Asynchronous Digital Multiplexing

Carlos F. Ojeda

University of Central Florida
ASYNCHRONOUS DIGITAL
MULTIPLEXING

BY

CARLOS FRANCISCO OJEDA

RESEARCH REPORT

Submitted in partial fulfillment of the requirement for the degree of Master of Science in Engineering in the Graduate Studies Program of Florida Technological University, 1972

Orlando, Florida
TABLE OF CONTENTS

LIST OF ILLUSTRATIONS ............................................................... iv

INTRODUCTION ................................................................................ 1

Chapter

I. ASYNCHRONOUS DIGITAL MULTIPLEXING ........................................ 3

   Introduction .................................................................................. 3
   Systems Presently Used ................................................................... 3
   Advantages of Asynchronous Transmission ...................................... 4
   Types of ADM Systems .................................................................. 5
   Asynchronous Digital Multiplexer System ....................................... 7
   Input Elastic Stores ....................................................................... 9
   Output Elastic Stores ................................................................... 12
   Format Selection .......................................................................... 12
   Summary ...................................................................................... 16

II. ADM DESIGN CONSIDERATION .................................................... 17

   Introduction .................................................................................. 17
   Overhead Command and Bit Integrity Analysis ............................... 17
   Deviation Handling Capability .................................................... 21
   Synchronization ............................................................................ 22
   Analysis of the Time to Synchronize ............................................ 24

III. EXPERIMENTAL RESULTS ......................................................... 27

IV. CONCLUSIONS ............................................................................ 29

APPENDIX

   Smoothing Buffer Techniques ....................................................... 31

LIST OF REFERENCES ....................................................................... 40
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Block Diagram of an ADM System</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>(Stuff Condition)</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Block Diagram of an ADM System</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>(Spill Condition)</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Input Elastic Store</td>
<td>11</td>
</tr>
<tr>
<td>4.</td>
<td>Output Elastic Store</td>
<td>13</td>
</tr>
<tr>
<td>5.</td>
<td>ADM Format</td>
<td>15</td>
</tr>
<tr>
<td>6.</td>
<td>Detailed ADM Format</td>
<td>18</td>
</tr>
<tr>
<td>7.</td>
<td>Complete PN Sequence Generator</td>
<td>23</td>
</tr>
<tr>
<td>8.</td>
<td>Frame Synchronizer Block Diagram</td>
<td>25</td>
</tr>
</tbody>
</table>
INTRODUCTION

For some time industry has been working on the feasibility of a complete digital transmission system to supply the increasing demand in communications. There are several reasons for the change from the classical analog transmission. Among them:

1. Digital communication can supply many channels of information at one time.

2. Present technology advances together with more readily available integrated circuits make digital communication more attractive.

3. Distant communications are facilitated by digital repeaters.

4. There are some savings in communication bandwidth, etc.

"Eventually both the military and commercial communications will gravitate toward the all-digital communications system for the major parts of their switched systems (1)." Bell Laboratory engineers have working models of a synchronous digital multiplexer (ADM) which include time division multiplex circuits that interleave streams of digital pulses, representing several different signals into one data stream for transmission. They have experimented with various types of communication signals including telephone, facsimile, television and digital data (2).

This study primarily deals with asynchronous digital multiplexing, subsequently referred to as ADM. The use of asynchronous adaptive sampling for multiplexing several channels of data over a
digital link is examined. This method has proved to be an accurate and efficient way to transmit data. System analysis and design consideration are described. The author has demonstrated the potentials of such a system with a working prototype.
CHAPTER I

ASYNCHRONOUS DIGITAL MULTIPLEXING

Introduction

Time division multiplexing of digital signals from unrelated sources, except for a very close frequency of operation, is a classical problem in digital communication. The transmission of digital data over relatively long distances is normally accomplished by sending the data in serial form through a single channel generally at higher bit rates for ease of transmission. To interleave various data channels for communication over a single link, it is necessary to make all of the data rates compatible, since the multiplexer master clock will sample each channel at a precise time. This process of equalizing the channels bit rate is termed synchronization. By proper synchronization techniques it is possible to multiplex sources derived from independent clocks varying slightly from a nominal bit rate.

Systems Presently Used (2)

Master Clock Method - The most obvious approach is to transmit a master clock to all locations and therefore force every multiplexer and demultiplexer to operate at exactly the same frequency. This system has some undesirable features. Since one clock provides the necessary timing to all units "its distribution system must be extremely reliable - protected by redundancy against technical failures as well as man-made or natural disasters (2)." Also, since precise relative
phasing must be maintained on all data streams entering the multiplexer, and delay differences encountered in cables length and changes in temperature must be overcome by some sort of "elastic delay" circuitry. This method is also called "Frequency Lock System" or "Synchronous Transmission."

Phase Averaging Method - This technique uses the fact that a two-way system of communication has pulse streams entering and leaving every terminal. Therefore, it is possible to establish a reference phase for each location which is the average of all phases entering that terminal. "If each location transmits the same phase or frequency to all other connected locations, it can be shown that the resulting reference frequencies established at the various locations are identical (2)." This approach is not well developed at present and little information about it is found in the literature.

Asynchronous Transmission Method - A third technique, which is the subject of this paper, is to add or delete bits in such a way as to make all the incoming data streams match a fixed rate set by the multiplexer clock.

Advantages of Asynchronous Transmission

For most applications, multiplexer and demultiplexer equipment complexity is minimized if asynchronous transmission is used. The fact that clock is not required to be transmitted with the data accounts for most of its simplicity. Clock information is necessary in synchronous transmission to locate data bits in the receiver. The asynchronous transmission offers a more reliable system since it is not vulnerable to a complete black out because of a master clock failure. The
tremendous amount of clock cabling is eliminated in asynchronous transmission, making it more cost attractive and less environment dependent. It is necessary to point out that for asynchronous transmission, data synchronization must be performed. This in turn decreases the efficiency of the system since a portion of the format must be allocated to synchronization data.

Types of ADM Systems

In general there are two ways to achieve rate compatibility of various independent data channels. They are called "Word Stuffing Techniques" and "Pulse Stuffing Techniques."

In Word Stuffing the "basic idea is to group the transmitted bits into words called data words. The data words are formed for transmission and are not related to any word structure that may exist in the customer's data stream (3)." Thus, transmission rates can be adjusted within the network by inserting or deleting stuff words.

In Pulse Stuffing Techniques the same principle holds except that instead of occasionally inserting or deleting stuff words, pulses are arranged to be inserted or deleted to make up for the frequency difference. Pulse Stuffing Techniques can be further divided in two basic ways, "Pulse Stuffing Synchronization" and "Pulse Spilling Synchronization."

In the first mode the master clock of the multiplexer is chosen to be higher than all the channels bit rates. Then the input data rates are increased to match the multiplexer bit rate by adding extra dummy bits to the input as often as required. These extra pulses
carry no information and are placed in known locations in the format. The information on whether a particular one of these data locations contains real data or dummy data is encoded into a overhead channel in the format.¹ This information contained in the overhead channel is then used in the demultiplexer to ignore the dummy data bits from the data stream.

In the Pulse Spilling Synchronization the input data rate of any channel is higher than the multiplexer bit rate. The multiplexer channel rates are decreased by extracting input data bits as needed and placing them in certain known locations in the overhead channel. The presence or absence of data in the overhead channel is encoded and transmitted also as part of the overhead channel. Thus, the information contained in the overhead channel is then decoded in the demultiplexer and used to restore the data bits back into the proper channel.

A particular system can be designed to operate with Pulse Stuffing Synchronization only, Pulse Spilling Synchronization only or with both modes simultaneously. This last condition is perhaps the major contribution of this report since no attempt has been made before to utilize both stuffing and spilling synchronization in one system.

¹Overhead channel is a channel reserved to carry information pertaining to stuff or spill operations.
Asynchronous Digital Multiplexer System Structure

A basic ADM system consists of:

1. Input channel electronics (input elastic store, data register, stuff or spill detector).
2. Multiplexer main frame.
3. Demultiplexer main frame.
4. Output channel electronics (output elastic store, data register, smoothing buffer).

Figure 1 represents a very simplified block diagram of a system with Pulse Stuffing Synchronization with added bit signaling (4). The digital input data to channel 1 with bit rate $f_1$ is written into a buffer memory located in the channel electronics. The multiplexer fixed bit rate is labeled $f_{\text{fix}}$, therefore, for Pulse Stuffing Synchronization $f_{\text{fix}} > f_1$. The multiplexer is shown as a commutator that combines all the data channels plus overhead and synchronization channels on each cycle of the commutator into a single data stream output. In this manner, all synchronized input signals along with overhead and framing information can then be time division multiplexed by interleaving the pulses from each source. The demultiplexer reverses the multiplexer functions. Framing information is first extracted and used to divide the high rate signal into its synchronized components. The overhead information identifies the location of the stuffed slots in each synchronized bit stream so that they can be removed from the data output. Finally, the signals out of the demultiplexer are replicas of the original signals applied to a distant multiplexer.
Fig. 1.—Block Diagram of an ADM System (Stuff Condition)
Figure 2 represents a simplified block diagram of a system with Pulse Spilling Synchronization. The operation is very similar to the Pulse Stuffing Synchronization case, except that this time $f'_1 < f_1$.

**Input Elastic Stores**

Each channel electronics contains a buffer memory called Input Elastic Store (5). This buffer memory allows time slots to be added or deleted at a rate equal to the difference between the synchronous and asynchronous bit rates.\(^2\) The position in the buffer store from which data is read out is compared to the position into which the data is read in.\(^3\) Refer to Figure 3. If the position of the data read out gets too far ahead of the data read in, a stuff command is generated and when the time comes to service that channel, the command inhibits one of the multiplexer clock pulses, thus preventing read out of the memory. In this manner the multiplexer is arranged to skip over occasional time slots so as to make up the frequency difference. If the data read into the elastic store is higher than the multiplexer rate, when the data read in gets too close to the read out point a spill command is generated in the channel electronics of that particular channel. The spill command is then sent to the main frame of the multiplexer. When time comes to service

\(^2\)The terms asynchronous bit rate and synchronous bit rate are defined here to denote the bit rate of the signals before and after it has been processed by the channel electronics.

\(^3\)This difference is normally referred to as the stuff rate or spill rate.
Fig. 2.--Block Diagram of ADM System (Spill Condition)
MUX = Multiplexer

Fig. 3.--Input Elastic Store
this channel, an extra data bit is read out of the channel electronics and placed in the overhead channel.

Output Elastic Stores

The Output Elastic Store in the demultiplexer in a way performs the inverse operation of the Input Elastic Store; refer to Figure 4. Since the extra dummy bits were removed and extra data pulses were added by the demultiplexer main frame, the channel data output has discontinuities. These discontinuities can be considered as jitter. The jitter will accumulate very rapidly on signals passing through tandem multiplexer units. Therefore, to reduce the amount of jitter in the output data stream the Output Elastic Store is combined with a smoothing buffer network. In the demultiplexer channel electronics the data bits are written into the buffer memory by the demultiplexer clock. The information is read out of the memory by an oscillator whose frequency is voltage controlled connected in a phase-locked loop (4). The loop acts as a low pass filter to the discontinuity jitter and generates a frequency average of the input bit rate and thus the original frequency of the signal is restored. The amount of jitter removal is a function of the loop parameters. Some design aspects of smoothing buffering techniques is found in the Appendix.

Format Selection

ADM format selection is one of the most important aspects to be considered. By choosing the appropriate format, the majority of the problems encountered in asynchronous transmission can be reduced.
DEMUX = Demultiplexer

Fig. 4.—Output Elastic Store
The parameters affected the most are:

1. Possibility of tandeming various ADM's systems.
2. Amount of discontinuities in the data stream.
4. Efficiency of the system.
5. Mean time of loss of bit integrity.
6. Frequency deviation handling capability.

The first three points call for a format in which each channel is evenly distributed. The distributed overhead format shown in Figure 5 eliminated all discontinuities due to inserting overhead bits since the overhead appears as another data channel. This is in contrast to the case where fast acquisition would demand that each data channel contain overhead information. Pulse Stuffing and Pulse Spilling Synchronization also helps in minimizing discontinuities since the stuffing and spilling operation is only one bit and that is the smallest discontinuity possible. Large discontinuity gaps are the major drawbacks of Word Stuffing Synchronizations.

The efficiency of the system requires minimal number of channels assigned to overhead information and synchronization. This greatly depends on the performance desired and is a function of the time to acquire synchronization and the amount of disturbances present in the link between multiplexer and demultiplexer.

---

4 Efficiency is defined as the ratio between the number of data bits and the total number of bits per frame.

5 The loss of bit integrity is defined as erroneously decoding the stuff or spill overhead command in the demultiplexer.
Fig. 5.—ADM Format
In the area of "mean time" of loss of bit integrity, there is a close relation between the amount of overhead coding necessary and the channel error rate. To minimize the loss of bit integrity the overhead channel is normally redundantly coded to provide a degree of immunity to errors introduced during transmission. Some error correcting codes like Gray codes or Hamming codes or simply redundancy are often used. The frequency deviation capacity is naturally very dependent on the format structure and the way to compute it is presented in the next chapter.

Summary

A very useful and powerful technique for digital transmission has been described. Its advantages and structure were discussed step by step. Various format structures were studied by the researcher until a suitable format for simultaneous stuff and spill synchronization was chosen and briefly presented. In the next chapter a more detailed explanation of the format performance and design considerations are given to supplement this presentation.
CHAPTER II

ADM DESIGN CONSIDERATIONS

Introduction

The synchronization principles described in the previous chapter are applied here to design an experimental multiplexer system. Even though a specific system is being considered, the same methods are applicable to a variety of digital transmission systems.

Overhead Command and Bit Integrity Analysis

The techniques thus far described utilizes stuffed and spilled pulses as means of achieving synchronization. Therefore, some way of identifying these fill pulses must be used, so that the receiving circuits may discard them or replace them to regenerate the proper pulse train. As the transmission takes place, errors may occur due to disturbances in the link, giving rise to faulty identification of the fill bits and resulting in loss of bit integrity from input to output. In order to minimize the erroneous identification some encoding of the overhead information is necessary. The overhead channel corresponding to the format shown in Figure 6 employs redundancy encoding by repeating each bit three times. The decoding is done by majority detection. The format has 16 columns and 16 rows. Column 16 is used for synchronization and column 8 is the overhead channel. Since there are 14 data channels left, 4 bits of binary address are necessary to address any one channel. Each binary bit is repeated 3 times which totals 12 bits. One bit is
Fig. 6.—Detailed ADM Format
used to describe if a stuff or spill command took place; this bit repeated 3 times makes a total of 15 bits. The 16th bit is the location of the spilled data bit. An error in this bit does not affect bit integrity so it is not protected.

In majority decoding an error occurs when 2 or all 3 bits are in error. Then the probability $P_e$ of this happening can be computed from the formula for the probability of exactly $k$ successes in $n$ repeated trials. For a bit error probability $p$ this formula is:

$$P_e (x = k) = \binom{n}{k} p^k (1 - p)^{n-k} \quad (k = 1, 2, \ldots, n)$$

summing from $k = 2$ to $k = 3$

$$P_e = \sum_{k=2}^{3} \binom{3}{k} p^k (1 - p)^{3-k}$$

then

$$P_e = 3 p^2 (1 - p) + p^3$$

for small $p$, in the order of $10^{-3}$ or less, $P_e$ can be approximated to

$$P_e = 3 p^2$$

Since the overhead channel has 5 bits of information then the probability of falsely decoding the 15 bit word ($P_w$) is then

$$P_w = 5 P_e = 15 p^2$$

The average number of code words $<N_w>$ that elapse on or before an error occurs, can be expressed as
\[ <N_w> = P_w + 2 \left(1 - P_w\right) P_w + 3 \left(1 - P_w\right)^2 P_w + \ldots \]

This expression can be reduced to
\[ <N_w> = \frac{1}{P_w} \]

The total number of bits between errors corresponds to a major frame, that is 256 bits; then the mean time in bits \(<T_e>\) between the occurrence of these errors is found by
\[ <T_e> = \frac{16 \text{ (bits/row)} \times 16 \text{ (channels)}}{P_w} = \frac{256}{15p^2} \]

assuming a bit error probability \(p = 10^{-5}\)
\[ <T_e> = 17 \cdot 10^{10} \text{ bits} \]

considering the bit rate to be 10 M bits, the average time to loss of bit integrity is
\[ <T_b> = \frac{17 \cdot 10^{10}}{10^7} = 4 \text{ hours and 43 minutes} \]

\[ N_w = P_w \{1 + 2 \left(1 - P_w\right) + 3 \left(1 - P_w\right)^2 + \ldots \} = P_w \frac{d}{dx} \left(1 - P_w\right) + (1 - P_w)^2 + \ldots \] = \[ P_w \left(1 - \left(1 - (1 - P_w)\right)^2 \right) = \frac{1}{P_w} \]

\[ <T_e> = (\text{Total number of bit per major frame}) \cdot \left(\frac{256}{P_w}\right) \]
Deviation Handling Capability

If all channels make the same identical demands for overhead access, the equation for the deviation can be written as:

\[
\begin{align*}
\text{Max. rate (spill condition)} &= f + \Delta f = \frac{n-m+s}{f} \\
\text{Min. rate (stuff condition)} &= f - \Delta f = \frac{n-m-s}{f}
\end{align*}
\]

where

\[f = \text{center frequency (multiplexer master clock rate)}\]
\[\Delta f = \text{deviation from center frequency}\]
\[m = \text{total number of overhead bits in a frame including overhead channel and synchronization channel}\]
\[n = \text{total number of bits per frame}\]
\[s = \text{maximum number of bits which can be stuffed or spilled in a frame. In this case, } s = 1 \text{ since single stuffing techniques are considered}\]

after some manipulations,

\[
\frac{\Delta f}{f} = \frac{1}{n-m}
\]

For the format under consideration \(n = 256, m = 32\), so the deviation rate is

\[
\frac{\Delta f}{f} = \frac{1}{224} \text{ or } .446\%
\]

This means that a system design around this format structure can handle 14 independent data sources, each within .446% deviation from the multiplexer bit rate.\(^8\)

\(^8\) This computation can be better understood by observing that only one channel can be serviced per each major frame. If only one channel needs to be serviced and all the rest run synchronously, the rate of service in the limit is every 16 bits. So in this case,

\[
\frac{\Delta f}{f} = \frac{1}{16} = 6.25\%. \text{ If } N \text{ channels need service in the limit } \frac{\Delta f}{f} = \frac{1}{16 N}, \text{ for } N = 14, \frac{\Delta f}{f} = \frac{1}{244} = .446\%.\]
The efficiency of the system can be calculated as follows:

\[ \eta = \frac{n-m}{n} = \frac{256-32}{256} = 87.5\% \]

Synchronization

In asynchronous transmission clock information is not transmitted with the data. Therefore, in order for the demultiplexer to properly distribute the message over to the corresponding channels, some means of synchronization must be provided with the data. The format taken as model reserves channel 16 for synchronization data. This channel contains a complete P. N. sequence. By "complete" it is meant that the all zeros state is added to the sequence. A P. N. sequence provides good synchronization properties and very easy implementation. Figure 7 depicts the basic circuit of a complete P. N. sequence generator.

The 16 bit P. N. sequence is generated by a 4 bit feedback shift register coder. The most important factor in this approach, is that when any consecutive 4 bits out of any column are forced into an identical 4 bit feedback shift register coder, if the coder is then allowed to free run, it will generate an identical P. N. sequence in the proper phase. The search is done on a column by column basis. Starting with any column, the first four bits of the column are forced into the shift register and then allowed to run. The coder output is compared with the sequence of bits of that particular column. If the test proves correct then the P. N. generator is gated to provide frame

---

9 A pseudo-random sequence is usually called a P. N. sequence. A P. N. sequence generator is also called maximum length counter.
Fig. 7.--Complete PN Sequence Generator
synchronization to the demultiplexer. If the test proves "wrong column", the system repeats the process in the next column. As described, every time a new column is searched, the system must wait 4 consecutive bits to be loaded into the coder. In order to speed the synchronization time a number of buffers could be used to store bits of the incoming columns so each time a new column is tested 3 of the 4 bits needed are already stored and ready to be loaded into the coder at the same time the fourth bit is read in. This procedure is shown graphically in Figure 8.

Analysis of the Mean Time to Synchronize

Assumptions made:

The system outputs data continuously and the probability of rejecting the right column is extremely small.

Definitions:

\[ Q = \text{number of rows needed to initialize the local P. N. sequence generator} \]
\[ \langle C \rangle = \text{average number of columns searched before reaching the proper column} \]
\[ \langle S \rangle = \text{average number of rows needed to reject a wrong column} \]
\[ N = \text{number of bits per row} \]
\[ \langle B \rangle = \text{average number of bits needed to find sync} \]

With these definitions stated, the following expression can be written:

\[ \langle B \rangle = \langle C \rangle Q N + \langle C \rangle \langle S \rangle N \]

Where the first term corresponds to the time to initialize the P. N. generator and the second is the time to reject a wrong column.

For an even number of bits per row and even number of columns, the average number of columns searched before the proper one is
Fig. 8.—Frame Synchronizer Block Diagram

DEMUX = Demultiplexer
\((N + 1) \left( \frac{N}{2} \right) \), and since each column has the same probability, then

\[
< C > = (N + 1) \left( \frac{N}{2} \right) \left( \frac{1}{N} \right) - 1 = \frac{N + 1}{2} - 1
\]

For a system which rejects a location if 15 agreements in a row are not achieved, \( S \) is given by

\[
< S > = \sum_{i=1}^{N-1} \frac{1}{2^i}
\]

If the upper limit of the summation were to go to infinity, \( S \) would converge to 2. So as an approximation

\[
< S > \approx 2
\]

then

\[
< B > = \left( \frac{N + 1}{2} - 1 \right) (Q + 2) N
\]

By adding storage as shown in Figure 8, it is possible to eliminate initialization time after the first initialization, so \( < B > \) can be reduced to

\[
< B > = < C > \cdot < S > N + N (Q + < S >)
\]

or

\[
< B > = \left( \frac{N + 1}{2} - 1 \right) 2 N + N (Q + 2)
\]

for \( N = 16 \) and \( Q = 4 \)

with storage \( < B > = 336 \) bits

without storage \( < B > = 720 \) bits
CHAPTER III

EXPERIMENTAL RESULTS

The synchronization principles described previously have been applied by the researcher to the design of an experimental multiplexer system. The system was built and tested. The bit rate was chosen to operate at medium speed 10 Mb/s, utilizing regular TTL logic. A maximum of 14 data channels were implemented. A suitable format able to accommodate pulse stuffing and pulse spilling synchronization was developed.

The ability to organize the bit stream in such a way to accommodate a wide range of input signals was achieved. In a fixed overhead format, the maximum deviation allowed for each channel was measured to be ± .4% of the multiplexer nominal bit rate, in agreement with the calculation. In a random access priority the maximum deviation tracked, provided no other channel asked to be serviced, was 6.25% as expected. Only one minor drawback was found in the random access mode. When a channel asks to be serviced, if no other channel is requesting access, some period of time elapses before the actual stuffing or spilling operation takes place; therefore, in the very next frame the same channel since it has not been serviced yet will ask for access. Up to three stuff pulses in a row were observed adding unnecessary jitter.

10 By fixed overhead format it is meant that each channel has equal overhead access capability.
to the data but not in any amount to cause any problem. A very successful jitter reduction circuit was implemented in the smoothing buffer section. A phase locked loop acting as a low pass filter removed the high frequency jitter component. Although it does not remove very low frequency jitter, this component does not significantly impair the data.

One of the areas not very well developed in digital transmission analysis is the problem of jitter build up, when large numbers of multiplexers are arranged in tandem. One of the main purposes in building a prototype system was to investigate the behavior of repeaters. To simulate a number of repeaters, a signal was routed through a multiplexer to demultiplexer channel. The demultiplexer channel output was used as another channel input in the multiplexer and this repeated for all 14 channels. Although an increase in phase excursions of the signals at lower rate was observed out of the last repeater, it did not cause any fundamental problems and an error free data was obtained.

Even though the prototype system is quite flexible, all the components used in the design are easily obtained and the unit has proven to be very reliable.
CHAPTER IV

CONCLUSIONS

The feasibility of processing various independent digital data sources into a single stream for transmission was established by this effort. Some new approaches in the area of synchronization and format structure were developed and implemented.

The major conclusions gained could be summarized as follows: Bit Stuffing and Bit Spilling Techniques are a very efficient and flexible method of achieving data synchronization, data bit interleaving proved to be a very good approach when deciding an ADM format.

A very fast and accurate method of frame synchronization using pseudo-random sequences was tested satisfactorily. It was also shown that accumulated phase jitter can be minimized to negligible level.

The design philosophy proved to be an acceptable solution to many problems.
APPENDIX

SMOOTHING BUFFER TECHNIQUES
Loop Analysis

Consider the block diagram of the PLL in Figure 2A. The loop is composed of a phase detector, an active filter element, a VCO, and a programmable divide by N counter. The closed loop transfer function is

\[ H(s) = \frac{\theta_0(s)}{\theta_1(s)} = \frac{K_p F(s) K_v}{1 + \frac{K_p F(s) K_v}{s N}} \]  

or

\[ H(s) = \frac{K_p F(s) K_v}{s + \frac{K_p F(s) K_v}{N}} \]  

A loop filter of the type shown in Figure 1A has a transfer function

\[ F(s) = \frac{\zeta_2 s + 1}{\zeta_1 s} = \frac{R_2 Cs + 1}{R_1 Cs} \]  

then

\[ H(s) = \frac{K_p K_v}{\zeta_1} \frac{\zeta_2 s + 1}{s^2 + \frac{K_p K_v \zeta_2 s}{N \zeta_1} + \frac{K_p K_v}{N \zeta_1}} \]  

From servo theory the characteristic equation of a second order loop becomes:

\[ CE(s) = s^2 + \frac{K_p K_v \zeta_2 s}{N \zeta_1} + \frac{K_p K_v}{N \zeta_1} = s^2 + 2 \delta \omega_n s + \omega_n^2 \]  

\[ ^{12} \text{voltage controlled oscillator} \]
Fig. 1A.—Phase Locked Loop Filter
Fig. 2A.—Block Diagram Phase Locked Loop
where $\delta = $ damping ratio
\[ \omega_n = $ natural frequency of the loop

\[
\omega_n = \left(\frac{K_p K_v}{N \zeta_l}\right)^{1/2}
\]

and

\[
\delta = \frac{\zeta_2}{2} \left(\frac{K_p K_v}{N \zeta_l}\right)^{1/2}
\]

\[
H(s) = \frac{K_p K_v}{\zeta_1} \frac{\zeta_2 s + 1}{s^2 + 2 \delta \omega_n s + \omega_n^2}
\]

\[
H(s) = N \frac{2 \delta \omega_n s + \omega_n^2}{s^2 + 2 \delta \omega_n s + \omega_n^2}
\]

The loop bandwidth is found by letting $s = j\omega$ and setting

\[
|H(j\omega)|^2 = 1/2
\]

this yields to

\[
\omega_{3DB} = \omega_n \left(2 \delta^2 + 1 + \sqrt{(2 \delta^2 + 1)^2 + 1}\right)^{1/2}
\]

The phase detector gain constant $K_p$ can be computed by noting that the phase detector output voltage is proportional to the phase difference between the signals entering the detector

\[
V_p = K_p (\theta_i - \theta_o)
\]
The VCO gain constant can be calculated by knowing or measuring the VCO transfer curve. This is done by plotting the change in output frequency over the change in input control voltage. Then the ratio of Δ frequency change over Δ of input voltage is the VCO gain constant.

**Design Criteria**

Normally a few parameters are chosen before the calculation can be carried out. First the loop bandwidth is determined. This is done by taking into consideration how narrow the loop should be made in order to perform an efficient smoothing task. Other factors like acquisition time and the amount of input buffer needed should also be considered.

Next the loop damping ratio is chosen. This is usually $\delta = .707$ which corresponds to a critically damped system.

With these two parameters we can then determine the loop natural frequency $\omega_n$ from equation (10).

It is also necessary to determine $K_v$ from the VCO transfer function curve and $K_p$ normally from manufacturer data sheets.

With all this data it is then possible to use equations (6) and (7) to determine the loop filter components $R_1$, $R_2$ and $C$, and this completes the design.

**Analysis of Transient Performance**

The error transfer function is

$$E(s) = \frac{s^2}{s^2 + 2\delta \omega_n s + \omega_n^2}$$

If the system is stressed by a ramp $R$ cycles per second the error transfer function in the frequency domain takes the form
\[ E_r(s) = \frac{R}{s^2 + 2\delta \omega_n s + \omega_n^2} \]

if \( \delta = 1 \) for simplicity, the error function in the time domain becomes

\[ E_r(t) = Rte^{-\omega_n t} \]

Differentiating and setting the result equal to zero gives the result

\[ \frac{dE_r(t)}{dt} = Re^{-\omega_n t} - R\omega_n e^{-\omega_n t} = 0 \]

therefore, \( \omega_n t = 1 \) and

\[ \text{Max } E_r(t) = \frac{R}{\omega_n} e^{-1} = 0.059 \frac{R}{\omega_n 2\pi} \]

This result gives an idea of the necessary number of bits the elastic store has to absorb to keep from losing bit integrity for a given frequency change.

**Analysis of Jitter Behavior**

The smoothing buffer loop has a servo transfer function

\[ H(s) = \frac{2\delta \omega_n s + \omega_n^2}{s^2 + 2\delta \omega_n s + \omega_n^2} \]

where \( \omega_n = \text{natural frequency of the loop} \)

\( \delta = \text{damping factor} \)
The jitter could be represented as the sum of an infinite ramp superimposed by a sawtooth waveform. The purpose of the loop is to track the ramp perfectly and ignore the sawtooth. Therefore, the amount of sawtooth tracked will be computed.

Consider a sawtooth of amplitude $A$ and period $T$

$$f(t) = \frac{A}{T} t, \ 0 < t < T$$

and

$$f(t + T) = f(t)$$

The complex Fourier series expansion of $f(t)$ is

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j n \omega_0 t}, \ (\omega_0 = \frac{2\pi}{T})$$

where

$$c_n = \frac{1}{T} \int_{0}^{T} f(t) e^{-j n \omega_0 t} \ dt$$

therefore

$$c_n = \frac{A}{T^2} \int_{0}^{T} t e^{-j n \omega_0 t} \ dt$$

$$= \frac{A}{T^2} \left( \frac{t e^{-j n \omega_0 t}}{-jn\omega_0} \right)_{0}^{T} + \frac{1}{jn\omega_0} \int_{0}^{T} e^{-jn\omega_0 t} \ dt$$

The second term goes to zero since $e^{-jn2\pi} = 1$ for $\omega_0 = \frac{2\pi}{T}$

$$c_n = j \frac{A}{n \omega_0 T} = j \frac{A}{2\pi n} = \frac{A}{2\pi n} e^{j \frac{\pi}{2}}$$
for the case \( n = 0 \)

\[
C_0 = \frac{1}{T} \int_0^T f(t) \, dt = \frac{A}{T^2} \int_0^T t \, dt = \frac{A}{2}
\]

hence

\[
f(t) = \frac{A}{2} + \frac{A}{2} \sum_{n=-\infty}^{\infty} \frac{1}{n} \, e^{j(n \omega_0 t + \pi/2)}
\]

for zero d.c. level and unity amplitude, the trigonometric form of \( f(t) \) becomes

\[
f(t) = -\frac{1}{\pi} \sum_{n=1}^{\infty} \frac{\sin n \omega_0 t}{n} = -\frac{1}{\pi} \sum_{n=1}^{\infty} g_n(t)
\]

From the theory of linear circuits, the total response to \( f(t) \) is the sum of the components outputs \( y(t) \)

\[
y(t) = -\frac{1}{\pi} \sum_{n=1}^{\infty} \{g_n(t)\} \{H(jn \omega_o)\}
\]

The magnitude of the system response to the \( n \)th harmonic of the sawtooth is given by the complex expression

\[
\left| \frac{1}{\pi} g_n(t) \right| \left| H(jn \omega_o) \right| = \frac{1}{n\pi} \left| \frac{2jn \omega_o \omega + \omega_n^2}{-n^2 \omega_o^2 + 2jn \omega_n \omega + \omega_n^2} \right|
\]

where \( \delta = 1 \) and it is assumed \( \omega_o \gg \omega_n \), then the expression can be approximated by

\[
\left| \frac{1}{\pi} g_n(t) \right| \left| H(jn \omega_o) \right| = \left| \frac{2jn \omega_o \omega_n}{(n^2 \omega_o^2) \pi} \right|
\]

Neglecting all but the first harmonic, the magnitude of the phase error is
$|\Delta \phi| \approx \frac{2\omega_n}{\pi \omega_o} \text{ cycles, } \omega_o >> \omega_n$
LIST OF REFERENCES


