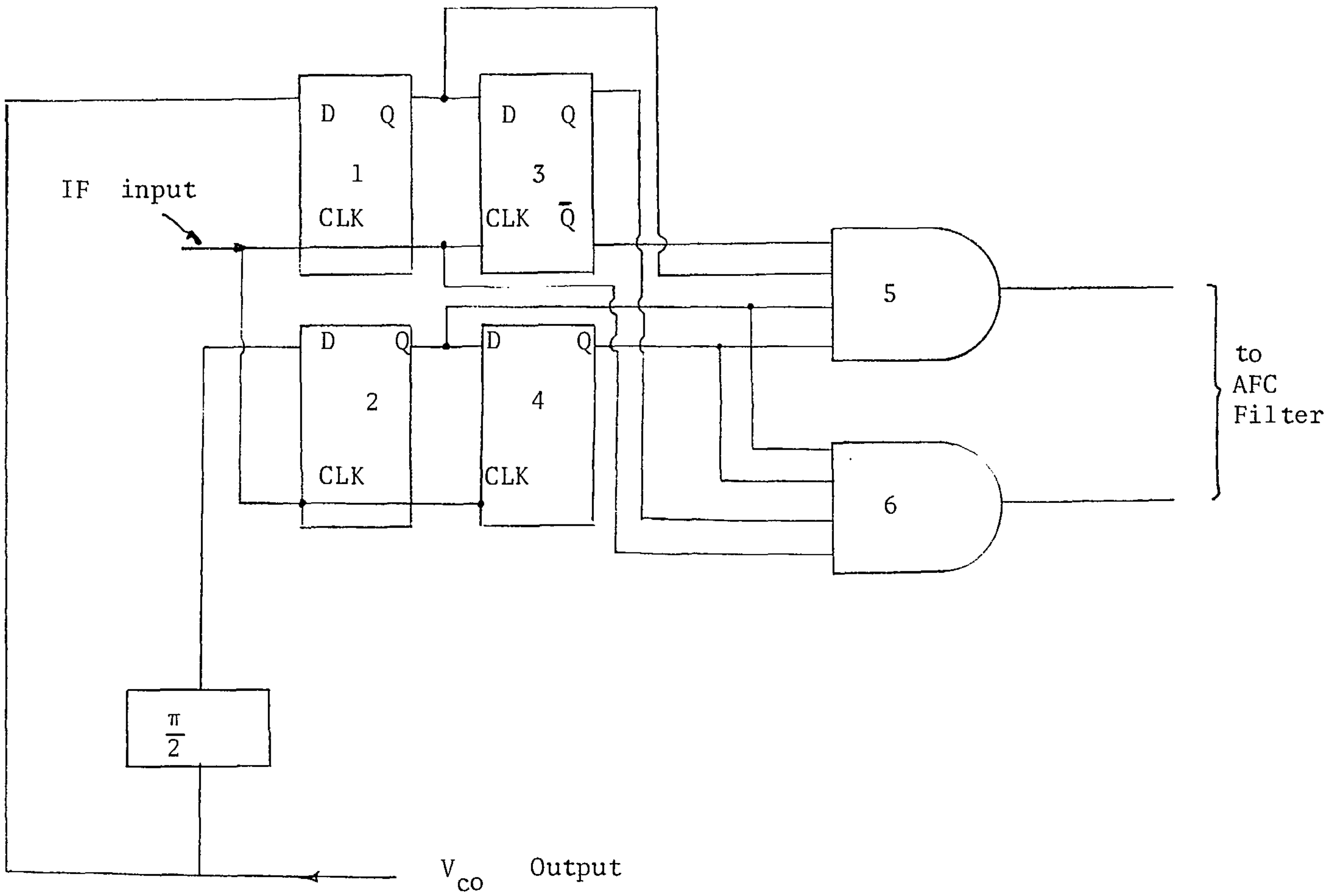


Figure 2.13

Frequency Comparater

strategy depicted in Figure 2.14.

Further simplification is possible by replacing the two binary counters by a programmable UP/DOWN counter as shown in Figure 2.15.

The up down counter in provided with external gating that enables it to be preset to $2^7 = 128$ whenever the reading reaches zero 2^8. Therefore, the counter reading at any time, t, will be a biased estimate for the integral of the frequency comparator output in the time interval starting at presetting the counter until time t. The counter reading is transformed into an analogue signal via an 8-bit DAC.

Model for the AFC filter

The AFC filter output at any time is proportional to the number of all the positive pulses generated up to this time less the total number of negative pulses generated in the same period. This output is incremented by Δ upon receiving a positive pulse, and decremented by the same amount upon receiving a "negative" pulse. If the instantaneous frequency offset is Δf, the positive pulses going to the count-up input will have a density equal to $\alpha\Delta f$ wher α is a constant. Therefore the filter output will be pumped by Δ volts energy $\frac{1}{\alpha\Delta f}$ seconds. If the equivalent integrator time constant is assumed to be τ , then

$$\frac{1}{\tau} \,.\, A\Delta f.\ (\alpha\Delta f)^{-1} = \Delta = \frac{1}{\tau} \,.\, \frac{A}{\alpha}$$

$$\therefore \frac{1}{\tau} = \Delta/(\frac{A}{\alpha})$$

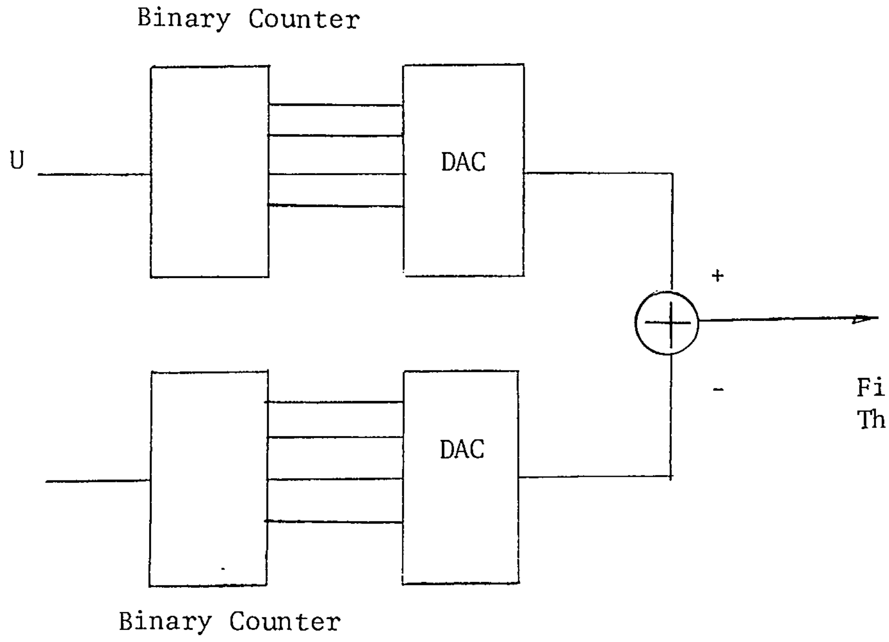


Figure 2.14
The AFC Loop Filter

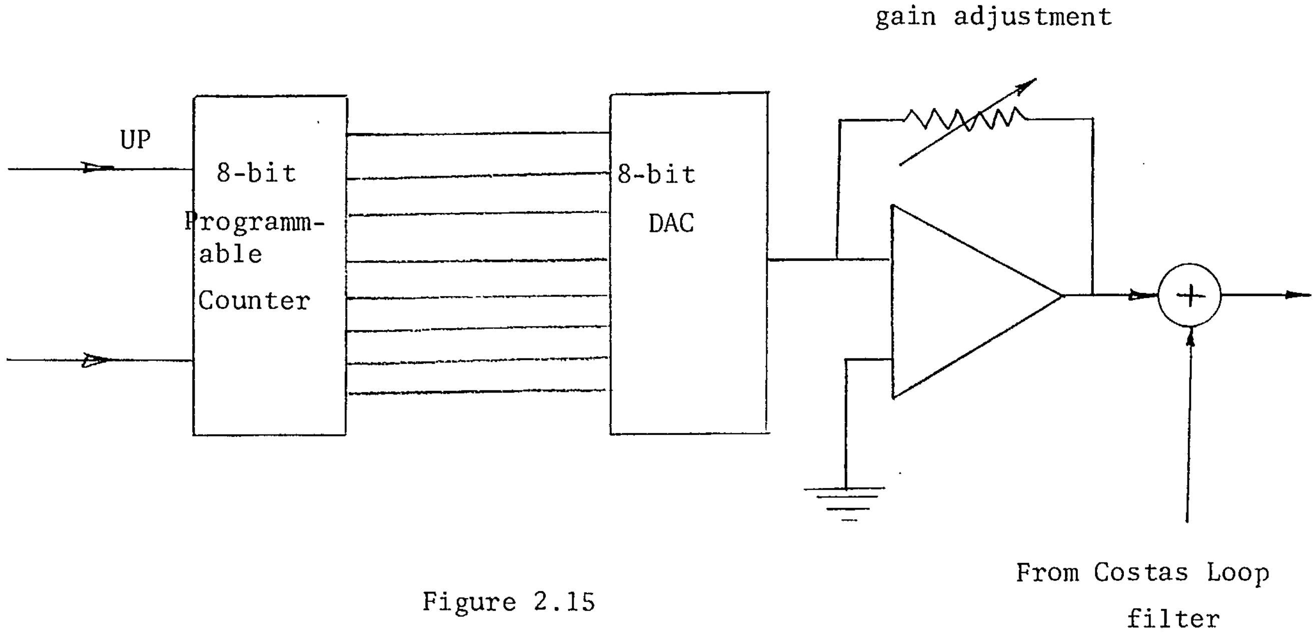


Figure 2.15

The AFC Loop Filter

Using Programmable Counter

where $\alpha \approx 1$, A = frequency comparator gain., Δ= (max voltage swing at the DAC output)/ 2^8.

$$\therefore \frac{A}{\tau} \simeq \Delta \qquad 2.25$$

Therefore one way to control the equivalent filter time constant is by adjusting the increment Δ. This is equivalent to controlling the DAC gain as shown in Figure 2.15.

2. 4. 3. 90° Phase Shifter

This part consists of 2 D-type Flip Flops connected in the manner shown in Figure 2.16 .

The state of this circuit is completely determined by Q_1, and Q_2. There are four possible states for Q_1Q_2 - i.e., 0 0, 01, 11, 10 as shown in Figure 2.17.

It can be seen that it takes four complete clock cycles for the 90° phase shifter output to complete a cycle. It can also be seen that the time difference between the positive going edge at point A and that at point B is exactly one clock cycle - therefore the phase shift between the outputs at A and B is exactly $\frac{2\pi}{4} = \frac{\pi}{2}$ which is independent of the clock frequency.

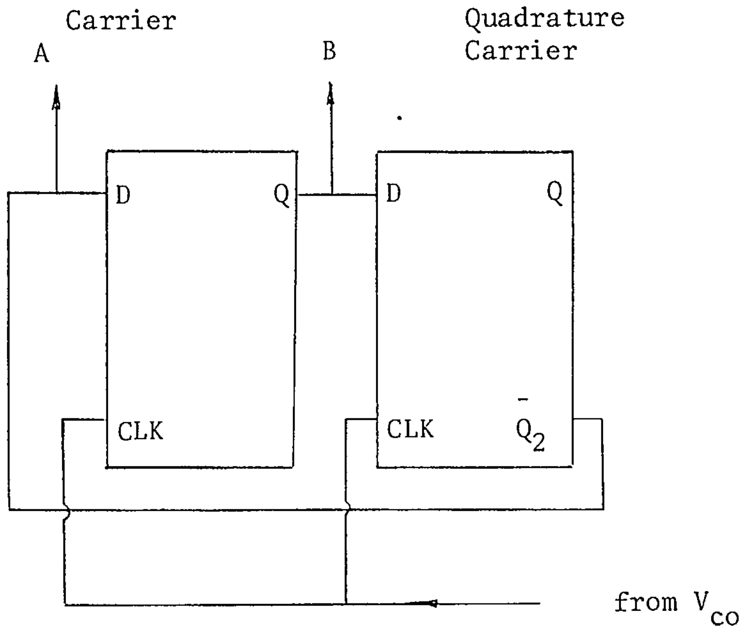


Figure 2.16

$\frac{\pi}{2}$ phase shifter

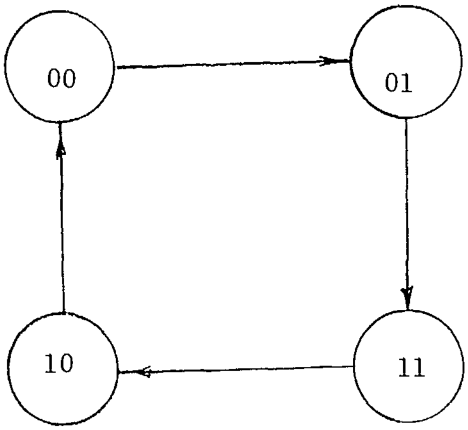


Figure 2.17

State Diagram for the phase shifter

2.5 References

[1] C.R. Cahn "Improving Frequency Acquisition of Costas Loop", IEEE Trans. on Communications, Vol. COM-25, No. 12, pp. 1453-1459. September 1977.

[2] D. Richman, "Color. Carrier Reference Phase Synchronization Accuracy in NTSC Color. Television", Proc. IRE, Vol. 42, January 1954.

[3] C.B. Dekker "On the Application of Tamed Freqeuncy Modulation to Various Fields of Digital Transmission via Radio", Proceedings of NTC, 1979.

[4] D.G. Messerschmitt. "Frequency Detector for PLL Acquisition in Timing and Carrier Recovery", IEEE Trans. on Communications, Vol. COM-29, No. 29, September 1979.

[5] A.J. Goldstein, "Analysis of the Phase Controlled Loop with a Saw-tooth Comparator, BSTJ, pp. 603-633, March 1962.

[6] A. Blanchard, "Phased-Locked Loops", John Wiley, 1976.

SECTION 3

CLOCK SYNCHRONIZATION

3.1. Introduction

Power efficient digital receivers require the existence of a digital clock synchronized to the received bit stream to control "the integrate and dump" detection filters, or to control the timing of the output data stream. Bit synchronization as discussed here is restricted to self synchronization techniques that extract the clock directly from a noisy Non-Return-to-Zero (NRZ) bit stream. This bit format has no spectral line component at the bit rate or its harmonic frequencies for random sequences with 50% transition density.

Scramblers are available to randamize the data-bit sequences preventing a long string of "ones" or "zeros". These devices eleminate the potential line component in the input bit stream by producing a 50% transition density. They also improve the performance of bit synchronizers of the self-synchronizing type.

In other bit-synchronization techniques, some of the signal energy is dedicated for synchronization. For example, a known pattern or additive sinusoidal component is transmitted along with the data.

3.2 Comparative study of self-synchronizing bit-synchronizers

In this section we discuss the most important four bit synchronizers and compare their acquisition and tracking performance in the presence of additive white Gaussian noise. We also compare their reliability and ease of implementation.

2.3.1. Nonlinear Filter Bit Synchronizers:

These synchronizers work on the received noisy bit stream. The whole idea is to filter the input data to eliminate part of the input noise. The filter output does not contain a discrete spectral line at the bit rate. Rather, it is operated upon by a non-linear device which generate a strong spectral component at the bit rate. The output from the nonlinearity is then passed through a narrow bandwidth bandpass filter or a phase locked loop which extracts the required clock. The most widely used types of nonlinearities are the even order ones (especially the second and fourth orders), delay and multiply, the absolute value nonlinearity, and log[cosh (x)] nonlinearity.

a. Square law nonlinearity:

Wintz [1] has shown results for a RC filter with cutoff frequency= 1/T, followed by a square law detector. The expected magnitude of phase error for a raised cosine input waveform is given by [2]

$$\frac{\overline{|\varepsilon|}}{T} \approx \frac{0.33}{\sqrt{KE_b/N_o}} \quad , \quad \frac{E_b}{No} > 5, \quad K \geqslant 18 \qquad (1)$$

where E_b/N_o is the bit energy to noise density ratio, KT is the PLL equivalent bandpass memory. At high signal to noise ratio the probability density of the timing error is approximately Gaussian, and the rms timing error is approximately

$$\frac{\sigma\varepsilon}{T} = 1.25 \frac{|\overline{\varepsilon}|}{T} = \frac{0.411}{\sqrt{KE_b/N_o}}; \quad E_b/N_o >> 1.$$

b. Delay-and-multiply Nonlinearities

This type of nonlinearity operates on a rectangular waveform by forming the product S(t)S(T-Δ) where S(t) is the received periodically clocked random sequence. The best value of the delay Δ to be used is Δ = T/2. This product contains a periodic component at the bit rate which can be filtered by a bandpass filter at f_c=1/T.T. Le Ngoc [3] has shown that the system performance is almost independent of the input signal to noise ratio for high SNR. He showed that the SNR within the PLL B.W. is given by,

$$\left(\frac{S}{N}\right)_L = \frac{1}{BW.T}$$

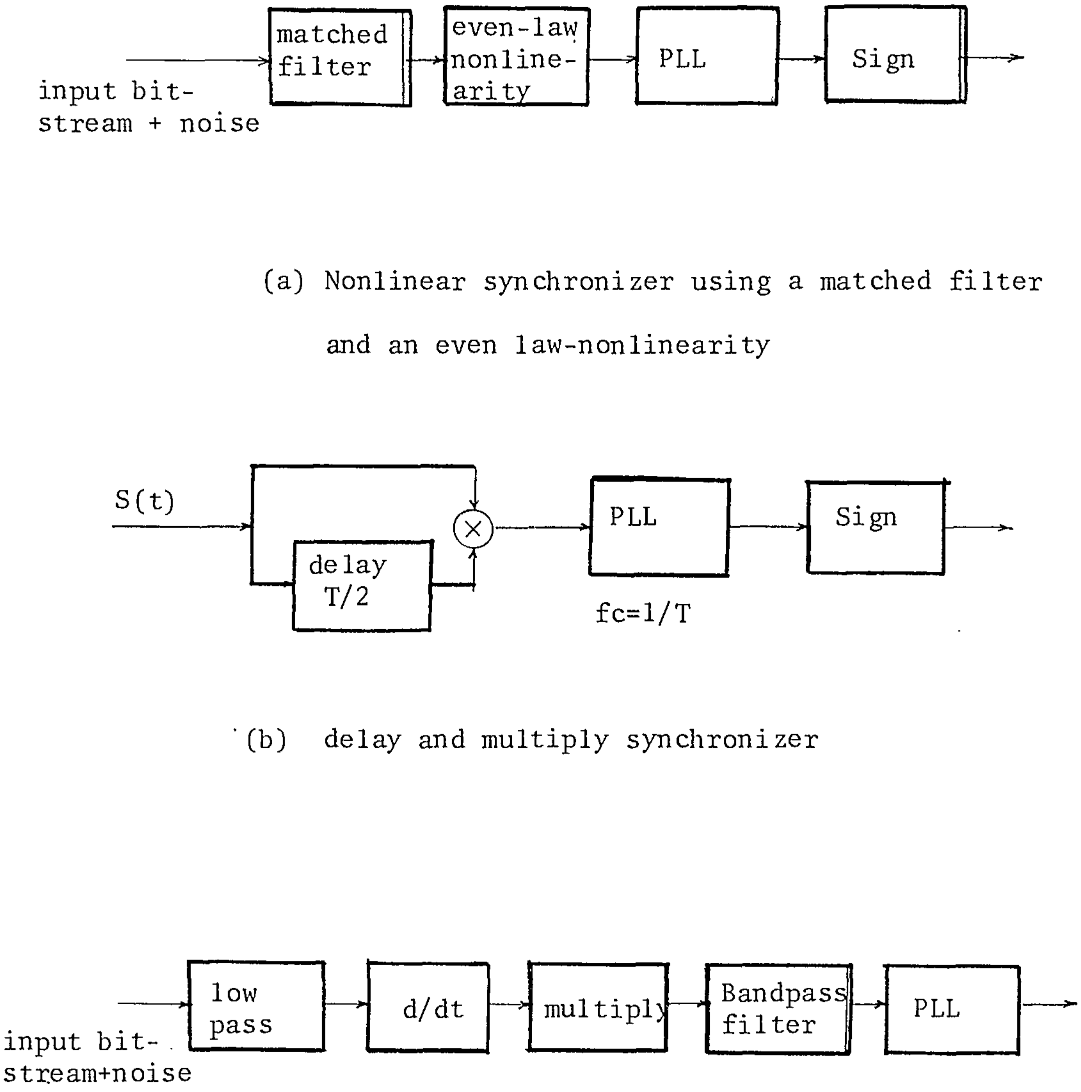
Therefore the rms timing error is,

$$\frac{\sigma_\varepsilon}{T} = \frac{\sqrt{BW.T}}{T} \cdot \frac{T}{2\pi} = \frac{1}{2\pi} \sqrt{BW.T}$$

This value is independent of the actual delay Δ.

C. Differentiator followed by a Square law device;

Spilker [2] has shown that this type of nonlinearity is identical to the delay and multiply for small values of Δ - from this and the discussion in b. We conclude that both nonlinearities have more or less the same performance.



(a) Nonlinear synchronizer using a matched filter and an even law-nonlinearity

(b) delay and multiply synchronizer

(c) differentiate and multiply synchronizers.

Fig. 3.1 - Three Types of Nonlinear Bit Synchronizers

d. Log (cosh x) type nonlinearity;

This type of nonlinearity acts as a square law device for small inputs, and magnitude device for larger input. This nonlinearity is superior to the nonlinearities (a-c), though much more difficult to implement.

Figure 3.1 illustrates the main three types of non-linear filter bit synchronizers. These synchronizers are easy to implement. Due to the existence of a discrete spectral component at the output of the nonlinearity, the acquisition time of these schemes is fast. On the other hand, the timing jitter of these systems is unacceptable for low E_b/N_0 values due to the timing jitter introduced by the "interference" signal at the nonlinearity output [3]. Another drawback of these systems is their falling-out-lock in the presence of long strings of "ones" or zeros".

3.2.2. In-phase Mid-Phase Bit Synchronizers

This synchronizer was first suggested by Lindsey and Tausworthe [5]. It has also been investigated by Simon [6] and Hurd and Anderson [7]. The synchronizer is also referred to as the DTTL or data transition tracking loop because of its method of operation. Both an in-phase channel and a midphase channel are utilized in providing a timing error discriminator (see Fig. 3.2). The inphase branch determines the polarity of the data transitions if and when they occur, while the midphase channel determines the magnitude of the bit timming error.

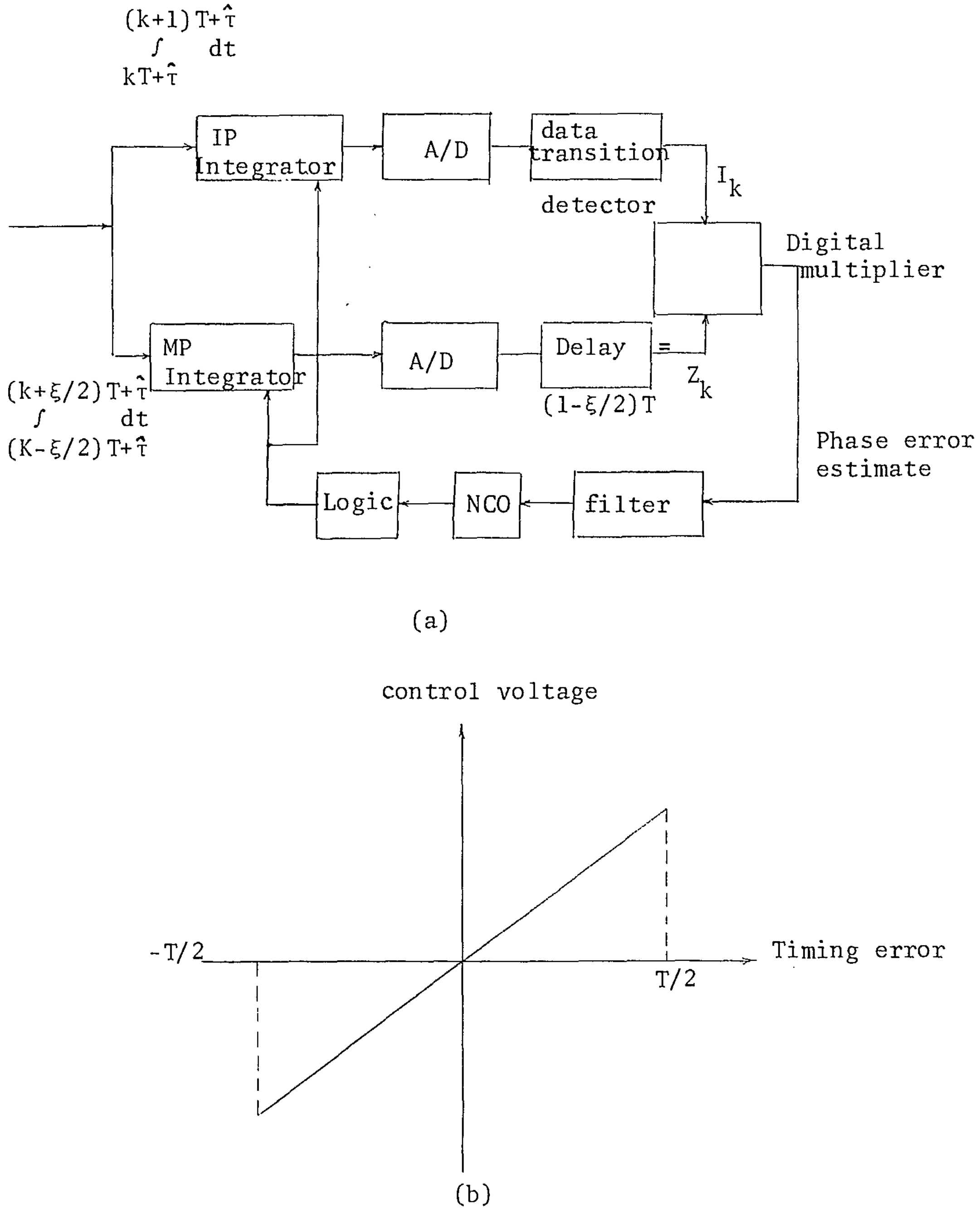


Fig. 32. - In-phase, Mid-phase Bit synchronizer
(a) Block Diagram
(b) error voltage versus timing error

The midphase error signal Z_k is multiplied by $I_k = \mp 1$ if a transition has been sensed, or by $I_k = 0$ if no transition has been sensed. The decision concerning the proper value of I_k is , of course, subject to bit error effects. The filtered output of the multiplier is used to drive the VCO and to control the integrate and dump operations. It is possible to improve the loop noise performance, while the SNR is above threshold, by narrowing the midphase integration window to T/4 An advantage of this scheme is that during long periods between transitions, caused by a long sequence of 1's or 0's, when there are no errors, the discriminator does not allow any noise to perturb the loop; it "hold" the last valid estimate. The loop control voltage for noiseless inputs is shown in Fig. 3.5 [2]. For noisy inputs, the expected value of the loop error voltage $D(\varepsilon)$, where ε is the loop timing error has been shown to be equal to ε [2]. The general formulas for the loop error signal as a function of the loop timing error is given by [6];

$$\frac{D(\varepsilon)}{T} = \frac{\varepsilon}{T} \quad \mathrm{erf}\ [\sqrt{R}(1-\frac{2\varepsilon}{T}) - \frac{1}{8}(1-\frac{2\varepsilon}{T})\left\{\mathrm{erf}\ \sqrt{R} - \mathrm{erf}\ [\sqrt{R}\ (1-\frac{2\varepsilon}{T})=\right\}]$$

where the signal energy is P_sT, the one sided noise spectral density N_o, and $R = P_sT/N_o = E_b/N_o$. The noise spectral density at the multiplier output($\varepsilon = 0$)for stationary input is

$$G_N\ (\omega,\varepsilon)\Big|_{\varepsilon=0} = E(N_k N_{k+m}) \qquad \text{where}$$

$$N_k = \int_{(k-1/2)T}^{(k+1/2)T} n(t)\,dt$$

for values of k where a transition occurs, and zero otherwise.Only the noise density in the vicinity of ω= 0 are of interest because of the narrow bandwidth loop filters.

The noise density at ω= 0 is

$$G_N(0,0) = \frac{1}{T} E(N_k^2) = \frac{1}{2} \left(\frac{N_o N_o}{2\ 4}\right) = \frac{N^2{}_o}{8}$$

For an equivalent closed-loop noise bandwitdth $B_L H_Z$ of the loop in Fig. 3.2 the inverse output SNR is,

$$1/(SNR)_o \cong \frac{GN(0,0)}{A^2\ P_s} B_L = \left(\frac{N_o/4}{A^2 P_s}\right) B_L \simeq \frac{N_o B_L}{4P_s(\mathrm{erf}\sqrt{R})^2} = \frac{B_L T}{4R(\mathrm{erf}\sqrt{R})^2}$$

Where $R = P_S T/N_o$.

the mean square timing error for large $B_L T$ is

$$\frac{\sigma\varepsilon^2}{T} = \frac{1}{(SNR)_o} \simeq \frac{B_L T\xi}{4R}\ , \quad \frac{R}{B_L T} >> 1$$

where ξT is the midphase channel window width. Simon [6] and Hurd and Anderson [7] have shown that use of a midphase window of only ξT=T/4 gives a 3dB improvement over the T/2 window case.

From this discussion we conclude that the DTTL are superior to the nonlinear synchronizers as far as the timing jitter is concerned. However, their acquisition is slower since the error signal is generated only at the moments when data transitions are detected. The DTTL jitter can still be reduced by 3dB without prolonging the loop acquisition time

[7]. This particular synchronizer will be investigated in more detail in section 3 of this chapter.

3.2.3 Early - Late - Gate Bit synchronizers

Another type of closed-loop bit synchronizer is the AVBS or absolute value Early-Late-Gate bit synchronizer [8]. This unit uses early-and-late gate integrate-and-dump channels, having an absolute value operation which makes it bit independent. Simon [8] has found that this scheme provides 3dB improvement in noise jitter over the DTTL, and that it is superior in terms of the mean time to first cycle slip. The acquisition time of the absolute value Early-Late-Gate bit synchronizer is longer compared to the DTTL [2]. This can be seen by comparing the bit synchronizer descriminator characteristic for both loops - it is noticed that the DTTL loop discriminator has stronger correction in the vicinity of $(\frac{T}{2})$ resulting in faster pull in for timing errors greater than T/4.

One important advantage of the AVBS over the DTTL is that its circuity is less complicated when implemented in the analogue domain since any dc drifts in the multipliers(assumed to be identical) will cancel when differencing the two channels' error signals. A schematic block diagram for a AVBS is shown in Figure 3.3.

3.2.4. Maximum Likelihood Bit Synchronizers (MLBS)

If the input pulse waveform is stricly confined to the interval T, then there is a maximum likelihood estimate. However, it is necessary to use a closed-loop tracking synchronizer to accomodate relative phase

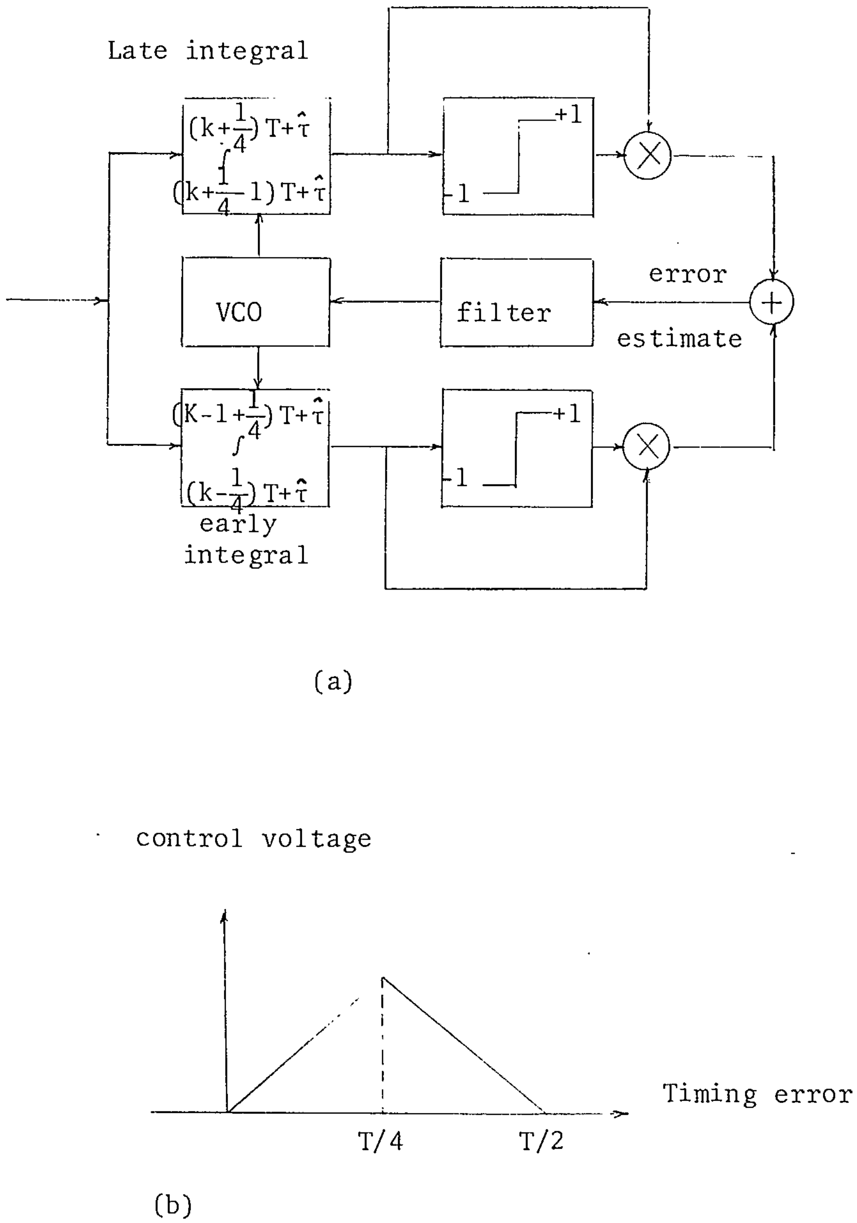

Late integral
$(k+\frac{1}{4})T+\hat{\tau}$
$\int$
$(k+\frac{1}{4}-1)T+\hat{\tau}$
+1
-1
VCO
filter
error
estimate
$(K-1+\frac{1}{4})T+\hat{\tau}$
$\int$
$(k-\frac{1}{4})T+\hat{\tau}$
+1
-1
early
integral
(a)
control voltage
Timing error
T/4
T/2
(b)

Fig. 3.3 - Early-Late-Gate Bit Synchronizer Block diagram (a) and discriminator characteristic (b)

drift between the incoming signal and the local clock. Stiffler [10] and Mengali [11] have devised tracking synchronizers which converge to the ML synchronizers and should have the same phase error statistics.

Mengali's Tracker is shown in Fig. 3.4. Apart from the Tanh(·) block which is difficult to implement, the loop implementation is straight forward. Mengali [11] has shown that this scheme jitter performance is only 20% worse than optimum. However Mengali's scheme does not work for rectangular pulses [9].

3.2.5. Concluding Remarks

The Bit synchronizers discussed previously are compared in terms of their noise jitter performance, acquisition time, circuit complexity and the acceptable signal shapes.

		signal shape	acquisition	noise jitter	complexity
I.	Non linear BS	not square	fast	poor	simple
II.	DTTL	square	good	good	simple
III	AVBS	square	poor	very good	simple
IV.	MLBS	not square	poor	excellent	complex
V.	delay and multiply BS sequences	square	fast	poor	simple

This survey recommends the use of the Absolute-value Early-Late-Gate synchronizer for our application due to the following reasons.

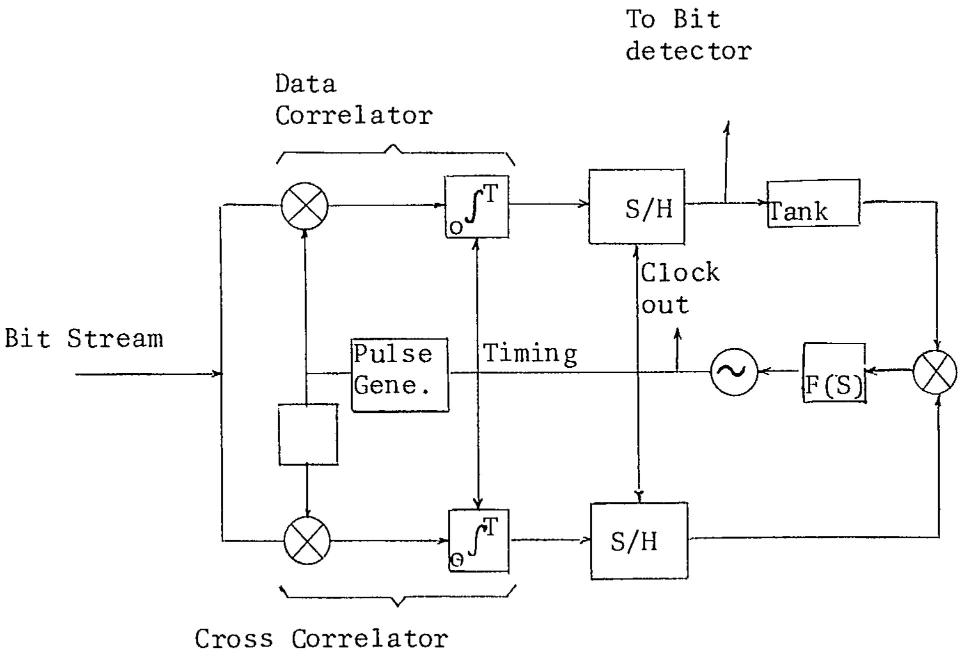


Fig. 3.4 Maximum Likelihood Bit Synchronizer

1. The noise immunity is best when compared to all other possible schemes (I, II & IV).

2. The loop lends itself readily to an all digital implementation.

3. The only drawback about the loop format presented here is its long acquisition time. To combat this difficulty, a new loop discriminator has been investigated, and implemented. As discussed in section 3.3, this modification increased the loop noise immunity in the absence of data transitions, and speeded up the loop acquisition

3.3 Proposed Clock Synchronizer

A block diagram for the clock synchronizer circuit we have developed is shown in Figure 3.6. This circuit will be used in the receiver section of the TFM modem.

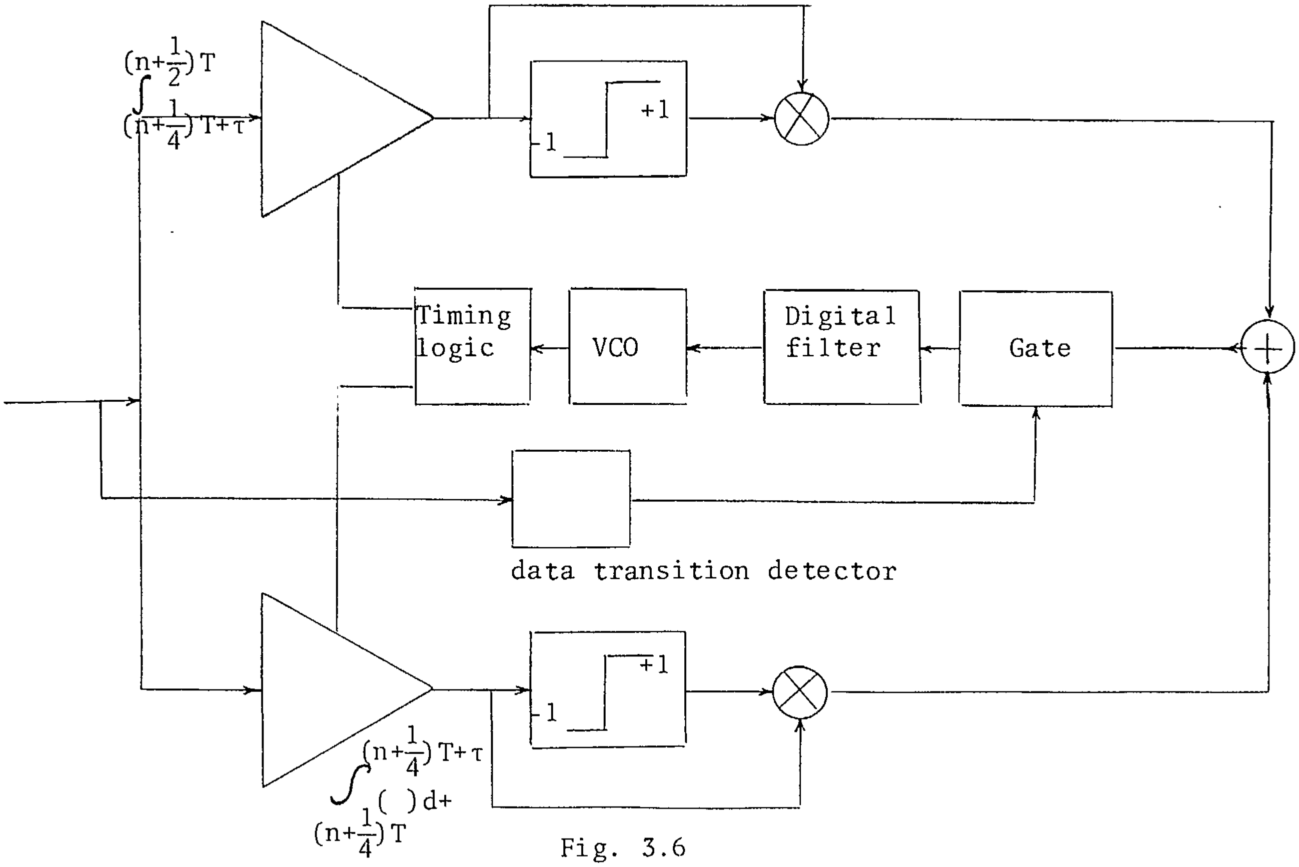

$\int_{(n+\frac{1}{4})T+\tau}^{(n+\frac{1}{2})T}$
+1
-1
Timing logic
VCO
Digital filter
Gate
data transition detector
+1
-1
$\int_{(n+\frac{1}{4})T}^{(n+\frac{1}{4})T+\tau}(\ )d+$

Fig. 3.6

The loop is a modified version of the Early-Late-Gate Bit Synchronizer discussed in section3.2.3. The difference is in the location and width of the integration windows in the Early and Late branches. In the scheme in section 3.2.3 the Early and Late channel windows have the same width, T/2, each. In our loop the early window starts at the rising edges of the quadrature clock and stops at the data transition, while the late-gate window starts at the data transition and stops at the quadrature clock falling edge. The sum of the width of the two windows is therefore, equal to $\frac{T}{2}$, where 1/T is the input data rate.

3.3.1. Loop discriminator characteristic

under the assumption of negligible noise and random NRZ input data with magnitude V_s and $-V_S$ and rate $\frac{1}{T}$, the early channel integrator output at the end of its nth cycle will be:

$$X_{nE} = \int_{nT}^{(\frac{1}{4}+n)T+\tau} a_{n-1} dt$$

$$= a_{n-1} [\tau + \frac{T}{4}] \tag{3.1}$$

and the late gate output will be:

$$X_{nL} = \int_{(n+\frac{1}{4})T+\tau}^{(n+\frac{1}{2})T} a_n \, dt$$

$$= a_n [\frac{T}{4} - \tau] \tag{3.2}$$

The discriminator output $Y_n(\tau)$ for $0<\tau<\frac{T}{2}$ will thus be:

$$y_n\ (\tau) \quad = |X_{nE}|-|X_{nL}|$$

$$= |a_{n-1}|\,[\tau+\frac{T}{4}] \ - \ |a_n|\,|\frac{T}{4}-\tau|$$

$$= \begin{cases} 2V_s\tau & 0<\tau<\frac{T}{4} \qquad (3.3)\text{a} \\ V_s\ \frac{T}{2} & \frac{T}{4}<\tau<\frac{T}{2} \qquad (3.3)\text{b} \end{cases}$$

equation (3.3)b follows from the fact that for $\frac{T}{4}<\tau<\frac{T}{2}$, the early-gate window vanishes, while the late-gate window is always $\frac{T}{2}$. For $\tau<0$, equations 3.1 to (3.3)b are still valid with V_s replaced by $-V_s$ and τ by $|\tau|$. The discriminator characteristic given by equations (3.3)a and (3.3)b is depicted in Figure 3.7.

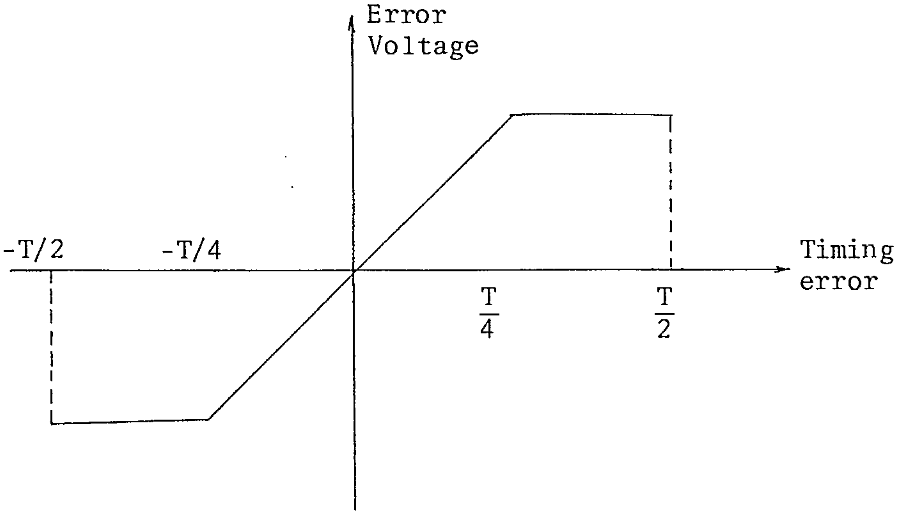


Fig. 3.7 - Discriminator characteristic

It can be seen that $y_n(\tau)$ reaches a maximum at $\tau= T/4$, and stays constant at that maximum for $\frac{T}{4}<\tau<\frac{T}{2}$, unlike the discriminator discussed in section 3.2.3. where the error signal is maximum at $\frac{T}{4}$, and decreases linarly until it reaches zero at $\frac{T}{2}$. This difference will be shown to improve loop acqusition time.

3.3.2 Loop jitter performance

Assume that the input data are corrupted by additive white Gaussian noise, with two sided spectral density N_o; and that the input data and the AWGN are statistically independent. Assume also that the input signal is V_s or $-V_s$, and the data transition probability (transition density) is d. The following is based on the approximate analysis technique suggested by Hurd and Anderson [7]. The authors used the technique for a DTTL after justifying it by means of computer simulation.

The early channel integrator output is given by:

$$X_{nE} = \int_{nT}^{(n+\frac{1}{4})T+\tau} a_{n-1}dt + \int_{nT}^{(n+\frac{1}{4})T+\tau} n(t)dt$$

$$= a_{n-1}\,(\tau + \frac{T}{4}) + \int_{nT}^{(n+\frac{1}{4})T+\tau} n(t)dt$$

the integral on the right hand side is equivalent to passing a WGN process through a linear filter with impulse response $h(t)=U(nT)-U(n+\frac{1}{4}T+\tau)$ where U(x) is the unit step function. It follows that X_{nE} is conditionally Gaussian with the following mean and variance

$$\left.\begin{aligned} E(X_{nE})\,\big|_{a_{n-1}} &= a_{n-1}(\tau+\tfrac{T}{4}) \\ E(X^2_{nE})\,\big|_{n-1} &= N_o(\tau+\tfrac{T}{4})/2 \end{aligned}\right\} \qquad (3.4)$$

Similarly, it can be shown that the output of the late channel integrator, X_{nL} is conditionally Gaussian with the following mean and variance:

$$\left.\begin{aligned} E(X_{nL})|_{a_n} &= a_n(\frac{T}{4} - \tau) \\ E(X^2_{nL})|_{a_n} &= \frac{N_o(T/4-\tau)}{2} \end{aligned}\right\} \tag{3.5}$$

The loop discriminator output, y_n, is given by,

$$y_n(\tau) = |X_{n-E}| - |X_{n-L}| \tag{3.6}$$

Substituting (3.4) and (3.5) into (3.6) and aobserving that the noise terms in (3.4) and (3.5) are independent, it follows that y_n is conditionally Gaussian with the following mean and variance

$$\left.\begin{aligned} E\ \{y_n(\tau)\}|_{d-t} &= 2V_s\tau \\ E\ \{y_n^2(\tau)\}|_{d.t} &= E(X^2_{nE}) + E(X^2_{nL}) = N_oT/4 \end{aligned}\right\} \tag{3.7}$$

where d.t is the condition of data transition occurrence. Assuming that the input symbol error probability is $P_E(R)$, where R is the symbol energy to noise ratio, is given by

$$\left.\begin{aligned} P_E(R) &= \frac{1}{2}\ (1-\mathrm{erf}(R^{1/2})) \\ \mathrm{erf}(x) &= \frac{2}{\sqrt{\pi}} \int_o^x e^{-t^2} dt \end{aligned}\right\} \tag{3.8}$$

A data transition occurs with probability d, and is detected with probability $1-P_E(R)$. Neglecting the interdependence between the phase channel's integral and the data, it is possible to write:

$$E\{y_n(\tau) = d[1-P_E(R)] - E\ \{y_n(\tau)\}|_{d.t}$$

$$= d[\frac{1}{2} + \frac{1}{2}\ \mathrm{erf}\ (R^{1/2})]\ .\ 2V_s\tau$$

$$= \begin{cases} V_s d[1+\mathrm{erf}(R^{1/2})]\tau\ ,\ \tau< \frac{T}{4} \\ \frac{V_s T}{4}\ d\ [1+\mathrm{erf}(R^{1/2})],\ \frac{T}{4}<\tau<\frac{T}{2} \end{cases} \qquad (3.9)$$

Equation (3.9) gives the loop S - curve for high SNR's in the vicinity of $\tau=0$. For large values of τ, the interdependence between the phase channel's integrator and the probability of error in the input data cannot be ignored. This case has not been analysed yet.

The loop discriminator gain, A, defined as in [7]:

$$A = \frac{\partial}{\partial\tau}\ E\{\ \hat{\tau}(T,R)\}|_{\tau=0}$$

where $\hat{\tau}$ is the loop estimate for the timing error. From equation 3.7, the timing error is formed by multiplying $E\{y_n(\tau)\}$ by $1/2V_s$. Thus, for the case of no noise, A will be equal to 1. In the presence of noise, A can be obtained from equation (3.9) as:

$$A = \frac{1}{2}d[1+\ \mathrm{erf}(R1/2)] \qquad (3.10)$$

Thus the AWGN has the effect of reducing the loop discriminator gain and hence increase the acquisition time.

in [2] and [8]. This result is expected since the noise component in the phase integrators output is proportional to width of the integration window [7][8]. Since the noise component from both channels add on a power basis [9], then the noise component at the loop discriminator output is expected to be proportional to the sum of the window lengths (if they are not overlapping). Since the integration windows in our synchronizer are T/2 compared to the T reproted in [2], then the naturally 3db reduction in the timing jitter follows.

3.3.3. Design of the Proposed Bit-Synchronizer:

Due to the high reliability, and low cost of digital IC components, an all digital implementation for the clock synchronizer has been selected. In the following we present a brief description of each of the blocks shown in Figure 3.6.

a. Digital "Voltage Controlled oscillator":

The loop "VCO" has a stable reference clock running at frequency f_0=9.6 MHz, followed by a programmable frequency division chain as depicted in Figure 3.8.

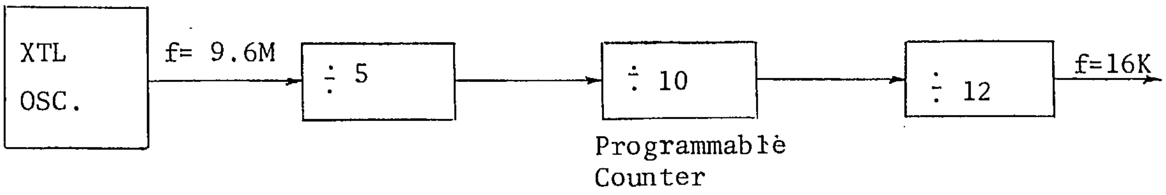


Fig. 3.8 Digital VCO

The noise spectral density at zero frequency is approximately given by,

$$\frac{S(0)}{T} = \left(\frac{1}{2V_s}\right)^2 E\{\hat{\tau}^2\} = \frac{1}{(2V_s)^2} \left[\frac{NoT}{4}\right] = \frac{NoT}{16V_s^2}$$

The loop signal to noise ratio is equal to:

$$\left[\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\,(R^{1/2})\right]^2 \cdot \frac{16V_s^2}{N_oT^2B_L}$$

$$= \left[\frac{1}{2} + \frac{1}{2}\mathrm{erf}(R^{1/2})\right] . \frac{16(E_b/N_o)}{T^3B_L}$$

The time jitter can be approximated as in [2],

$$\sigma_\tau^2 = \frac{1}{(SNR)_o} = \frac{T^2B_L}{1/2[1+\mathrm{erf}(R^{1/2})].16(E_b/N_o)}$$

$$\frac{\sigma_\tau^2}{T^2} = \frac{B_L}{1/2[1+\mathrm{erf}(R^{1/2})].16(E_b/N_o)}$$

$$\frac{\sigma_\tau^2}{T^2} = \frac{B_L}{[1+\mathrm{erf}(R^{1/2})]8(E_b/N_o)}$$

for high signal to noise ratios, erf $(R^{1/2}) \simeq 1$, thus

$$\frac{\sigma_\tau^2}{T} = \frac{B_L\ T}{16\ E_b/N_o} \qquad (3.11)$$

Equation 3.11 suggests that an improvement of 3db in jitter performance can be attained over the Early-Late-Gate Synchronizer analyzed by Simon

The reference oscillator frequency is first divided by 5, resulting in a square wave at 1.96 MHz. The latter is passed through a frequency division chain which consists of 4-bit synchronous counter. When no correction commands are received the counter divides the 1.96 MHz resulting in a square wave at 196 KHZ. When there is a correction command the counter is preset to a control word generated from the loop discriminator (after filtering). Upon receiving the correction command, the programmable counter is preset just once in a clock cycle. The output from the programmable counter is divided by 12 to generate the 16KHz clock.

Now suppose that the programmable counter is set to divide by 11 at some point . This is equivalent to presetting the counter to the count.7 The counter will divide by 11 once, and after it keeps dividing by 10 for the rest of the present clock cycle. Therefore the present clock period will be given by,

$$T'_C = 11T_o+11x10T_o=121T_o \tag{3.12}$$

If no correction commands are received, division by 120 is provided,

$$\therefore \quad T_c = 120T_o \tag{3.13}$$

where $T_o = 1/1.96 \times 10^6$ sec

by comparing 3.12 to 3.13, the resulting change in the output clock phase is found to be

$$\Delta\phi = \frac{1}{120} \times 2\pi = \frac{2\pi}{120} = 3^o \tag{3.14}$$

Summarizing, the output clock phase can be adjusted by an integer multiple of $\Delta\phi$ upon receiving a correction command, and the adjustment in phase is completely determined by the control word generated by the loop error discriminator.

b. Data Transition Detector

The function of this block is to detect the occurrence of data transitions. This is easily implemented as a zero crossing detector. To do so, the input data is delayed slightly by passing it through a D-type Flip Flop clocked by the system referecne clock (1.92 MHz) So, the output data delay will be 0.52 μS at the most.

By Exclusive ORing the delayed data with its undelayed version we get short pulses (0.5μS wide) at all data transition moments. These pulses are used to generate pulses extending from the data transition instants untill the next rising edge of the in-phase clock is encountered (that is the instant at which the programmable counter is supposed to be preset). Those pulses are used to enable the preset function of the divede by 10 programmable counter. The way these functions are implemented is depicted in Figure 3.9..

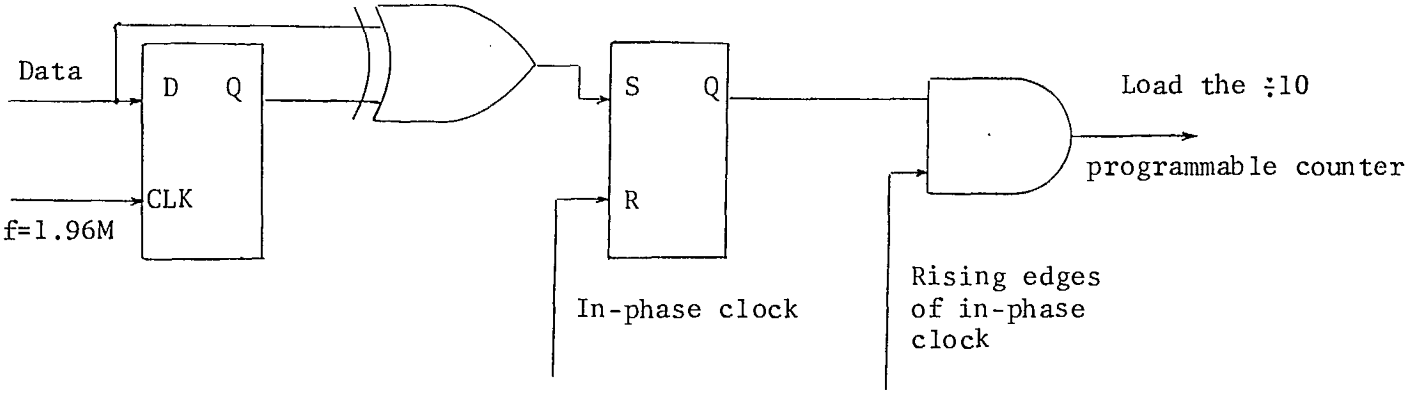

Data
D
Q
CLK
f=1.96M
S
Q
R
In-phase clock
Rising edges of in-phase clock
Load the ÷10 programmable counter

Fig. 3.9 - Data Transition Detector

c. Integrators Gating

The Early-Gate integrator integrates on a window extending from the mid-phase clock rising edge to the data transition moments while the late channel integration window covers the time interval between the data transition and the next falling edge of the midphase clock. This is illustrated in Figure 3.10. The hardware implementation is given in Figure 3.11.

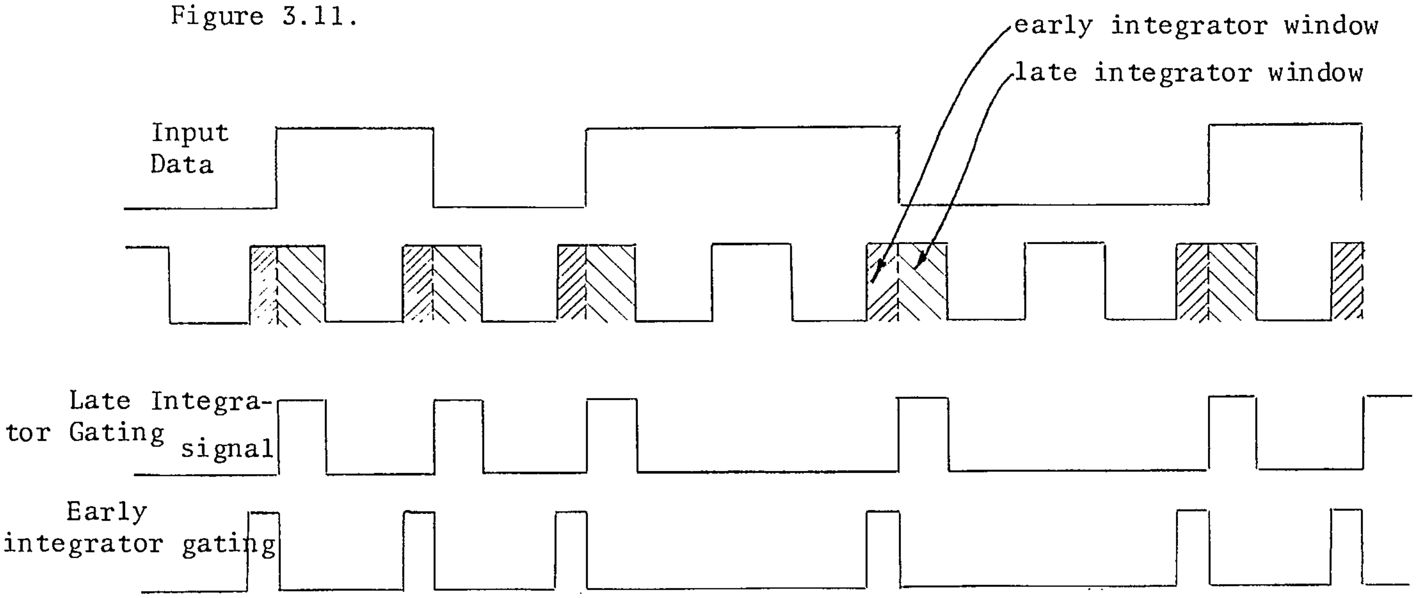


Fig. 3.10

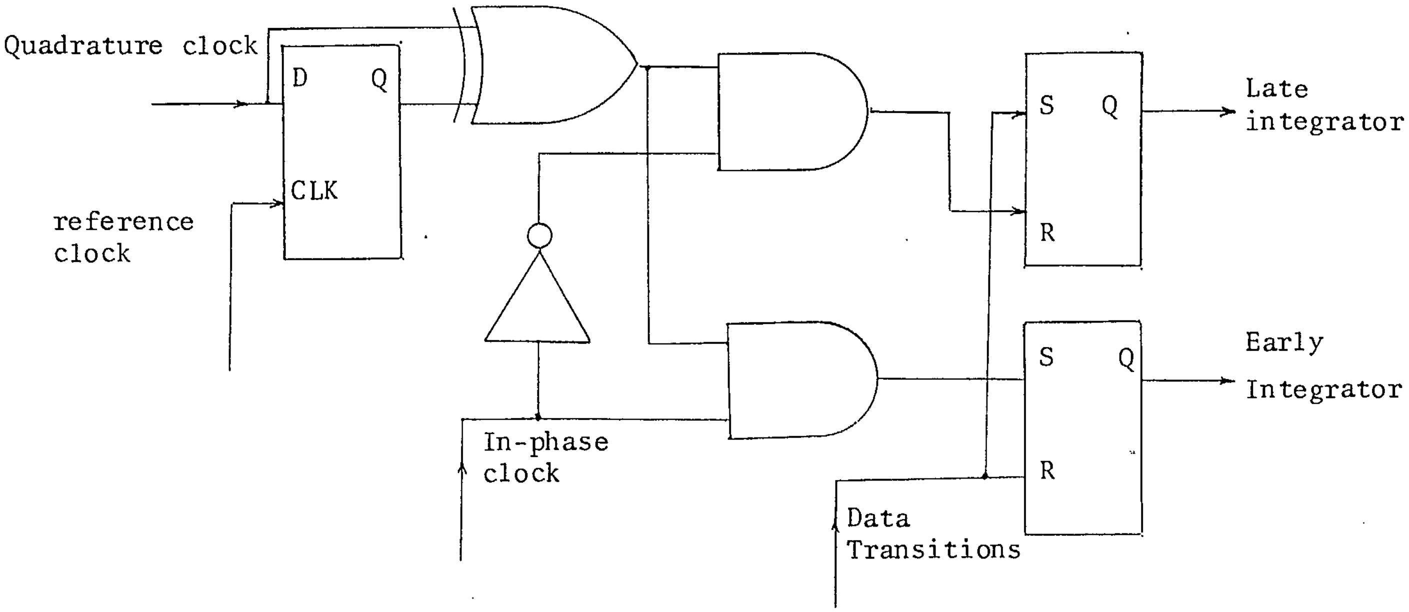


Fig. 3.11

d. Loop Error Discriminator

As discussed previously,the loop error signal is formed by integrating the input data across the integration windows depicted in Figure 3.10 using integrate and dump filters,and taking the difference between the integrators outputs upon completion of each integration cycle (T/$_2$ sec.conincident with the positive half cycle of the midphase clock). If a data transition is detected in the T sec. interval centered around the data transition, then the error signal is used to control the clock phase, otherwise it is excluded. Recalling that the input data is square pulses, then its integral over any interval of time such that no data transition occurs during the integration process will be proportional to the integrations interval duration, with the proportionality factor equal to V_s/τ', where V_s is the data voltage and $\tau'$the integrator time constant. So, the integration process and taking the absolute value, is equivalent to estimating the integration window width. A quantized version of the integration window estimate is obtained by counting the number of a reference clock cycles occuring during the integration window, and the window width will be proportional to the counter reading at the end of its counting period (assuming that the counter was preset at 0 at the beginning of its counting cycle). The only constraint on the reference clock frequency is the counter length, and it has to be in phase coherence with the midphase clock. Implementing the Discriminator followed the strategy shown in Figure 3.12.

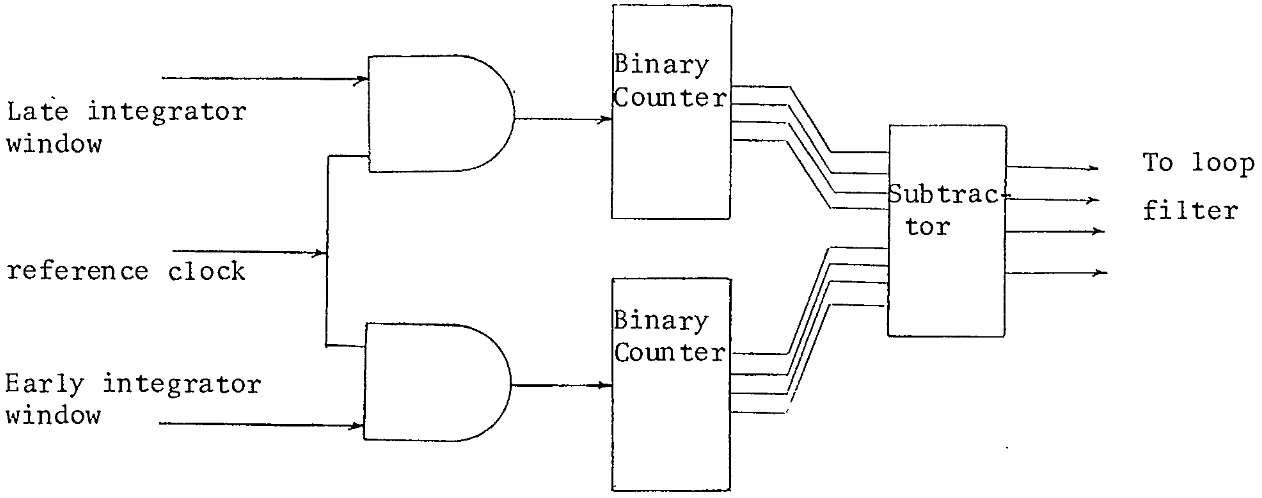


Fig. 3.12

Further simplification for that section is still possible by replacing the two binary counters and the subtractor by an Up/Down programmable counter. In the hardware implementation of the loop, the reference clock frequency is 24 times the output clock frequency. To ensure phase coherence between the reference clock and the output clock, the first is generated from the same master clock by frequency devision. The actual implementation for the loop discriminator is depicted in Figure 3.13.

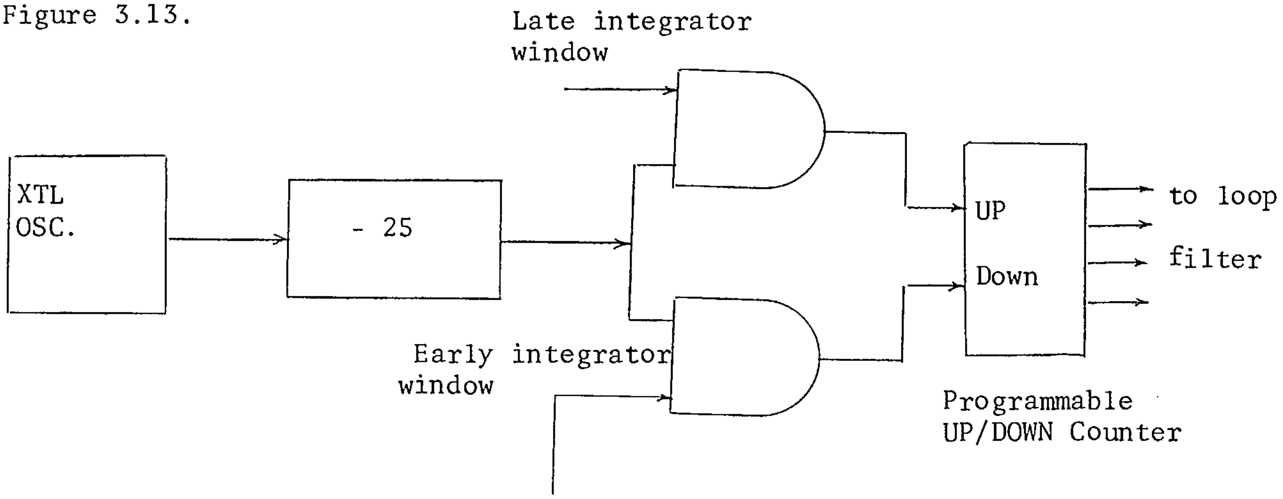


Fig. 3.13

e. Loop Filter

In order to get a better estimate for the loop timing error, the control word at the output of the loop discriminator has to be filtered. The filter has to be designed to serve two purposes, i.e.

1. reduce the loop timing jitter due to input noise
2. to enable the loop to accomodate slight frequency offsets, e.g. if we assume a frequency stability of 10^{-4}, then the loop should be capable of acquiring ∓16Hz frequency offset. A first order loop with noise Bandwidth 32Hz can serve the second requirement at the expense of a steady state phase offset and increase in the steady state phase jitter due to noise. However, the Data clock is namely generated from a crystal stabilized reference. If we assume a crystal stability of 10^{-5}, which is common, the expected frequency offset will be within ∓0.16. Hence a first order loop having a bandwidth equal to 10^{-5} of the bit rate might be preferrable if extremely short acquisition times are necessary. The loop transient and steady state response to a phase step input and frequency step input in the absence of noise is discussed in detail later.

The loop filter has been implemented as a proportional path plus integral, though the integrator part can be switched off at will if the first order loop is preferred. The loop filter implementation follows the strategy depicted in Figure 3.14.

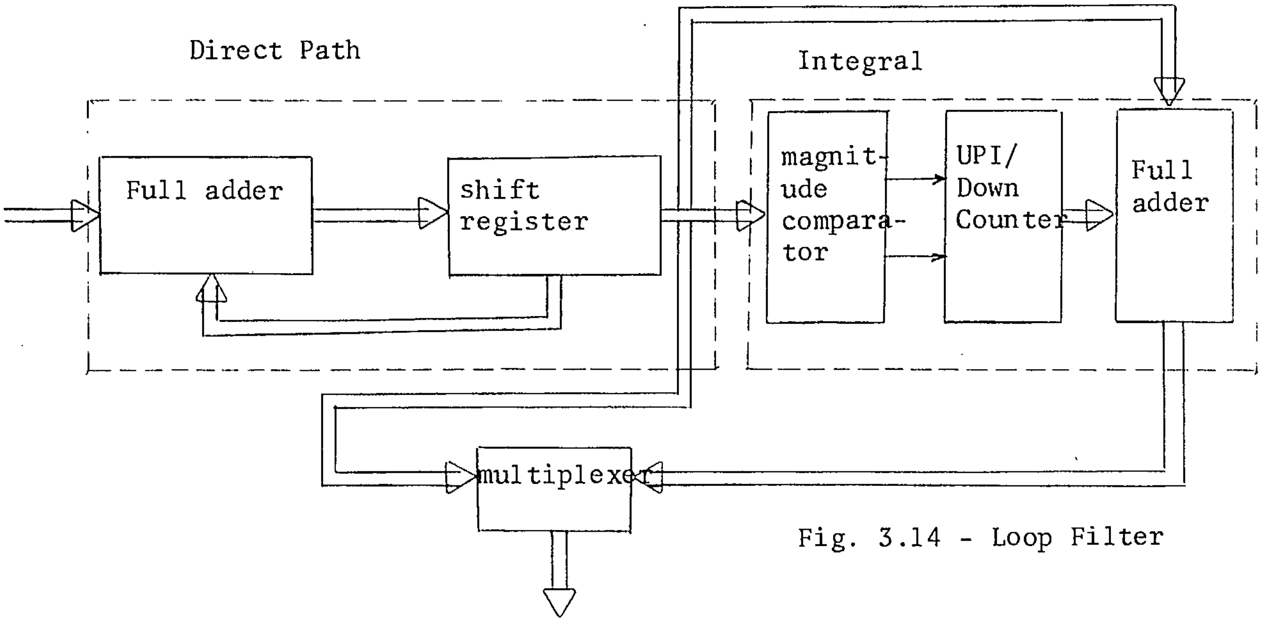


Fig. 3.14 - Loop Filter

In the direct path, the control word is accumulated over a number of data transitions, then fed to the loop integrator. The output of the Direct path"accumulator" is added to the integrator output. The resulting control word is multiplexed with the direct path output.

The multiplexer output is dependent on the state of its control input i.e. The output can be identical to the direct path word or the direct path plus integral word . This provides the capability of switching the integrator on an off at will.

The detailed loop hardware implementation follows closely the strategy we just discussed. It has been all implemented using TTL logics to avoid the excessive propagation delays in the frequency division chains associated with CMOS logic. The detailed circuit diagrams are given in Appendix A.

3.3.4 Loop acquisition performance

In this section the loop transient response to a step in phase, and a step in frequency is investigated based on an analytical model of the actual loop components.

a. Loop Discriminator model

With reference to Figure 3.14, the loop control word is generated by gating the reference clock signal, which is 2M times the data clock in frequency, with the midphase clock. The number of cycles preceeding the data transition, X_D, is fed to the count Down, input while the number of cycles following the data transition, X_u, is fed into the count-up input of the programmable Up/Down Counter. If the data transition occurs in time slot L.., then

$$\begin{aligned} X_D &= L, \\ X_u &= M-(L-1) \end{aligned} \qquad (3.12).$$

and if the transition occurs in time slot L', then

$$\begin{aligned} X_D &= L \\ X_u &= M-L \end{aligned} \qquad (3.13)$$

assuming the maximum count of the UP/DOWN COUTNER to be 2K-1, the counter reading at the end of any Counting cycle, during which a data transition occurs, is given bt

$$\begin{aligned} R &= [[(K-1)-X_D] \bmod 2K+X_u] \text{ modulo } 2K \\ &= [(K-1)+X_u-X_D] \text{ modulo } 2K \end{aligned} \qquad (3.14)$$

where (K-1) is a predetermined preset value for the UP/DOWN Counter at the beginning of each counting cycle. Substituting equations (3.12) and 3.13 into 3.14 yields,

$R = (K+M-2L) \text{ mod. } 2K$ for data transitions in time slot 1 (3.15)

$R = (K+M-2L-L) \text{ modulo } 2K$, for data transition in time slot 1'(3.16)

To avoid counter overflow in any counting cycle M should be chosen such

$$K + M \leq 2K - 1$$

$$\& \quad K - M \geq 0$$

The maximum allowable vlaue for M to avoid overflow is $M = K - 1$. Rewriting 3.15 and 3.16 for $M = K - 1$;

$$R = (2K-2L-\ell) \bmod 2K \tag{3.15a}$$

$$R = (2K-2\ell-2) \bmod 2K \tag{3.16a}$$

For timing errors, τ, such that $T/4<\tau<\frac{T}{2}$, $X_D=0$, $X_u= 1_{max}$ $= M = K - 1$. This results in:

$$R = 2K-1 \qquad \frac{T}{4}<\tau< \frac{T}{2} \tag{3.17}$$

for timing errors such that $-\frac{T}{2}<\tau< -\frac{T}{4}$, $X_u=0$, $X_D= \ell_{max}$

$$\therefore\ R = 2K-2-2(K-1)=0 \qquad -\frac{T}{2} <\tau<-\frac{T}{4} \tag{3.17a}$$

If the loop estimate for the data clock phase is correct, the data transition will always occur in the time slot $1'=\frac{M+1}{2}$, while the counter reading will be $2K-2-K+1=K-1$. This fact suggests that the loop estimate of the clock phase will be proportional to $\overline{R}$ where,

$$\overline{R} = R-(K-1) \tag{3.18}$$

We observe that if the data transition occurred in time slot 1, the corresponding phase offset will be $\Delta\phi$ where

$$\frac{\pi}{2M}(L-\frac{M+1}{2})>\phi> \frac{\pi}{2M}(L+1-\frac{M+1}{2}) \tag{3.19}$$

equations (3.15a) - 3.19 yield the loop S-curve which is shown in Figure 3.15 for the case K=8

b. Model of the first order Loop Filter

The first order loop filter is essentially an accumulator that averages the loop discriminator reading at the end of the counting cycles during which data transitions occur. For N counting cycles, this process takes a time NT, where T is the clock period if the input data were alternating. After each N consecutive data transitions the accumulator contents is fed back to the ÷ 10 programmable counter [see Figure 3.8] and conseo quently the local clock phase is pumped by $\Delta\overline{R}$ in order to reduce the loop phase offset, where Δ is the increment in clock phase due to adding or deleting one pulse of the reference clock.

c. Loop Response to a step in Phase

It has been mentioned in the introduction that the loop input consists mainly of random NRZ binary data plus additive white Gaussian noise. This, coupled with the fact that the loop discriminator output is valid only when data transitions occur, the loop transient response will be strongly dependent on the particular input data sequence which is not expressible in any closed form expression due to its random nature. Cosnequently we have to resort to numerical and graphical techniques.

Graphical Techniques

According to the previous discussions, the change in the Digital VCO phase versus the loop phase is shown in Figure 3.15, with the verticcal axis units multiplied by Δ . This is redrawn as Figure 3.16. Assuming

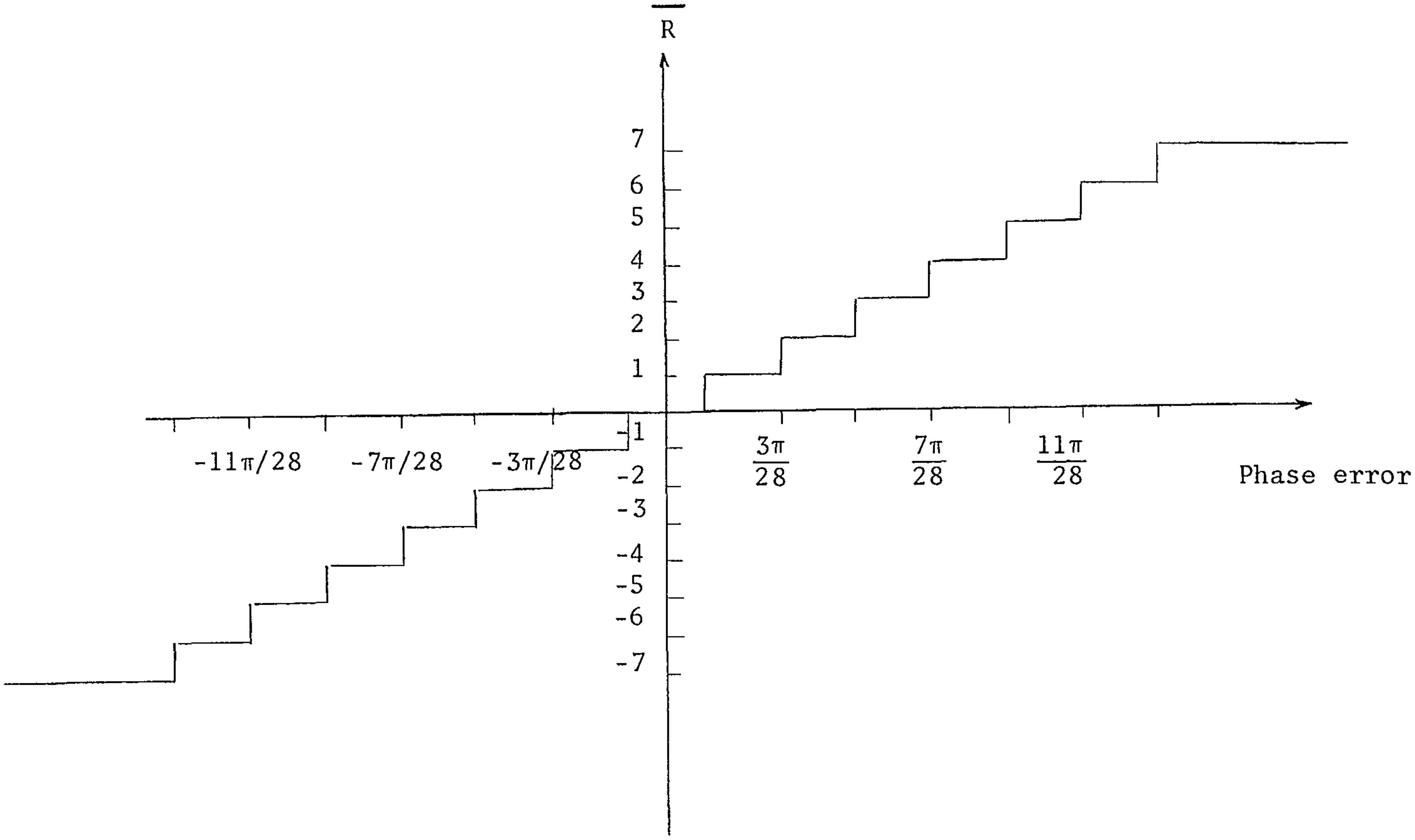


Fig. 3.15 Loop Discriminator characteristic

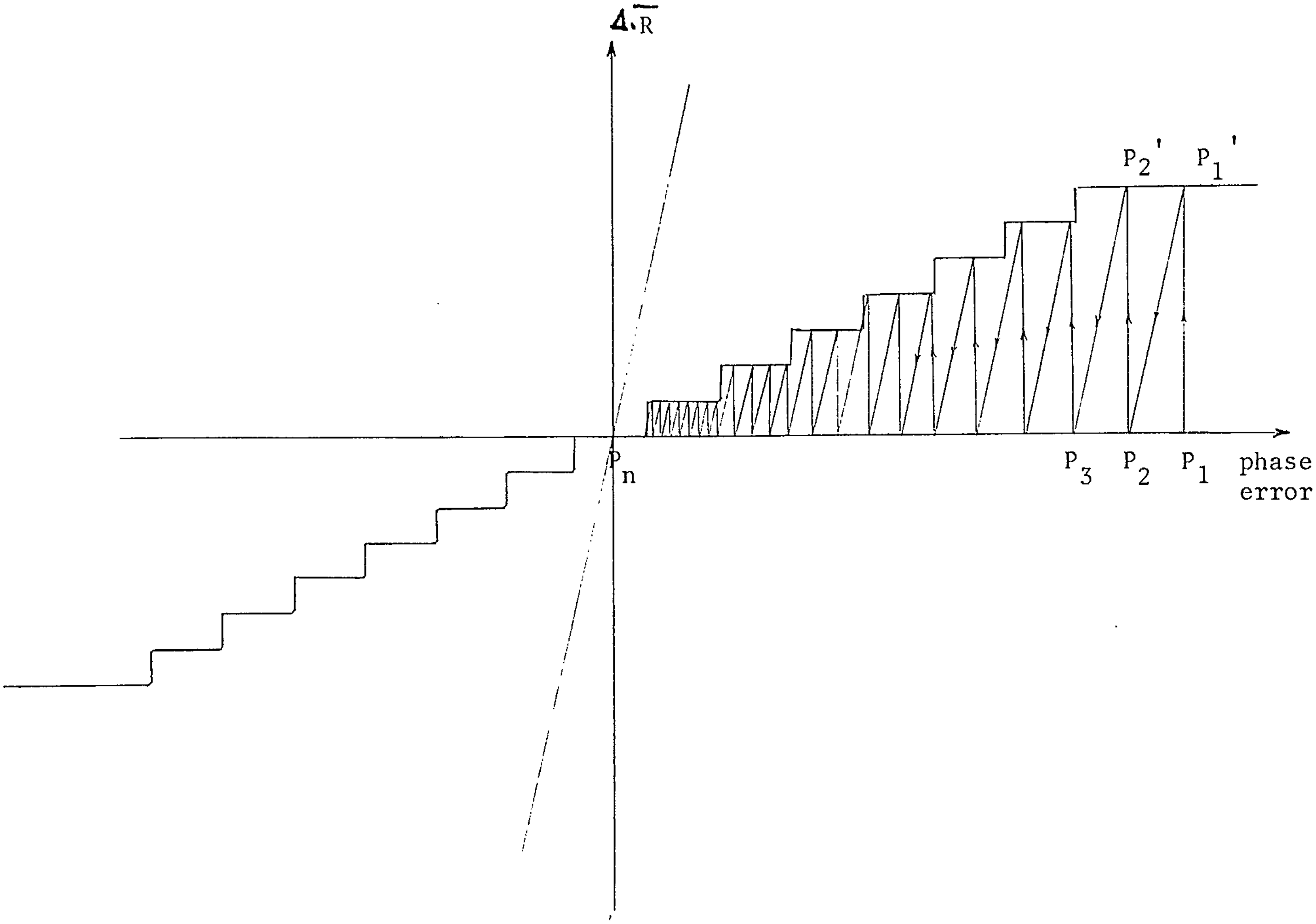


Fig. 3.16 Loop Response to a step in Phase

that the horizontal axis scale is X/rad, and the vertical axis scale is Y/rad. Draw line C with slope* = X/Y. Let the input step in phase be P_1, project P_1, vertically on P_1' - project P_1' parallel to C on P_2. P_2 will be equal to the loop error following the first phase correction cycle. We repeat this process until the discriminator' dead zone is reached for the first time at P_n. The loop phase offset will remain equal to the value corresponding to P_n as long as the input data transitions are jitter-free. The number of data transitions required for the loop to achieve phase lock is equal to the number of vertical lines crossed until the phase error reaches the dead zone for the first time. A closed form expression can be obtained for the loop phase acquisition time if we ignore the effect of the discriminator quantization. The corresponding characteristic is shown in Figure 3.16. Assuming that the initial phase step is less than $\pi/2$, and that the phase offset at P_R is ϕ_R, and at P_{R+1} is ϕ_{R+1} we have,

$$\phi_{R+1} = \phi_{R-g\phi_{R+1}} \qquad (3.20)$$

where g = loop filter gain multiplied by $\Delta/(\pi/2M)$ (3.21)

$$\therefore \quad \phi_{R+1}(1+g)=\phi_R$$

$$\phi_{R+1} = \phi_{R/}(1+g)$$

$$\phi_1 = \phi_0/(1+g)$$

$$\phi_2 = \phi_1/(1+g) = \phi_o/(1+g)^2$$

$$\phi_b = \phi_o/(1+g)^n \qquad (3.22)$$

Where ϕ_o is the input phase step.

* $(X/Y = \frac{\pi}{2M}/\Delta N$ loop filter gain)

The loop is said to have reached lock when the phase offset is less than or equal to some threshold, δ. To get the number of data transitions required for the loop to reach lock we solve (3.22), for n, subject to the condition $\phi_n < \delta$

$$\log \phi_n = \log \phi_o - n \log (1+g);$$

$$\log \phi_o - n \log(1+g) < \log \delta;$$

$$n \log(1+g) > \log(\phi_o/\delta) \quad ; \quad \text{Thus}$$

$$N_{acq} = \log(\phi_o/\delta)/\log(1+g) \qquad \phi_o < \frac{\pi}{2} \qquad (3.23)$$

and for $\frac{\pi}{2} < \phi_o < \pi$, we obtain:

$$N_{acq} = \frac{\log\left(\frac{\pi}{2\delta}\right)}{\log(1+g)} + \left\lfloor (|\phi_o| - \frac{\pi}{2})/g\frac{\pi}{2} \right\rfloor + 1 \qquad (3.24)$$

Using equation 3.24, N_{acq} versus ϕ_o, with g as a parameter is plotted in Figure 3.17.

Computer Simulation

Using the computer simulation technique suggested by C.P. Reddy and S.C. Gupta [12], with appropriate modifications, the loop response to an input phase step has been investigated for different combinations of M & N. The results are shown in Figures 3.18-3.21.

$g = 10^{-3}$	ϕ_o	0.1	0.5	1.0	1.5	2	3	2.5	
	$N_{acq.}$	1	805	1152	1355	1652	2289	1971	
$g = 10^{-2}$	ϕ_o	0.1	0.5	1	1.5	1.57	2	2.5	3
	$N_{acq.}$	1	161	231	272	277	304	336	368
$g = 10^{-1}$	ϕ_o	0.1	0.5	1	1.5	1.57	2	2.5	3
	$N_{acq.}$	1	16.8	24	28	29	32	35	38
$g = 0.5$	ϕ_o	0.1	0.5	1	1.5	1.57	2	2.5	3
	$N_{acq.}$	1	4	6	7	7	8	9	9

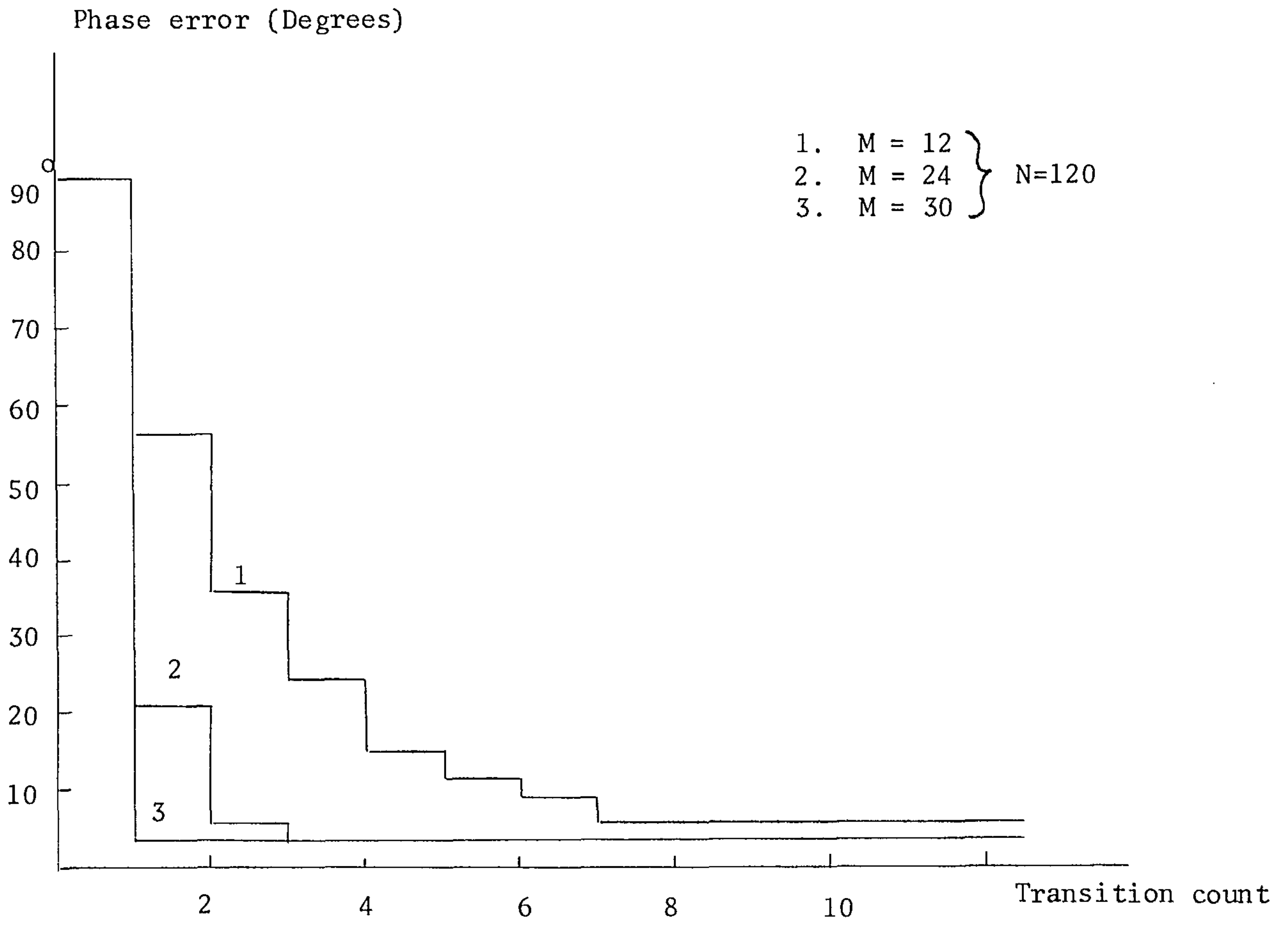


Fig. 3.18 Effect of No. of qunatization levels on the loop acquisition

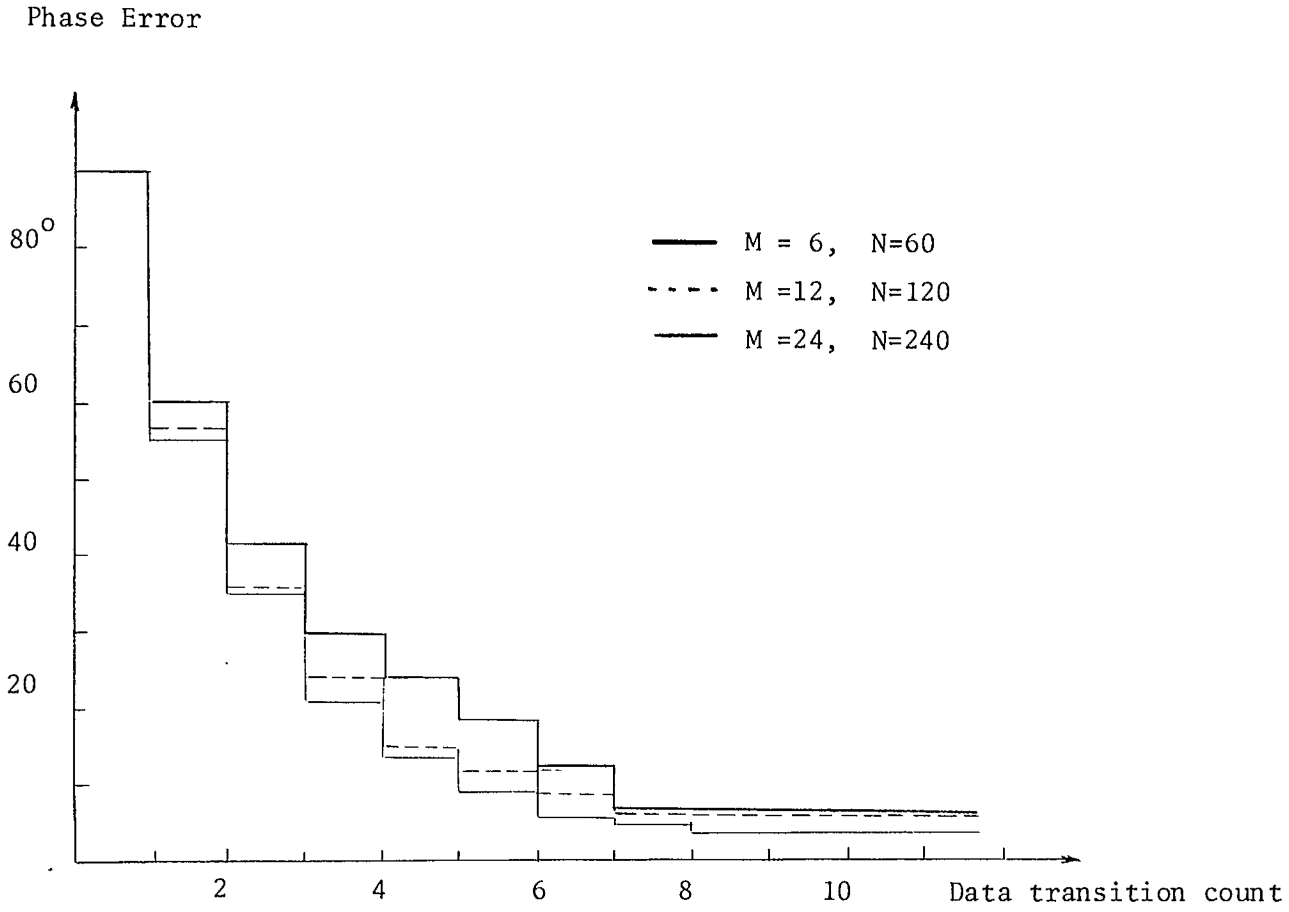


Fig. 3.19 - M & N are varied such that g is constant = 0.4

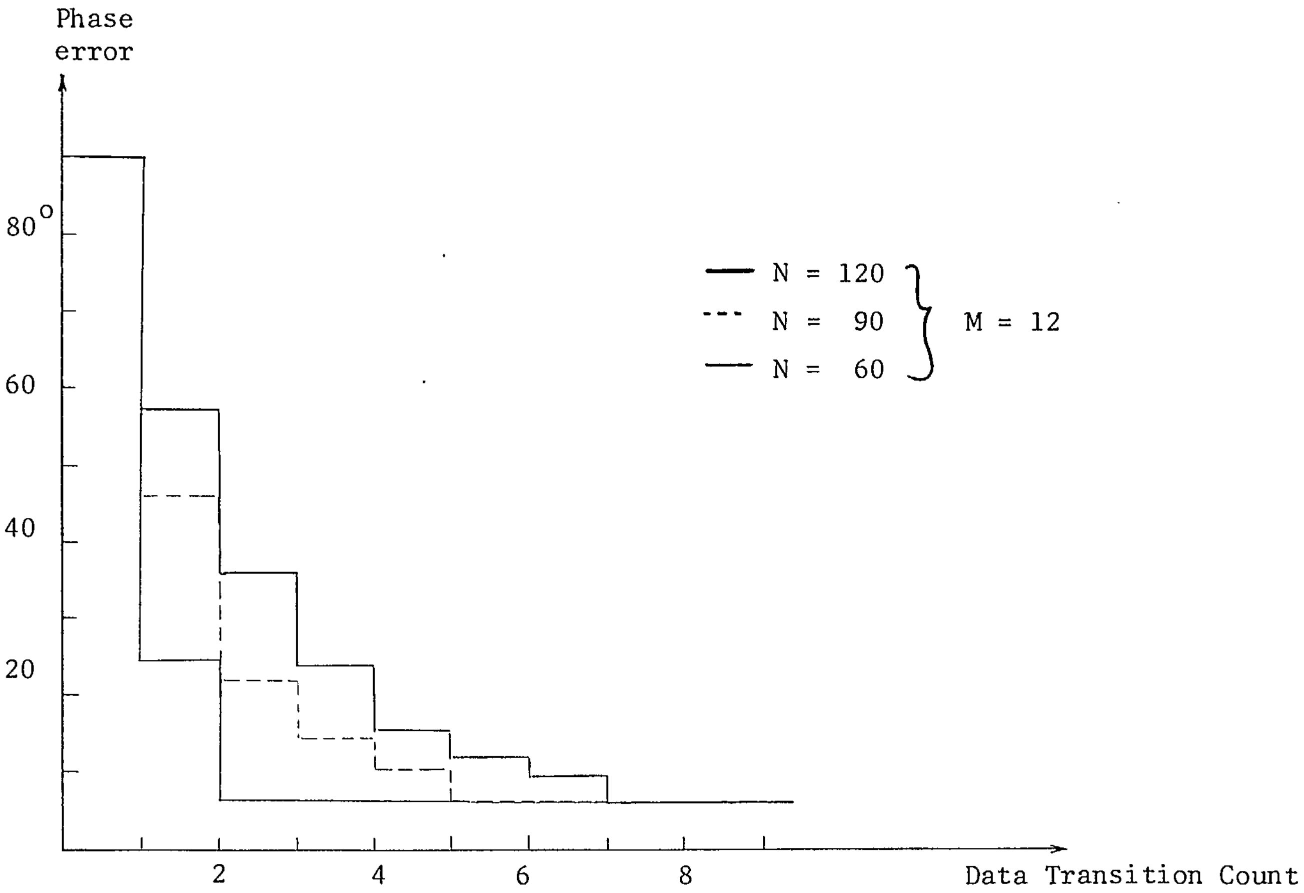


Fig. 3.20 Effect of step size on loop acquisition

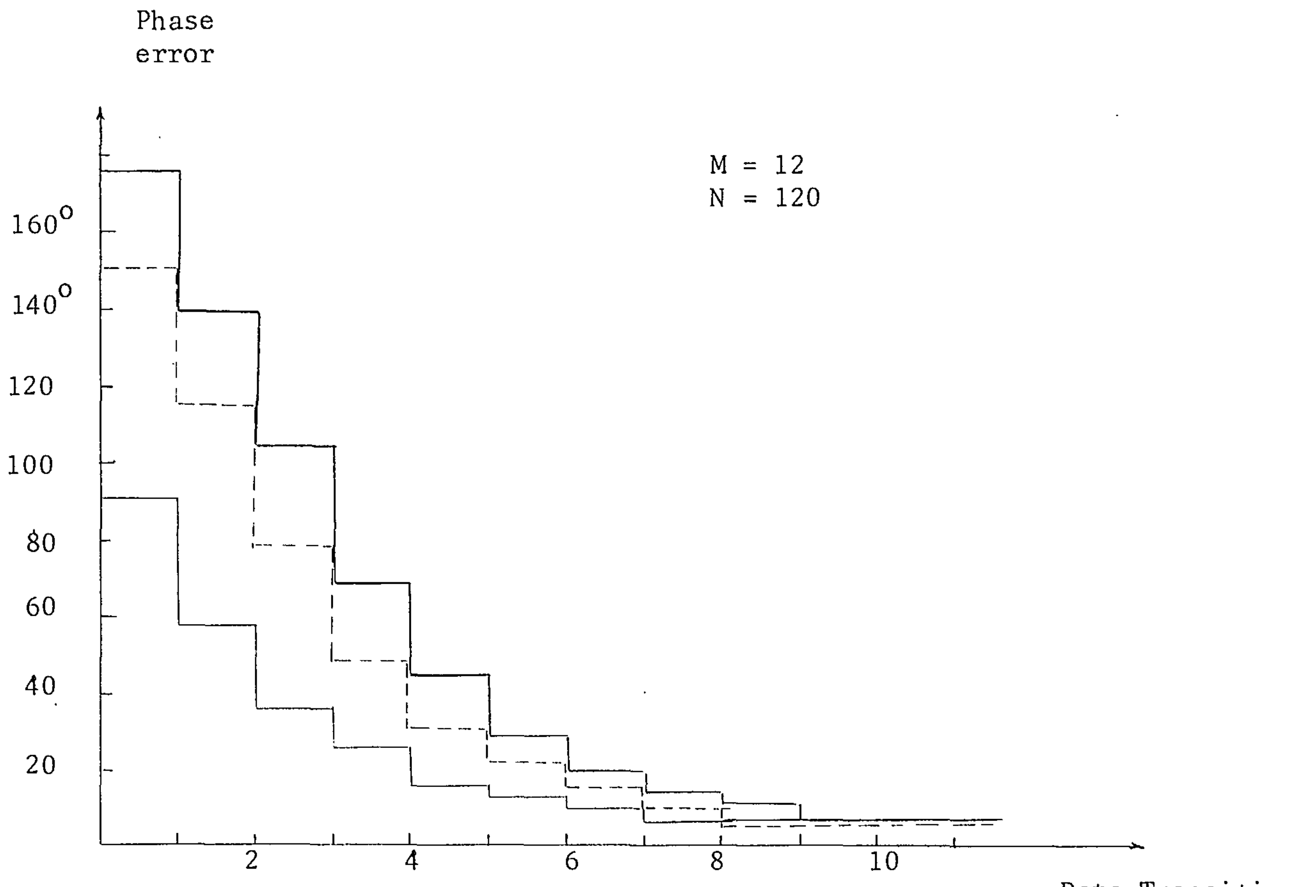


Fig. 3.21 Effect of Phase Step Size on Loop acquisition

Remarks:

from these results we note that:

1. for constant M/N (i.e. constant loop gain) the phase acquisiton time is almost the same.
2. for a given M, the loop acquisition time decreases by increasing the quantizer step at the expense of increasing the loop noise bandwidth.
3. for a given quantization step, the steady state phase uncertainty

(due to the existence of the loop dead zone) decreases by increasing M.

Loop response to a step in frequence

Some simulation technique has been used for the loop frequency acquisition study . The results shown in Figure 3.22 are for a periodic pattern 11001100 .... note that the loop steady state phase offset reaches a nonzero value proportional to the input frequency step size.

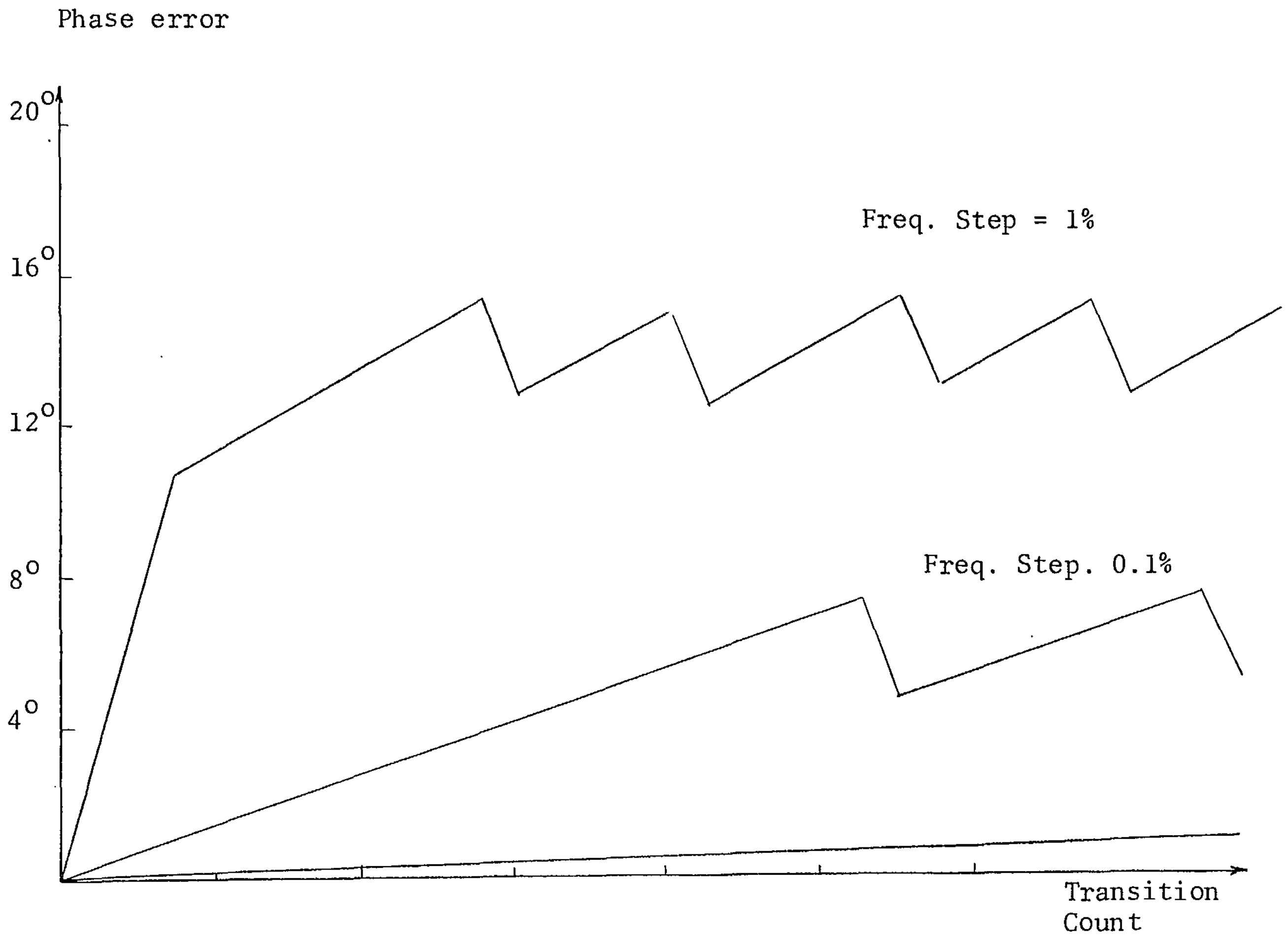


Fig. 3.22 - Loop Response to a frequency Step
Note the steady State phase offset and phase jitter due to the frequency offset.

Experimental Results

A loop has been implemented with M = number of quantization levels = 12, and $\Delta = 3^{\circ}$. To investigate the loop response to a step in phase, a random data generator was clocked externally using a signal having a frequency equal to the loop output clock frequency, but in phase quadrature with it. In the absence of Frequency offset it has been demonstrated that the loop acquisition time is expressible in terms of the number of data transitions required for the loop to achieve lock. Therefore there is no loss of generality if the input data were simply alternating between 1 and 0 periodically. Consequently a data pattern 001100 ... was chosen to test the loop response to a step in phase. Fig. 3.23 shows the loop phase error for initial phase step equal to $\pi/2$. The bottom trace is the output of an S-R Flip Flop which was set at the loop output clock rising edges, and reset at the data transitions. It is obvious that the resulting pulses width is equal to the loop timing error. The first pulse width is equal to T/2, and it keeps decreasing till it reaches a constant value within the loop discriminator dead zone. The number of data transitions passed till this steady state value is reached is the number required for the loop to reach lock. The top trace in Fig. 3.23 is the filtered version of the bottom trace.

To check the effect of frequency errors on the steady state loop performance, the Data generator clock frequency was offset from the loop output clock, and the loop phase error monitored on a scope. Fig. 3.24-3.26 show the loop steady state phase error for frequency steps equal to 0.35%, 0.273% and 0.195% respectively. Note that the mean value of the steady state phase offset.increase with the frequency step size, and the steady state phase "ripples" which also increase with the frequency step size.

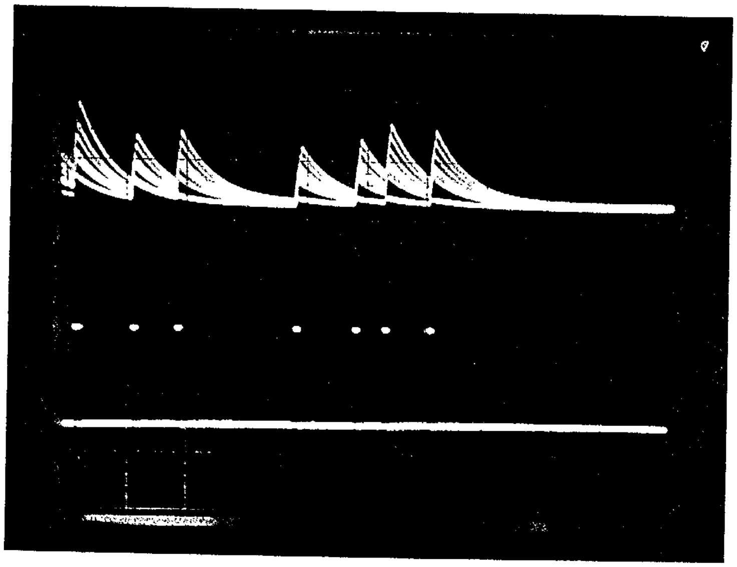


Fig. 3.26 Steady state phase offset due to a frequency step equal to 0.35% of the free running frequency.

3.4 References

[1] Wintz and Luecke "performance of optimum and suboptimum synchronizers" IEEE Trans. Comm. Tech. June 69, pp. 380-389

[2] Spilker "Digital Communications by satellite" 77, Prentice Hall, P. 433

[3] D.Le Ngoc & K. Feher" A digital approach to symbol timing recovery systems", IEEE Trans. Comm. Vol Com. 29, number 12, Dec. 1980, pp. 1993-1999.

[4] A. Blanchard "Phase Locked Loops" 1976, J.Wiley, p. 158

[5] Lindsay & Tausworthe "Digital Data transition tracking loop" JPL, SPS 37-50, Vol. III, Ap. 68

[6] Simon, M.K. "An analysis of the steady state noise performance of a digital-data-transition Tracking loop " 1969 Int. Comm. Conf. Record: 20-9-20-15.

[7] Hurd, W.S. and T.C. Anderson "Digital transition Tracking Symbol Synchronizer for low SNR Coded Systems" .IEEE Trans. Comm. Techn. Apr. 70: pp. 141-146.

[8] Simon. M.K. "Nonlinear Analysis of an Absolute Value Type of anEarly-Late-Gate Bit Synchronizer" IEEE Trans. Comm. Tech, Oct. 1970, pp. 589-596

[9] F. Gardner "Phaselock Techniques" 1979, J. Wiley & Sons, pp. 231-234

[10] J. Stiffler "Theory of Synchronous Communications", Prentice Hall, Englewood Cliffs, NJ, 1971, Chap. 7

[11] V. Mengali "A self bit synchronizer matched to the signal shape", IEEE Trans. AES - 7, pp. 686-693, July 71.

SECTION 4

INVESTIGATION OF SPEECH DETECTION

4.1. Introduction

Any scheme for interpolation of data packets in the silent periods of speech requires an efficient speech detector for discriminating silent and non-silent (active) periods of human speech. In the average conversation a speaker will be silent more than 50% of the time. Most of this silence will be during periods of listening to the other speaker, but there will also be a significant number of short silent periods in the form of pauses between words and phrases. It is important to note that in a given conversation, the durations of the silent periods and also the percentage of total time classified as silence, and therefore available for data transmission, depends on the speech detector's response to conversational speech on a background of noise. One can specify the activity factor of a speech detector as the percentage of total time it classifies its input as something other than background noise, under specified background noise conditions and averaged over many speakers and conversations.

The function of a speech detector is to classify its current input as either speech plus background noise or background noise alone. Clearly the discrimination problem becomes more difficult, and the activity factor has a tendency to increase, if the background noise level increases. In a mobile radio environment, the background noise level may be high. Moreover, mobile radio conversations may have rather different speech/silence statistics from ordinary telephone conversations. Thus attainable activity factors may be different from those typically found in telephone traffic.[1]

The design of a speech detector for a mobile telecommunications environment will be affected not only by the noise and speech characteristics, but also by speech quality and intelligibility requirements and by cost constraints. Subjective requirements have not yet been established for the mobile packetized speech/data system. However low cost is essential, and unlike speech detectors used in telephone TASI systems, time-shared processing of many trunks at one location is not possible. Thus a simple robust speech detector algorithm is required, which is suitable for LSI or VLSI realization. Because of the variability of the background noise level the speech detection algorithm must be able to adapt its sensitivity to the background noise level.

A number of adaptive speech detectors have appeared in the literature, most being intended for application in TASI or digital speech interpolation systems. They can be roughly classified into two main classes,those making heavy use of amplitude measurements to set a speech detection threshold above the noise level, and those relying heavily on waveform characteristics, such as zero crossing statistics and envelope variability, to discriminate speech and noise. References [2] and [3] are typical of the first class, and references [4] - [7] are typical of the second class.

To minimize sensitivity of the speech detector's performance to background noise waveform statistics, we have elected to consider a speech detector design in the first class, based largely on waveform amplitude measurements. However to classify a signal as background noise to obtain a measure of its level, our proposed speech detector forms a "constancy" measure similar to one used in reference [6] in the second class. In reference [2], Jankowski describes a speech detector which adaptively adjusts its threshold to just above the 96% - percentile point of low-level signals (presumed to be noise). LaMarche et al in reference [3] describe a detector which performs a weighted average of the outcomes of several tests, one of which tends to classify signals with small envelope variation ("constancy") as noise, as in reference [6]. The speech detector proposed here uses a "constancy" test to help identify and measure the level of segments of background noise, so as to adjust the threshold for discriminating speech and noise.

4.2 Basis for the Speech Detector Algorithm

The speech detector and its variations which we propose and evaluate is based on several premises regarding speech and noise characteristics in the mobile radio environment:

(1) Speech is non-stationary and has a large dynamic range; its amplitude probability distribution approximates a Gamma or Laplace distribution.

(2) Although there is inadequate statistical characterization of typical background noise in a mobile radio environment, it is assumed to be almost stationary with zero-mean and to be approximately gaussian - distributed. Its dynamic range is smaller than that of speech. Any time variation in its level (variance) is assumed to be slow (changing only over periods of several seconds or more). There will of course also be impulse noise bursts, which if of sufficient energy and duration, will unavoidably be classified as speech.

(3) In a mobile radio application it will be assumed that no echo of the far-end talker will be present at the speech detector's input. Thus echo detection and suppression would not be necessary, as it is in TASI speech detectors.

(4) The level of the background noise may be sufficiently high, and the similarity or difference of its waveform from some speech waveforms is sufficiently uncertain that waveform - specific methods such as zero-crossing measurements are deemed unreliable. Accordingly we focus mainly on measurements involving amplitude levels and variability of amplitude levels.

(5) The input signal is assumed to be presented to the speech detector in the form of PCM-coded (linear or companded) samples at a sampling rate of say 6.6 kHz or 8 kHz. This allows the speech detector to be developed independent of any particular coding algorithm, such as delta modulation, which could be used to digitize speech for transmission.

(6) In the interest of simplicity the speech detection algorithm should not require multi-bit multiplication; multiplying coefficients should be restricted to be powers of 2.

These premises lead us to envision the temporal variation over several seconds of the envelope of conversational speech plus background noise as shown in Figure 4.1.Probable "silent" periods, in which the signal consists only of background noise, are shown as shaded areas. During these periods, the envelope level is at a minimum and relatively constant. Of course, low-pass filtering of the signal magnitude is implied in Figure 4.1, to define a relatively constant envelope measure.

The speech detector we propose and investigate here is based on the simple idea of identifying periods of the signal which are "almost surely noise", estimating the noise level in such periods, and adjusting the speech detector threshold accordingly, just above say the 95th percentile point of the noise distribution function. Periods of "almost surely noise"(ASN) would be those judged to be at a minimum and fairly constant signal level over say several hundred milliseconds. Such periods would only be used for estimating the speech detector's threshold.

Figure 4.1 shows bursts of speech plus noise. some not much above the noise level, interspersed with "silent" periods of noise only. The statistics of the duration of active and silent periods of speech strongly depends on the speech detector which defines them. Measurements by Brady [1], in conditions of low background noise and fixed detector threshold, suggested active and silent period distributions which are both approximately exponentially distributed and with means of about 1 to 1.3 sec and about 1.8 sec. respectively. Depending on the threshold used, the measured activity factor varied from .35 to .44. Other investigators have found shorter average durations and different speech activity factors for different speech detectors,background noise conditions, and talker samples.

4.3 Description of Speech Detector

(a) detection and hangover

Figure 4.2(a) shows the major components of the speech detector. The speech detector declares its input to be speech if N consecutive sample magnitudes exceed its current threshold. Values of N from 1 to say 4 will be tried. Once this threshold criterion is satisfied, the detector remains locked in this "active speech declaration state" for at least the next H samples, where H, the hangover period, is chosen to bridge low level sample periods within speech bursts. The value of H chosen will be one or two hundred milliseconds. Figure 4.2(b) shows the detector's state transition diagram for a given threshold value and as input sample magnitudes $|x(n)|$.

(b) determination of current average amplitude level

A simple recursive low-pass filtering operation is performed on the sample magnitudes to estimate the current average level, as shown in Figure 4.3.The output p(n) is

$$\begin{aligned} p(n) &= \beta\, p(n-1) + (1-\beta)\ |x(n)| \\ &= (1-\beta) \sum_{k=0}^{\infty} \beta^k |x(n-k)|\,. \end{aligned}$$

where β is less than but close to 1. Assuming the noise samples are statistically independent, we are led to assume that p(n) is approximately gaussian-distributed, at least to within several standard deviations from its mean. It is argued in Appendix B that a value $\beta = 1-2^{-7}$ is reasonable, since then under the gaussian and independence assumptions, the standard deviation of p(n) will be only about 5% of its mean. Thus p(n) can be a fairly stable estimator of the current mean magnitude of the input signal, as averaged roughly over the past $\frac{1}{1-\beta} = 128$ sample intervals (16 msec. at a sampling rate of 8 khz).

The short term average level p(n) is used only to identify and determine the level of periods of background noise. Thus there isno need to allow p(n) to exceed the highest background noise level at which the system is still considered usable. A suitable maximum value p_{max} might correspond to a noise level about 20 dB below the long term average speech power. Then the updating of p(n) is modified as follows:

$$p(n) = \min \left\{ p_{max},\ \beta \cdot p(n-1) + (1-\beta)\,|x(n)| \right\}$$

Saturating p(n) in this way relaxes the digital storage requirements on p(n) and minimizes the effect of high-level speech on p(n).

(c) detection of "almost surely noise " (ASN) segments

The speech detector algorithm must process the short-term-average input amplitude p(n) to determine if it is minimal - valued and fairly constant like the shaded segments of Figure 4.1 and can thus be assumed to contain only background noise. Note that this processing is not the speech detector's ultimate discrimination between speech-plus-noise and noise alone, but instead is aimed at identifying ASN segments, so that the estimate of the background noise level and therefore the speech detector's threshold can be updated. In general the fraction of time identified as "almost surely noise" would be expected to be much less than the fraction of time ultimately identified as silence by the speech detector.

The ASN detector is looking for segments of p(n) that are:

(1) near or below its current estimated minimum;

and (2) nearly constant over several hundred msec; as typified by the shaded segments of Figure 4.1. To test for condition (1) the ASN detector must store the current estimated noise level, designated CNLE. Condition (1) is satisfied if p(n) dips below CNLE. If both conditions(1) and (2) are satisfied, the ASN flag is raised and CNLE is reset to p(n). To allow CNLE to increase(to slowly adapt to an increase in the background noise level) it must be allowed to increase slowly between resets. This can be accomplished by adding a small positive quantity δ to CNLE for every sample p(n) that exceeds the current CNLE.

i.e. if $\text{ASN} \neq 1$, $\text{CNLE}(n) = \text{CNLE}(n-1)+\delta$

The choice of δ is a compromise: too large a value may cause on undesirably large increase in the speech detector's threshold during a long talk spurt. Too small a value will slow the rate of adaption to increases in the background noise level. Appropriate values of δ and other techniques for allowing CNLE to increase will be investigated further.

In order that "almost surely noise" be declared, and CNLE and the speech detector's threshold be reset, condition (2) must also be satisfied simultaneously: p(n) must have undergone relatively little variation over the past several hundred milliseconds. The variation can be measured by filtering p(n) to produce a long term level estimate $\hat{p}(n)$ (averaged over say 256 msec.) The magnitude of the difference between p(n) and $\hat{p}(n)$ is averaged (filtered) again to yield the average deviation magnitude of p(n) from its long-term average. The resulting signal, multiplied by appropriate constant k, is subtracted from p(n). If the result d(n) is positive, the deviation is "small" and p(n) is judged to be "constant". A flag c is set to 1. Values of the parameters and filter time constants are to be established by computer simulation. Figure 4.4 shows the proposed test for constancy.

(d) resetting the speech detector threshold

A condition of ASN is identified whenever c=1 and p(n) $\leq$ CNLE(n). Then CNLE(n) is reset to p(n) and the speech detector's threshold CT is reset. When these two conditons do not both hold, the ASN flag is set to zero, and CNLE(n) is allowed to increase by δ , while CT remains fixed.

The quantity CNLE is the current noise level estimate. It can be shown see(Appendix A) that if the speech detector's input consists of independent samples with variance σ^2, the mean value of p(n), which is CNLE, is $\sqrt{\frac{2}{\pi}}\sigma$. Thus whenever the ASN flag is raised, the current speech detector threshold CT can be reset to say $3\sqrt{\frac{\pi}{2}}$ CNLE, which would be three times the estimated standard deviation of the background noise.

(e) <u>other aspects of the speech detector implementation</u>

To minimize complexity, all multiplying factors in the speech detector algorithm can be made powers of two, so that only simple shifting operations are necessary for multiplication. Further hardware simplification may be possible by sub-sampling outputs of the various low pass filters at rates of about twice their nominal bandwidths, rather than at the orginal high sampling rate of say 6.6 or 8 khz.

The digitized speech detector input may have DC or low frequency (<100hz) noise components which should be removed by inserting a simple digital high pass filter prior to the detector.

The speech detector will not be infallible, especially under the high background noise conditions of the mobile radio environment. The use of a hangover period H of 100 to 200 msec. should bridge very short gaps between speech bursts. The speech detector algorithm will tend to produce some clipping of the beginnings of low level utterances. This clipping can be reduced at the expense of a slightly increased activity factor and speech delay by applying the speech detector threshold criterion to speech that has been delayed by several milliseconds.

4.4 Simulation of the Proposed Speech Detector Algorithm

Investigation and demonstration of the speech detector is through computer simulation followed by construction with digital hardware. The computer simulation is performed on the PDP 11/55 system of the Department of Systems and Computer Engineering. In the later hardware phase the algorithm will be implemented in real time with a 8 khz sampling rate, using an INTEL 2920 programmable digital signal processing chip. This should allow considerable flexibility to test modifications of the algorithm in hardware.

The PDP 11/55 system used in the simulation phrase has a CPU under the RT-11 operating system. It has a memory management unit which extends the 32k instruction - addressing space to 124k. Peripheral units include two RK05 disk drives with one RK11 disk controller (DMA), a floppy disk unit and a terminal interface consisting of a DECWRITER, and a VR17 CRT monitor with VT11 graphics processor. There is also an AR11 unit for data acquisition, which can handle two independent D/A channels, 16 multiplexed 10 bit linear A/D channels, and a programmable clock. Finally, an associated audio section comprises on 8 track tape monitor, pre-amp, power amplifier and wiring panel.

The software effort has so far proven to be more extensive and time-consuming than orginally anticipated, due to limitations in existing memory and display software. In what follows and in appendix C we describe the orginal software approach and a modified version of it that is presently being developed. The new software simulation facility will allow detailed study of speech detection algorithms and also other signal processing algorithms for voice-frequency signals.

The speech samples used in the simulations are recordings of actual two-way mobile radio conversations. These will permit a preliminary investigation of speech detector performance in a fairly realistic environment.

(a) SDESSI and SDESSII

SDESS is an acronym for Speech Detector and Simulation Software The version I of this package -- SDESS I -- was written in Oct-Nov of 1980. SDESS I was soon found to be inadèquate in that it only processed speech segments of 2.3 seconds, a time too short for subjective judgement. A second version, SDEESS II is being developed which processes much longer speech segments and offers more versatile waveform display. Various alternatives were considered for implementation: Memory management to access all 92k physical memory; virtual memory system to swap speech samples to/from disks; waveform display on a storage scope via the AR11; and waveform display by programming the VT11 The second and last approaches were chosen and their implementation for SDESS II is being developed.

(b) Function of SDESS I

The primary objective of SDESS I was to process speech segments up to the CPU memory capability (about 20000 samples) and to replay the segments as seen by the speech detector for subjective evaluation.

(c) Features of SDEES I

(i) speech segments about 2.3 seconds long

(ii) sample at 8 khz

(iii) segments are stored on individual diskette files. File I/O is provided by FORTRAN.

(iv) Simultaneous (synchronized) replay of the processed and non-processed segments on two D/A channels in real time (8 khz).

The segments are repeated indefinitely until stopped by the user from the keyboard. This allows the two signals to be displayed on a dual-channel scope.

(v) Non-real time processing of the speech samples with parameters supplied by the user. The user can set break points at any sample locations to examine the internal "registers" of the speech detector and the speech detector "states" coded in the 6 highest-order bits of each sample.

(d) Technical overview of SDESS I

SDEES I is relatively simple in comparison with SDESS II for the following reasons:

(i) File I/O is supported by RT-11 FORTRAN

(ii) Since a speech segment is in memory in its entirety, file I/O is decoupled from any processing = D/A, A/D, simulations.

(iii) The system is not interupt-driven except for the reception of a key entry to stop the repetitiye D/A process, since one thing is done at any given time.

The size of SDEES I is 1.7k with about .3k written in MACRO and the rest in FORTRAN.

(e) Evaluation of SDESS I and Features of SDESS II

(i) Speech segments too short. Still, one is able to perceive the"clicks" caused by the speech detector switching on and off while processing some speech segments A/D ed from a tape recording of actual mobile dialogues.

(ii) With a speech segment repetitiously scanned on a scope, one can hardly correlate what is heard with what is actually seen. The new software package SDESS II eliminates these shortcomings of SDESS I.

The following aspects of SDESS I are retained in SDESS II:

(i) Sampling rate at 8 khz

(ii) Sample storage -- 10 bit A/D word and 6 bit "state" code

(iii) the manner by which the user is prompted for commands.

(iv) Except for the address of a "sample location", the specification of breakpoints and "windows" remain unchanged.

New features include:

(i) Extended speech segment to 15 seconds

(ii) In SDESS I, a speech segment resides in memory in its totality. In SDESS II, a speech segment resides in a "workspace" on the disk transparent to the user.

The user needs only load a file, which is also on the disk, into the workspace.

(iii) Disk files of sampled speech are created by double-buffering samples into the disk workspace. (the two RK05 disks are driven by a hardwired controller, the RK11 which will perform direct memory access (DMA) data transfer until transfer is completed, at which point the CPU is interrupted.)

(iv) Visual Audio effect: The user can specify a spurt of a speech segment, listen to it, and view it on the screen. The user can search for fine details within the spurt defined by translation and time scaling. The processed and non processed speech spurt are juxtaposed. Alternatively, the user can look at signals internal to the speech detector, e.g. the long term power of speech versus the the non-processed speech itself.

(v)"Mixing": The user can create a white noise file or for that matter any other signals, and add it to speech files.

(vi) Utilities: Utility commands are used to search for samples (possibly coded) that satisfy a stated condition and to compile statistics for speech segments.

Not available on SDEES II:

(i) A comprehensive file system: no diverting is maintained; the user must keep track of files.

(ii) (May yet be implemented) File transfer between disk and diskettes: The speech files are designed such that one file just fit one diskette. Due to a lack of memory and excessuvely slow transfer, this feature may not be implemented.

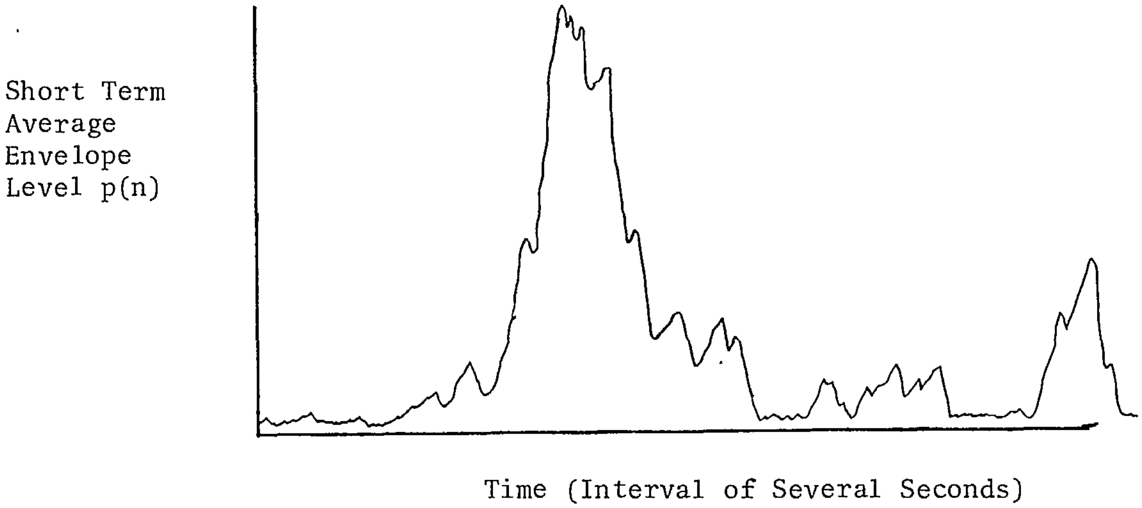


Fig4·1 Typical Temporal Variation of Signal Envelope

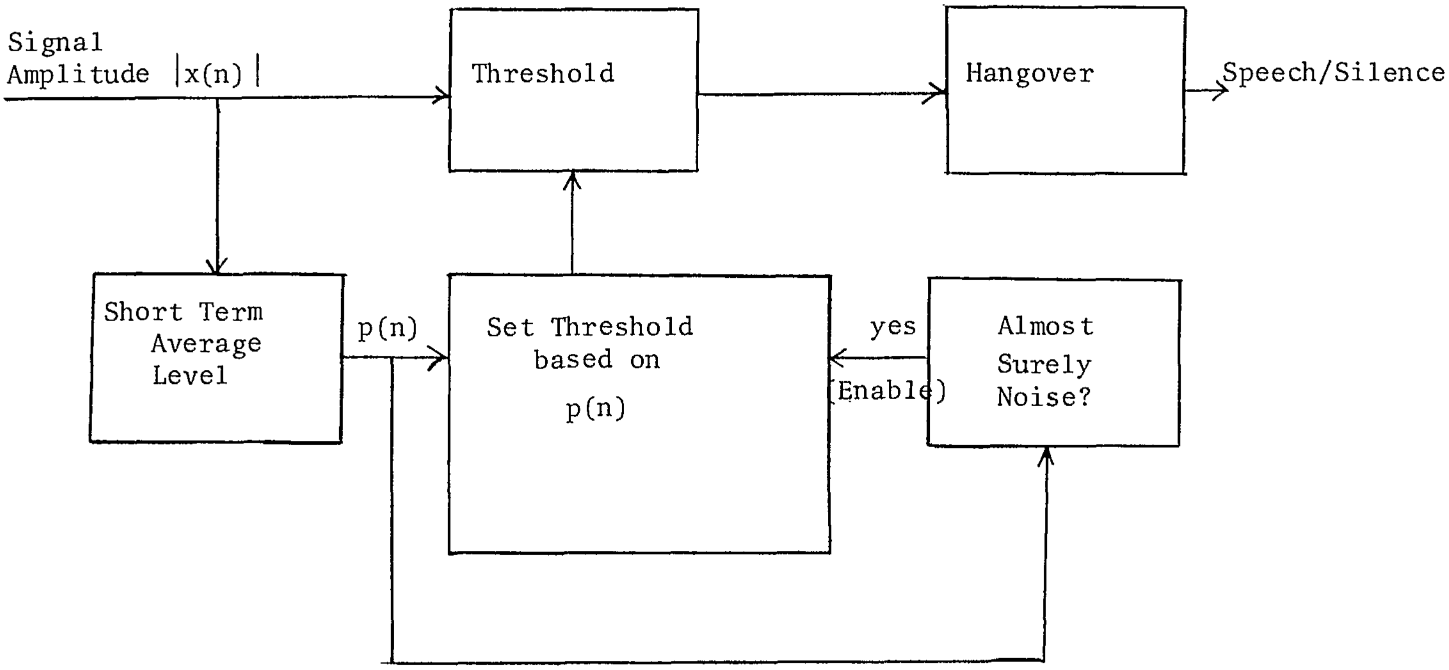


Fig. 4.2(a) Speech Detection Algorithm - Major Components

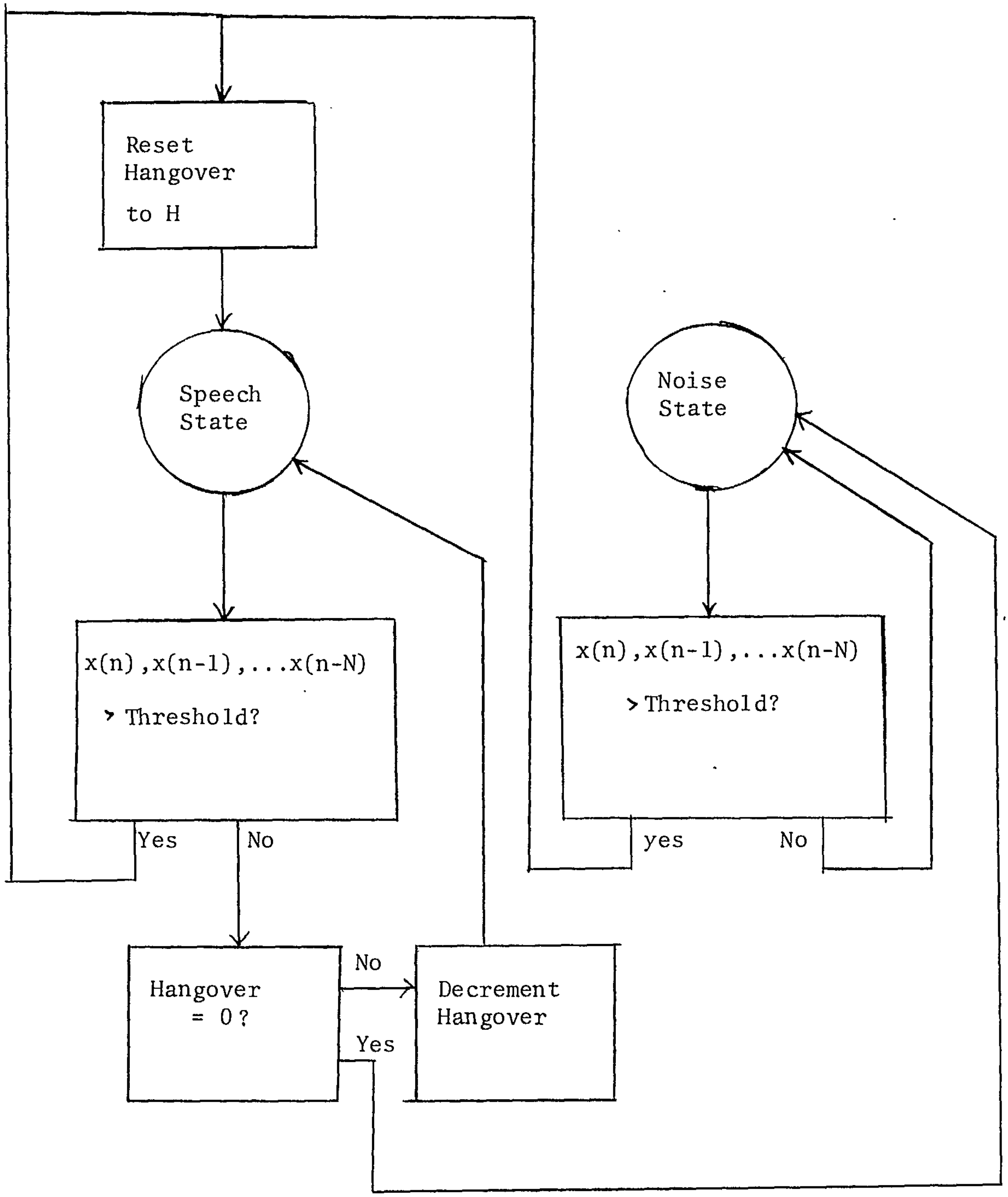


Fig. 4.2(b) Speech Detector Flow Chart

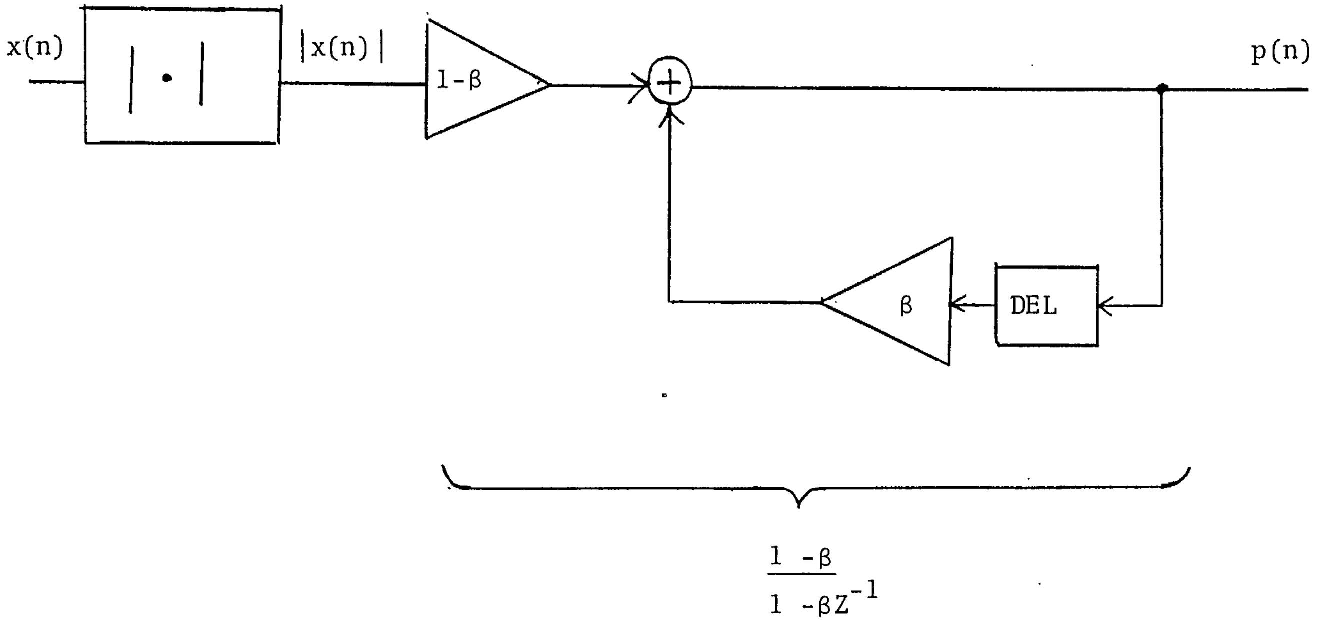
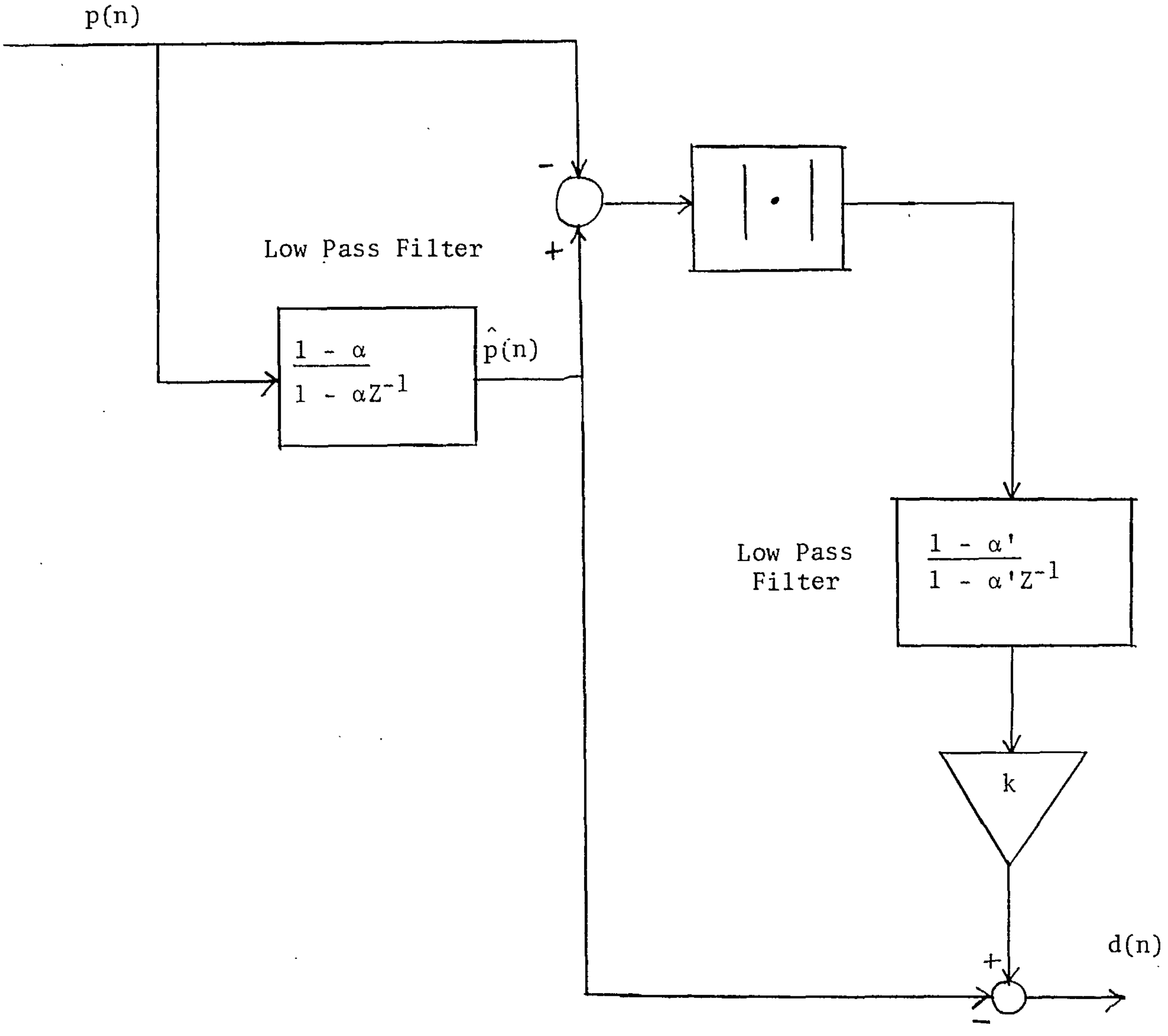


Fig.4·3 Determination of Short Term Average Level


p(n)
-
+
Low Pass Filter
$\frac{1-\alpha}{1-\alpha Z^{-1}}$
$\hat{p}(n)$
$|\cdot|$
Low Pass Filter
$\frac{1-\alpha'}{1-\alpha' Z^{-1}}$
k
+
-
d(n)

d(n)<0 : SET C = 1
d(n)⩾0 : SET C = 0

Fig. 4.4 Determination of Constancy

REFERENCES

[1] P.T. Brady, "A Statistical Analysis of On-Off Patterns in 16 Conversations", Bell Sys. Tech. J., January 1968, pp. 73-91

[2] J.A. Jankowski, Jr., "A New Digital Voice-Activated Switch", COMSAT Technical Review, Spring 1976, pp. 159-178

[3] R.E. LaMarche, C.J. May, Jr., & T.J. Zebo, "Digital Speech Detector", U.S. Patent # 4, 028, 496,, June 7, 1977.

[4] E. Fariello, "A Novel Digital Speech Detector for Improving Effective Satellite Capacity", IEEE Trans. on Communications, Feb. 1972, pp. 55-60.

[5] R.W. Schafer, K. Jackson, J.J. Dubnowski, and L.R. Rabiner, "Detecting the Presence of Speech Using ADPCM Coding", IEEE Trans. on Communications, May 1976, pp. 563-567.

[6] P.G. Drago, A.M. Molinari, and F.C. Vagliani, "Digital Dynamic Speech Detectors", IEEE Trans. on Communications, Jan. 1978, pp. 140-145.

[7] C.K. Un & H.H.Lee, "Voiced/Unvoiced/Silence Discrimination of Speech by Delta Modulation", IEEE Trans. ASSP, Aug. 1980, pp. 398-407.

SECTION 5

TRANSMITTER AND RECEIVER DESIGN

5.1 Transmitter and Receiver Structure

The mobile radio transmitter design shown in Fig. 5.1 provides 4 watt saturated output power at 840.455 MHz with a frequency stability of 5 x 10^{-5} allowing for a maximum frequency drift of 42 KHz. The transmitter structure is based on a double up-conversion design using bandpass filters to suppress undesired frequency components, rather than a single side band, single up-conversion design. This choice was made since greater carrier and lower-side-band suppression can be obtained for the same level of design cost. The double up-conversion process is necessary in order to avoid unrealistically narrow bandpass filter bandwidths which would be required in a single up-conversion process. Commercially available filters are used with percentage bandwidths of less than 1%. In addition the RF up-converter is driven by a crystal controlled oscillator running at the second subharmonic and pumping a frequency doubler. This design choice reduces the transmitter cost significantly. The mobile radio receiver structure, shown in Fig. 5.2, is based on a low noise front end amplifier design using a single down-conversion stage, rather than a double down-conversion stage using an IF amplifier. This choice was made due to the lower cost of the low noise amplifier at 840 MHz when compared with a second mixing stage involving an additional crystal controlled local oscillator, mixer, filter, and IF low noise amplifier. The down-converter is driven by a crystal controlled oscillator running at the second subharmonic and pumping a frequency doubler, as in the transmitter, in order to reduce cost.

The pertinent signal power levels, and frequencies at various locations in the transmitter and receiver are shown in Figs. 5.1 and 5.2. The signal spectrum plans located at points A, B, C, D and E in Figs. 5.1 and 5.2 are shown in Fig. 5.3.

A list of the components purchased for the transmitter and receiver design is given in Table 5.1. Each component was selected from typically four manufacturers, based primarily on low cost and fast delivery, and secondarily on best performance.

TABLE 5.1

System Components

Component	Model	Manufacturer	Description	Quantity
crystal controlled oscillator	Y-657G6J	Greenray Industries	30MHz, 10dBm	1
crystal controlled oscillator	Y-657G6K	Greenray Industries	405MHz, 13dBm	1
crystal controlled oscillator	Y-657G6K	Greenray Industries	420MHz, 13dBm	1
Frequency Doubler	MK-2	Minicircuits	1-500MHz input 13dB loss	2
Mixer	DMM-2-500	Merrumac		3
IF Amplifier	GAM-30-150	Merrimac	0.5-400MHz 30dB gain	1
HP Amplifier	LWA 510-4	Microwave Power Devices	4W output	1
LN Amplifier	AK-1000	Avantek	2.5dB noise figure	1
Band Pass Filter	3/CH/30.5/.2/KL	Texscan	30.5MHz± .1MHz	1
Band Pass Filter	3/CR/840.5/2./KL	Texscan	840.5MHz± 1MHz	2

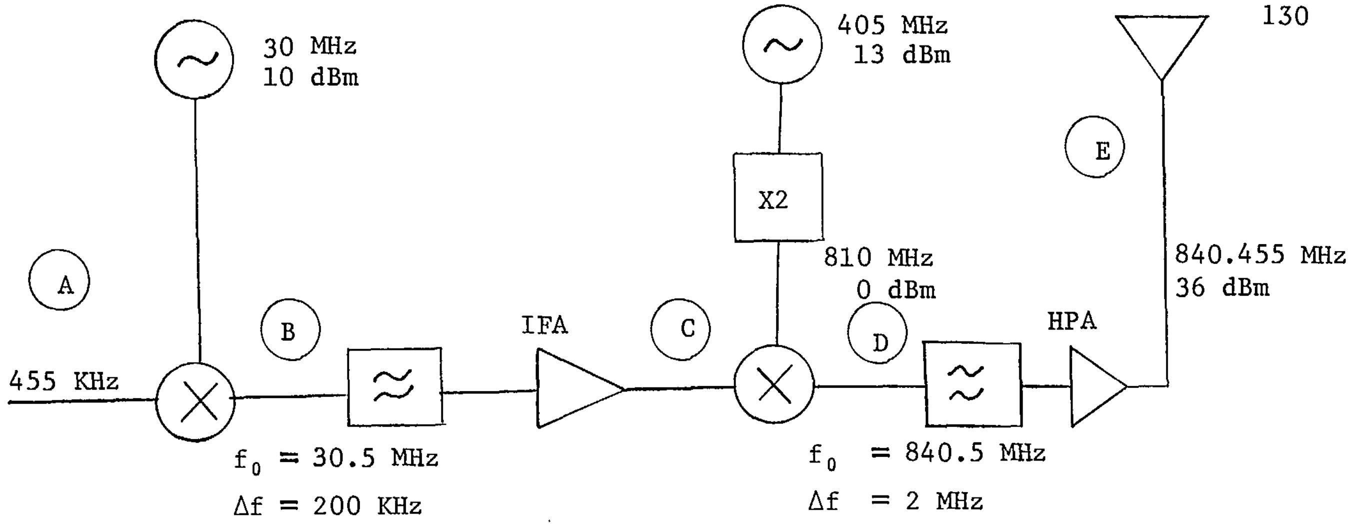


FIG. 5.1 TRANSMITTER STRUCTURE

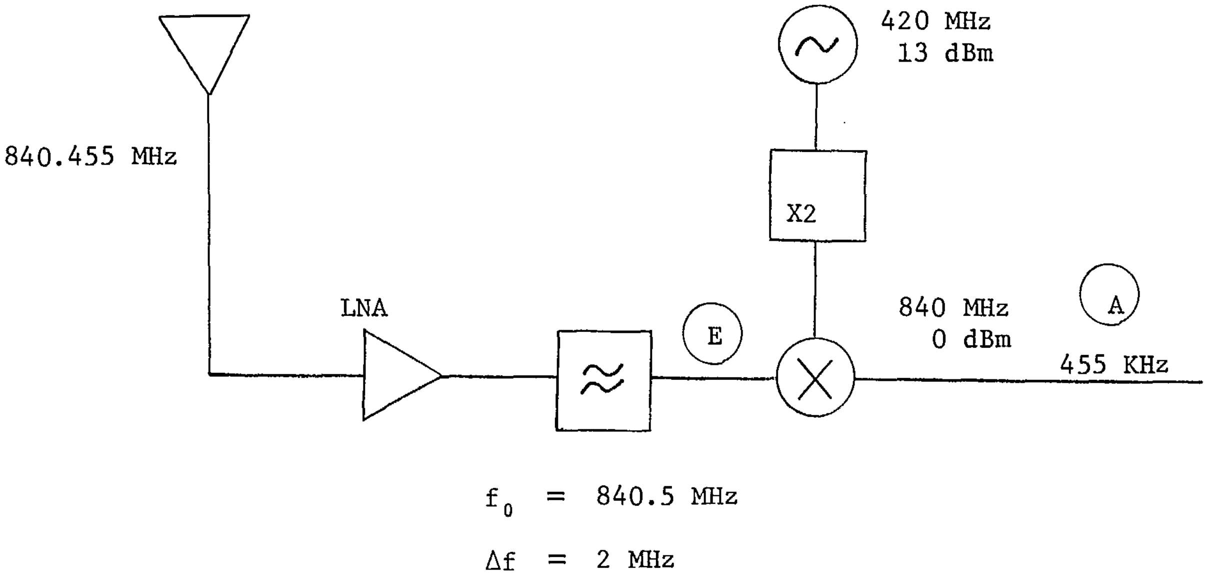


FIG. 5.2 RECEIVER STRUCTURE

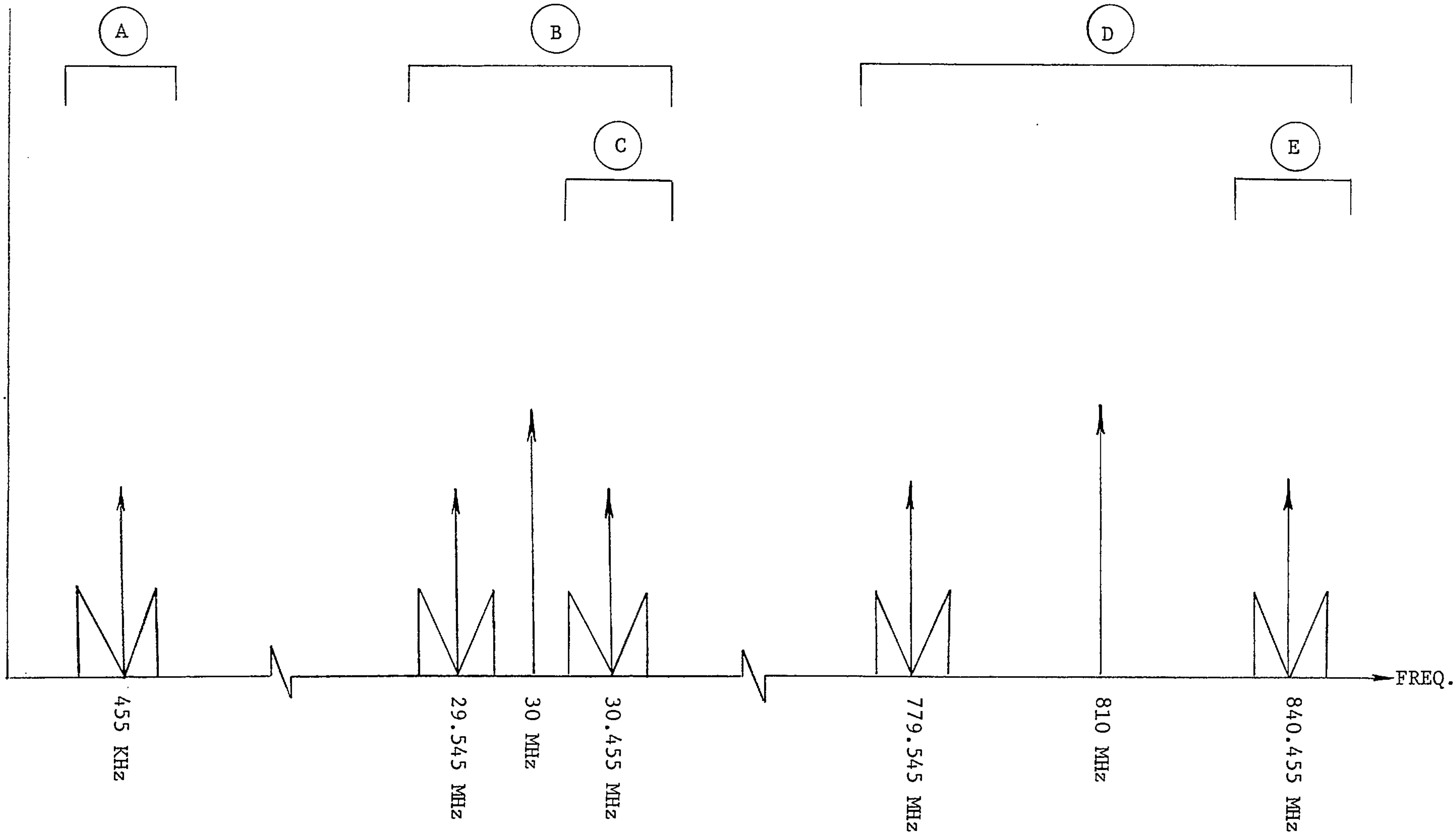


FIG. 5.3 SIGNAL SPECTRUM PLAN

5.2 Base Station and Mobile Unit Antenna Designs

The simplest antenna designs which yield omnidirectional patterns in the horizontal plane are the half wavelength dipole and the quarter wavelength monopole. The dipole is useful in situations where a flat metallic "ground" is not available, such as at a base station; while the monopole is useful where a flat metallic ground is available, such as on the roof of a mobile vehicle. These antennas form the starting point in the design of the base station and Mobile Unit Antennas.

Base Station Antenna

The two major limitations of the half wavelength dipole for use as a base station antenna are its balanced transmission line feed requirement, and its broad vertical plane pattern. The balanced feed requirement is in apparent conflict with the use of coaxial lines which are unbalanced in nature. However this difficulty can be surmounted by use of the skirt dipole shown in Fig. 5.4. Here the current on the centre conductor feeds one monopole as in a conventional dipole feed network, while the current on the inside of the outer conductor feeds the skirt. The current flowing down the skirt encounters a transmission line open circuit at the skirt bottom. This open circuit is created by the coaxial line formed by the skirt and the outer conductor of the feed line, and occurs since the wavelength in an air dielectric coaxial geometry is identical to that of freespace. Since the currents flowing on the monopole and the skirt are in phase and are centre fed, and since an open circuit is encountered at the top of the monopole and the bottom of the skirt, a standing wave pattern is set up on the Skirt Dipole identical to that which appears on a balanced fed dipole.

The second limitation of the half wavelength dipole, that of its broad vertical plane pattern can be overcome by stacking an array of Skirt dipoles as shown in Fig. 5.5. The narrowing of the vertical plane pattern results in an increased antenna gain with a consequent savings in required transmitter power. Here the lower skirt of the bottom diple is excited as before by the current flowing on the inside of the outer conductor of the coaxial feed. The upper skirt of the bottom dipole is not however excited directly by the centre conductor of the coaxial feed. Instead the current flowing along the centre conductor induces current to flow on the inside of the outer conductor and hence on the top skirt of the bottom dipole. This induced current is of equal amplitude as the current on the centre conductor, and flows in the opposite direction in accordance with the characteristics of all transmission lines. This process is repeated to induce current on both skirts of the centre dipole and on the bottom skirt of the top dipole.

Due to attenuation of the current on the centre conductor resulting from energy radiated by successive dipoles, a practical limit of three skirt dipoles is found. By the use of a coaxial feed line with Teflon as the dielectric, the ratio of free space wavelength λ_o to guide wavelength λg is given as:

$$\frac{\lambda g}{\lambda_o} = 0.694$$

This allows a collinear spacing between dipoles of $.194\lambda_o$ with a favourable mutual impedance of

$Z_{ij} = 6-j7.5$ and a self impedance of

$$Z_{ii} = 73 + j44$$

The resulting antenna gain for three stacked skirt dipoles is 6.4 dB, quite respectable for a horizontally isotropic radiator.

The transmission line equivalent circuit of the three stacked skirt dipoles is shown in Fig. 5.6. This circuit yields an input impedance of

$$Z_{in} = 243 + j\ 102$$

In order to maximize the radiated power and in order not to exceed the output SWR rating of the High Power Amplifier, this input impedance must be matched to 50 Ω. This impedance can be brought sufficiently close to 50 Ω using the 4:1 impedance transformation network shown in Fig. 5.7. It should be noted that this transformer is inherently broad band and hence does not require tuning, and also uses only 50 Ω transmission lines, which are readily available. The coaxial embodiment of this transformer along with that of the complete base station antenna is shown in Fig. 5.8. It should be noted that the antenna provides lightning protection for the base station electronics since the antenna is dc grounded.

Mobile Unit Antenna

The major limitation of the quarter wavelength monopole above a ground plane for use as a mobile unit antenna is its broad vertical plane pattern. This pattern can be narrowed in a similar manner as used for the base station by simulating a collinear array of three dipoles. This array is established by use of a sleeve monopole above ground as shown in Fig. 5.9. The input impedance for this antenna is estimated as:

$$Z_{in} = 42-j130\ \Omega$$

The uncertainty in this impedance due to the presence of a finite ground plane of $3.5\lambda_o$ (a conservative estimate for the size of the roof of a mobile vehicle) is given as:

$$\Delta Z < \frac{30\ \lambda_o}{\Pi\ d} = 2.7\Omega$$

In order to match the above input impedance to 50 Ω a reactance of +j130 Ω can be placed in series with the antenna input. This reactance can be formed by constructing a short circuited 50 Ω coaxial line in series with the sleeve antenna terminals as shown in Fig. 5.10. The length of the series stub required to present j130Ω is 0.441 guide wavelengths.

The antenna gain for the sleeve antenna is estimated as 5 dB.

A possible improvement on the sleeve antenna which affords lightning protection to the mobile unit electronics is shown in Fig. 5.11 However, due to the difficulty in estimating the value of the input impedance, this design has not been considered.

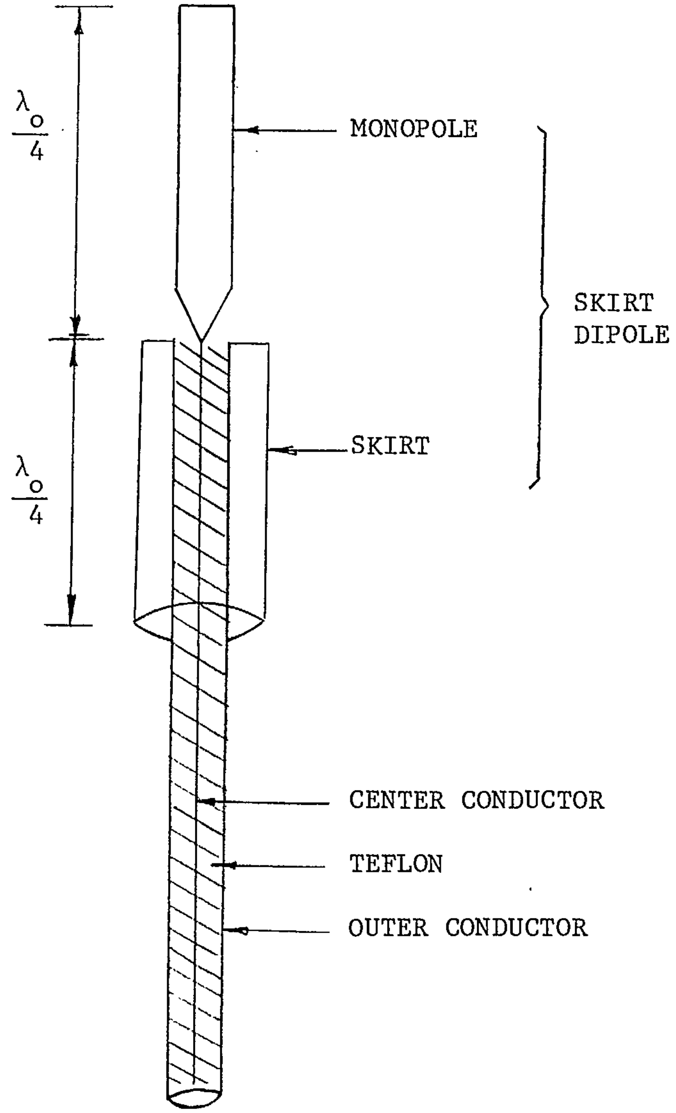


FIG. 5.4 SKIRT DIPOLE ANTENNA

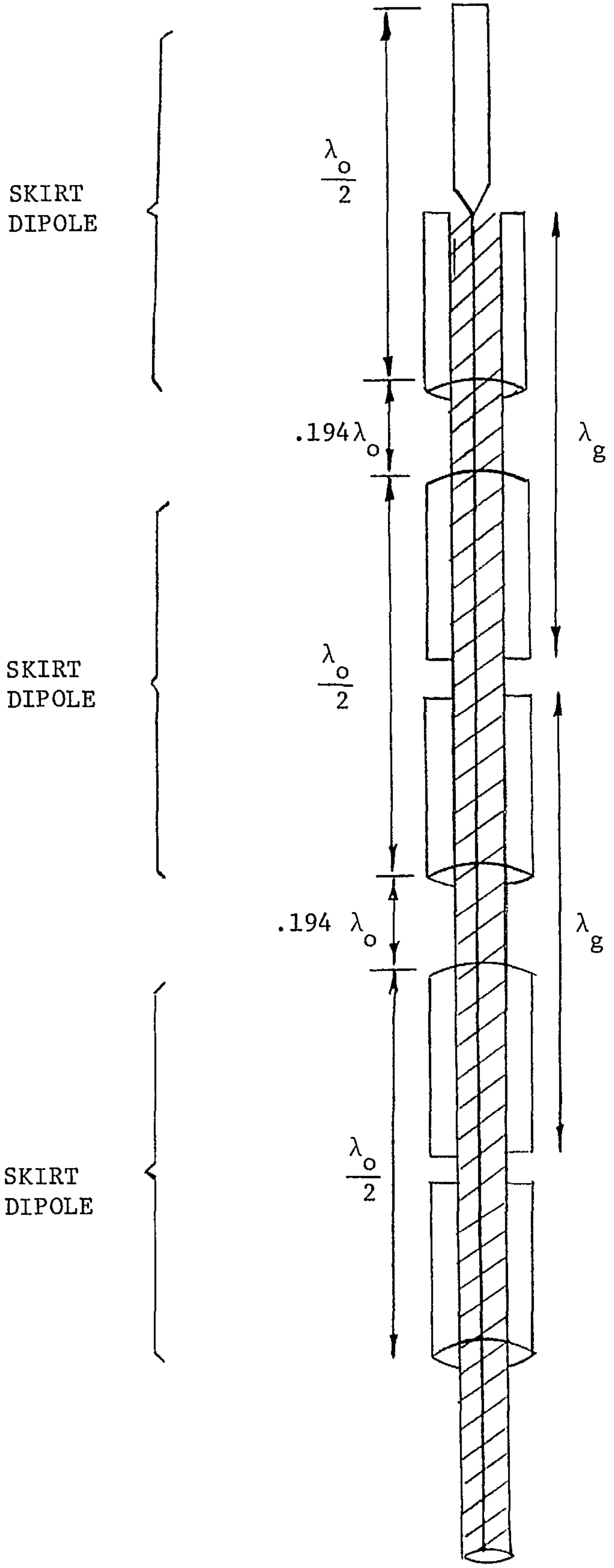


FIG. 5.5 SKIRT DIPOLE ARRAY

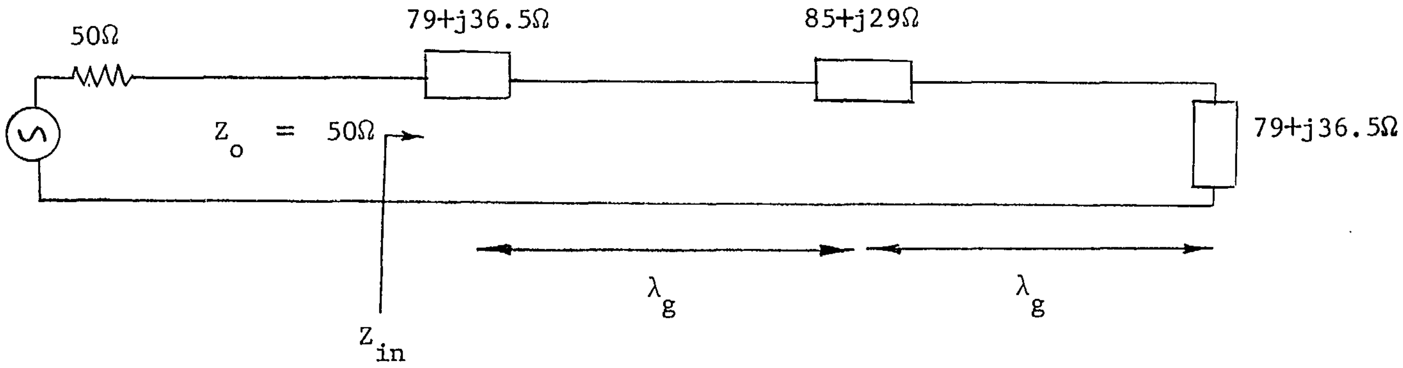


FIG. 5.6 TRANSMISSION LINE EQUIVALENT CIRCUIT

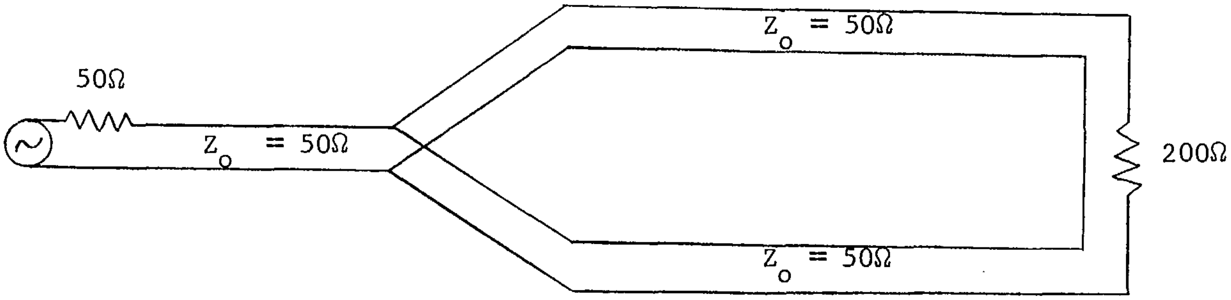


FIG. 5.7 4:1 IMPEDANCE TRANSFORMATION NETWORK

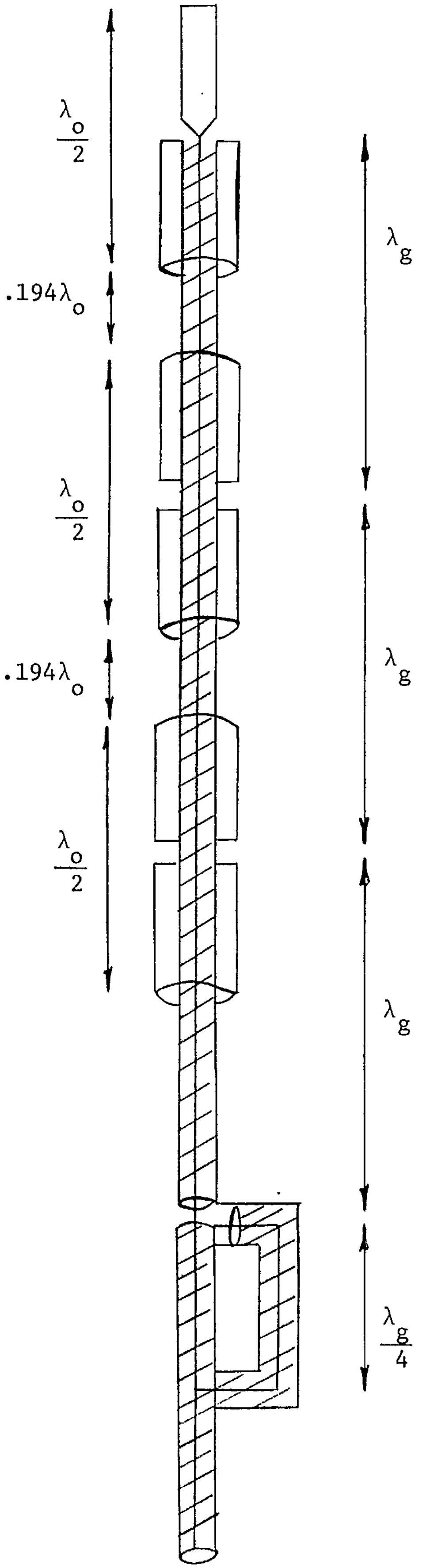


FIG. 5.8 BASE STATION ANTENNA

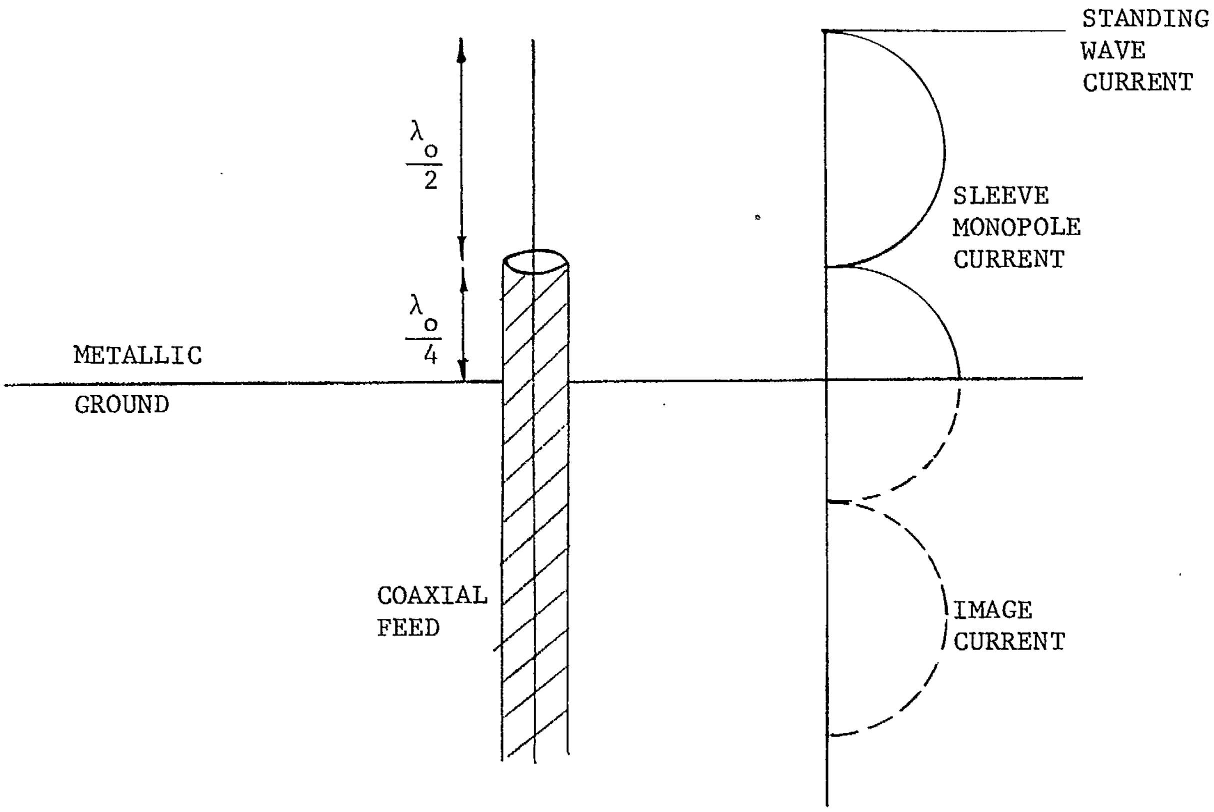


FIG. 5.9 SLEEVE MONOPOLE

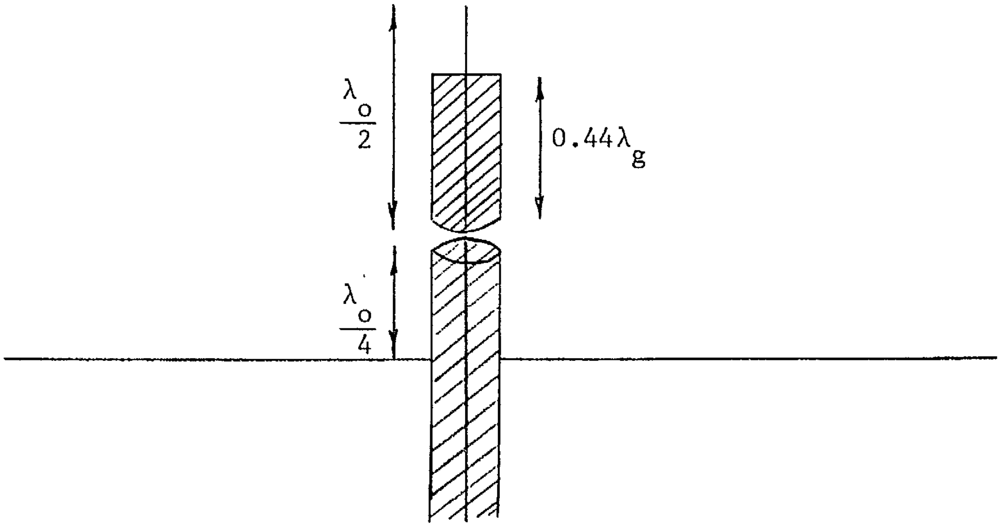


FIG. 5.10 SLEEVE MONOPOLE WITH SERIES STUB

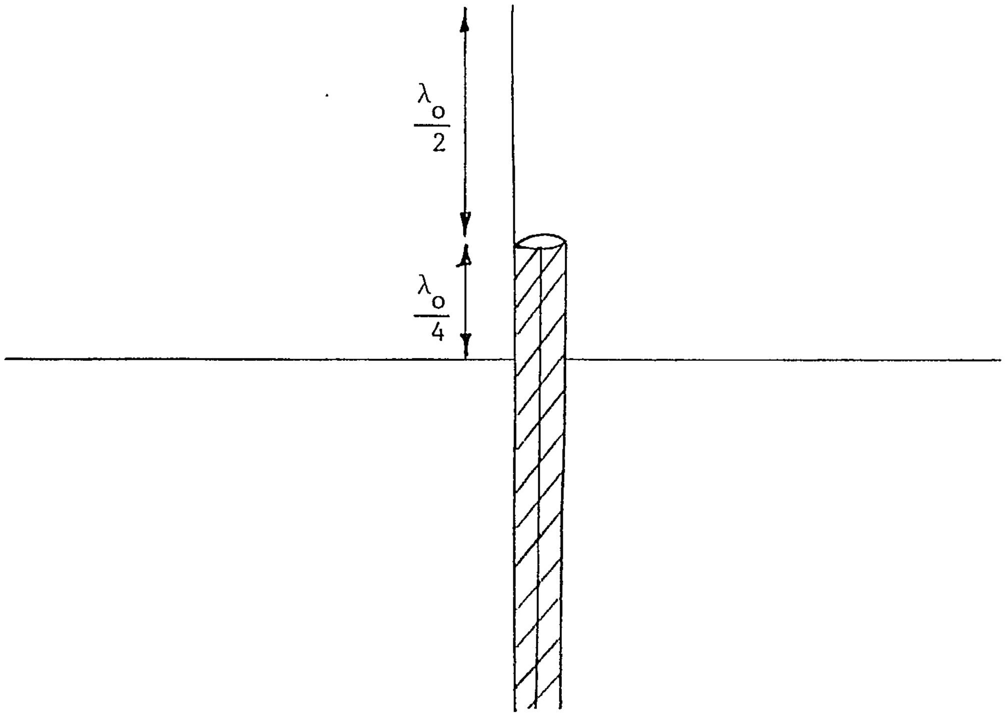


FIG. 5.11 SLEEVE MONOPOLE WITH LIGHTNING PROTECTION

SUMMARY 6

SUMMARY AND FUTURE RESEARCH

6.1 Summary

The major findings of this phase of research can be summarized as follows:

1. The integration of voice and data in a carrier sense multiple access scheme requires a spectrum efficient modem with fast clock and carrier recovery circuits and a low cost, efficient voice detection circuit.

2. For mobile radio applications, TFM appears strongly to be the optimum technique due to its constant envelope property, small out of band radiation characteristics and spectrum efficiency. This modulation technique has been demonstrated to be capable of transmitting at a rate of 16 k bits/sec over 25 KHz channels with the spectrum at 67 dB down near the edge of the adjacent channel. TFM has a slight degradation in its BER performance compared with FFSK or PSK.

3. A carrier recovery circuit has been designed and implemented based on a modified Costas loop structure. The loop has two bandwidths: (1) a narrow bandwidth for the tracking mode to minimize the carrier phase jitter (75 Hz), and (2) a large bandwidth for the acquisition mode (400 Hz). The transient response for the loop was found to range from 4 m.seconds at no noise to 8 m.seconds for carrier to noise ratio of 10 dB.

4. It was decided that the Absolute-value Early-Late-Gate clock synchronizer was best suited for our application because of its noise immunity and the ease of of implementing it in an all digital circuit. A new loop discriminator was implemented to speed up the acquisition time of this method. The circuit was shown to require 20 data transitions to achieve lock for a 180° phase step.

5. An algorithm for speech detection was implemented using the PDP11/45 minicomputer. The algorithm runs in real time for recorded speech which is digitized and stored on the processor's disk. The complete test bed has been prepared so that different versions of the algorithm can be tested and evaluated.

6.2 Future Research

It is recommended that this research be continued along the following lines.

(1) Constructions of the demodulator section of the TFM system so that BER measurements can be obtained both for a simulated channel at baseband frequency as well as in line-of-site.

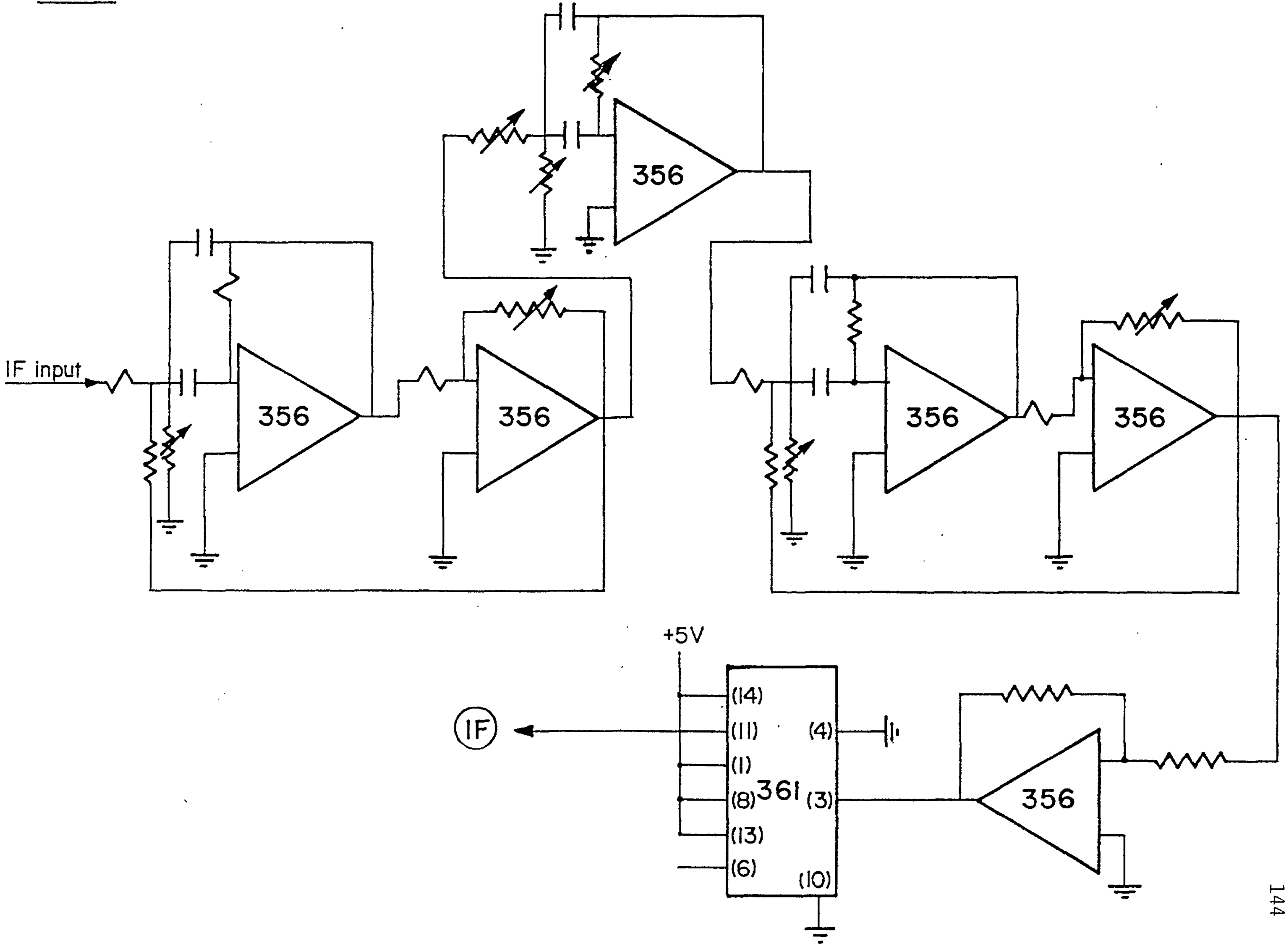
(2) Implementation of the speech detector algorithm using the recently available signal microprocessors (Intel 2920 and Nippon microprocessor).

Appendix A

Carrier and Clock Recovery Circuit Components

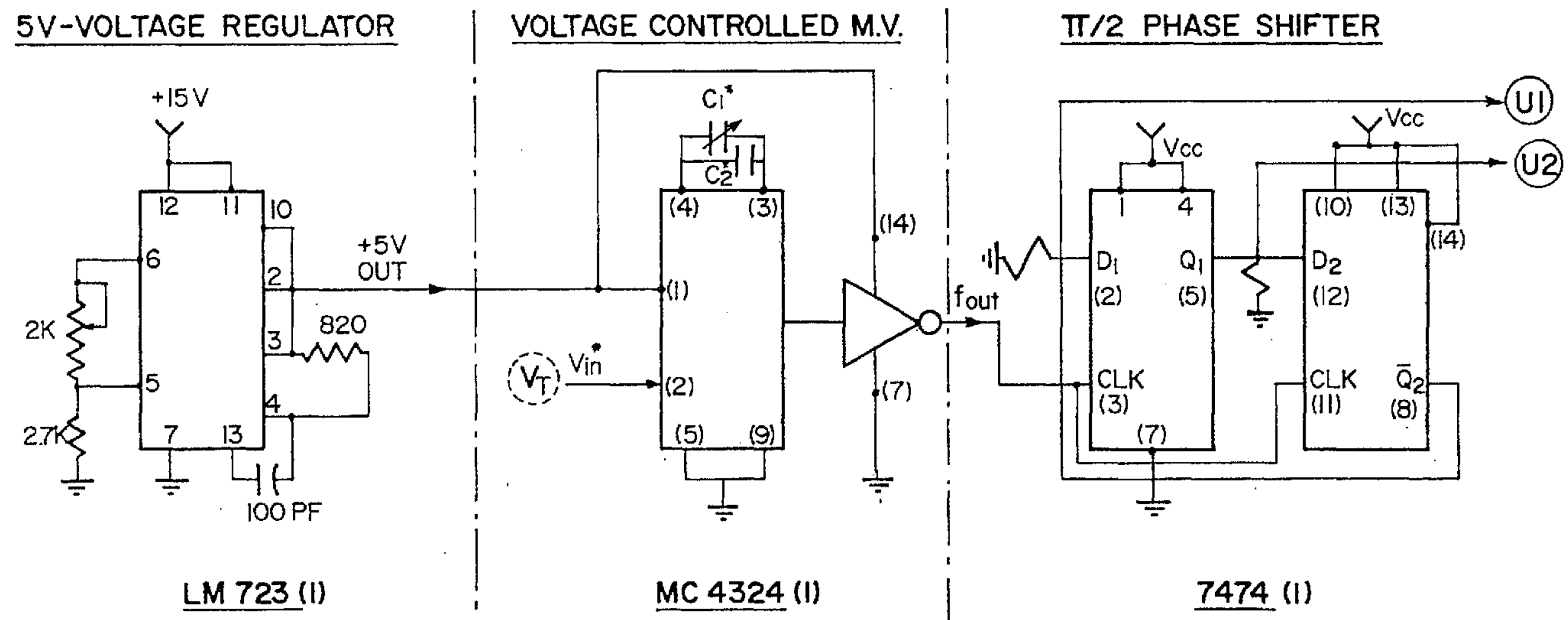
IF FILTER & HARD LIMITER

Input: IF modulated signal
Output: IF (2&3)



INPUT: the VCO control voltage ≡ output from summing amplifier (7)

OUTPUT: carrier (u1) and quadrature carrier (u2) → (2 & 3)



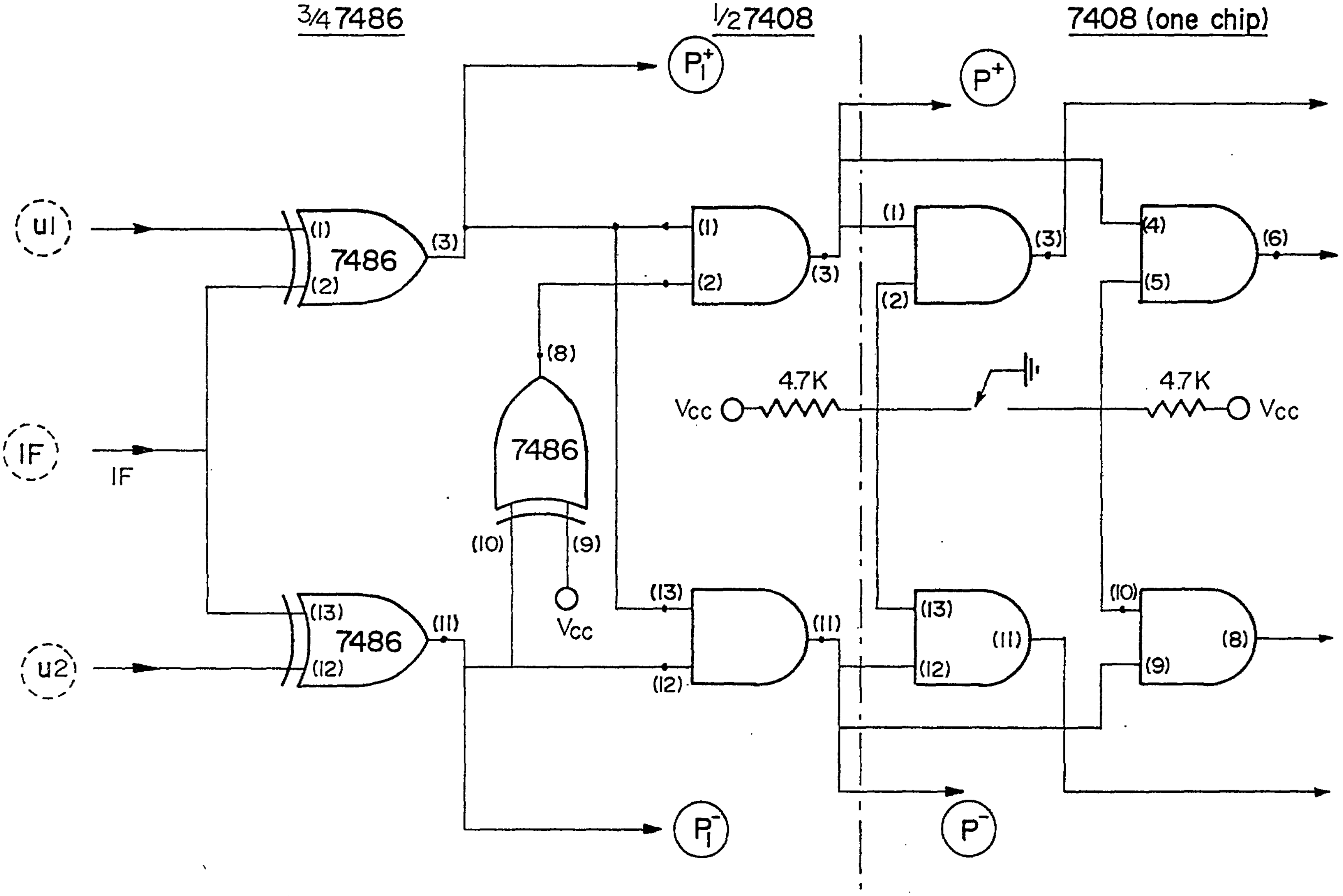
* 1- C_1 - 30 PF Trimming Capacitor

2- C_2 * 47 PF Ceramic Capacitor

3- V_{in} ≡ VCO Control voltage

PHASE DETECTORS

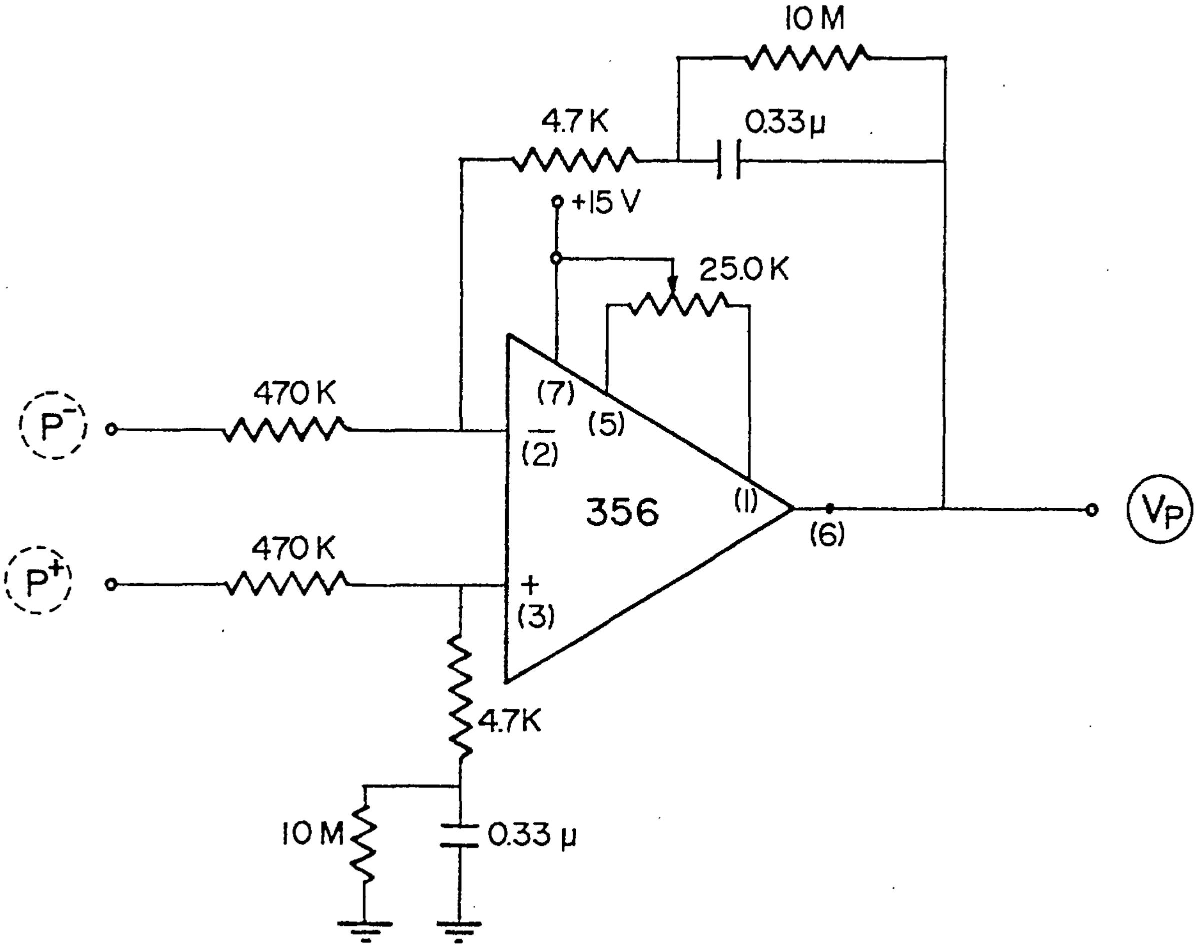
INPUT: u1 & u2 (1) & IF (0)
OUTPUT: 1· P_1^+ & P_1^- (9)



COSTAS LOOP FILTER

INPUT: P^- & P^+ (2)

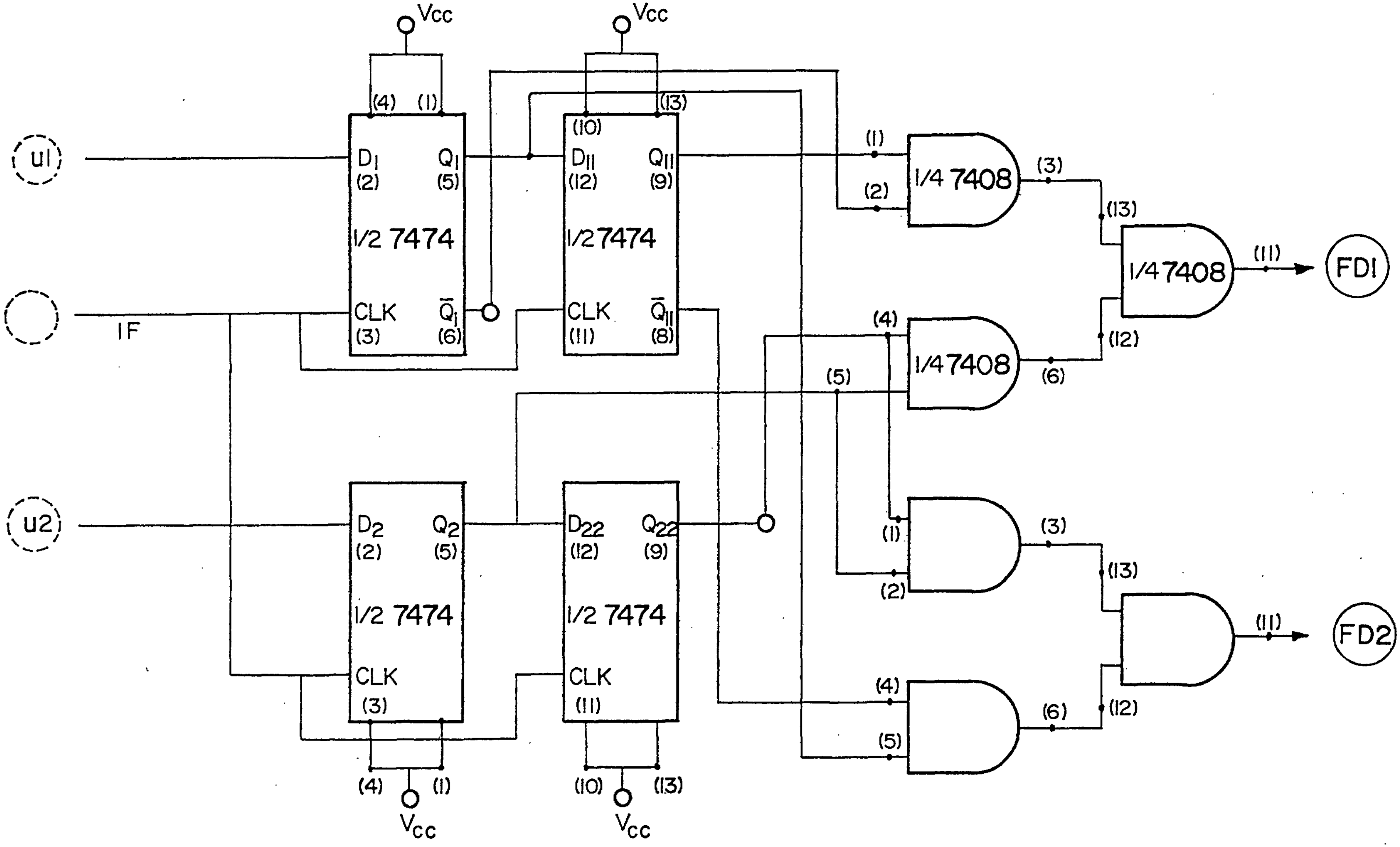
OUTPUT: V_P (1)



FREQUENCY COMPARATOR

INPUT: 1. u1 & u2 (P· 1) 2. IF Signal (0)
OUTPUT: FD1 & FD2 (4)

PARTS
7474 x 2
7408 x 2

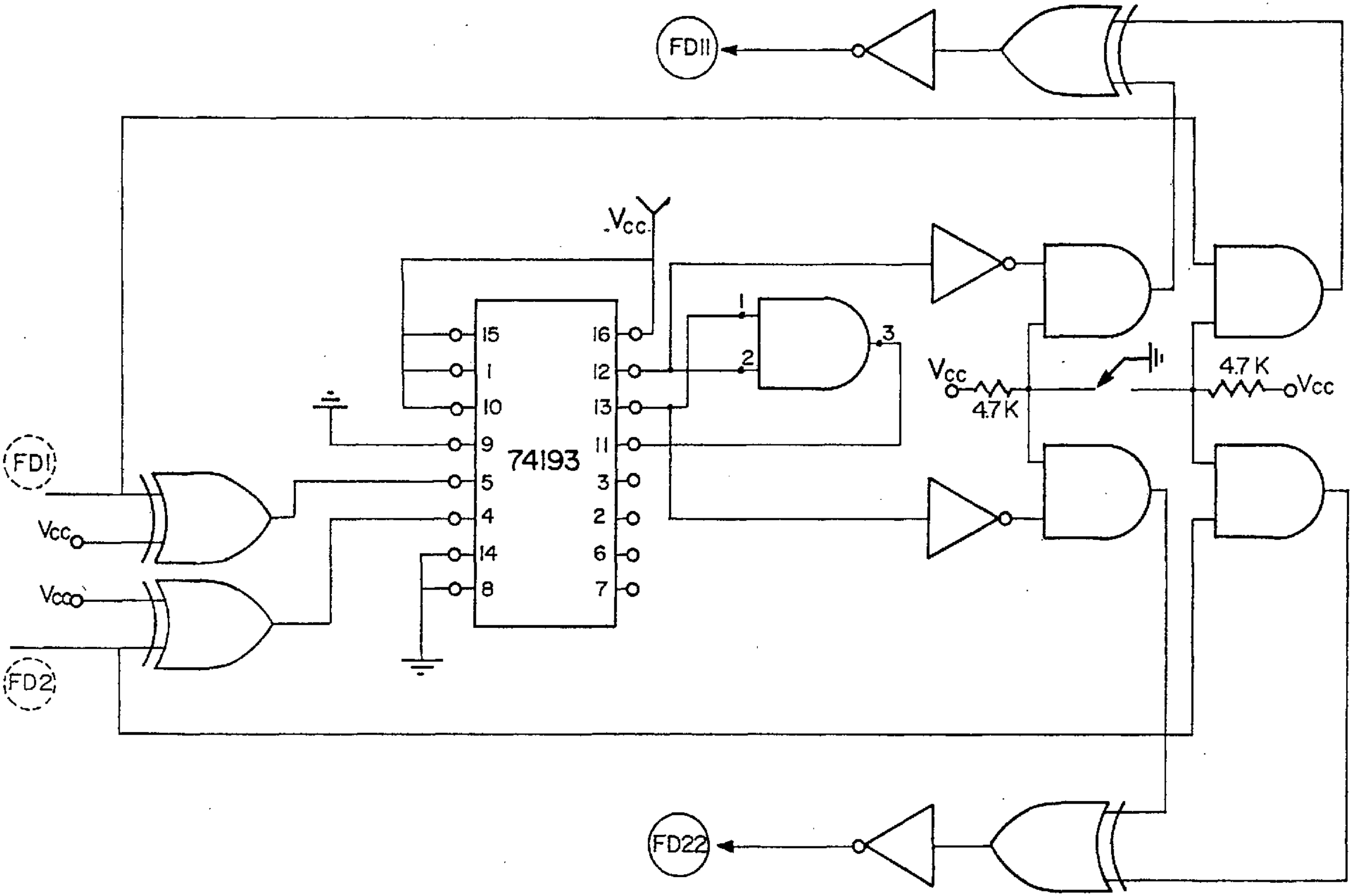


The logic function to be achieved by the right hand half (Combinational) is, FD1 = $(\bar{Q}_1 Q_{11})(Q_2 Q_{22})$ & FD2 = $(Q_1 \bar{Q}_{11})(Q_2 Q_{22})$

AFC LOOP FILTER

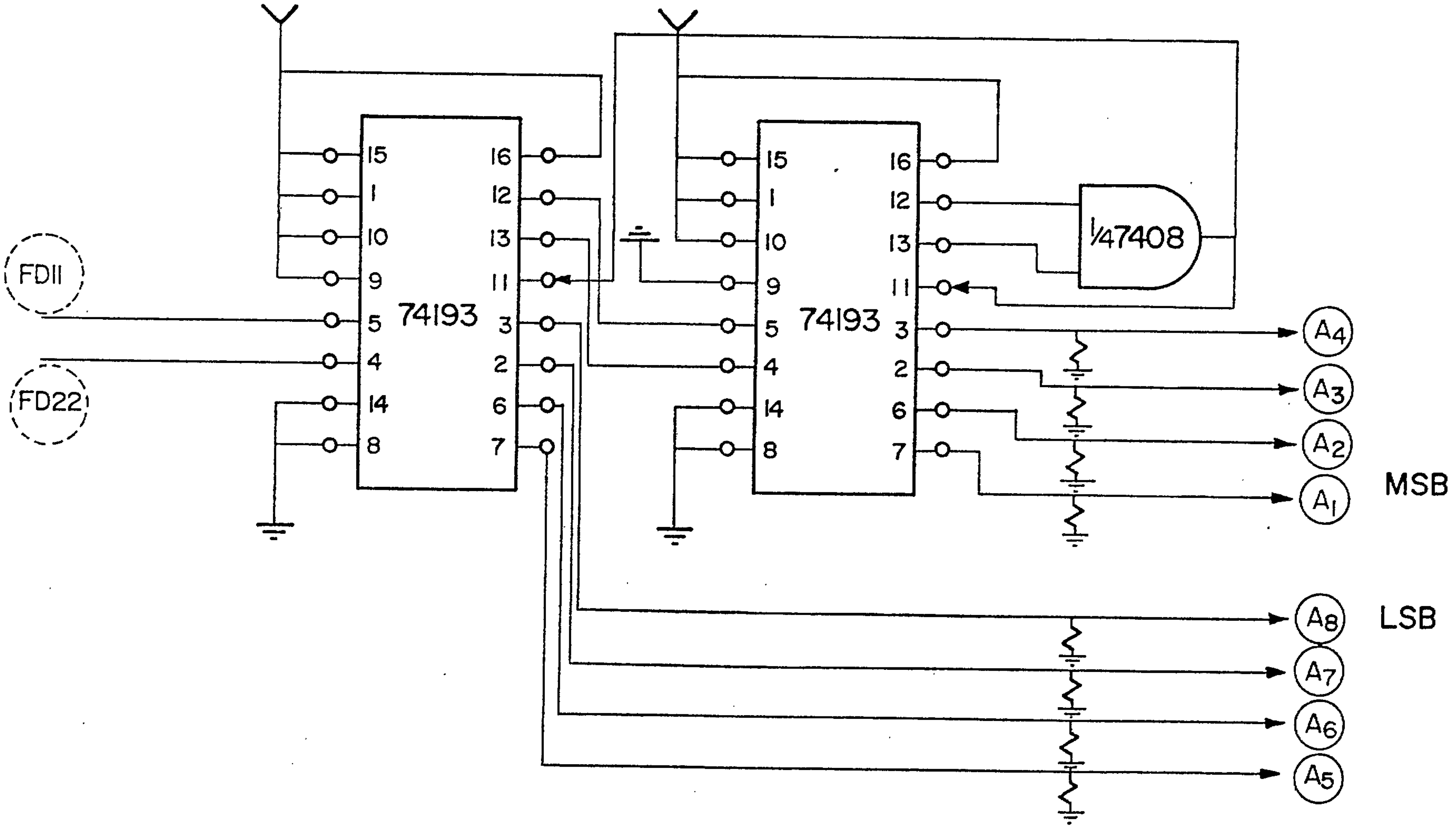
INPUT: F D1 & FD2 (3)

OUTPUT: F D 11 & F D 22 (5)



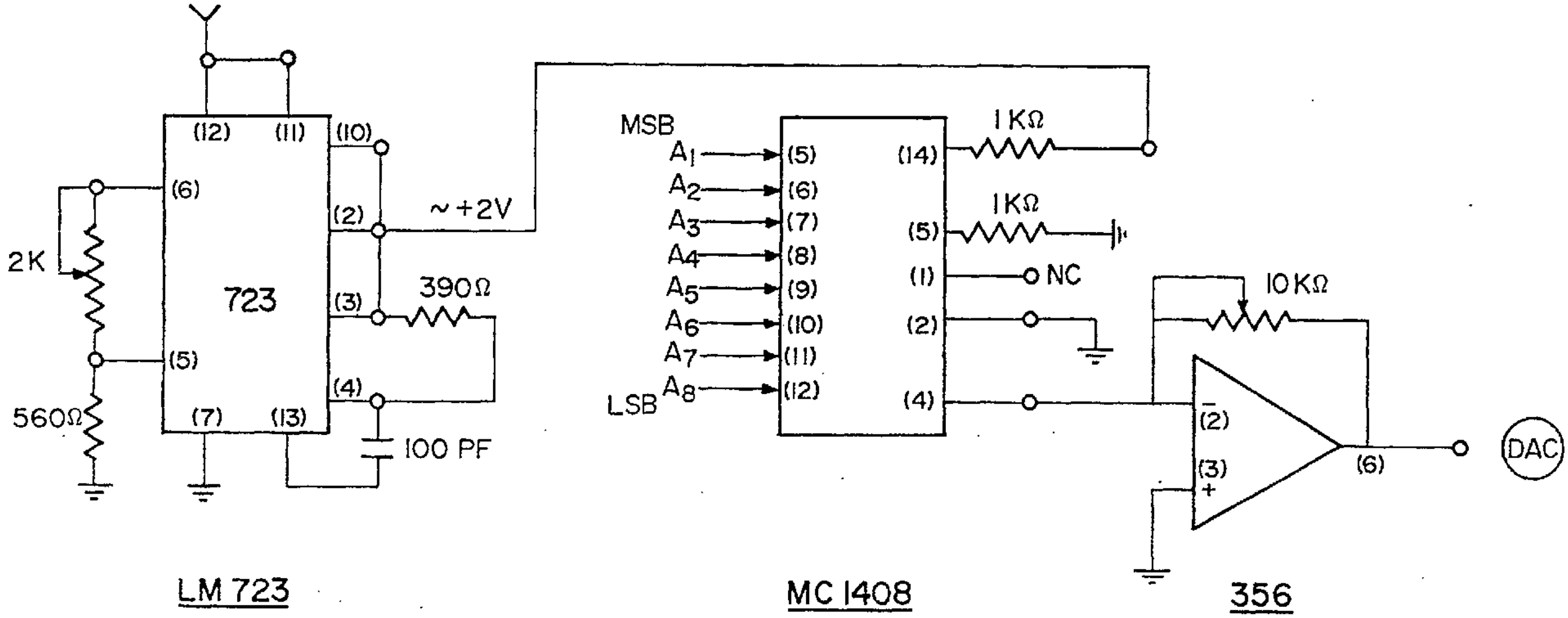
AFC LOOP FILTER

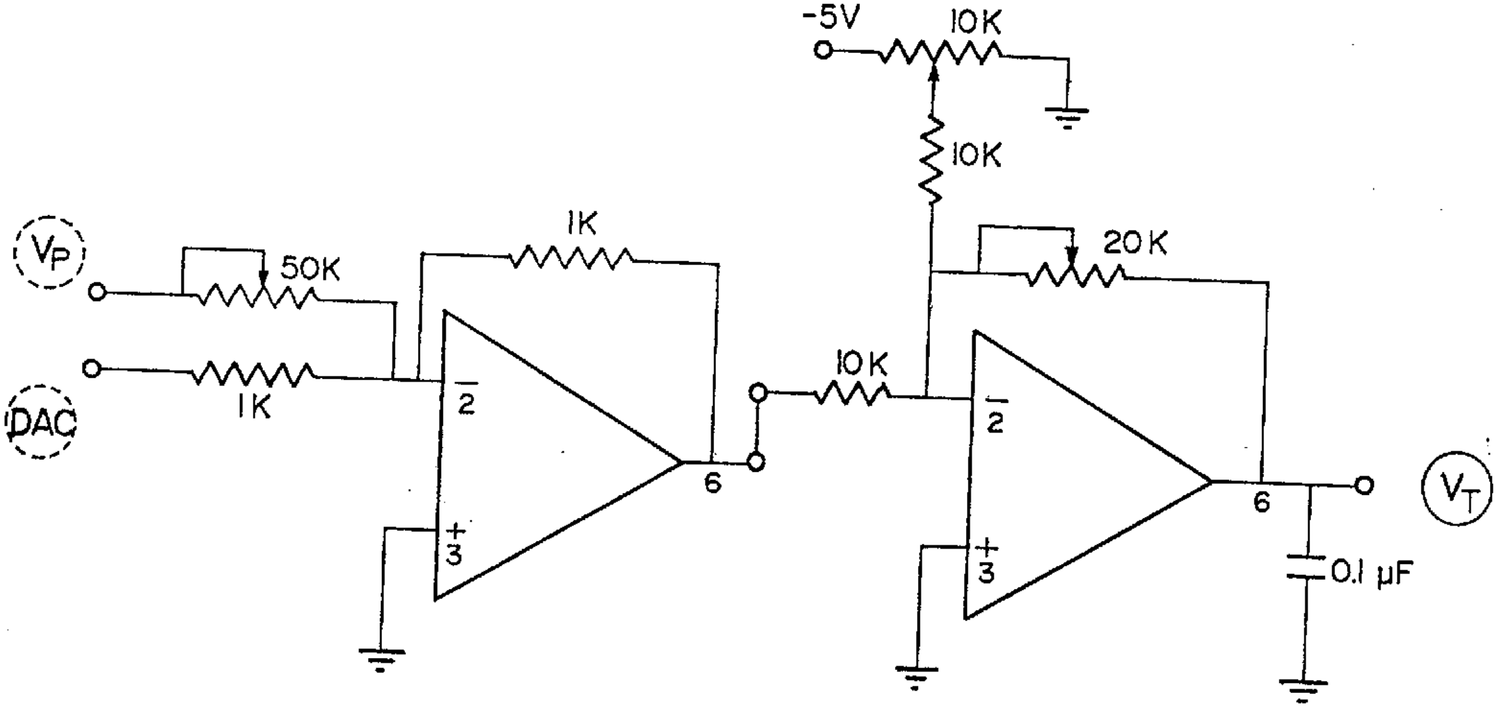
INPUT: FD11 & FD22 (4)
OUTPUT: A_1...................A_8 (6)

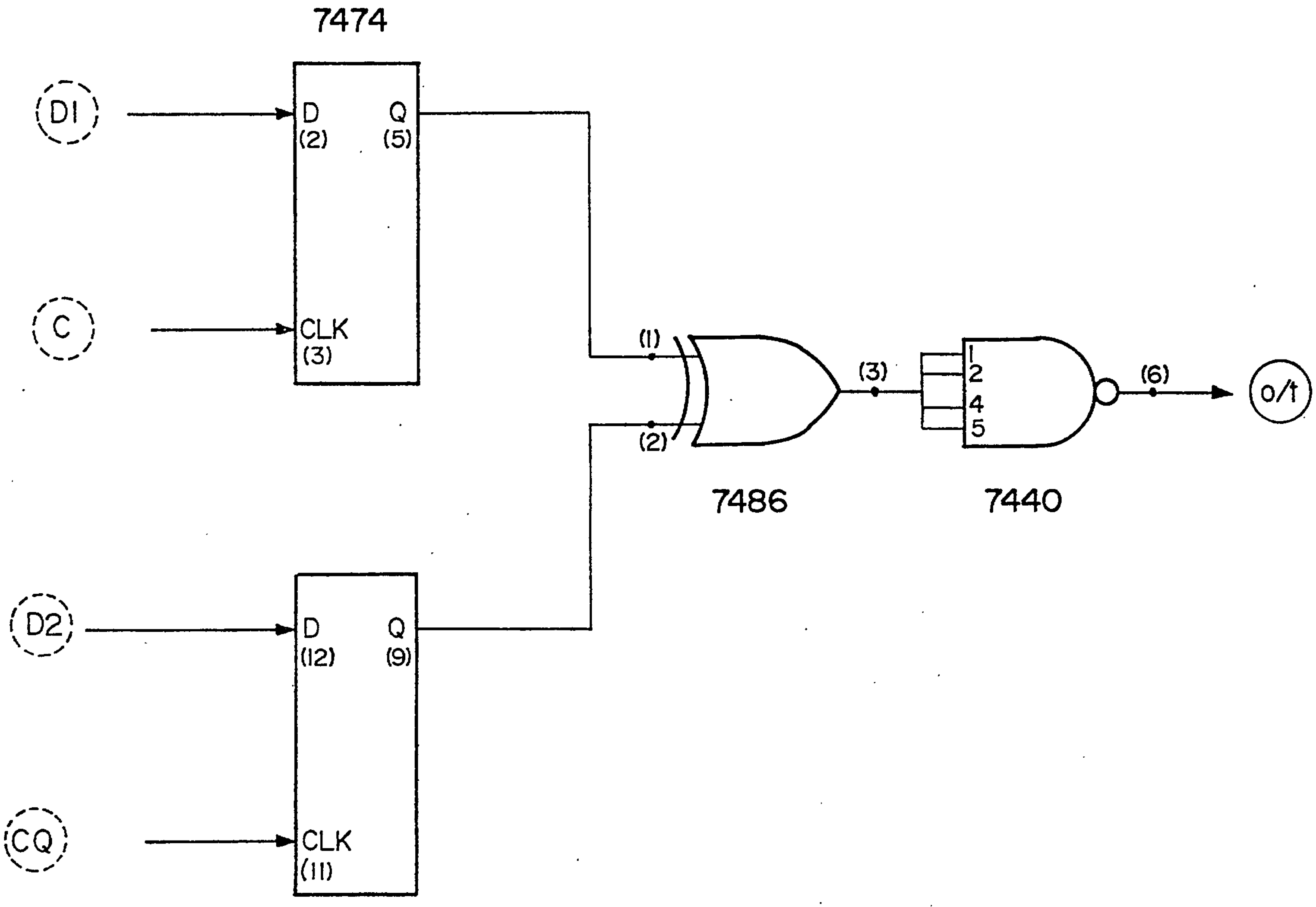


AFC LOOP FILTER

INPUT: $A_1, A_2, \ldots\ldots A_8$ (5)
OUTPUT: DAC (7)

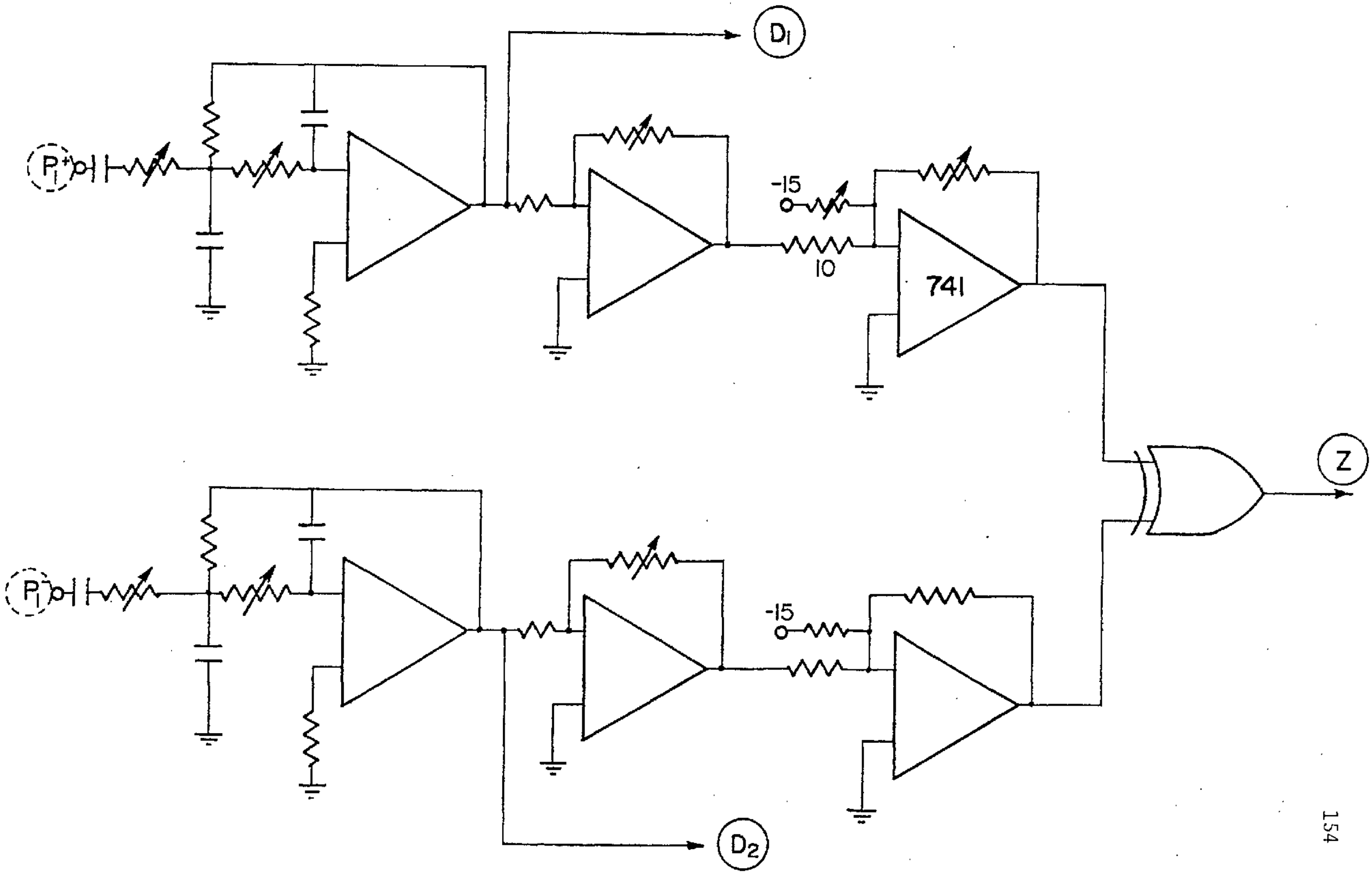



-5V
10K
10K
1K
20K
VP
50K
DAC
1K
2
10K
2
6
6
VT
3
3
0.1 μF


7474
D1
D
(2)
Q
(5)
C
CLK
(3)
D2
D
(12)
Q
(9)
CQ
CLK
(11)
(1)
(2)
7486
(3)
1
2
4
5
7440
(6)
o/t

INPUT: P_1^+, P_1^- (2)

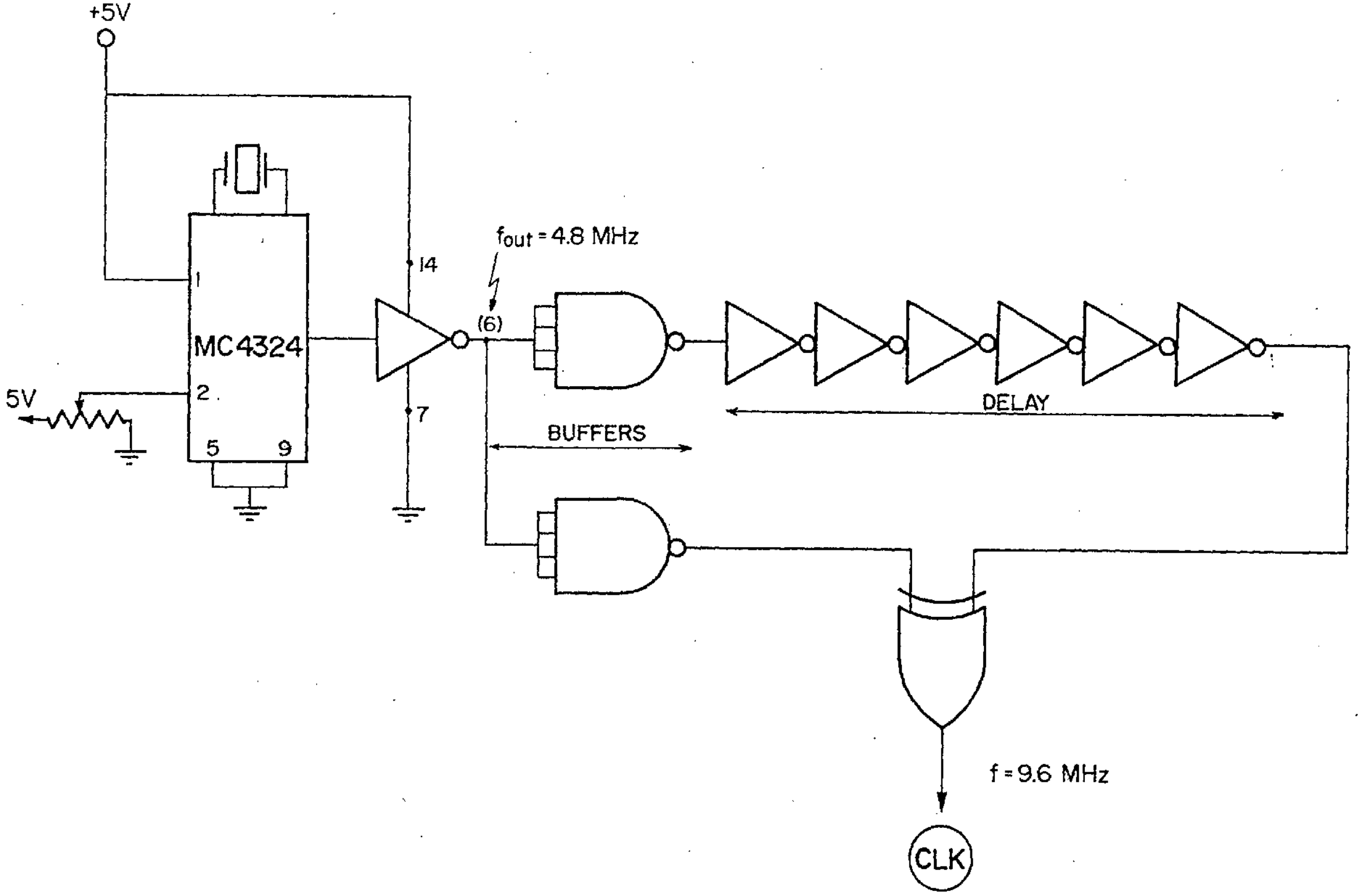
OUTPUT: D_1, D_2 ( ) & Z → clock synchronizer



MASTER CLOCK & FREQUENCY DOUBLER

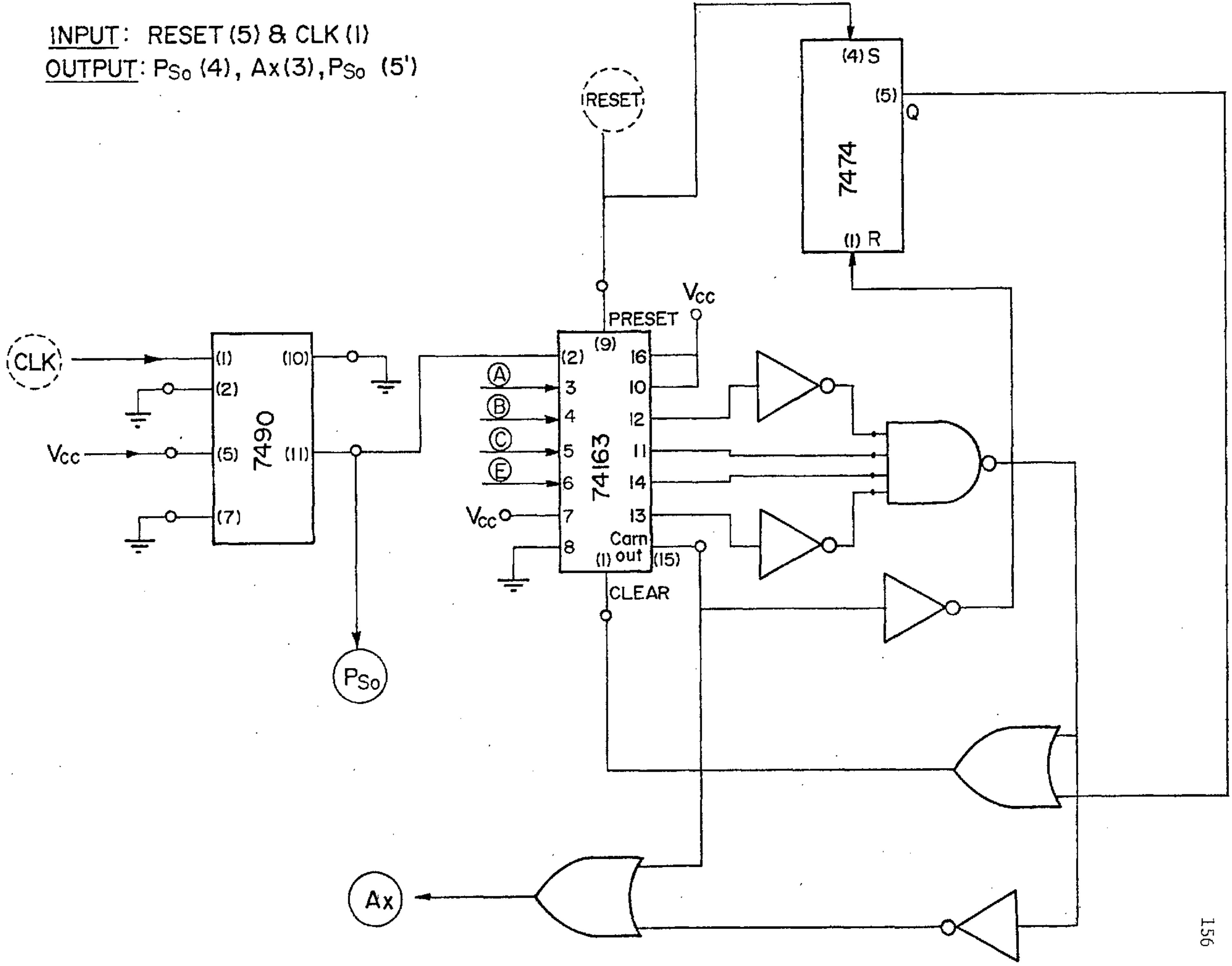
INPUT: ———

OUTPUT: 9.6 MHz refence, CLK (2&3)



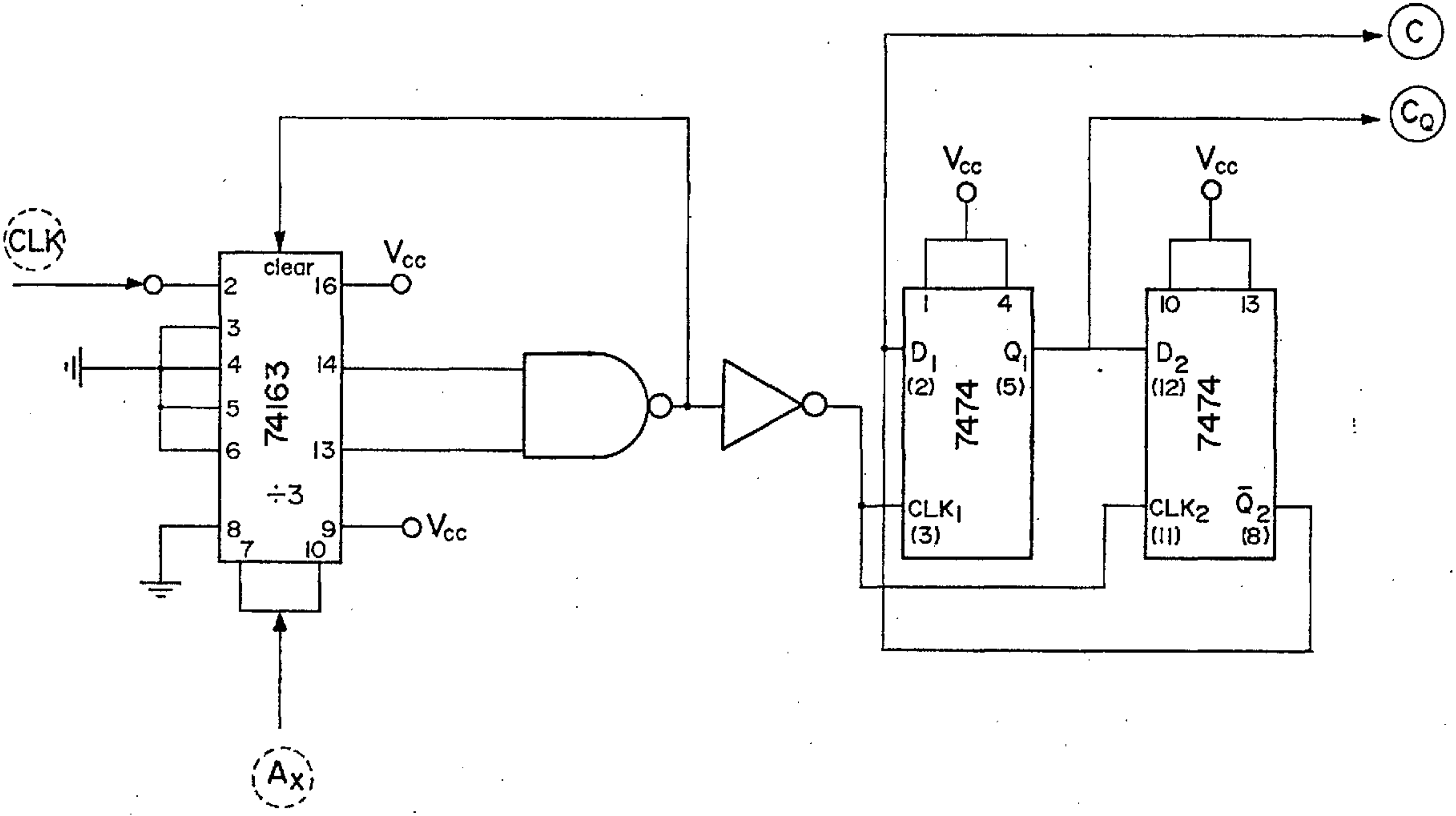
INPUT: RESET (5) & CLK (1)

OUTPUT: P_{So} (4), Ax (3), P_{So} (5')

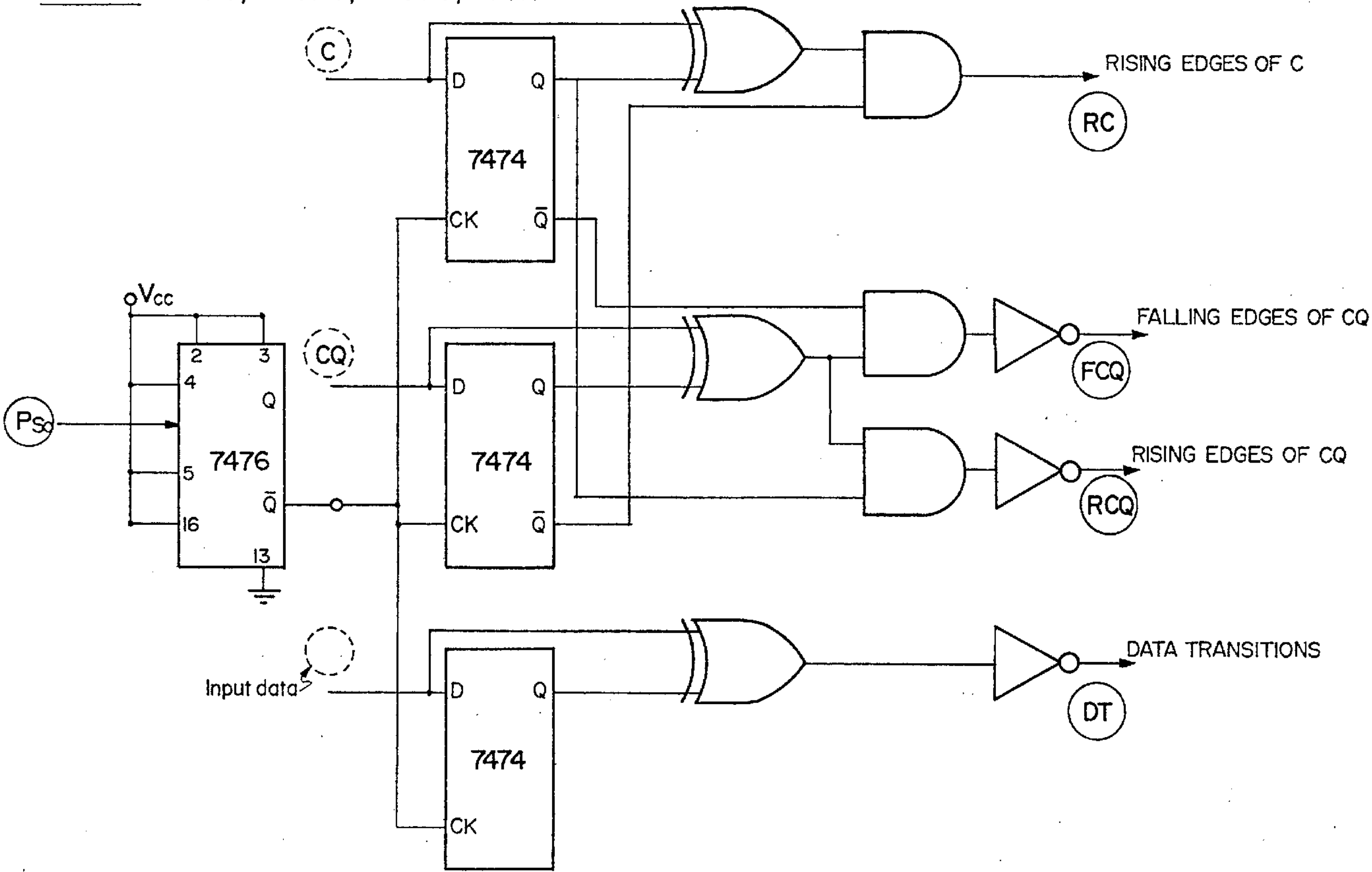


INPUT: CLK(1) & A_X(2)

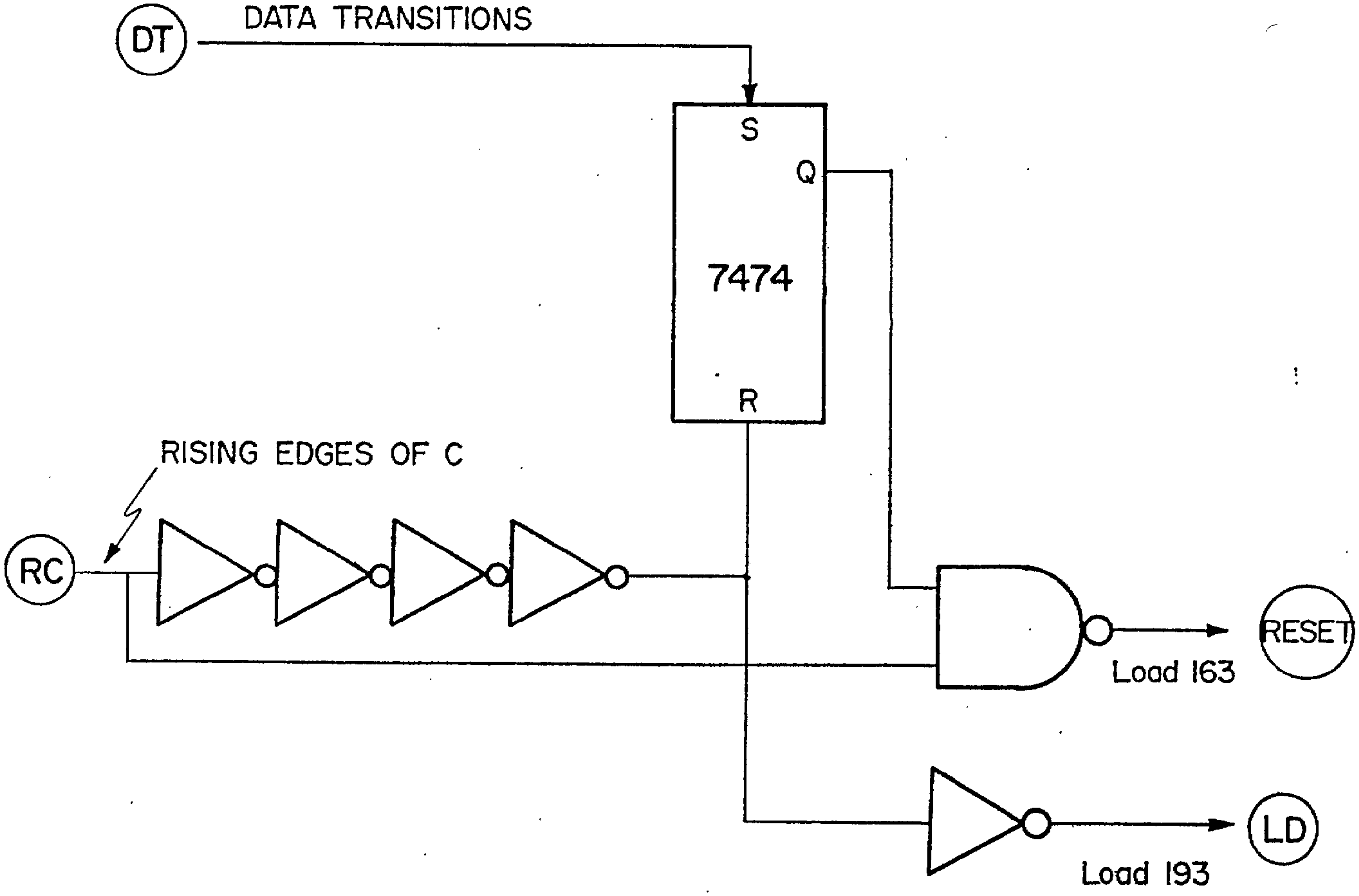
OUTPUT: C(4) & C_Q(4)&(5')



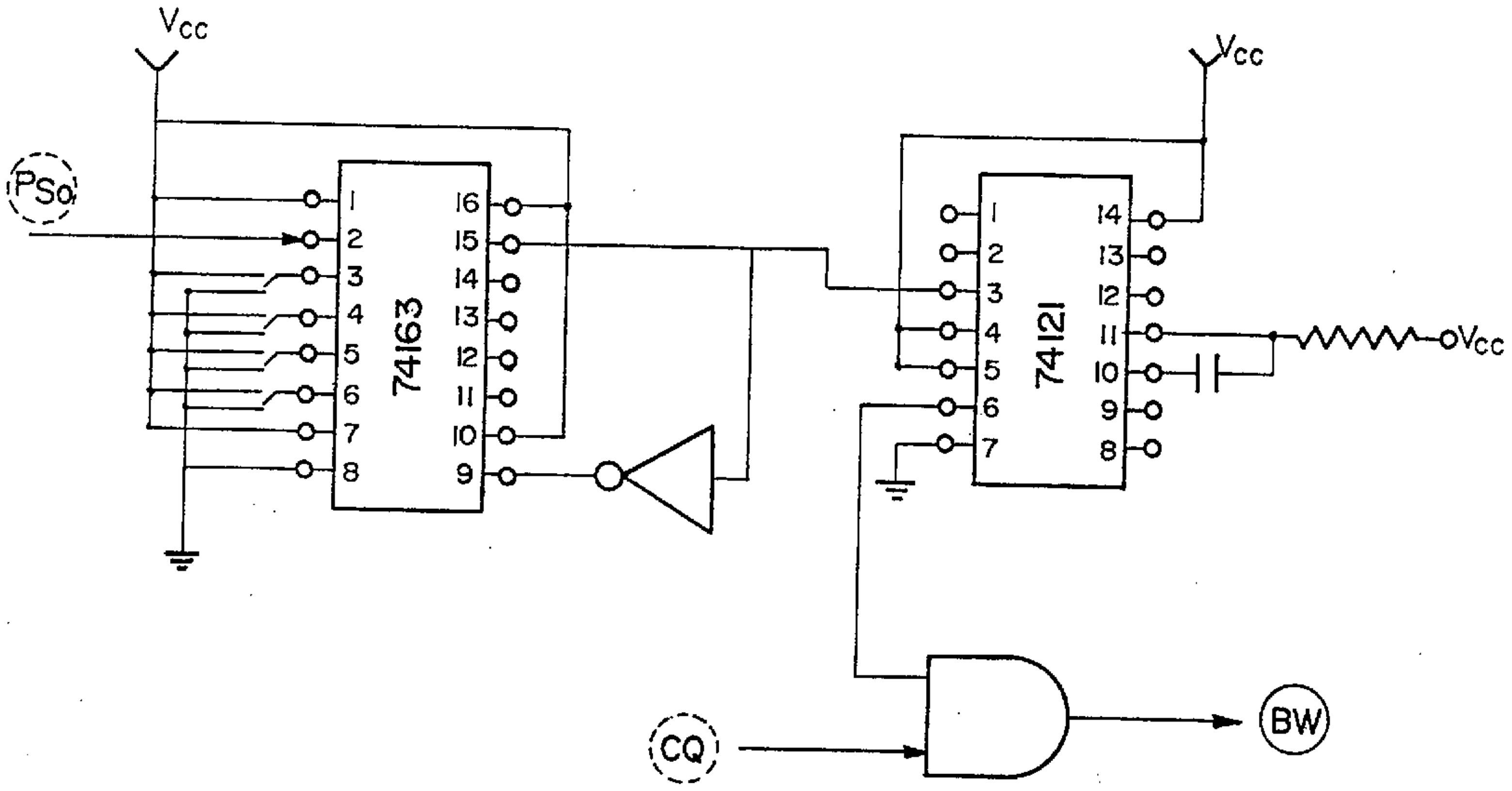
INPUT: C (3), CQ (3), P_{So}(2), Input data (off board)
OUTPUT: RC (5), RCQ ( ), FCQ (6), DT (5)



INPUT: DT (4), RC (4)
OUTPUT: RESET (2), LD (6)

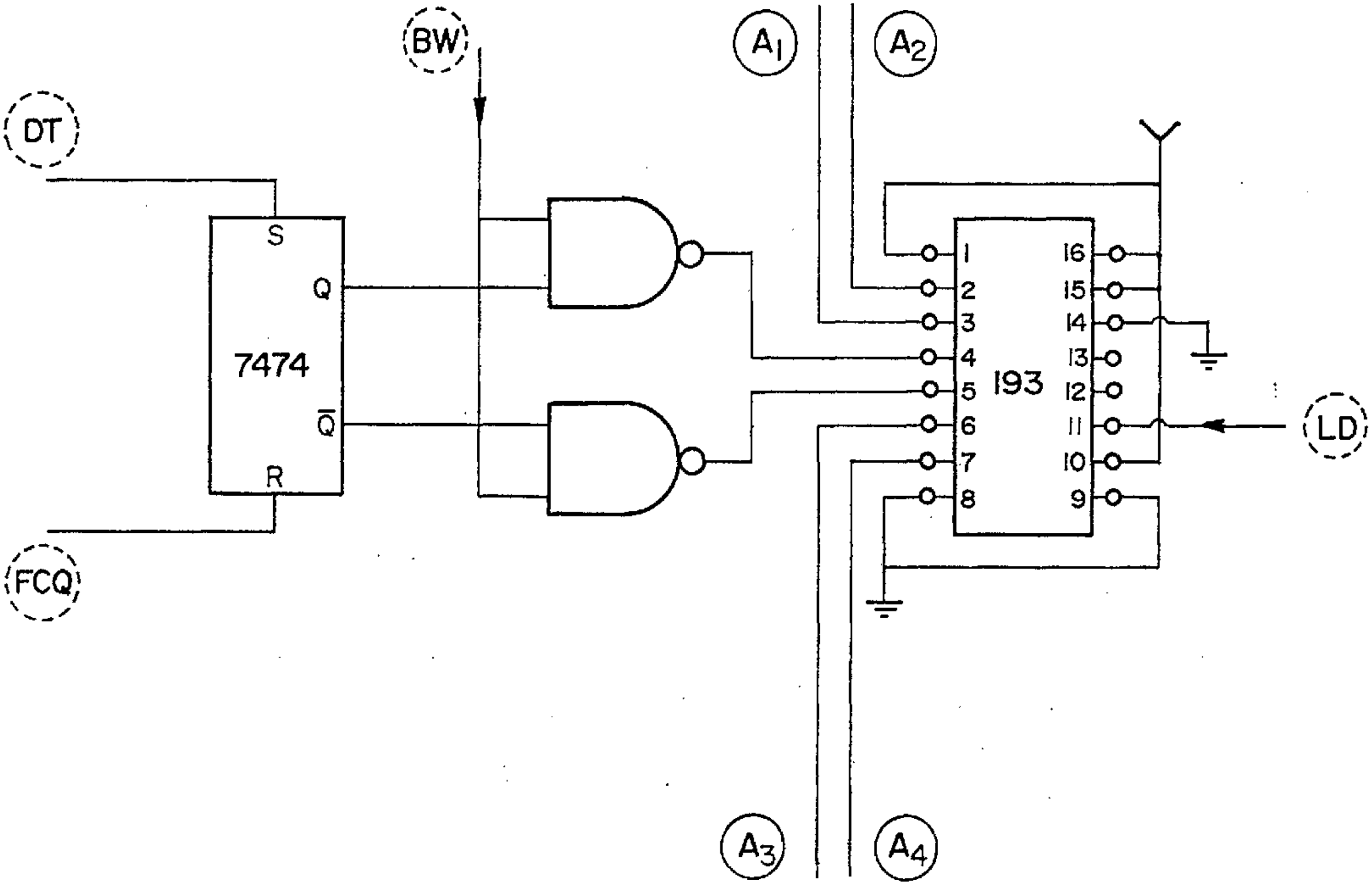


INPUT: PSo (2), CQ (3)
OUTPUT: BW (6)

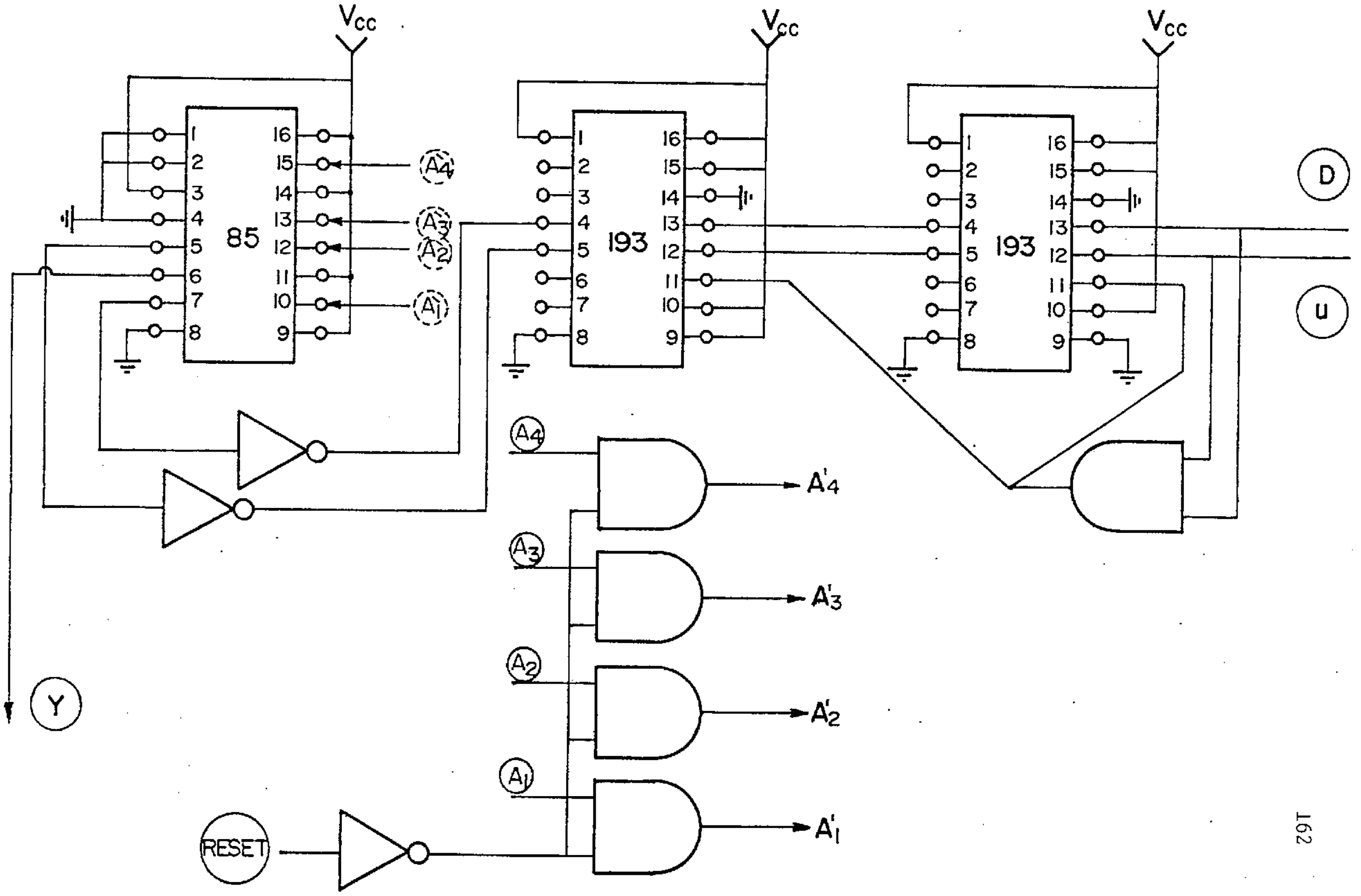


INPUT: LD (5), DT (4), FCQ (4), BW (5')

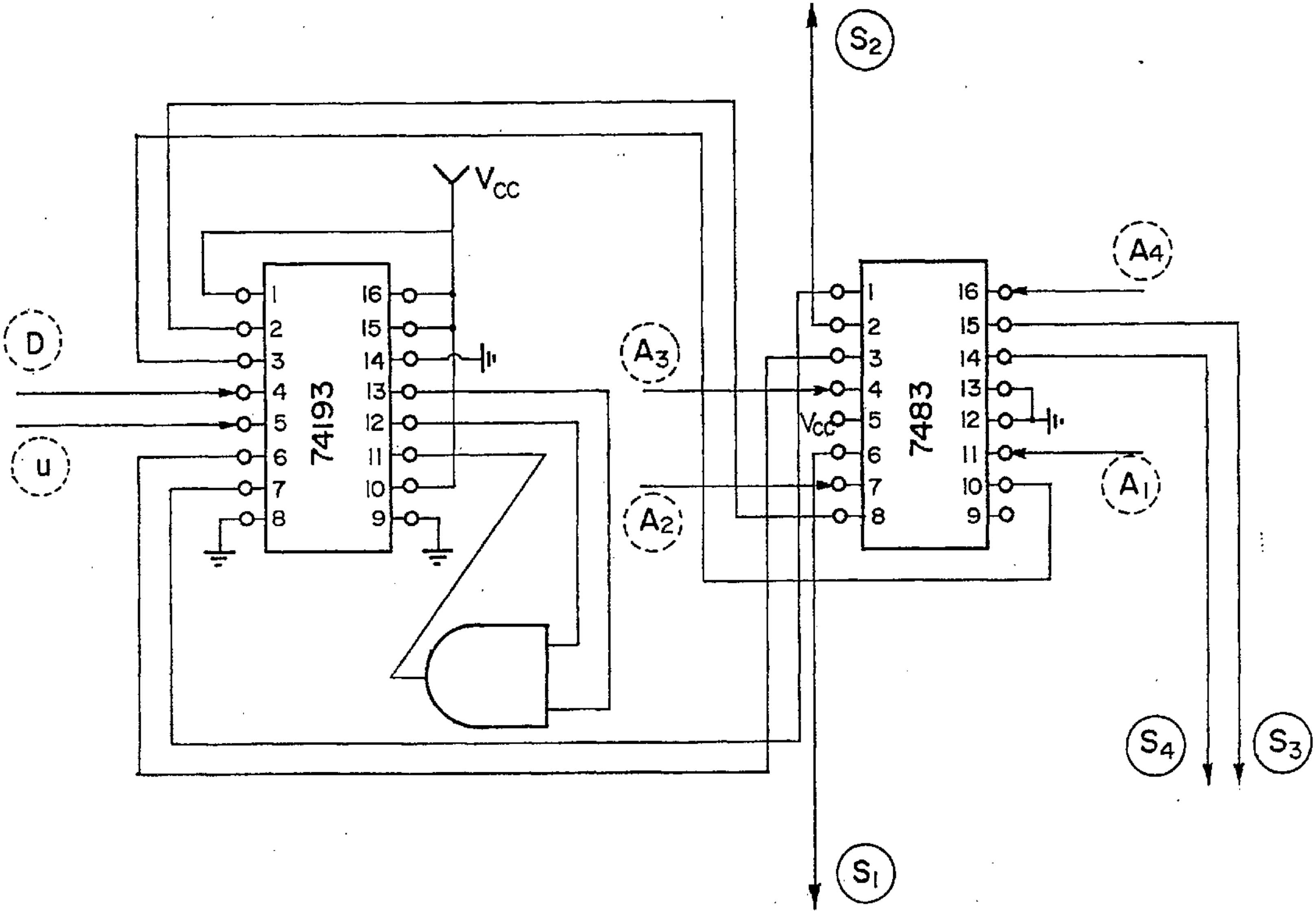
OUTPUT: A_1, A_2, A_3 & A_4 (7)



INPUT : A_1— A_4 (6) , Reset (5)
OUTPUT: D & u (8) , Y (10)

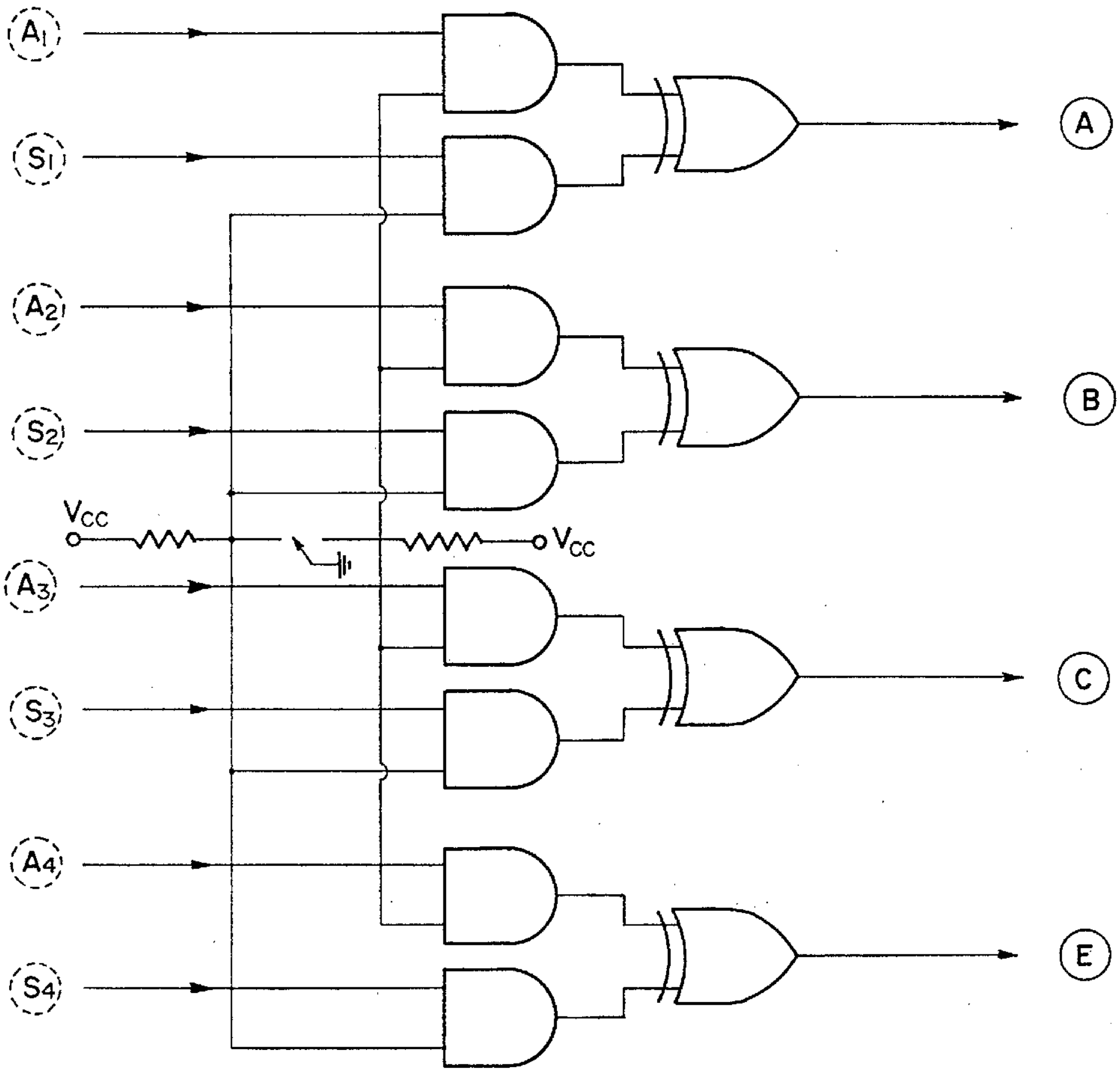


INPUT : D & u (7), A_1 —— A_4 (6)
OUTPUT : S_1 —— S_4 (9)

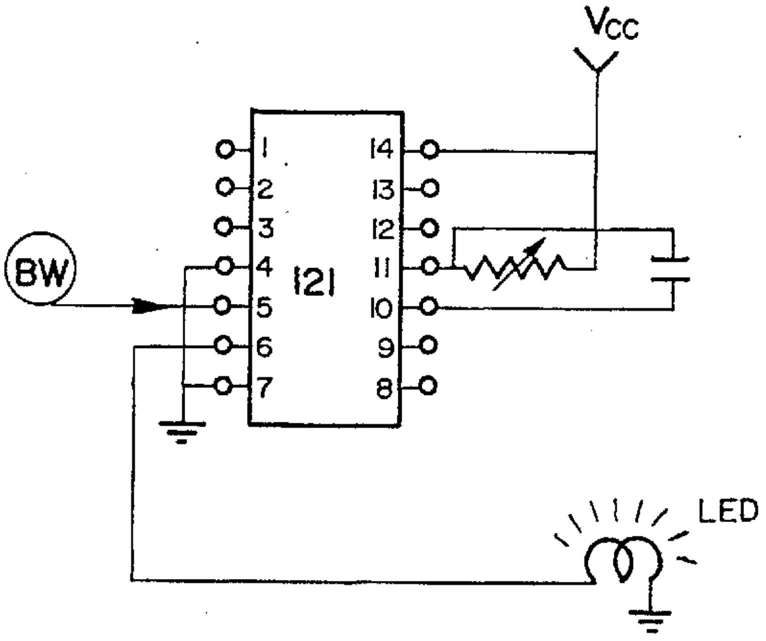


INPUT : A_1 — A_4 (6) & S_1 — S_4(9)

OUTPUT : A,B,C,E (2)



INPUT : Y (7)



10

Appendix B - Background Noise Amplitude Level Measurement

The short-term average amplitude level of the speech detector's input is obtained by low-pass filtering the succession of sample magnitudes as shown in Fig.4.3. With n^{th} input sample magnitude denoted by $|x(n)|$, the estimated level at the n^{th} sampling instant is

$$p(n) = (1-\beta) \sum_{k=0}^{\infty} \beta^k \; |x(n-k)|$$

where β is the feedback gain parameter of the simple one-pole recursive filter. Assuming the input is stationary, the mean of p(n) is

$$M_p = \langle p(n) \rangle = \langle |x(n)| \rangle$$

$$= M_{|x|}, \text{ the mean of } |x(n)|.$$

Assuming successive inputs x(n) are uncorrelated the variance of p(n) is

$$\sigma_p^2 = \langle x(n)^2 \rangle - m_{|x|}^2$$

$$= \frac{1-\beta}{1+\beta} \; \sigma_{|x|}^2, \text{ where}$$

$\sigma_{|x|}^2$ is the variance of $|x(n)|$.

Now we suppose that the successive inputs x(n) are not only uncorrelated, but gaussian with zero mean and variance σ^2. This seems to be a reasonable assumption for the background noise. Then $m_{|x|}$ the mean of $|x(n)|$ is

$$m_x = \frac{1}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{\infty} |x| \; e^{\frac{-x^2}{2\sigma_x^2}} \; dx = \sqrt{\frac{2}{\pi}} \; \sigma_x$$

and the variance

$$\sigma_{|x|}^{2} = \frac{1}{\sqrt{2\pi}\sigma_x} \int_{\infty}^{\infty} x^2 \, e^{\frac{-x^2}{2\sigma_x^2}} \, dx - m_x^{\,2} = (1 - \frac{2}{\pi}) \, \sigma_x^{\,2}$$

Thus the ratio of the standard deviations of p(n) to its mean is

$$\frac{\sigma p}{m_p} = \sqrt{\frac{\pi}{2} - 1} \cdot \sqrt{\frac{1 - \beta}{1 + \beta}} = .755 \sqrt{\frac{1 - \beta}{1 + \beta}} \qquad .$$

The parameter β , governing the low pass filter's time constant, or bandwidth, should be chosen so that the standard deviation is a small fraction of the mean. A value of $\beta = 1 - 2^{-7}$, gives $\sigma_p/m_p = .0472$, about 5%. With this value of β, the filter time constant is $\frac{1}{1-\beta} = 2^7 = 128$ samples, or 16 msec for a sampling rate of 8 khz.

Appendix C

C.1 User information for SDESS I

With SDESS I, the user has two modules, CRSMFL and SDSIML, the former to create samplefiles and the latter to process the samples. A set of commands is available on each module. When no command is in execution, the system is always under CD, the Command Decoder. CD lists the repertoire of commands and prompts for a reply. When execution of a command is finished, the system again return to CD. Commands can be issued in any order, i.e. CD does not check for semantics. CRSMFL commands:

1. Create file: Saves the samples in memory on a diskette file. The user is prompted for a standard RT-11 filename.
2. View sample: Looks at the samples in memory, effectively a memory dump.
3. Re-sample : When CRSMPL is run, it prompts the user for a go-ahead signal before sampling a speech segment. When the user decides to scrap the samples currently in memory, this command can be issued.
4. Halt: Exit to RT-11.

SDSIML commands:

1. Process sample: Runs the speech detector algorithm. The user is asked to specify all the parameters and to set a breakpoint. During a break point the"internal registers" of the speech detector are displayed and the user is free to View-Sample. The user can set the next breakpoint and restart simulation.

2. Store sample: Same as Create-file. Note that no distinction is made between processed and raw samples.

3. View sample: described above.

4, 5, 6. Play-original, Play-processed, Play-both: These commands are equivalent in that the speech segment in memory is D/A ed repetitively on two channels until the user hits a key on the keyboard.

Command	channel X	channel y
4	original	processed
5	processed	original
6	original	processed

7. Read file : retrieve a file from the diskette. The user is prompted for a file name.

8. Halt: returns to RT-11

C.2 General design considerations prior to designing version 1 of the simulation software.

Memory: PDP-11 architecture has 32K-words (2bytes) addressing space, the following amount of which is unavailable for users' programs:

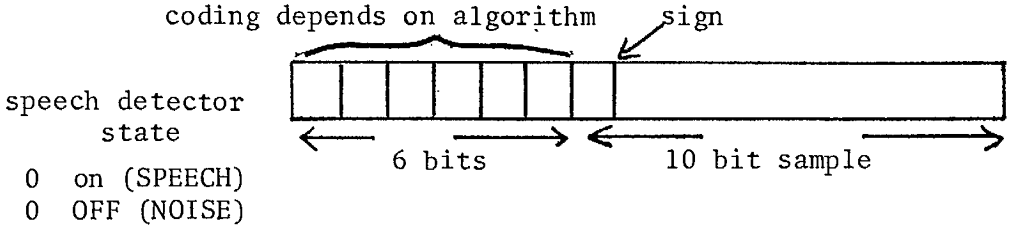
4K	IO-mapping space
1/2K	interrupt vectors
1/2K	stack
2K	Resident monitor
2 1/2K	System: User Service Routines object Time System
1/2 to 1K	I/O (variable): Device Handlers Buffer Channel Tables
Σ≈10-10 1/2K	

For user: 32-(10 or 10 1/2K) $\approx$ 22K

A 2K program would leave us with about 20K memory space. At 8khz sampling rate, 20K corresponds to 2.5 sec of speech assuming one sample word . The actual length in implementation is 2.75 sec.

Time: The 125 μsec real time requirement is no cause for concern since nothing else is being done while D/A or A/D is in process. A/D and D/A is automatically actuated by AR11's real time clock every sampling period. A program needs only monitor the "Done" bit of the A/D or D/A status register to read in or send out a sample.

Sample storage : It was initially thought that to save storage, a sample should be stored in a byte by dropping the least significant 2 bits of an A/D word. In addition to a degradation of signal-to-quantization-noise by 12 dB, such a scheme would leave no space for storage of speech detector states. Although lengthening speech segments to about s sec each, the idea was unworkable due primarily to the latter reason. A sample word thus looks like this:



With five bits, one can code 32 different states for the speech detector at each sample point. Of course, the actual values of the detector's "registers", if any, can only be viewed at breakpoints.

Filters: All filters are sampled=data forms of the analog single pole LPF

$$\frac{1}{1-\tau s} \xrightarrow[\text{invariant}]{\text{impulse}} \frac{1}{1-e^{\frac{-T}{\tau}}z^{-1}} \xrightarrow[\text{gain}]{\text{unit}} \frac{1-\beta}{1-\beta z^{-1}}$$

where $\tau=$ is sampling period

$\frac{1}{\tau}=$ 3 dB Bandwidth

$\beta= e^{-T/\tau}$

Since it is desirable to implement multiplication as shifting in the final prototype, SDESS I should use shifting in place of multiplication, even though the 11/45 is equipped with fixed point and floating point multiplication hardware.

$$\begin{aligned} y(n) &= \beta\ y(n-1) + (1-\beta)x(n) \\ &= x(n) + \beta(n-1)-x(n)) \\ &= x(n) + \text{shift } (y(n-1)-x(n)) \end{aligned}$$

$$\text{where} \quad \beta = 2^{n} < \begin{array}{l} \text{left shift } n>o \\ \text{right shift } n<o \\ \text{sign extended)} \end{array}$$

C-3 Software Organization

As it turns out, the organization of SDESS I is much simpler than that of SDESS II. It has a hierarchical structure as shown in Figure 5.

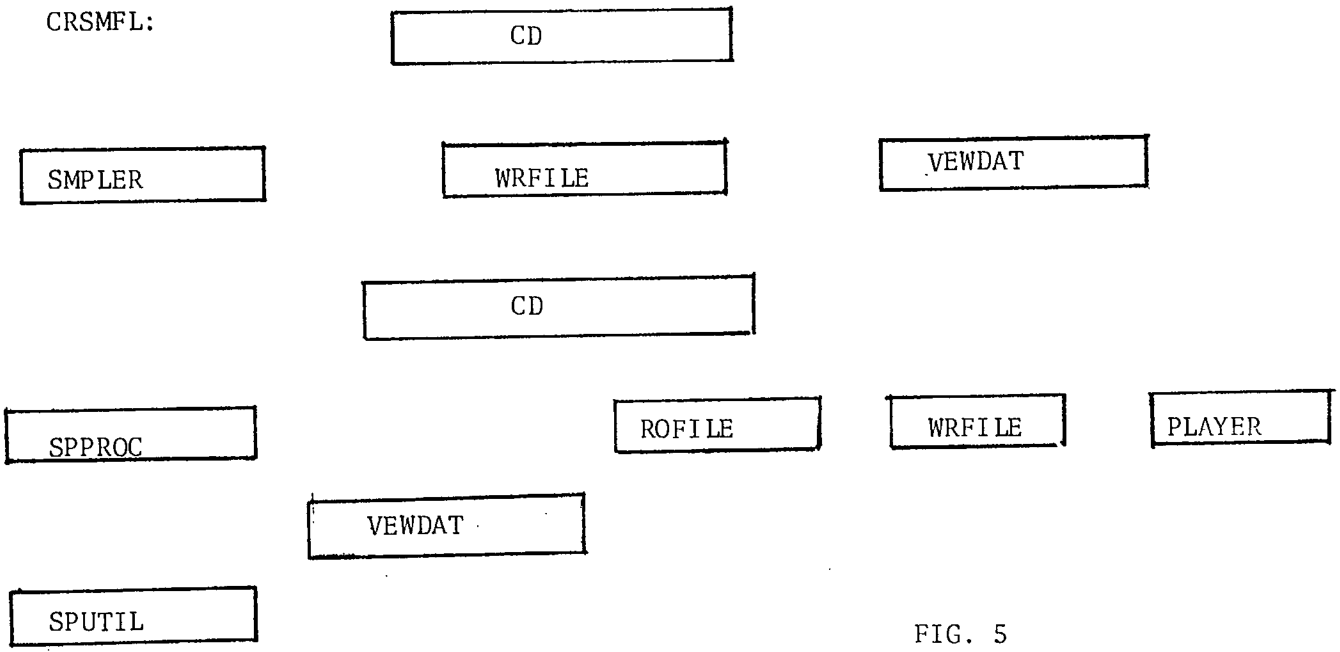


FIG. 5

CD:	Command Decoder
SMPLER:	performs A/D
WRFILE:	creates sample files
VEWDAT:	views samples in memory through "windows"
SPPROC:	drives the speech detector; breakpoint facility

SPUTIL: various speech processing utilities such as:

LPF: Low Pass Filter

MARKSM: Mark the code workds onto samples

REINIT: Clear all code words

SHIFT : right or left shifts

MAG : compute magnitudes from offset A/D words

RDFILE: Load memory from a file

PLAYER: D/A speech (see SDSIML commands)

C·4 The transition to SDESS II

Several alternatives to improve SDESS I were studied, the first of which was to employ the Memory Management Unit (MMU) to utilize all 92K words of physical memory.Such an approach could be very sound since one does not have to be concerned with file I/O during any memory operation. But the following factors defeated this argument:

The extended memory monitor (XM) which supports the MMU and the newer versions of RT-11 under which XM runs are not part of local expertise.

It is doubtful that the portion of XM which manages the set of Page Address Registers and Page Description Registers can operate in real time.

Special attention must be given to interrupt service routines. In particular, anything related to interrupts must be static.

The newer versions of FORTRAN which support extended memory addressing are not in use locally.

The idea finally adopted for extended storage was to double-buffer samples into a workspace on a disk (RK05). The two RK05's are driven by a hardwired countroller, the RK11, which once its registers are initialized, will perform data transfer by DMA (Direct Memory Access) until the transfer is completed; at that point the CPU is interruped. The design considerations will be presented later on.

As for waveform display, it was deemed that flexibility and generality were most desirable. We sought a display wherein two waveforms are juxtaposed while the user issues commands to scale and shift the waveform. One should be able to, say, select a .5 sec spurt, listen to it, and look at its fine details on the screen with dynamic scaling. Furthermore, the signals plotted should include "signals" internal to the device being simulated instead of confined to speech.

Two approaches were considered: D/A the waveforms at variable rate and scan them on a storage scope; program the VT11 graphics processor. In the former approach, one would adjust the time scale of the scope such that when the stored signal is expanded for viewing, aliasing would be avoided; this is so because the sampling rate varies with the time base. The approach was judged unsatisfactory because:

(i) In comparison with the VR17 screen, display on a scope is relatively coarse.

(ii) One cannot post numeric information on the screen e.g. it is desirable to know the effective time scaling

(iii) It is very difficult to trigger the scope at any desirable point of the signal. Manual triggering is necessary.

So the VTII was chosen because it does not suffer the above shortcomings. The design consideration becomes primarily one of memory allocation, sicne VTII's display file is also stored in memory. More on this issue will be presented later on.

In the sections that follow, the user view of SDESS II will be given, followed by its design considerations and technical documentation.

C-5 User Information

Files

Disk files are numbered 1. 2, ...,9. Within the workspace (wksp), a sample location is defined by two numbers:

DTN	$0 \leqslant \text{DIN} \leqslant 19$
LOC	$0 \leqslant \text{LOC} \leqslant 6143$

Explanation: The workspace is viewed as composed of 20 DTN-segments (contiguous and mutually exclusice), each is 6144 words in length. During breakpoints, the user is not allowed to view any location below LOC=64 within every DTN-segment.

A file is defined as having certain attributes associated with it, even though a directory is not kept:

	Type	
1.	Empty	-- an empty file
2.	Raw	-- raw from A/D
3.	Processed	- "marked" after processing

These attributes also apply to the workspace. Commands incompatible with the attributes of a file or workspace are aborted.

Associated with each processed file are the parameters used to simulate the speech detector. These parameters are stored in a 64-word header at the beginning of every even-numbered DTN-segment. Also stored in the header is state information of the speech detector just before processing that segment. The format of this header is described later on.

Commands

SDGSS II is run as a monolithic package called SDESS2. Upon startup and completion of each command, the system is under the Command Decoder (CD). The user is prompted with a list of commands, the user can issue commands in any order what soever i.e. no semantic checking.

1. Load Wksp (file #): Loads a file into Wksp.
2. Save Wksp (file #): the reverse of 1. if the file es non-empty, the user is asked to confirm.
3. Sample speech: A/D and puts samples in Wksp.
4. Replay original speech segment: D/A speech in Wksp.
5. Replay processed speech segment: D/A speech (if processed) in Wksp. The speech is muted where the speech detector is marked OFF

6. Process speech: User supplies parameters to simulate the speech detector with samples in the Wksp. The user may set breakpoints. At each breakpoint, the user's view is restricted to the DTN-segment where the breakpoint belongs.

7. File status (file #): Request that the attributes of a file be displayed. If the file is "processed", the parameters used for simulation are displayed.

8. Display signals: The user is first asked to define a time frame in terms of sample-addresses. A time frame is specified by the sample-location of its first sample and the length of the frame in samples. The minimum frame length is 1024 words. The user view is restricted to that time frame in the course of executing the command.

   Two waveform are displayed one of which the user may have specified. The fixed waveform is the original speech segment for reference purpose. The selected waveform may be processed speech or from a given set of internal "signals" of the speech detector the set being dependent on the algorithm implemented.

   The user may hit the following keys to translate, expand, and contract the signals:

   "R" : translates signal to the right. Translation stops at left margin of frame.

   "L" : same as above except for left translation.

   "E": expands signal about the centre of the screen. Expansion stops when there are about 100 samples on the screen.

"C" : Contracts signal about the centre of the screen. Contraction stops when both boundaries of the time frame are reached.

'S" : stops graphics.

Due to a lack of memory space, only 1024 samples are stored in memory for each time frame of a waveform. If the time frame is too large, when the signal is expanded, the user sees the aliased version of the signal. This, of course, depends on the bandwidth of the signal viewed.

9. Mix (file #): Add a file to the Wksp. The command is not valid if:

(i) the file is not a raw file

(ii) either the file or Wksp is empty.

If the Wksp is "processed", the user is asked to confirm the action. After mixing, the status of the file is "raw". The user may specify a scaling factor applicable to the file prior to mixing.

10. Various data collection utilities: This feature is not completely defined yet. Various options may be selected:

- search for the point at which the detector switch between "ON" and "OFF" - this corresponds very simply to a sign change for the samples.

- sum the # of state transitions -- this # divided by two corresponds to the # of spurts and gaps.

C.6 General Design Considerations for SDESS II

Files -- as dictated by speed and memory constraints

The following factors must be taken into account:

(i) Without a directory, the names, disk locations, and size of files are must be fixed.

(ii) A RK05 disk stores 1.23 M words, divided into 200 cylinders, with both top and bottom surfaces for each cylinder i.e. altogether 400 tracks. Each track is further subdivided into segments, but this is of no interest since the overhead of a segment transfer is too high (as will be clear). Thus there are 3K = 3072 words to a track.

(iii) A disk must be partitioned so that there are "comfortable" number of files and "sufficiently lengthy" speech segments to work with.

(iv) A file should be further subdivided into units equal in length to the size of a buffer into memory. Sequential numbering of these units should correspond to the order in which head movement is automatically advanced by RK05. These units should not be so small that the overhead of controlling excesses approaches the idle time of CPU during D/A or A/D.

(v) A file should not be bigger than what a diskette can hold.

It is now apparent that the size of each unit of data transferred is governed by real time requirements. Thus let us first consider timing. The following factors should be noted:

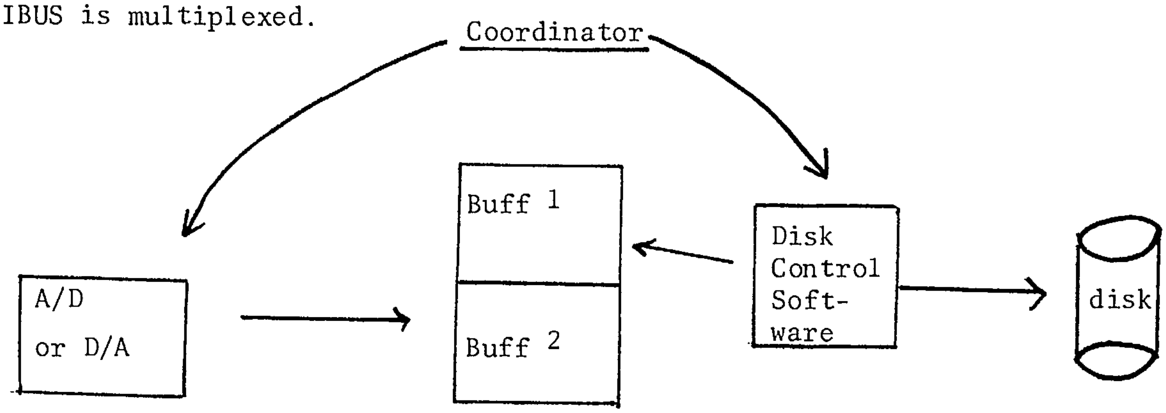
(a) The worse case latency time is 40 msec (for 1 revolution). Track seek time (per track) = 10 msec (including settling time). Data transfer = 11.1 μsec/byte.

(b) To control the RK05, a software disk controller must be executed for each access.

(c) An interrupt service routine for A/D or D/A must be executed every 125 μsec.

(d) A monitor process must be executed whereever the disk software or interrupt service routine is not being executed. The process is needed to coordinated double buffering (see Fig. below).

(e) The CPU is effectively slowed down by DMA cycle-stealing i.e. the UNIBUS is multiplexed.



Disk time per unit transfer

$$= \left(11.1 \frac{\mu\text{sec}}{\text{word}}\right) (n \text{ words})$$

$$+ \left(10 \frac{\text{msec}}{\text{track}}\right) \quad (m \text{ tracks})$$

$$+ \left(40 \frac{\text{msec}}{\text{track}}\right) \quad (m \text{ tracks})$$

The difference between this time and (n words)(125 $\frac{\mu\text{sec}}{\text{word}}$)

is the time left for (c), (d), (e), and (b) above.

The difference, called D, must satisfy

$$D > [n \times T(ISR)] + T(DSW)$$
$$+ [n \times T(COOR)] + m$$

where T(ISR)=time for 1 cycle of the Interrupt Service Routine taking into account (e) above.

T(COOR) = time for coordinator to check the buffer status and switch the 2 buffers and inform the disk software, taking into account (e) above.

T(DSW) = time for one cycle fo disk software

M = safety margin

Using $= (m \text{ tracks})(30+2 \frac{\text{words}}{\text{track}})$

$D = .384m - .0841m \cong .3m$ (sec)

where m= # of tracks.

If, say, T(ISR) = 50 μsec

T(COOR)= 20 μsec

T(DSW) = 200 μsec

then

$$.3m > .215m + .0002 + M$$

If m=1

$$m < .085 = 85 \text{ msec}$$

Therefore any transfer -unit greater that 1 track should do.

Note= 1 track = 3K words

In discussing the memory allocation for SDESS II, we found that approx. 21 to 22K words are available for the user's program. Assuming the worse case and allot 6K words for SDESS II, one is left with 15K for buffers. Two 1K-buffers are required for displaying two waveforms and we are left with double-buffering, then a transfer-unit = 6K = 2 tracks. 2 tracks correspond to one cylinder = two surfaces of a platter (RK05 has only 1 platter). This means that a disk head makes no physical movement at all while transfering one unit. At the end of a unit-transfer, the disk controlling software is executed while RK11 automatically positions the head to next cylinder -- full parallelism. I transfer - unit is given the name <u>D</u>ouble <u>T</u>rack (DT)(1 DT=6K)

How many DT's should make up a file?

A RK05 has 203 DT's

A standard floppy stores 128K words

128K words = 21.3 DT's

So allocate 20 DT's to a file and we come up with 10 files to one disk-pack and 1 file to a floppy. 10 files, with 1 allocated to the Wksp. left us with 9, a fairly "comfortable" number.

Are speech segments long enough?

20 DT's = 120K words

At 8 khz, we have

$$\frac{120 \times 1024}{8000} = 15.36 \text{ seconds of speech}$$

15.36 seconds roughly accomodates a long sentence.

Summary= 1DT = 1 transfer unit = 1x6K words

= 1 buffer load = 1 cylinder

= .77 sec of speech

1 file = 20 DT's

Since "signals" internal to the speech detector are not stored, they must be computed in order to be plotted. If the user desires to view a time frame at the last DT of a file, it would be unreasonable to respond after the time taken to process the entire file -- response time is about 4 - 5 minutes. To circumvent the problem, a approach should enable simulation on a processed file to begin at any point within the file. One such solution is to store state information fo the speech detector in a small header at the beginning of every DT. The values of the information are that just prior to processing the DT.

A 64 word header is judged sufficient for most simple gadgets to be simulated -- these would be insufficient if there were high order filters or FIR filters. The general organization is determined, even though the particular values are algorithm dependent:

word		
1	status	empty, raw, processed
1	leng. of filled header	$= L_1+L_2+L_3+L_4$
L_1	user supplied parameters	
L_2	control variables: pointers, timers, counters, indexes, ... etc.	
L_3	essential state variables	
L_4	redundant state variables	
	empty	

Σ= 64 words

- status values must be the same for all DT's in a file.
- the effective speech segment length is reduced to 15.2 sec, only .16 sec degradation.