NTIA Report 82-111

Proposed Direct Sequence Spread Spectrum Voice Techniques for the Amateur Radio Service

J. E. Hershey

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

Bernard J. Wunder, Jr., Assistant Secretary
for Communications and Information

November 1982

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PROPOSED DIRECT SEQUENCE SPREAD SPECTRUM VOICE TECHNIQUES FOR THE AMATEUR RADIO SERVICE

J. E. Hershey*

General Docket 81-414, Notice of Inquiry and Proposed Rulemaking, proposes allowing the Amateur Radio Service to use spread spectrum techniques in three bands. This report reviews the Docket's proposals and the public's reaction, reviews direct sequence spread spectrum techniques, and proposes (for purposes of further discussion) a direct sequence spread system suitable for voice communications.

Key words: Amateur Radio Service; direct sequence spread spectrum; General Docket 81-414; spread spectrum

1. INTRODUCTION

Title 47, Section 303 of the US Code sets forth some of the powers and responsibilities of the Federal Communications Commission (FCC). Among numerous responsibilities is ensconced the directive:

> ...the Commission from time to time, as public convenience, interest, or necessity requires, shall-
>
> ...
>
> (g) Study new uses for radio, provide for experimental uses of frequencies, and generally encourage the larger and more effective use of radio in the public interest...

On September 18, 1981, the FCC released a Notice of Inquiry and Proposed Rule Making under General Docket No. 81-414 which addressed the concept of Amateur Radio Service (ARS) use of spread spectrum techniques within certain of their frequency bands. The question has sparked a healthy public comment both from neighboring band users who profess to be legitimately concerned over potential interference problems and members of the ARS who are themselves divided on the issue or key parts thereof.

*The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80303.

The issue is an extremely interesting one as no radio technique is surrounded by more mythology than is spread spectrum. This class of modulation techniques was swaddled in secrecy at its inception over three decades ago and still maintains its aura despite numerous open literature examinations and primers. Essentially, spread spectrum is any technique that demands a substantially greater spectral domain than that required by its information baseband. The two most common techniques are frequency hopping (FH) and direct sequence (DS). In an FH system members of a wide range of discrete frequencies are used for short "dwell" times and thus the signal appears to hop about through an extended frequency domain. In a DS system, the relatively narrow spectral baseband is added to a wide spectral pseudorandom (and therefore deterministic or predictable) digital process. The baseband energy is thus "spread" over a wide frequency range.

We have prepared this report in an effort to coalesce the proposed actions and the public's reactions. We have included a number of appendices, written in the style of primers, that explore some of the basic tools of spread spectrum systems. Finally we have also attempted to review one class of spread spectrum systems, the DS biphase technique, and we have suggested an architecture for the ARS. We hope our proposed architecture will prompt interest, thought, and comment to help lead to a viable vehicle for ARS experimentation. We believe that the ARS can make significant contributions to the study of spread spectrum techniques particularly in the Code Division Multiple Access (CDMA) and local networking arenas. The ARS has long been recognized as a responsible and contributing member of the body of radio spectrum users and has been a consistent source of innovation--a valuable national asset for furthering technological development in the radio engineering disciplines.

For the interested reader, there are a number of excellent and easily understandable pieces of literature on spread spectrum. In particular, the following are recommended: Scholtz (1982), Dixon (1976a), Viterbi (1979), AGARD (1973) and Utlaut (1978). There is even a paper on the subject at hand: Rinaldo (1980). For further technical indoctrination see: Holmes (1982), Dixon (1976b), Scholtz (1977), Pursley (1977), and Pickholtz et al. (1982).

2. CHARACTERIZATION OF THE ISSUE

The issue is the Notice of Inquiry and Proposed Rule Making General Docket 81-414. The issue proposes to affect ARS operations in the three frequency bands

50-54 MHz, 144-148 MHz, and 220-225 MHz. This chapter attempts to collate the facts without comment.

2.1 The Present Status of the Three ARS Bands in Question

The presently allowed types of modulation in the three bands (and their sub-bands, where appropriate) are presented below per Section 97.61, Authorized frequencies and emissions, of Part 97 of the FCC's Rules and Regulations.

Frequency Band	Allowed Emissions
50.0 - 54.0 MHz	A1
50.1 - 54.0 MHz	A2, A3, A4, A5, F1, F2, F3, F5
51.0 - 54.0 MHz	A0
144 - 148 MHz	A1
144.1 - 148.0 MHz	A0, A2, A3, A4, A5, F0, F1, F2, F3, F5
220 - 225 MHz	A0, A1, A2, A3, A4, A5, F0, F1, F2, F3, F4, F5

Type A0 emission where not designated may be used for short durations if required for remote control purposes or experimental work. Other exceptions are presented in Section 97.65, Emission limitations, of Part 97. The maximum authorized power input is not to exceed one kilowatt to the plate circuit of the final amplifier stage of an amplifier oscillator transmitter or to the plate circuit of an oscillator transmitter as per Section 97.67, Maximum authorized power, of Part 97.

The above emission designators are described in Table 1 (from the Reference Data for Radio Engineers, 1968, pp. 1-16 and 1-17).

2.2 Brief Comments on the Proposed Changes to the FCC's Rules and Regulations

The proposed changes are recorded in full detail in Appendix D. This present section attempts to review, with comment, the significant aspects of the proposed changes.

The main aspect, indeed the crux of the issue, is that the FCC has proposed to allow the ARS the use of spread spectrum techniques in three frequency bands: (1) 50-54 MHz, (2) 144-148 MHz, and (3) 220-225 MHz. What is not addressed, however, is guidance relative to spread spectrum genre. The FCC's proposed rules will admit frequency hopping, direct sequence, time hopping or hybrid spread spectrum techniques. Further, a method of synchronization is not specified nor

Table 1. Abbreviated Description of Emission Designators

EMISSION DESIGNATOR	MAIN CARRIER MODULATION	TRANSMISSION TYPE	OTHER DETAILS
A0	AMPLITUDE	UNMODULATED	
A1	AMPLITUDE	ON-OFF KEYED TELEGRAPHY (NO AUDIO FREQ. MODULATION)	
A2	AMPLITUDE	ON-OFF KEYED TELEGRAPHY (AMPLITUDE MODULATED AUDIO FREQ. OR FREQS.)	
A3	AMPLITUDE	TELEPHONY	DOUBLE SIDEBAND, FULL CARRIER
A4	AMPLITUDE	FACSIMILE (MODULATION OF MAIN CARRIER DIRECTLY OR BY FREQ. MODULATED SUBCARRIER)	
A5(C)	AMPLITUDE	TELEVISION	VESTIGIAL SIDEBAND
F0	FREQUENCY*	UNMODULATED	
F1	FREQUENCY*	TELEGRAPHY (FSK, I.E., ONE OF TWO FREQS. AT ANY INSTANT)	
F2	FREQUENCY*	TELEGRAPHY (ON-OFF KEYING OF A FREQ. MODULATING AUDIO FREQ.)	
F3	FREQUENCY*	TELEPHONY	
F4	FREQUENCY*	FACSIMILE BY DIRECT FREQ. MODULATION OF THE CARRIER	
F5	FREQUENCY*	TELEVISION	

*Can also be phase modulation.

suggested. Rather, the method of synchronization is merely to be detailed in the station's log along with the other signal parameters of concern such as center frequency, code rate, chip rate, and others. Indeed, the only technical requirements directly affecting the pseudorandom portion of the spreading mechanism concern those "pseudorandom sequences [that] may be used to generate the transmitted signal." The FCC specifies that only "binary linear feedback shift register[s]" of particular lengths and modulo-two added feedback connections be used.

The second, and the only other major aspect, concerns station identification. The FCC proposes that "identification in telegraphy shall be given on the center frequency of the transmission." Identification of a transmitter is a classical concern of all public radio usage. The FCC's intent is clear but the specifics are somewhat murky. Does the spread spectrum station identify itself using spread spectrum modulation or by a conventional, more nearly universal, narrowband transmission? If the former, can we expect someone who is not spread spectrum equipped to identify an interfering transmitter? If the latter, would it not be better to agree on a common "check-in" frequency vice the center frequency of the spread spectrum transmission?

All things considered, it seems, in sum, that the FCC's proposals are novel and in the best interest of the radio arts; however, they appear to lack specificity in some critical areas. The issue is a most complex one as the equities involved touch, at least tangentially, the national security and, more mundanely, the rights of other radio amateurs of the "narrowband persuasion." For these reasons, it is perhaps advisable that any new allowed mode of ARS operation be carefully posited, evaluated, and rigorously specified at least in the early phases of its use.

2.3 Concerns

The following report most of the concerns expressed in the public commentaries submitted to the FCC regarding the General Docket.

2.3.1 FCC Monitoring Capability

Interference: A number of persons have expressed worry that the FCC will have difficulty locating spread spectrum transmitters if they interfere with other users. One commentator stated that because of the "difficulty of detection (of spread spectrum signals)" there would be a concomitant difficulty of discerning the source of interference.

Monitoring Message Content: The concerns here are perceptions that clandestine radios inimical to the national security might spring up and also that other, less exotic interlopers, such as business concerns, might arrogate ARS spectrum by using spread spectrum modulations that are difficult to monitor.
Cost: One respondent stated his belief that proper implementation of the proposals contained in the Docket might lead to outlandish expenditures of funds by the Government.

2.3.2 Interference

The 50-54 MHz Band and Channel 2 (54-60 MHz): A number of respondents profess concern that spread spectrum activities within the 50-54 MHz band might adversely impact the quality of Channel 2 reception.
Locating Interferers: Concern was expressed that the ARS as well as the FCC would have significant difficulties in determining the location and identities of spread spectrum interferers.
Repeaters: There is some concern that spread spectrum activity would adversely affect repeater service especially in the 144-148 MHz band.
Marginal Terrestrial Links: Concern was expressed that low power VHF line-of-sight and diffraction paths should be protected from undue interference related to spread spectrum activity.
Alert Frequency: It was suggested that the spot frequency of 145.695 MHz be kept clear of spread spectrum energy. This frequency is used by the Radio Amateur Civil Emergency Service (RACES) as an alert frequency.
Moonbounce: Much concern was expressed by moonbounce experimenters in the 144-148 MHz band.

2.3.3 Overlay

The concerns here are the questions of practical coexistence of spread spectrum transmissions overlaid on narrowband emissions, i.e., sharing the same spectrum space. One commentator was concerned that the FM capture effect might cause the suppression of either the channelized FM station or the spread spectrum station. The general question of spectrum efficiency, its meaning, and its potential for realization for overlaid systems was raised.

2.3.4 Miscellaneous ARS Sensitivities

Some ARS commentators expressed:

a) the view that the proposed shift register codes were not sufficiently flexible.

b) the view that the spread spectrum privileges should also be extended to Technician Class licensees.

c) the hope that the FCC would permit international spread spectrum experimentation in some cases. This point addresses Article 32 of the Radio Regulations of the International Telecommunication Union (ITU). Paragraph 2732, Section 2, part (1) of the 1982 Edition reads as follows:

> When transmissions between amateur stations of different countries are permitted, they shall be made in plain language and shall be limited to messages of a technical nature relating to tests and to remarks of a personal character for which, by reason of their unimportance, recourse to the public telecommunications service is not justified.

As pointed out in the public commentary, Paragraph 2734 (3) does provide as follows:

> The preceding provisions may be modified by special arrangements between the administrations of the countries concerned.

2.4 The General Issues Surrounding DS Spread Spectrum and the Purpose of This Report

There are five large problem areas that need attention when considering DS spread spectrum usage:

a) spectral efficiency

b) the overlay problem

c) the near/far problem

d) synchronization

e) user identification.

We will briefly consider these five items and then outline what this report will attempt to cover and contribute towards the resolution of some of these areas.

Spectral efficiency is a measure of the maximum number of users that can simultaneously use a portion of spectrum. (It is often defined relative to a geographical area.) The DS spread spectrum method we propose in Chapter 3 is designed to exhibit some degree of spectral efficiency. This is proposed to be attained through the use of Gold codes as the spreading sequences. A Gold code family is a set of sequences of bits that possesses bounded crosscorrelations between the family members. The crosscorrelations are bounded in the absolute value sense and thus are independent of bit sense. The Gold code family used has sequences whose periods are half as long as a data baud of the DATA stream; the narrowband information which is to be spread.

The overlay problem addresses the compatibility issues involved with using spread spectrum and narrowband, channelized communications within the same spectrum space in the same, or in a closely neighboring, geographical area. Juroshek (1979) and others have studied this problem, although the DS spread spectrum system that they posited used a pseudorandom spreading sequence vice a Gold, or other, short periodic code. This will lead to very distinct and important differences that must be accounted for in a similar analysis for the proposed system of Chapter 3. This analysis has not yet been made for our proposed system, but it is, in our judgment, worth recounting what Juroshek determined. Juroshek's analysis indicates that DS spread spectrum overlay with channelized FM communications is not practical as the spread spectrum signals will cause unacceptable interference to the narrowband FM links. It is not worthwhile to be any more than qualitative in this brief recount because of the peculiar spectral differences between Juroshek's DS power spectral density model and that which would result from the system proposed in Chapter 3. We do believe, however, that even the system proposed in Chapter 3 will not overlay well with narrowband communications, i.e., there will probably be interference to some of the narrowband links. However, as we will later recommend, the potential of this interference should not summarily rule out the use of DS spread spectrum techniques by the ARS as the experimental results and experience gained is expected to be of especial benefit to the ARS and to the radio engineering arts. There is a clear need to conduct a thorough study of the overlay problem for the proposed system and also to aggressively pursue other avenues such as the spectrum shaping technique advanced in Appendix C.

The near/far problem is perhaps the most difficult problem facing ARS usage of DS spread spectrum techniques. Simultaneous usage of spectrum in the same geographical area by two or more DS spread spectrum links is possibly only because the spreading codes can be made to exhibit low crosscorrelations. This "processing gain" is easily offset, however, by the power disadvantage that obtains when a transmitter is much closer to a victim's receiver than the victim receiver's transmitter. In many situations the near/far problem is simply not present. The case of multiple earthbound users communicating via DS spread spectrum through a satellite is an example of such a case. In this instance, the users all keep their power levels equal. The fact that the users are all essentially equidistant to the satellite results in equal transmitter powers incident at the satellite. For the ARS, unfortunately, the situation is not so simple. Receivers and transmitters are distributed in helter-skelter relationship and the near/far problem is a cruel reality. The system proposed in Chapter 3 has a provision that should help overcome, at least to a limited degree, the near/far problem. By choosing Gold codes vice a pseudorandom spreading sequence, and by further assigning two (or perhaps more) sequence periods per narrowband data baud, we will leave equally spaced interstices in the power spectrum. Other users can, by a minimal adjustment of their center frequencies, fit their power spectra into the interstices of the interfering transmitter's power spectrum.

The synchronization problem is, in general, a very difficult problem for DS spread spectrum systems. The difficulty is further heightened for those systems that must operate in a jamming environment. The system proposed for the ARS need not, of course, be so rigorous and we have therefore opted to use a classical epoch determination scheme (see Appendix B) and train the receiver's clock so that the receiver can operate open-loop in the post-synchronization phase. This will provide a further degree of system immunity to interference.

The identification problem, i.e., determining the identify of a transmitter, is of critical importance to the FCC and the ARS. We propose an automatic identification mechanism which is resident in the synchronization process. The method proposed is robust, with regard to interference, and simple in deference to engineering complexity and cost.

3. A DS SPREAD SPECTRUM SYSTEM PROPOSED FOR CONSIDERATION FOR ARS USE

3.1 Introduction

In this chapter we present a short review of DS spread spectrum techniques. We follow with a candidate DS spread spectrum that we developed in order that the following advantages accrue.

a) Station identification and monitoring of transmission content can be done with a minimum of effort by the enforcement authority and by the ARS.

b) The near/far problem*, although not eliminated, can be possibly overcome at least for a small number of users.

c) The spectrum can be shared more easily among a small number of users, i.e., there will be a modicum of spectral efficiency.

d) There will be ample latitude for the ARS experimenters to be creative and have the potential to contribute towards the solution of important, contemporary problems and add to the "state-of-the-art."

e) A workable system should be able to be constructed within a reasonable budget albeit with much dedicated labor.

f) The system can be improved to higher and higher levels of quality through investment in the receiver system alone.

Some disadvantages are the following:

a) There is no latitude available for varying the spreading codes. They are limited for two reasons. The first is to provide a degree of spectral efficiency. The second is to ensure that content monitoring can be easily effected.

b) The chip rate of the transmissions must be approximately uniform for all the users.

c) The synchronization and automatic identification architecture is fixed for the same two reasons as per (a).

d) Some special hardware items may have to be commercially developed.

3.2 The DS Spread Spectrum Technique

The simplest block diagram of a DS biphase spread spectrum transmitter is shown in Figure 1. (It is a slight modification to that used in AGARD, 1973, p. 5-7.) What Figure 1 depicts is a relatively slow rate DATA stream; let us

*The near/far problem refers to the problems encountered by a receiver attempting to demodulate a weak spread spectrum signal in the presence of a stronger one.

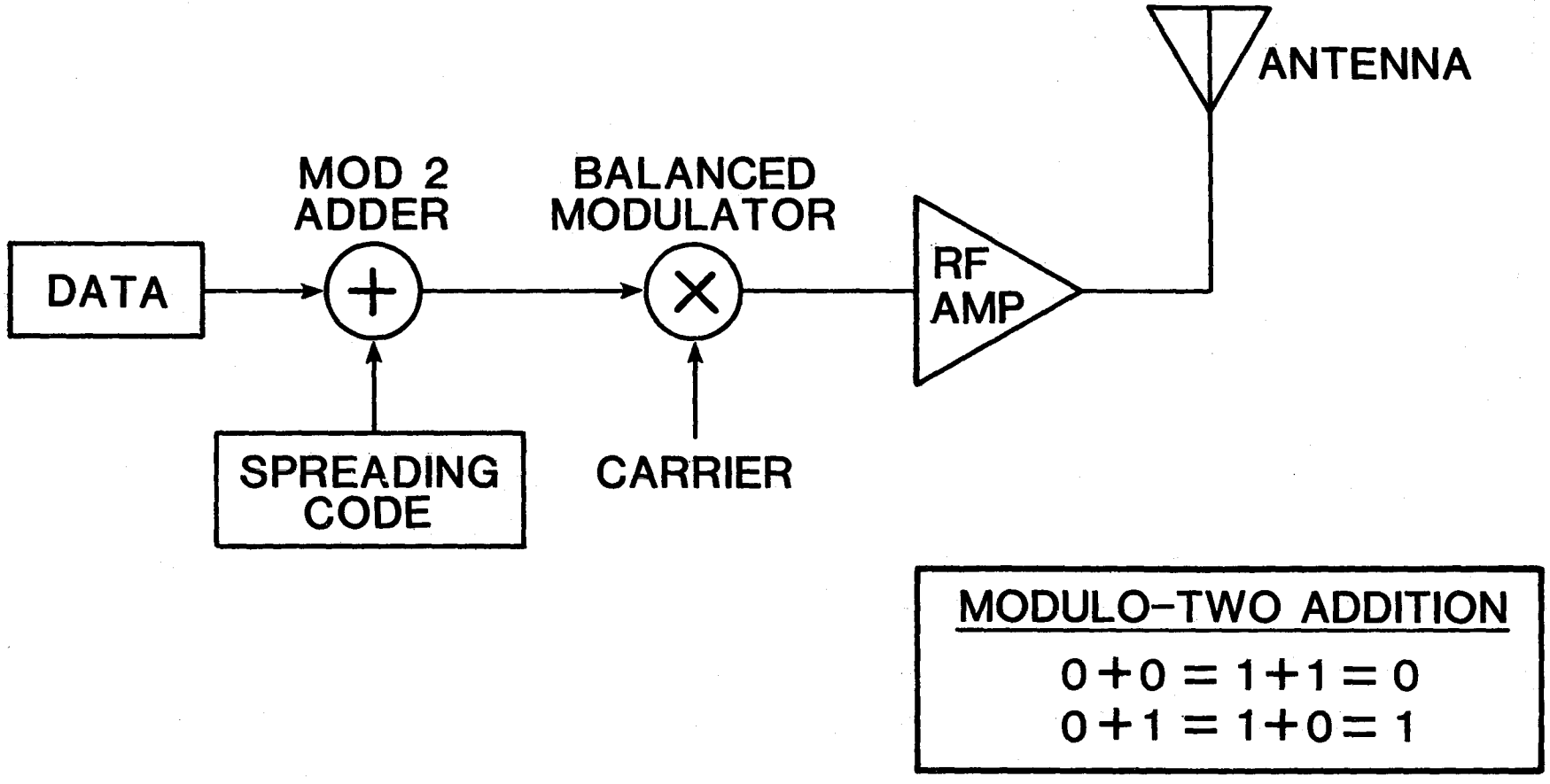


Figure 1. Generic DS biphase spread spectrum transmitter.

assume the data are bauds, zeros and ones, with duration T_b. These data are modulo-two added (exclusive-ored) with a relatively high speed binary spreading code--a stream of zeros and ones, sometimes called 'chips,' with bit durations T_c. The period of the spreading code is denoted by T_p. The sum of the data and spreading code bit streams is presented to a balanced modulator with its other input set to the carrier frequency f_c. The output of the balanced modulator is amplified and radiated.

Let us now examine the above parts in a bit more detail. First, it is desirable to have an integral number of chips per data baud and, further, the chip clock should be phased with the data clock so that the data baud transition times coincide with a chip transition time. To do otherwise would lead to pulses "skinnier" than T_c and thus would cause a wider bandwidth process than is desirable. Second, let us assume that the balanced modulator allows the carrier to pass without a phase change if the digital stream bit into the balanced modulator is a zero and changes the phase by π radians if the digital stream bit is a one. The carrier should be phased with the chip clock in the sense that the carrier will not change phase until near a zero crossing when a chip transition occurs. To do otherwise will also result in needless and wasteful high frequency components. (Notice, incidentally, as Pasupathy (1979) and others point out, this scheme is equivalent to AM modulation of a carrier by a stream of plus and minus ones.) If the spreading code bit stream is independent of the data stream and resembles a random and balanced bit stream (a stream of bits such that the probability is one-half that any particular bit is zero) such as a long m-sequence (see Appendix A), then the normalized power spectral density at the antenna is

$$S(f) = T_c \left[\frac{\sin \pi(f-f_c)T_c}{\pi (f-f_c)T_c}\right]^2 \quad . \tag{1}$$

This power spectral density is depicted in Figure 2. Note that the main lobe is $\frac{2}{T_c}$ Hz wide. Dixon (1976a) relates that 90 percent of the signal power is contained within the main lobe. The main lobe is all we need to demodulate the spread spectrum signal and thus we may filter out the higher lobes before transmitting to prevent their spilling into unauthorized spectrum space.

The receiver for a biphase spread spectrum signal may take a number of forms. Figure 3 depicts two of the most common approaches (Unkauf, 1977). The two schematisms of Figure 3 seem straightforward enough but they contain one component that is usually "easier said than done" and this is the SYNCHRONIZATION MODULE. Listen to what Dixon (1976a, p. 177) has to say about synchronization:

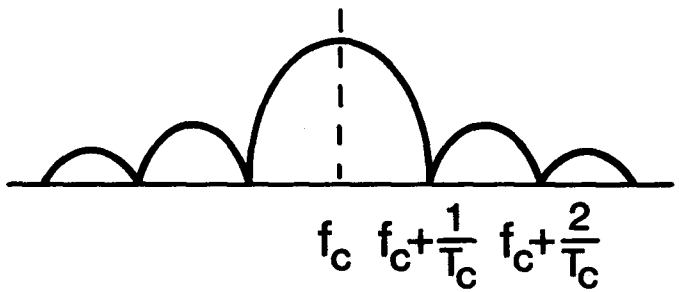


Figure 2. Spectrum shape of a DS spread spectrum signal.

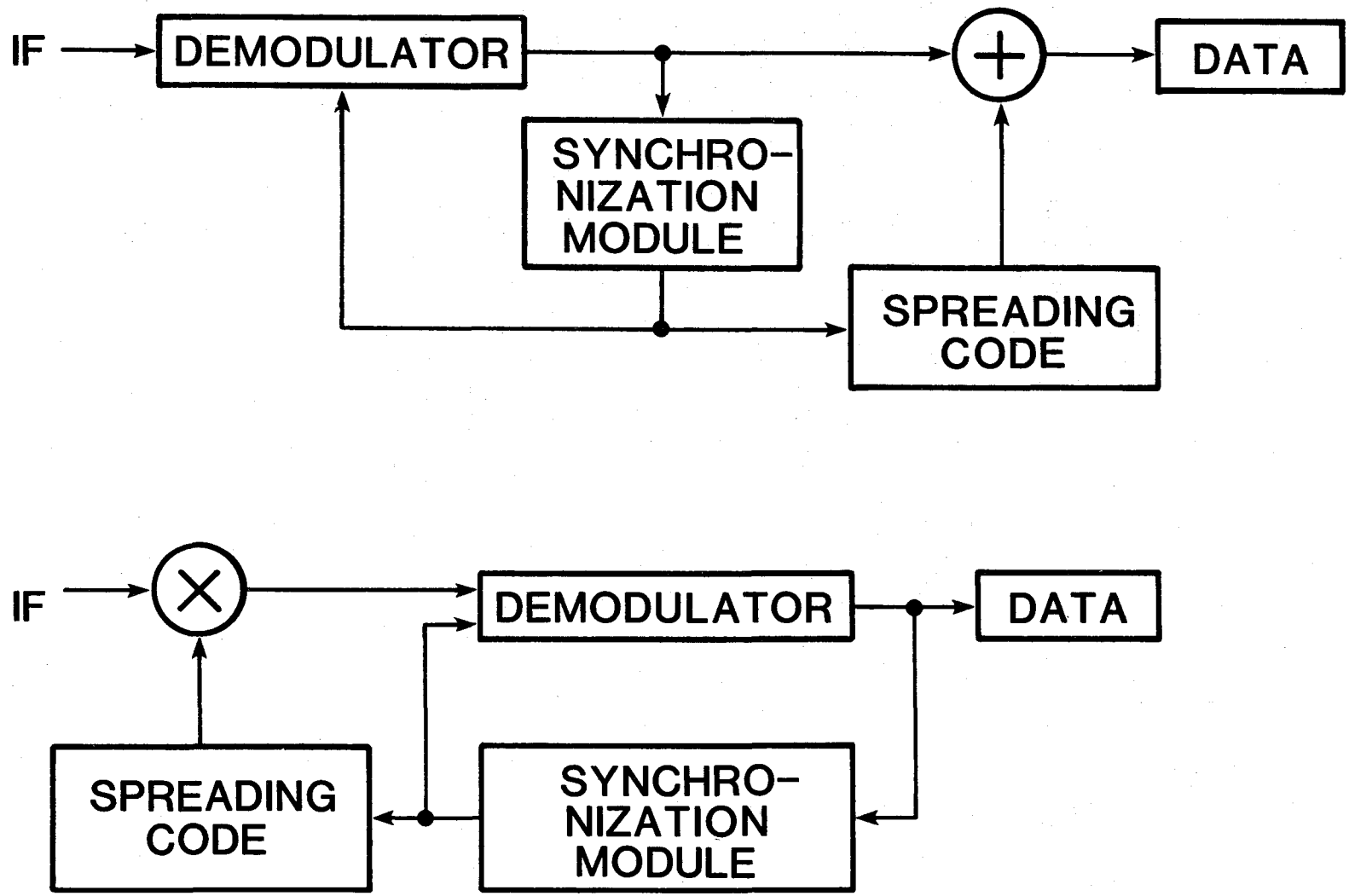


Figure 3. The most common DS spread spectrum receivers.
Top receiver performs postdemodulation despreading.
Bottom receiver performs predemodulation despreading.

"Now we must talk about the hardest part. Throughout this book we have assumed good synchronization (on the part of the code) between transmitters and receivers ... we have assumed that the codes in the systems ... were already synchronized and would remain so.

What an assumption! More time, effort, and money has been spent developing and improving synchronizing techniques than in any other area of spread spectrum systems. There is not reason to suspect that this will not continue to be true in the future."

Synchronization has two components, the coarse and the fine. Coarse synchronization gets the receiver's spreading code synchronized with the transmitter's within one chip time, T_c. Fine synchronization reduces the error to within a small fraction of a chip time and dynamically works to keep it there throughout the reception. This dynamic refinement is accomplished by a feedback loop structure which is easily discerned in the diagrams of Figure 3.

But even before we work on synchronizing the spreading codes, however, we must first be sure we know what the center frequency of our spread spectrum signal is. Recall that a balanced modulator removes (actually it suppresses) the carrier frequency and therefore we cannot use a phase locked loop (PLL) to track the carrier as a PLL requires at least a residual carrier. (For an excellent discussion of this and what is to come see Chapters 4 and 5 of Holmes, 1982.)

What we can use, however, is either a squaring loop or a Costas loop; the loops are equivalent and optimum at low signal-to-noise ratios (Riter, 1969). Figure 4 (from Dixon, 1976a) shows a Costas loop operating on a baseband modulated carrier. As shown, the OUTPUT is $\frac{A}{2}\cos\phi$. When the loop has achieved lock, $\phi \simeq 0$ and $\cos\phi \simeq 1$. What is important to note is that we have recovered the sign of the input process without explicit knowledge of ω and therefore the Costas loop is functioning as a demodulator. Dixon (1976a) notes that there is an ambiguity between zeros and ones. The loop will allow us to extract the difference between the bit at time t and the bit at time t+1 but not the absolute values of the bits and therefore differential schemes such as differential phase shift keying (DPSK) are used in transmitting the data. By observing the shortest period between the transitions we can recover the clock. Now, having recovered carrier and clock, we are ready to work on synchronizing the receiver's spreading code sequence to the transmitter's. We examine various techniques

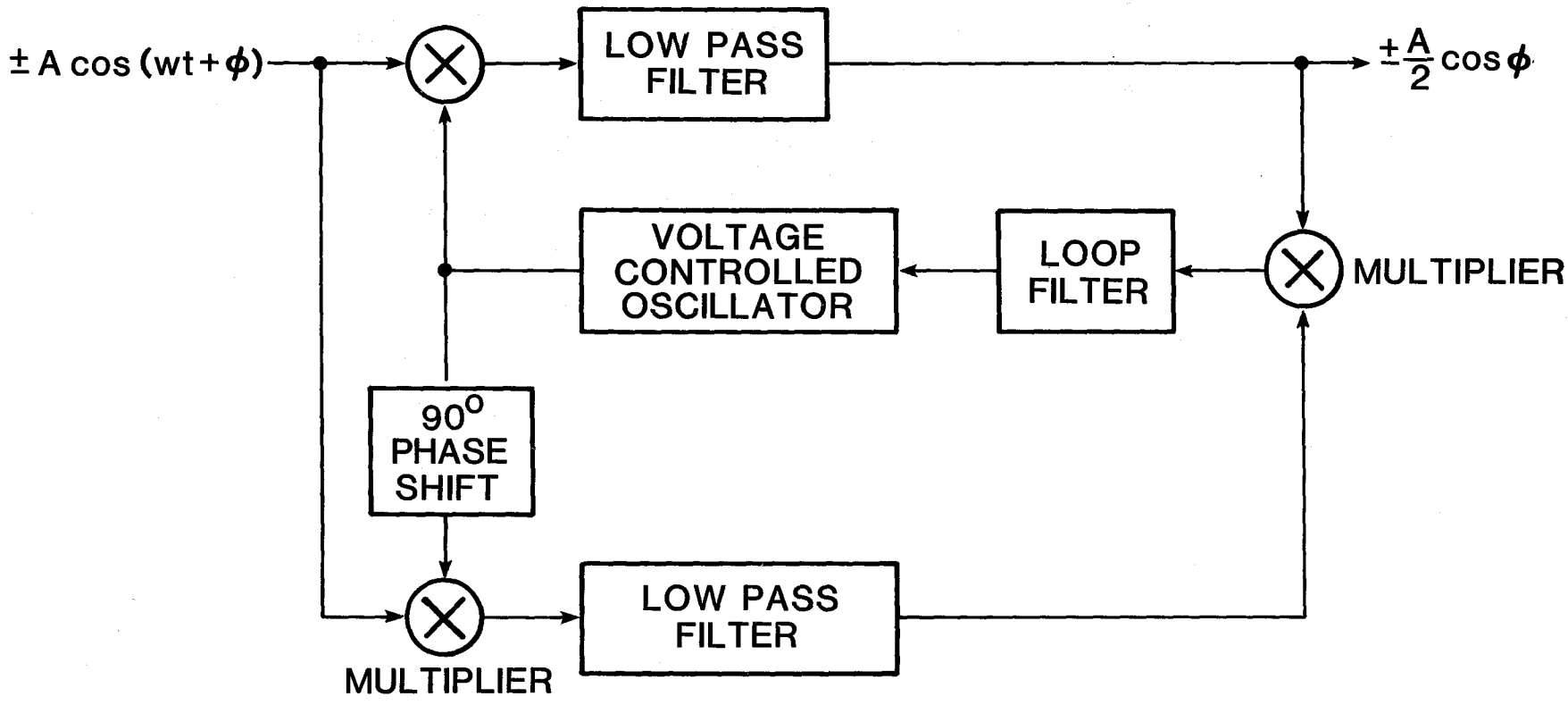


Figure 4. Costas loop demodulator.

for doing this in Appendix B.

3.3 Proposed DS Design

3.3.1 Introduction

There are fundamental decisions that must, or at least should, be made before a DS spread spectrum system is designed; fundamental parameters that must be decided. We are not talking about clock rate, center frequency, or any single, simple parameter but rather the philosophy of the system--what its purpose is to be and how it should be designed to fulfill that purpose. This is a most basic question and it is not well answered, or even asked, in the present case concerning the ARS. The American Radio Relay League, in their comments to the FCC (received 1 March 1982) concerning General Docket No. 81-414, crystallized this thought:

> "At this time, the interest of the amateur community in spread spectrum techniques is primarily experimental. While those techniques are of considerable interest to inquiring amateurs now that advances in technology have brought them within practical reach of individual experimenters, their major advantages do not particularly promote the communications objectives of the Amateur Service. Looking at the matter from the point of view of the amateur qua communicator, message privacy is not a desired feature of amateur communication. To the contrary, as the Commission has noted, techniques which provide privacy raise difficulties in monitoring and enforcement. Selective addressing and multiple access are desirable in certain situations, but may be achieved more easily by other means and with presently authorized modulation techniques. Thus, the primary motivation for amateur utilization of spread spectrum techniques is simply the desire to better understand and develop the concepts which make spread spectrum a useful communications medium. While it is unlikely that these techniques will be widely used in the Amateur Service in the near future, even a modest level of experimentation in this service will significantly expand the body of knowledge about spread spectrum techniques in non-government applications."

The answer to the question of system purpose will deeply affect three areas: synchronizing procedure, spreading code and the ratio T_b/T_p. Let us comment on these items in order.

Synchronizing Procedure: The easiest synchronization method to implement in hardware is epoch determination via a unique word as discussed in the first part of Appendix B. If, however, we are concerned with communications subject to jamming or spoofing, then a unique word system may be very vulnerable. Dixon (1976a, p. 185) puts it very well:

> "With the exception of the vulnerability problem, however, preamble synchronization is by far the least critical, easiest to implement, least complex, and best for all around use."

Spreading Code: Many authors have stated that if the spreading code is an m-sequence or Gold Code or similar linearly generated sequence, then any other suitably equipped party could, after just a modicum of analysis of a small stretch of the code, predict the future spreading code sequence without error. Such knowledge would enable the "outsider" to read the communications and intelligently, and therefore effectively, jam the communications whenever desired.

The Ratio T_b/T_p: It is in consideration of this ratio that the greatest differences in philosophy are reflected. There are in reality two cases. The first displays $T_b/T_p \ll 1$; the second $T_b/T_p \simeq 1$.

One use of spread spectrum techniques is to allow communications to be conducted in secrecy, not just denial of message comprehension by an interceptor, but also denial of the knowledge that communications are indeed taking place. This is sometimes referred to as LPI for Low Probability of Intercept. The acronym is poorly used since in most cases, the communications are intercepted, i.e., gathered in by the antenna and receiver systems. They are just sufficiently buried in the noise that the interceptor does not recognize their presence. Some folks have argued that to more accurately reflect this fine difference one should use the acronym LPR for Low Probability of Recognition. The author believes that it is this type of spread spectrum that most people consociate with DS spread spectrum and for which $T_b/T_p \ll 1$. This is the type of usage that Scholtz (1977) had in mind when he advised:

> "(a) Make sure that the data modulation bandwidth is much larger than ... the reciprocal of the SS [spread spectrum] code modulation's period. ...
>
> [or]
>
> (b) If it is impossible to guarantee data modulation of sufficient bandwidth, then make sure that [the code modulation period] is very large."

What Scholtz is warning us about is a fallout of Fourier analysis. Because our

spreading code is periodic (because it is deterministic) it will, if unmodulated by the DATA stream (e.g., consider that the DATA stream is an unchanging stream of all zeros or ones), result in a process exhibiting a line spectrum rather than a continuous spectrum. The lines in the power spectrum will be separated by a frequency of $1/T_p$. The larger T_p, the closer the lines will be together. Also, as the DATA stream begins to vary and modulate the spreading code, the lines will broaden and thereby move closer together until, in the limit, the continuous power spectral density of (1) obtains.

If, on the other hand, we are not interested in coverture for our communications but rather spectral efficiency, then we may elect to set $T_b/T_p=1$, or a small integer, and employ spreading codes that exhibit excellent crosscorrelation properties. Such procedures are outlined by Pursley (1977) among others.

3.3.2 System Architecture

The system architecture that we are proposing will at first blush appear to be an unusual hybridization of various techniques. As we stated in the Introduction to this chapter, we wanted to design a system that could be realized within a reasonable budget, would allow at least some bandsharing under the near/far problem, allow much latitude for creativity and novel designs (particularly in the receiver system), and allow for easy monitoring and identification of transmitters.

The system we designed is characterized as follows:

- Bi-Phase Shift Keying/Differential Encoding
- Costas Loop Training Sequence
- Epoch Synchronization
- Open Loop Timing
- A Gold Code Family of Spreading Codes with Randomized PPM Coded Selection of Family Members*

*Gold codes are codes of length 2^n-1 which can be easily generated by adding together, term-by-term two m-sequences generated by "preferred" polynomials. The Gold codes have the following properties: (a) a Gold code family has 2^n+1 members (sequences of length 2^n-1) and (b) the (full-period) crosscorrelation R(k) of two Gold code family members satisfies the following

$$|R(k)| \leq \begin{cases} 2^{(n+1)/2} + 1 & , \quad n \text{ odd} \\ 2^{(n+2)/2} + 1 & , \quad n \text{ even and not divisible by 4.} \end{cases}$$

Suitable references are contained in Gold (1967), Gold (1968) and a very helpful step-by-step ("cookbook") approach given in Holmes (1982).

We chose biphase shift keying essentially because it is easy to implement. We will have a few more words to say about this later in the section entitled Implementation. Figure 5 shows the proposed transmitter structure. At the figure's bottom is a sequency flow depicting the transmission preamble (which begins at t=0) and the eventual start of traffic. The preamble starts with a stretch of alternating ones and zeros denoted by A. This is continued until $t=t_1$ at which time a Unique Word (UW) (see Appendix B) is sent. This is followed by τ_1 random bits and another UW. This second occurrence of the UW is followed by another stretch of random bits and another UW and so on until the UW following τ_6 has been sent after which τ_c random bits are sent followed by a final UW which marks the start of traffic. The function of this "kludgey" looking procedure, which is in a sense Pulse (UW) Position Encoding is twofold. It identifies, with relatively strong, i.e., noise-resistant, coding, the transmitter's identity and his spreading code. It does these things by the following conventions:

a) τ_1, τ_2, τ_3, τ_4, τ_5, τ_6 are each 0 to 36 chip times in length and decode according to the following.

τ(chip times)	Meaning	τ(chip times)	Meaning
0	0	18	I
1	1	19	J
2	2	20	K
3	3	21	L
4	4	22	M
5	5	23	N
6	6	24	O
7	7	25	P
8	8	26	Q
9	9	27	R
10	A	28	S
11	B	29	T
12	C	30	U
13	D	31	V
14	E	32	W
15	F	33	X
16	G	34	Y
17	H	35	Z
		36	(NULL)

The quantities τ_1-τ_6 serve to identify the transmitter. As an example, assume the transmitter's call sign is K2QRM. For this case τ_1=20, τ_2=2, τ_3=26, τ_4=27, τ_5=22, τ_6=36.

b) The final pulse position encoded variable, τ_c, ranges from 0 to 2^n-1 where n is the number of stages of one of the two (equal length) Gold Code Generator (GCG) linear feedback shift registers. The way τ_c

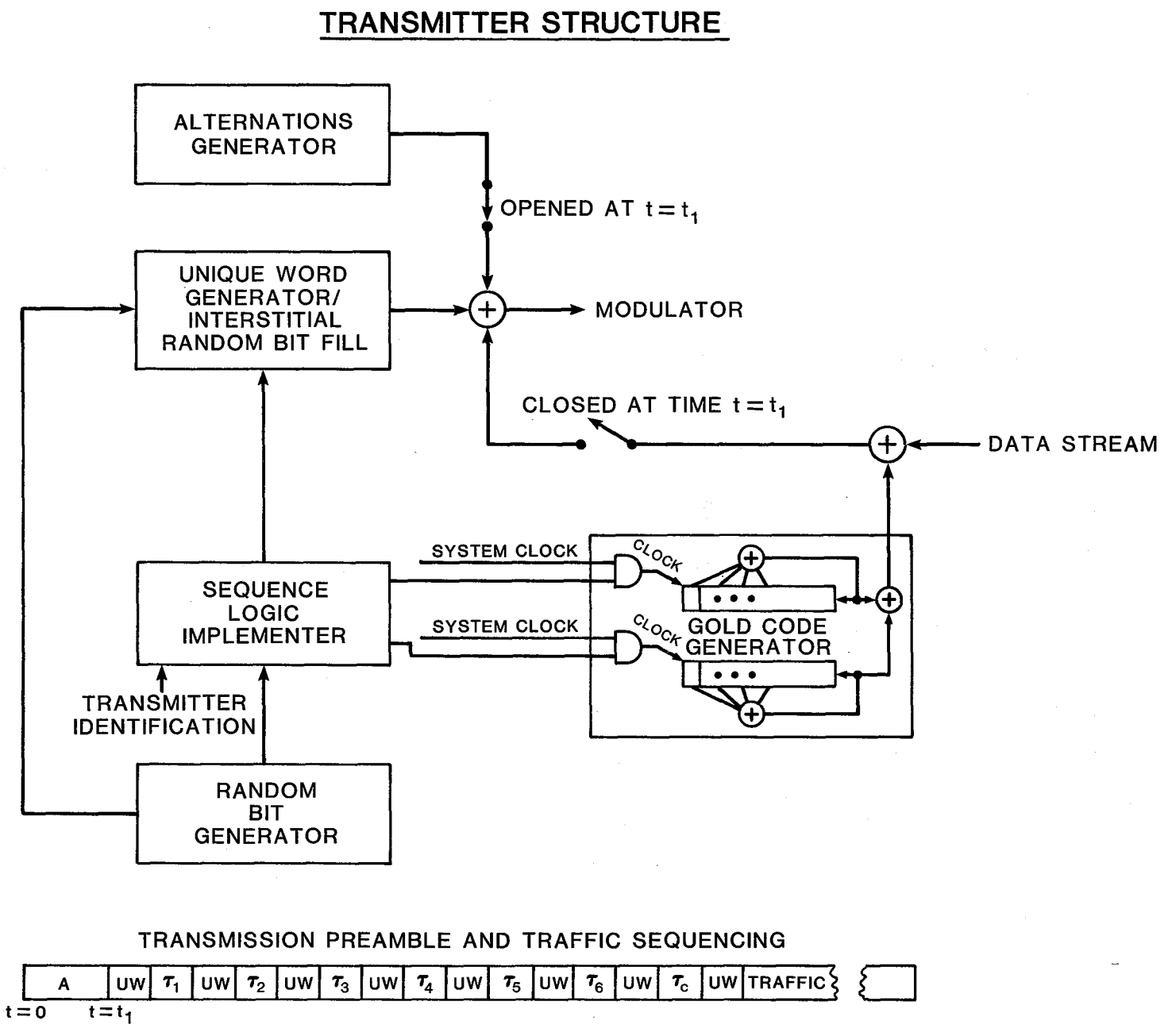


Figure 5. Proposed transmitter structure.

selects the code is as follows. At t=0, both the upper and the lower feedback (m-sequence) shift registers in the GCG modules are set to all ones. After the UW following τ_6 has been detected (this UW determines τ_6), the top feedback shift register in the GCG starts clocking at the chip rate clock. When the next, and final UW is detected, the bottom register of the GCG also starts clocking at the chip rate clock. This point in time also determines the start of a baud time for the DATA stream.*

c) We chose to use two periods of the GCG per data baud, for reasons to be discussed later, and therefore the chip rate is equal to two times the DATA stream rate times the number of bits in a GCG period which is of the form 2^n-1.

d) The alternations generator is part of the synchronization preamble and is used to train the receiver's Costas loop.

e) The random bit generator provides randomly derived bits where needed, i.e., to specify the length of τ_c and also the chips necessary to fill in the periods τ_1-τ_6.

Figure 6 depicts the proposed receiver's structure. We chose a Costas loop to recover center frequency and chip timing and boundaries. We train the Costas loop, however, by sending a preamble to the traffic and other preamble parts of our transmission. The training sequence is simply a string of alternations 01010101... . The output of the modulator will be a very narrowband (a spectral pair of lines in the main lobe in the limit) energy signal centered at $f_c \pm \frac{0.5}{T_c}$. To help the loop estimate f_c and T_c as accurately as possible, we preface the input to the loop with a band-pass-filter (BPF). The receiver opens its loop before the training segment (A) comes to an end. The receiver then uses an estimate of f_c and the chip boundaries to process the remainder of the synchronization preamble and the traffic. We chose open loop operation to allow the receiver to obtain and retain synchronization even in a high noise environment. An open loop operation should not be difficult or costly to implement when we consider that the chip rates are quite tame and that the longest an operator will speak ("key down") is probably a minute at most.

*This range gives a very slight preference to one of the Gold Code Family members. It is easy to implement in hardware, however.

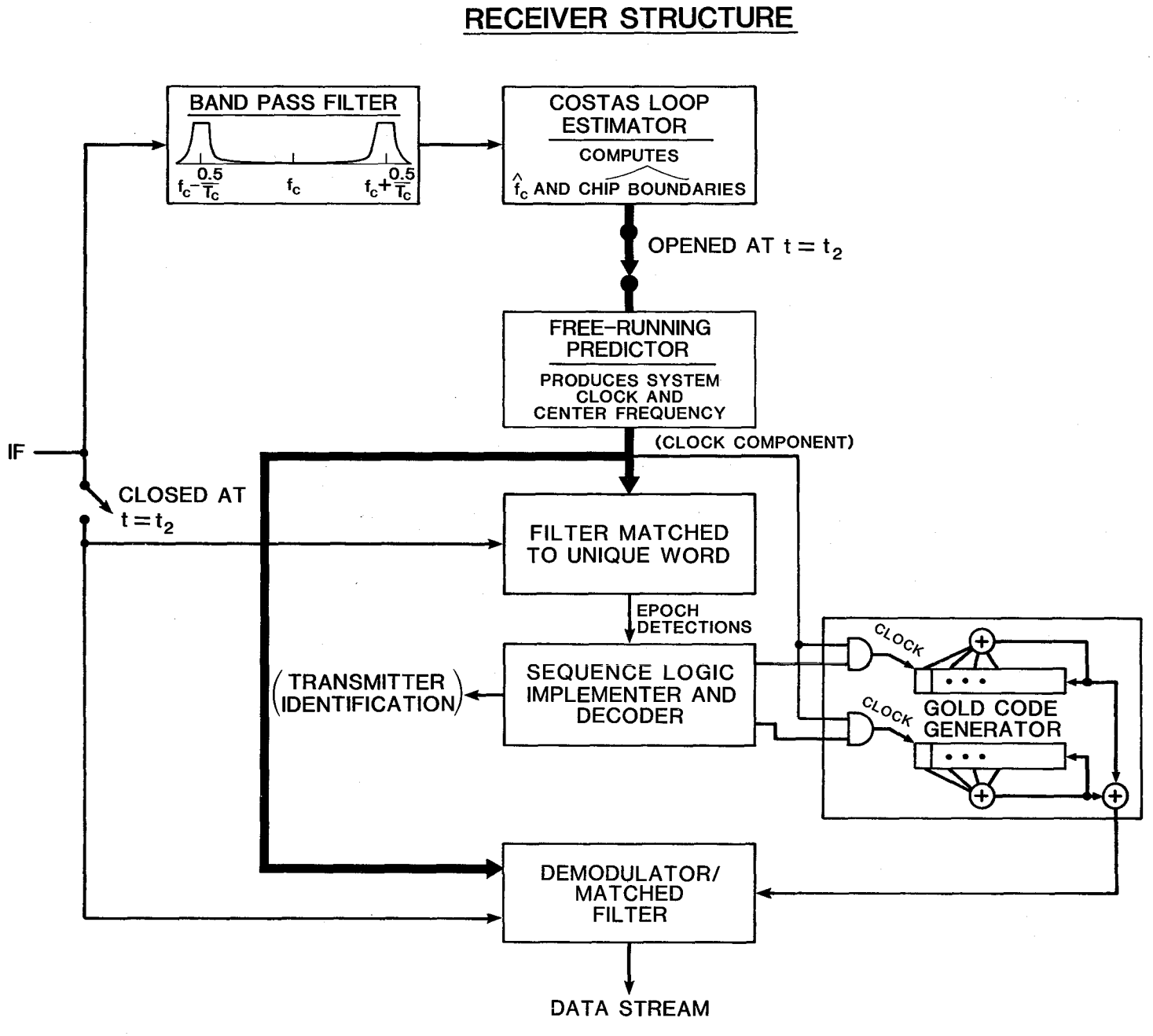


Figure 6. Proposed receiver structure.

3.3.3 Implementation

3.3.3.1 Introduction

We believe it is fitting to say a few words about implementation of our proposed structure. There is no getting away from some analog components, however, a surprisingly large number of the necessary functions can be performed digitally or with hybrid analog and digital techniques such as charge coupled devices.

3.3.3.2 Open loop architecture and related components

Key to our open loop concept is the Costas loop demodulator. Much headway has been made on digital synthesis of this loop and similar loops. A milestone paper in the field was presented by Natali (1972) at the International Telemetering Conference in 1972. Natali reported on analytical methods and supporting experimental data to show that it was possible to construct an all digital coherent demodulator of biphase PSK signals. Natali's bit synchronization algorithm provides near optimal performance at low signal-to-noise ratios. Natali concluded that his techniques were practical (in 1972) for data rates up to 1 megabit-Hz.

A key component to any open loop system is a stable frequency source. For our modest requirements of bandwidth and open loop timing, crystals should perform quite well. Three excellent references are: (a) a National Bureau of Standards Technical Note (Walls and Stein, 1976) that reviews analog servo techniques for oscillators, (b) a study of an electromechanical system which, after synchronization, could run open loop for twelve hours and exhibit an rms time error of only 70 nanoseconds (Allan, et al., 1968), and (c) a paper by Walls and Stein (1978) that reviews and improves upon the traditional technique of slaving and frequency locking crystal oscillators.

Another possibility is to use a Surface Acoustic Wave (SAW) device. A recent article (Mitchell, 1982) reports on a SAW-resonator oscillator with short-term frequency stability for 2 to 3 seconds of about one part in 10^{-11}.

Even though our frequency stability requirements are modest, we must bear in mind what many people have alluded to and that is the extreme sensitivity in performance of BPSK systems to frequency offset. (See Bhargava, et al., 1981, Chapter 5, for example.) One way of looking at this is via the ambiguity function of radar theory. In radar, one of the variables is motion and hence a Doppler frequency. In the AGARD (1973) publication we find that the peak amplitude of a matched filter for a BPSK receiver is

$$\left| \frac{\sin \pi f_\phi T}{\pi f_\phi T} \right| \qquad (2)$$

where f_ϕ is the frequency offset and T is the duration of integration.

For performing the digital clocking functions, the author subscribes to the belief that standard, off-the-shelf, inexpensive and widely available TTL logic should be sufficient. It is probably best (Hayward, 1982) to use frequency division techniques (divide down counters) rather than frequency multiplication. Noise in oscillators is, unfortunately, frequently disregarded in analysis. This noise may be dichotomized into amplitude and phase components. Phase noise grows on the order of 20 $\log_{10}N$ where N is the multiplication factor in a frequency multiplication technique.

Finally, the full or partial use of digital techniques brings extra attention to the need for good engineering practices such as isolation and shielding. This is especially true for circuits such as those performing phase-locked synthesis. An excellent discussion is to be found in Chapter 10, entitled "Spectral Purity," of a recent book by Egan (1981).

3.3.3.3 The balanced mixer

Dixon (1976a) reports that Carson obtained a patent on the balanced modulator in 1915. Since then, this remarkable circuit has been refined many times with increasing innovation. Apropos of this and as a bit of proffered evidence that the ARS has been a valuable asset to the engineering arts, the reader should peruse an article by Rohde (1977) in which Rohde, a member of the ARS, describes his and others' work on achieving double-balanced mixers of high dynamic range.

In general, simple divide-ring mixers will exhibit typical conversion losses of only 6 to 7 dB. A bandwidth of 0.5-500 MHz for the rf and local oscillator ports is easily obtained as is an IF port response of from 0 to 500 MHz with about 40 dB of balance over a considerable portion of the bandwidth. (See Hayward, 1982.)

3.3.3.4 Matched filters

Two very important papers appeared more than two decades ago: Davenport (1953) and Cahn (1961). These two papers dealt with the very interesting question of degradation of signal-to-noise ratio in the presence of hard limiting. Abstruse as this sounds it has a very decided input to the problem at hand. In Appendix B we talk about synchronization words and we compute crosscorrelations by first hard quantizing each bit and then computing total agreements minus total disagreements. This is a good way to study synchronization words and what

Davenport and Cahn tell us is that, although it is not an optimal way to implement the synchronization process, it is not too bad for low signal-to-noise ratios in Gaussian noise. Unfortunately, in a multi-user CDMA system with a near/far problem, the interference is certainly not Gaussian and what we should be doing is to perform our crosscorrelations linearly, i.e., without hard limiting until total disagreements have been subtracted from total agreements. This is a much more difficult job to do in hardware. There are two very promising techniques to aid us, however, surface acoustic wave (SAW) devices and charge coupled devices (CCDs). These two families of solid state devices are quite different and each possesses its own advantages. For example, CCDs are somewhat easier to interface with ancillary logic circuits but harder to fabricate. SAWs can operate at rf; CCDs only at baseband. The following literature is recommended as introductory. Milstein and Das (1979) give an excellent, easily read, view of theory, hardware, and snapshots of the state-of-the-technology for SAWs. Bell et al. (1973) and Milstein and Das (1977) show how SAWs are of direct benefit to spread spectrum problems. Morgan et al. (1976) discuss a spread spectrum synchronization problem using a SAW that processes 730 microseconds of a 9 megabit-Hz signal and which realizes a 40 dB processing gain. Another good reference is Unkauf (1977). The CCD literature is also abundant. Two worthwhile pieces are Grant (1981) who reviews SAWs and CCDs as to their performance when used for fixed and programmable analog matched filters for spread spectrum work, and Collins et al. (1972), who also present empirical results of using CCDs to perform the analog matched filter function.

4. AN EXAMPLE AND A CURSORY ANALYSIS

4.1 Example Parameters

For our example we:

-- choose a unique word (UW) of length 1023 bits (perhaps an m-sequence phased for best results - see Appendix B)

-- assume that the DATA stream is 8000 bits/second. This rate would support fair quality voice that was digitized by an adaptive delta modulation technique such as a continuously variable slope delta modulator (CVSD). Such devices are commercially available in LSI (chip) form at nominal cost. (For a good tutorial and review of delta modulation techniques see Rabiner and Schafer (1978) and Flanagan et al. (1979) respectively.)

-- choose a Gold code of length 127 derived from the preferred primitive

polynomials

$$x^7+x^3+x^2+x+1$$

and (3)

$$x^7+x^5+x^4+x^3+x^2+x+1$$

as shown in Figure 7.

-- choose the center frequency of the DS spread spectrum transmissions to be adjustable over the range 222.492 MHz $\leq f_c \leq$ 222.508 MHz.

-- require that all power outside the main lobe be filtered out.

4.2 Operational Parameters

-- The length of the Alternations (A) segment of the transmitter's preamble is yet to be determined by the channel conditions. Theoretically and empirically supported guidelines must be developed.

-- The operator shall, in addition to the auto-identification performed by the transmitter, identify himself in Morse code, using narrowband A1, A2, F1 or F2 modulation, by his call sign, his center frequency and other pertinent information on a standard "check-in" frequency within the 220-225 MHz band before beginning spread spectrum transmissions, at the end of a spread spectrum session and at intervals not to exceed 10 minutes during any single transmission or exchange of transmissions of more than 10 minutes duration to keep within the spirit of Section 97.84, Station identification, of Part 97 of the FCC's Rules and Regulations.

4.3 A Cursory Analysis of the Above Example

For our proposed system, the ability to easily monitor the spread spectrum transmissions, which is of paramount importance, is totally resident in the synchronization process. The quantities τ_1-τ_6 identify the transmitter and τ_c identifies the transmitter's code. It is important, then, that the Pulse Position Encoding delimiter, i.e., the UW, survive much channel degradation. We chose a length for the UW that is long compared to the system baud period for this reason. As we stated previously, we chose an open loop system because most one way transmissions would probably not last for more than a minute. This is a very modest requirement for open loop estimation. The advantage of an open loop system is that interference of any type will not affect clock generation after the preamble or synchronization phase. Thus, in improving the receiver, the operator need not be concerned about loop pull-out or other false-lock or capture problems.

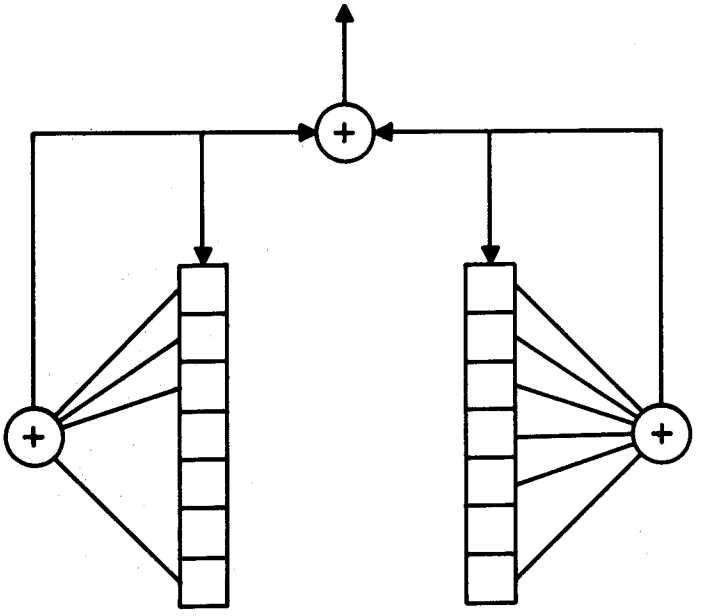


Figure 7. Gold code generator.

Our mainlobe will be 4.064 MHz wide as we are using two 127 bit code periods per 8 kilobit-Hz data bauds. The form of power spectral density, is, of course, what is of interest. Following Scholtz (1977), let the spreading code sequence be denoted by $c_0, c_1, c_2, \ldots c_{126}$. We define the normalized (full period) autocorrelation of the spreading code sequence by

$$R(k) = \frac{1}{127} \sum_{i=0}^{126} c_i c_{i+k} \tag{4}$$

where k is the "lag" or degree of slip. (We have mapped 0 into a 1 and a 1 into a minus 1 for these computations; also note that the values of {R(k)} are all real and we have therefore dropped the conjugation operation.) The power spectral (line) density, S(k), for the spreading code is found by taking the Fourier transform of (4) and we obtain

$$S(k) = \begin{cases} \frac{1}{127} \sum_{i=0}^{126} R(i), & k=0 \\ \\ \frac{2}{127} \sum_{i=0}^{126} R(i) \cos \frac{2\pi ik}{127}, & k \neq 0 \end{cases} \tag{5}$$

The power spectral (line) density (5) is displayed in Figure 8. The top set of points are the {S(k)} and the bottom set are the {S(k)} weighted by the "sinc-squared" envelope, i.e.,

$$S(k) \left[\frac{\sin(\pi k/127)}{\pi k/127} \right]^2 . \tag{6}$$

The maximum cross-correlation of the Gold code family members* for length 127 is 17. A fair question would be: "Why not just use pseudorandomly generated strings of 127 bits (or even 254 bits, if we forget about the $T_b/T_p=2$ requirement) such as those generated by taking successive stretches of a long m-sequence?" This is a good and legitimate question. Consider that we were to do this. After all, we have the process synchronized so using a long m-sequence, or even an exotic nonlinear generator will not, in any way, impair our ability to monitor the transmissions and identify the transmitter. Consider now that we had indeed done so. The crosscorrelation of one segment with that from another transmitter would be described, in first order analysis, by a variable, x, that possesses the statistics of a Binomial process with range x=-127, -125, ...

*Note, incidentally, that we are using only 127 out of the 129 possible Gold code family members.

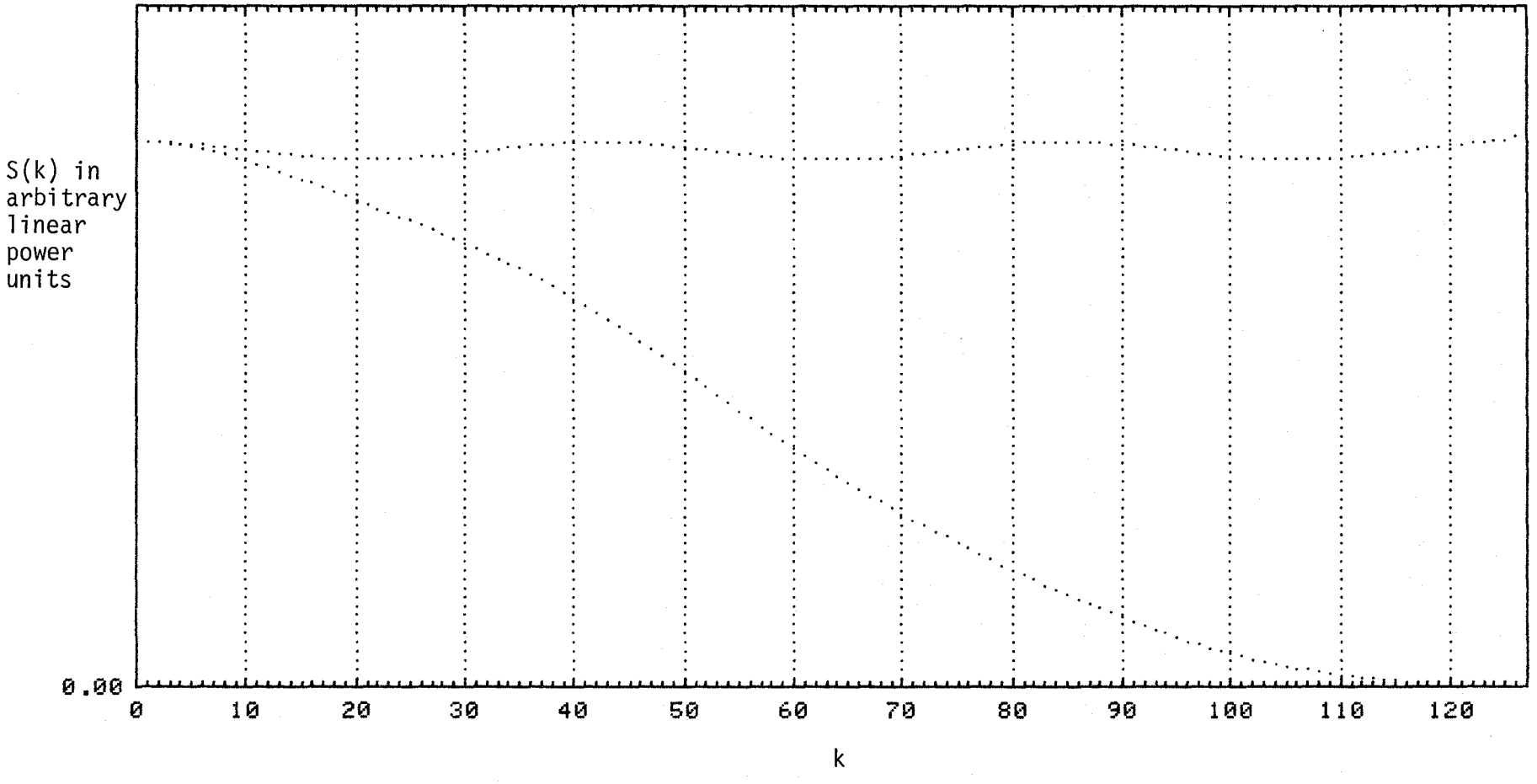


Figure 8. Gold Code power spectral density (upper curve), weighted by sinc-squared (lower curve).

-3, -1, +1, +3, ... +125, +127; agreements minus disagreements. The probability that $|x|>17$ is only about 13%. This is not a highly significant percentage and we conclude that using short Gold codes, while we do accrue some correlation gain advantage, we do not have a compelling edge over the pseudorandom technique. Why then bother with Gold codes and, especially, why set $T_b/T_p=2$? The answer lies in the "near-far" problem. Consider the situation of Figure 9. Assume that A_1 and A_2 are engaged in half-duplex communications and similarly for B_1 and B_2. The code isolation or "processing gain" provided by 127 or even 254 pseudo-randomly chosen bits is at <u>best</u> $10 \log_{10} 254 \simeq 24$ dB. If $d_2>16d_1$*, communications will be impossible or severely degraded. The nearness of A_1 to B_1 or A_2 to B_2 increases the effective interfering power received. Cahn (undated) puts it very well:

> "Interest in such techniques [Code Division Multiple Access; CDMA] typically is relative to random access communication systems where the number of potential users (subscribers) is much larger than the maximum number of users simultaneously active in the channel.
>
> .
> .
> .
>
> A pure CDMA approach is not capable of accommodating signals with large power differentials, which we term the 'near-far' problem since it typically arises from the large variation in ranges between users in a geographically dispersed network. In other words, the processing gain of the receiver determines the tolerable total interference, and it does not matter whether there are 1000 signals at the same power level, 100 signals each at ten times the power level, or 10 signals each at 100 times the power level as the desired signal."

But. By using two periods of a Gold code per DATA stream baud we have an interesting fallback position. We have already examined, in Figure 8, the power spectral (line) density of one period of one phase of the Gold code. The chip rate at which the 127-bit Gold code is run is 16 kilobits-Hz. This yields a line spacing of 16 kHz. Because we use two periods per DATA stream baud, we will cause the Gold code power spectral density to broaden and resemble that shown in

*This assumes a very simple line-of-sight case.

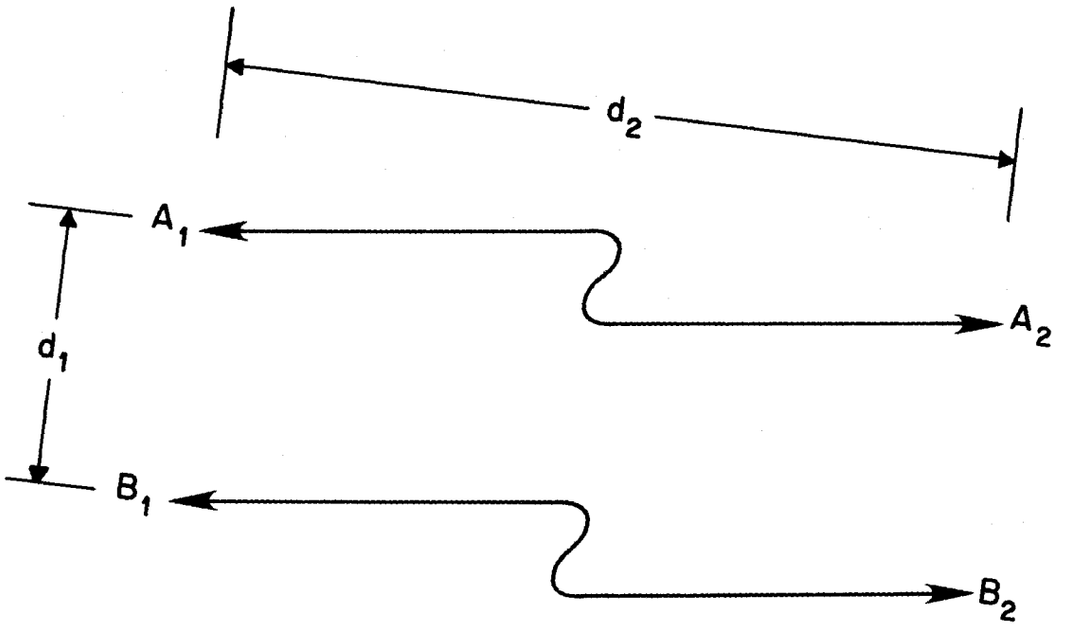


Figure 9. The near/far problem.

Figure 10. Note that we have interstices between our bands of energy. It is within these interstices that another, similar power spectral density, can be inserted with a relatively high degree of interference isolation. All the second half-duplex user pair have to do is to offset their center frequency by 8 kHz from the first half-duplex user pair.

It is even conceivable that an operator could construct a second receiver system to subtract out an interfering signal. All that would be needed would be a DATA stream baud estimator that ran in real time. If the interfering signal were strong, such an estimation should be practicable.

Finally, although it is doubtful that collisions, i.e., two users attempting to synchronize during the same time, would ever be a serious problem, the mathematics applicable to its analysis can probably be borrowed from analyses of the ALOHA system. (See Abramson and Kuo (1973), Chapter 14.)

5. BELIEFS AND RECOMMENDATIONS

The author believes that:

a) DS spread spectrum is not profitably overlaid with channelized communications (see Juroshek, 1979, and deHaas and Watterson 1981) and is not a good choice for ARS communications qua communications but is a valid vehicle for research by the ARS.

b) the Technician Class licensees should be allowed any spread spectrum privileges allocated to higher class licensees as the license class is specifically for competent ARS experimenters.

c) monitoring and identification can be taken care of through a well chosen synchronization mechanism that includes an auto-identification technique.

The author recommends that:

a) if the Amateur Extra, Advanced, and Technician Class licensees are allowed to experiment with the DS spread spectrum techniques outlined in Chapter 3, the bands, or portions thereof, must be carefully chosen. The following are cursorily outlined key considerations. The 50-54 MHz band is adjacent to a TV channel allocation and while the operators of spread spectrum units may conscientiously keep their emissions within their assigned limits, TV receiver selectivity is not a uniform parameter. (See Allnatt et al. (1963).) The extent and nature of potential interference is unknown and should be studied. The 144-148 MHz band is partly used for communications that operate under very low signal-to-noise ratios. The experimentation

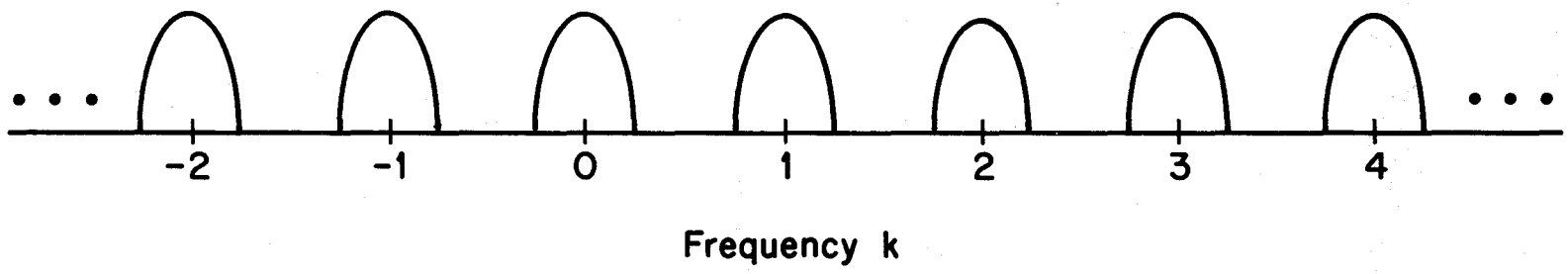


Figure 10. The power spectral density showing interstices.

associated with moonbounce, path losses of up to 254 dB (see Tynan, 1981), and other herculean communication tasks that have low S/N deserve protection. While FH spread spectrum techniques might be reasonably tailored for this band, and the author is impressed with the partial but encouraging results reported by AMRAD to the public file, a serious question remains concerning the suitability of DS systems for this band. Finally, we note that the 220-225 MHz band is being jointly planned by the NTIA and FCC for Amateur, Fixed and Mobile Services. The joint planning will consider technical standards and sharing requirements between the authorized services for maximum effective use of the band. This planning will also consider the use of spread spectrum modulation by the ARS.

b) other spreading code schemes be investigated in a search for higher spectral efficiency. An interesting, and largely unknown concept, has been proposed by Olsen (1977). It is also suggested that spectrum shaping techniques, such as Markov filtering proposed by the author in Appendix C, be researched.

c) section 97.67, Maximum authorized power, of Part 97 of the FCC's Rules and Regulations, which presently states, in part,

...

(b) Notwithstanding the provisions of paragraph (a) of this section, amateur stations shall use the minimum amount of transmitter power necessary to carry out the desired communications.

...

be rewritten to stress the extra need for ARS compliance when using spread spectrum communications.

d) detailed study be undertaken to determine whether DS spread spectrum users should be subject to lower power limits than those imposed on narrowband communications. The study should probably be sensitive to local geography and local traffic types and densities.

e) DS spread spectrum be operated from fixed sites only, i.e., that mobile operation not be permitted.

f) a scheme similar to that suggested in Chapter 3 be studied for use with narrowband (nonvoice) DATA streams such as TTY. Such a system could probably overcome the near/far problem for a many user community if T_b/T_p were set to k where k was 10, 20 or perhaps even greater.

g) a prototype system, making extensive use of digital techniques, be constructed and tested as an aid to further studies supporting this new frontier.

6. ACKNOWLEDGMENTS

The author especially thanks Dr. Peter McManamon for technical direction on this project, Dr. William Utlaut for his encouragement, and Mr. Douglass Crombie (NTIA) for his interest and guidance. The author also expresses his gratitude to the following, randomly ordered, people for giving their time and thoughts in conversation: Martin Nesenbergs, Mike Kennedy (FCC), Don Spaulding, Gene Ax (NTIA), Les Berry, Bill Hartman, Gene Adams, Bill Pomper, Val Pietrasiewicz, Rao Yarlagadda (Oklahoma State University), Dave Wortendyke, Ted deHaas, John Juroshek, Clark Watterson, Dwight Melcher, Carole Ax, and Charlene Cunningham.

7. REFERENCES

Abramson, N., and F. Kuo (1973), Computer-Communication Networks (Prentice-Hall, Inc., Englewood Cliffs, NJ).

AGARD (1973), NATA Advisory Group for Aerospace Research and Development, Lecture Series No. 58, Spread Spectrum Communications.

Allan, D., L. Fey, H. Machlan, and J. Barnes (1968), An ultra-precise time synchronization system designed by computer simulation, Frequency, January.

Allnatt, J., D. Mills, and E. Loveless (1963), The subjective effect of co-channel and adjacent-channel interference in television reception, The Institution of Electrical Engineers, Paper No. 3941E, originally presented at the International Television Conference, 4 June 1962, pp. 109-117.

Bell, D., J. Holmes, and R. Ridings (1973), Application of acoustic surface-wave technology to spread spectrum communications, IEEE Trans. Microwave Theory Tech. MTT-21, No. 4, pp. 263-271, April.

Bhargava, V., D. Haccoun, R. Matyas and P. Nuspl (1981), Digital Communications by Satellite (John Wiley & Sons).

Cahn, C. (1961), A note on signal-to-noise ratio in band-pass limiters, IRE Trans. Inform. Theory, pp. 39-43, January.

Cahn, C. (Undated), Notes from a short course on spread spectrum systems lecture entitled: Theoretical concepts and advanced techniques, Section 4, Multiple Access.

Collins, D., W. Bailey, W. Gosney, and D. Buss (1972), Charge-coupled-device analogue matched filters, Electron. Letters 8, No. 13, pp. 328-329, June.

Davenport, W. (1953), Signal-to-noise ratios in band-pass limiters, Applied Physics 24, No. 6, pp. 720-727, June.

deHaas, T., and C. Watterson (1981), An analysis of the compatibility of spread-spectrum and narrowband FM mobile radio systems in the 156 to 162 MHz band, U.S. Dept. of Commerce Report MA-RD-940-81011, April.

Dixon, R. (1976a), Spread Spectrum Systems (John Wiley & Sons).

Dixon, R. (Editor) (1976b), Spread Spectrum Techniques (IEEE Press).

Egan, W. (1981), Frequency Synthesis by Phase Lock (John Wiley & Sons).

FCC Rules and Regulations (1981), Part 2, July.

FCC Rules and Regulations (1981), Part 97, 1 October.

Flanagan, J., M. Schroeder, B. Atal, R. Crochiere, N. Jayant, and J. Tribolet (1977), Speech coding, IEEE Trans. Commun. COM-27. No. 4, pp. 710-737, April.

Gold, R. (1967), Optimal binary sequences for spread spectrum multiplexing, IEEE Trans. Inform. Theory, pp. 619-621, October.

Gold, R. (1968), Maximal recursive sequences with 3-valued recursive cross-correlation functions, IEEE Trans. Inform. Theory, pp. 154-156, January.

Grant, P. (1981), Application of analogue signal processors to matched and adaptive filtering for spread spectrum communications, Proceedings (No. 50) of the Clerk Maxwell Commemorative Conference on Radio Receivers and Associated Systems, 7-9 July.

Hayward, W. (1982), Introduction to Radio Frequency Design (Prentice-Hall, Inc.).

Holmes, J. (1982), Coherent Spread Spectrum Systems (John Wiley & Sons).

ITU (International Telecommunication Union) Radio Regulations (1982).

Juroshek, J. (1979), A compatibility analysis of spread-spectrum and FM and mobile radio systems, U.S. Department of Commerce, NTIA Report 79-23, August.

Milstein, L., and P. Das (1977), Spread spectrum receiver using surface acoustic wave technology, IEEE Trans. Commun. COM-25, No. 8, pp. 841-847, August.

Milstein, L., and P. Das (1979), Surface acoustic wave devices, IEEE Communications Magazine, September, pp. 25-33.

Mitchell, B. (1982), SAW oscillators: an alternative to quartz-crystal sources, Microwaves, February, pp. 14, 18.

Morgan, D., J. Hannah, and J. Collins (1976), Spread-spectrum synchronizer using a SAW convolver and recirculation loop, Proc. IEEE, pp. 751-759, May.

Natali, F. (1972), All-digital coherent demodulator techniques, Proceedings of the 1972 International Telemetering Conference, Vol. VIII, 10-12 October, pp. 89-107.

Olsen, J. (1977), Nonlinear Binary Sequences with Asymptotically Optimum Cross-correlation, PhD. Dissertation, USC, December.

Pasupathy, S. (1979), Minimum shift keying: a spectrally efficient modulation, IEEE Communications Magazine, pp. 14-22, July.

Pickholtz, R., D. Schilling, and L. Milstein (1982), Theory of spread-spectrum communication - a tutorial, IEEE Trans. Commun. COM-30, No. 5, pp. 855-884, May.

Pursley, M. (1977), Performance evaluation for phase-coded spread-spectrum multiple-access communication - Part I: system analysis, IEEE Trans. Commun. COM-25, No. 8, pp. 795-799, August.

Rabiner, L. and R. Schafer (1978), Digital Processing of Speech Signals (Prentice-Hall).

Reference Data for Radio Engineers (1968), (Howard W. Sams & Co. Inc.).

Rinaldo, P. (1980), Spread spectrum and the radio amateur, QST, LXIV, No. 11, pp. 15-17, November.

Riter, S. (1969), An optimum phase reference detector for fully modulated phase shift keyed signals, IEEE Trans. Aerospace and Electron. Systems AES-5, No. 4, pp. 627-631, July.

Rohde, U. (1977), High-dynamic range active double-balanced mixer, Ham Radio 10, No. 11, pp. 90-91, November.

Scholtz, R. (1977), The spread spectrum concept, IEEE Trans. Commun. COM-25, No. 8, pp. 748-755, August.

Scholtz, R. (1982), The origins of spread-spectrum communications, IEEE Trans. Commun. COM-30, No. 5, pp. 822-854, May.

Tynan, W. (1981), A VHF/UHF primer - EME, QST, LXV, No. 11, p. 84, November.

Unkauf, M. (1977), Surface Wave Devices in Spread Spectrum Systems, Chapter 11 of Surface Wave Filters, H. Matthews (Editor), (John Wiley & Sons), pp. 477-509.

Utlaut, W. (1978), Spread-spectrum principles and possible application to spectrum utilization and allocation, ITU Telecommunication J., January.

Viterbi, A. (1979), Spread spectrum communications-myths and realities, IEEE Communications Magazine, pp. 11-18, May.

Walls, F., and S. Stein (1976), Servo techniques in oscillators and measurement systems, NBS Technical Note 692, December.

Walls, F., and S. Stein (1978), A frequency-lock system for improved quartz crystal oscillator performance, IEEE Trans. Instr. Measr. IM-27, No. 3, pp. 249-252, September.

APPENDIX A: m-SEQUENCES: WHAT THEY ARE AND HOW THEY CAN BE IMPLEMENTED*

1. INTRODUCTION

This appendix is both a primer on maximal length recursive sequences (m-sequences) and a discourse on their implementation architectures and properties they exhibit under certain manipulations. The style is that of an annotated bibliography that attempts to cover most of the significant contributions of the last two decades. The appendix also contains some new material, however, and new ways of looking at old material. The authors have deliberately chosen this style as it has been their experience that innovation most likely proceeds by viewing a problem or situation from many and diverse perspectives.

The appendix, then, attempts to bring together most of the developments of m-sequence architectures and properties following the publication of Golomb's milestone book (1967). This appendix is concerned with binary m-sequences only, and therefore the mathematics is to be understood as almost exclusively modulo-two computations. Further, we are concerned with single m-sequences and not interactions of different m-sequences. The subject of interactions is rich in its own right and has been recently well examined by Sarwate and Pursley (1980a, 1980b). In addition to attempting to collocate results which have been scattered through scores of different sources, we have attempted to present some new items and, perhaps more important, to recast some of the extant theory in a different form. Specifically, the authors believe that m-sequence theory can often be profitably viewed in matrix terms vice strictly polynomial algebra. The trend has been away from matrix representation and this has been motivated largely by the error correction coding theorists who maintain, and rightly so, that most useful properties can be handled exclusively by polynomial algebra and to employ matrices is both cumbersome and inefficient. Yet, circuit engineers, system engineers, and sequential machine theorists find matrix formulation a far more intuitive vehicle than abstract algebra. We have therefore attempted to balance our approach in an attempt to motivate as well as to coalesce the important aspects of the subject for a broad audience.

*This Appendix coauthored with Professor Rao Yarlagadda, School of Electrical Engineering, Oklahoma State University, Stillwater, Oklahoma 74078.

The appendix has four main parts: (a) a propaedeutic on recursive sequence theory; (b) sequential machine architectures for implementing m-sequences; (c) manipulation of m-sequences; and (d) implementation of high-speed m-sequences.

2. SEQUENCE THEORY

The theory of maximum length (linearly generated) binary sequences, or m-sequences is one of the most mathematically aesthetic disciplines of finite field theory. The theory offers far more than aesthetics, however, as m-sequences are extensively used by electrical engineers, particularly in the communications, radar, navigation, and computer disciplines. The theory behind m-sequences is sufficiently well developed and, for lack of a better word, "modular," so that even one who is not a mathematician can manipulate and apply powerful results in order to create new useful architectures and uncover new truths.

The sequences are an important subclass of recursively generated binary sequences which are defined by

$$s(t)=f(s(t-1,s(t-2),\ldots s(t-n)), \qquad s(i)\varepsilon\{0,1\} \tag{A-1}$$

which states that the bit at time t is precisely dependent on the n-bits preceding it. The sequence so produced is sometimes said to be of "span-n" (Golomb, 1980). We see from (A-1) that the sequence is deterministically generated and we immediately deduce that the sequence will eventually give rise to a cycle, or recurrent (in the Markov sense) set of states where a state is defined as the n-tuple

$$(s(\tau-1),s(\tau-2),\ldots s(\tau-n)) \tag{A-2}$$

We also deduce that the maximum possible cycle length must be bounded by the number of possible tuples of the form (A-2) which is 2^n. We also observe that because f in (A-1) is a function of n binary terms, f may be viewed as a boolean function of n variables. There are 2^{2^n} possible functions and thus 2^{2^n} possible recurrences. (See p. 12 of Golomb, 1967.)

If and only if f is expressible as a modulo-two sum of terms, i.e.,

$$f=\sum_{i=1}^{n} \alpha_i s(t-i) \quad , \quad \alpha_i\varepsilon\{0,1\} \tag{A-3}$$

then f is said to be a linear function and the recursively generated sequence is said to be linearly generated. We can, incidentally, view a linearly generated

recursive sequence as a nonlinear difference equation using regular (nonmodular) mathematics. For example, the modulo-two linear recursive sequence

$$s(t)=s(t-3)+s(t-5) \tag{A-4}$$

where, according to our convention, the plus sign is modulo-two addition can be written as

$$s(t)=s(t-3)+s(t-5)-2s(t-3)s(t-5) \tag{A-5}$$

where the plus and minus signs in (A-5) imply regular addition and subtraction. When, and only when, all the s(i) are zeros and ones will (A-5) be the same as (A-4). In this one special case, a periodic solution obtains to the nonlinear difference equation (A-5).

One further general remark is in order. If and only if f can be written as

$$f=g(s(t-1),\ s(t-2),\ldots s(t-n+1))+s(t-n) \tag{A-6}$$

where g may be any boolean function of n-1 variables, will the sequence of tuples (A-2) be such that every tuple has a unique predecessor. This is an obvious, yet very powerful truth and is well presented by Golomb (1967, p. 116). Note that all span-n linear functions are of the form (A-6).

The study of sequences is in and of itself a tremendous undertaking. We do not pretend to even try. Why then do we select one particular class of sequences, viz the m-sequences? The answer is twofold. First, the theory behind m-sequences is seasoned, tractable, and rich. Second, and most important to an engineer, m-sequences are useful, primarily because of their random-like qualities. To motivate further we again return to Solomon Golomb, who, in his famous book (1967, pp. 25-26), sets three "randomness postulates" or three properties or characteristics one would expect or demand from sequences purported to be random. Before recounting these properties we must comment that there is a subtle "doublethink" involved. Because our sequences will be deterministically generated, they will exhibit a period. They are thus anything but random. What we are addressing is a study of their "short-term" behavior which is taken to be their statistical analysis over a single period only. Thus we must (silently) preface our use of the word random with the prefix "pseudo." Golomb's postulates for sequence randomness are, then,

a) $p-2\sum_{i=1}^{p} s(i) \leq 1$ where p is the sequence period.

b) To the extent that the period can be subdivided, the number of runs of zeros and ones exhibited must fall in inverse geometric proportion to their lengths, i.e., half the runs should be 1-long, one quarter 2-long, etc.

c) The autocorrelation, $R_{ss}(\tau)=\sum_{i=1}^{p} s(i)s(i+\tau)$, must be two valued, i.e., $R_{ss}(0)=p$ and $R_{ss}(\tau)=\rho\neq p, \tau\neq 0$.

If a sequence comports to the above requirements, it is termed a pseudonoise or PN sequence. The following 31 bit period sequence (this sequence and all other sequences should be read from left to right)

$$1111100011011101010000100101100 \qquad \text{(A-7)}$$

meets all three postulates, i.e.,

a) There are sixteen ones and 15 zeros.

b) There are sixteen runs distributed as follows:
   1) four runs of zeros and ones each of length 1
   2) two runs each of length 2
   3) one run each of length 3
   4) one run of zeros of length 4
   5) one run of ones of length 5.

c) The autocorrelation is 16 for $\tau=0$ and 8 otherwise.

The PN sequence (A-7) was generated by the recursion

$$s(t)=s(t-3)+s(t-5) \ . \qquad \text{(A-8)}$$

Because f is of the form (A-3), the recursion is linearly generated.

We thus have a hint that linearly generated sequences might be useful as PN sequence generators. Let us try another (arbitrarily chosen) linear generator, say, the recursion

$$s(t)=s(t-4)+s(t-5) \ . \qquad \text{(A-9)}$$

We easily find that (A-9), if started with the all ones tuple, gives rise to the sequence

$$111110000100011001010 \ . \qquad \text{(A-10)}$$

The sequence (A-10) is of period 21. Checking, we find that it satisfies randomness postulate (a). The sequence (A-10) exhibits a total of 10 runs. Of the 10 runs, half are of length one but it is required that the number of 1-long runs of ones

must equal the number of 1-long runs of zeros which is impossible as 5 is an odd number. Hence, the sequence (A-10) fails to meet the second randomness postulate. The sequence (A-10) also fails to meet the third postulate as it exhibits autocorrelation values of {10,5,4 and 3}.

Why has one linear generator of span equal to 5 produced a PN sequence and another linear span-5 generator failed? With further experimentation, we would come to the hypothesis that only and all linear generators of span-n whose sequences exhibit periods equal to 2^n-1 produce PN sequences. Golomb (1967, pp. 43-45) establishes the sufficiency of the hypothesis, i.e., a linear generator of span-n that exhibits a period of length 2^n-1 must indeed produce a PN sequence. Not established, but conjectured by Golomb (1980) is the (even strengthened) necessity, viz, the PN sequences are solely composed of maximum length linearly generated sequences. Consider for example, the nonlinearly generated sequence produced by the recursion

$$s(t)=s(t-1)+s(t-5)+\bar{s}(t-1)s(t-2)s(t-3) \qquad \text{(A-11)}$$

(where the super-bar on the s denotes complementation). Starting (A-11) with the all ones tuple we obtain the sequence

$$1111100101110101001101100010000\ . \qquad \text{(A-12)}$$

The sequence (A-12) meets randomness postulates (a) and (b) but exhibits autocorrelation values of {16,9,8 and 7} and hence fails postulate (c).

The longest, or maximum length, sequence that can be produced by a recursion of the form

$$s(t)=\sum_{i=1}^{n} \alpha_i s(t-i) \qquad \text{(A-13)}$$

is clearly 2^n-1 (as the all zero n-tuple will immediately perpetuate itself), hence the term m-sequence is given to such a sequence. (The authors believe that the term 'm-sequence' was coined by Zierler (1959, p. 39).)

The period of a linearly generated recursive sequence can be straightforwardly analyzed by using generating functions (Golomb, 1967, pp. 30-33) or by z-transform theory (Charney and Mengani, 1961). The essence of the theory is that the polynomial

$$1+\sum_{i=1}^{n} \alpha_i x^i \qquad \text{(A-14)}$$

where the α_i's in (A-14) are the same as in (A-13), is either reducible, divisible without remainder by a polynomial of degree d, $1<d<n$, or irreducible. The recursion corresponding to a reducible polynomial, such as $x^4+x^2+1=(x^2+x+1)^2$, cannot produce an m-sequence. The recursion corresponding to an irreducible polynomial will not necessarily produce an m-sequence, however. Irreducible polynomials are further dichotomized into those that are termed primitive, such as x^4+x+1, and those that are not, such as $x^4+x^3+x^2+x+1$. An irreducible degree n polynomial is primitive if and only if the smallest m for which it divides x^m+1 is $m=2^n-1$. It is therefore redundant to say "irreducible and primitive" as primitivity implies irreducibility. Irreducibility implies primitivity only for Mersenne primes, i.e., when 2^n-1 is prime.

The number of primitive polynomials of degree n is well known and equal to

$$\frac{\phi(2^n-1)}{n} \qquad \text{(A-15)}$$

where ϕ is Euler's "totient" or phi function, an extremely important number theoretic function. This function when applied to a positive integer m, $\phi(m)$, gives the count of the number of integers relatively prime (sharing no factors save unity) to m starting with 1 (which is relatively prime to all positive integers) and incrementing by 1 up to m. Thus $\phi(6)=2$ and $\phi(8)=4$, for examples. For a prime, p, $\phi(p)=p-1$. The function ϕ is said to be "weakly multiplicative." This means that $\phi(mn)=\phi(m)\phi(n)$ if m and n are relatively prime. It is easy to show that $\phi(p^n)=p^{n-1}(p-1)$ for p a prime. Knowing this, we can easily calculate $\phi(q)$ for any positive integer q. All we need do is:

a) Canonically decompose q into its (unique) product of powers of primes, i.e., $q=p_1^{\alpha_1}p_2^{\alpha_2}\ldots p_r^{\alpha_r}$

b) Use the fact that ϕ is a weakly multiplicative function and write $\phi(q) = \phi(p_1^{\alpha_1})\ \phi(p_2^{\alpha_2})\ \ldots\ \phi(p_r^{\alpha_r})$

c) Sequentially evaluate all right-hand terms using the result $\phi(p_k^{\alpha_k}) = p_k^{\alpha_k-1}(p_k-1)$.

For example, $\phi(108)=\phi(2^2\cdot 3^3)=\phi(2^2)\phi(3^3)=2\cdot(2-1)\cdot 3^2\cdot(3-1)=36$.

The count of primitive polynomials (A-15) can be easily derived from an algebraic argument (Golomb, 1967, pp. 48-49) or by a less formal and intuitive argument such as that given by Birdsall and Ristenbatt (1958, pp. 34-35).

Finding primitive polynomials, then, is of great importance. Attempting to factor polynomials is, of course, the first step in proving primitivity as a primitive polynomial must be irreducible. Factoring has become an exciting field in and of itself. An early important method is due to Berlekamp (1967, 1970). A good review of the early methods along with some excellent exercises is given in Knuth (1969). For more recent work, the reader is invited to review Moenck's method (1977) which incorporates a time-saving refinement of Berlekamp's method. Another recent contribution is a new factoring algorithm by Cantor and Zassenhaus (1981) that is of the "probabilistic genre." The probabilistic algorithms are presently in vogue in all sorts of fields and portend, in the authors' opinion, to be a powerful and revolutionary approach to some classically difficult computations.

Tables of primitive polynomials are readily available. The earliest, and most famous compendium, is Marsh's set of tables (1957) which contains an exhaustive listing of all primitive polynomials through degree 19. Watson's list (1962) provides a primitive polynomial for each $n \leq 100$ and n=107 and n=127. Watson's list was followed by Stahnke's list (1973) which presents a primitive polynomial for all $n \leq 168$. Unlike Watson's table, Stahnke's uses trinomials when a primitive trinomial exists for a particular n and a pentanomial otherwise. (All tetranomials are, of course, reducible.) The trinomial listed is of the form x^n+x^a+1 and the 'a' listed is as small as possible. When Stahnke was forced to choose a pentanomial, he chose it to be of the form $x^n+x^{b+a}+x^b+x^a+1$ with $0<a<b<n-a$ and 'a' as small as possible with 'b' also as small as possible following the selection of 'a'. Stahnke chose this pentanomial form in order to comport with Scholefield's architecture (1960) which we shall examine in the next part of the report. If we examine Stahnke's valuable list we are struck by the lack of regularity or patterns in the polynomials. What, for instance, determines whether or not there exists a primitive trinomial for a given n? This and other similar questions broach the frontiers of knowledge of this corner of abstract algebra. There have been rents in the curtain of ignorance, however. Swan (1962) uncovered a number of criteria under which certain polynomial forms must be reducible and hence not primitive. Perhaps Swan's most general, at least most easily remembered, and, to the authors, most exciting, rule is that the trinomial

$$x^{8k}+x^m+1, \quad m<8k \tag{A-16}$$

is always reducible. Thus we at least understand the absence of primitive trinomials for n=8,16,24,....

Zierler and Brillhart (1968, 1969) have catalogued all of the irreducible trinomials for $n \leq 1000$ and, where the factorization of 2^n-1 was known, have indicated those irreducible trinomials which were found to be primitive. Zierler (1969) adding mainly to work done by Rodemich and Rumsey (1968), has catalogued all primitive trinomials for the first 23 Mersenne primes. Finally, Zierler (1970) has shown that the trinomial x^n+x+1 is primitive for n=2, 3, 4, 6, 7, 15, 22, 60, 63, 127, 153, and 532.

3. M-SEQUENCE ARCHITECTURE

3.1 Introductory Remarks

Once we have a primitive polynomial we can construct a sequential machine out of memory elements and exclusive-or gates which will exhibit a cycle of 2^n-1 distinct states. Our first construction is a "natural" construction and to do it we use elementary matrix theory. As an introduction, consider the following statement by Garrett Birkhoff (1964, p. 299):

> "Each square matrix A has a determinant; though the determinant can be used in the elementary study of the rank of a matrix and in the solution of simultaneous linear equations, its most essential application in matrix theory is to the definition of the characteristic polynomial of a matrix."

The characteristic polynomial of a matrix, M, is, of course, obtained by evaluating the determinant

$$|M + \lambda I|. \tag{A-17}$$

One magnificent result from matrix theory is the Cayley-Hamilton theorem. Simply stated by Perlis (1952, p. 136), "Every (square) matrix satisfies its characteristic equation." The characteristic equation is created by simply setting the characteristic polynomial (A-17) to zero:

$$|M+\lambda I_n| = \lambda^n + \lambda^{n-1} c_{n-1} + \ldots + \lambda c_1 + c_0 = 0. \tag{A-18}$$

By the Cayley-Hamilton theorem, then, we have

$$M^n + c_{n-1} M^{n-1} + \ldots + c_1 M + c_0 I_n = 0 \quad . \tag{A-19}$$

We observe that (A-19) guarantees (constructively) that powers of M greater than or equal to n can be expressed by a linear combination of the set of matrices

$$\{I_n, M, M^2, \ldots, M^{n-2}, M^{n-1}\} . \tag{A-20}$$

If the characteristic polynomial is primitive then each member of the set

$$\{I_n, M, M^2, \ldots, M^{2^n-2}\} \tag{A-21}$$

will be distinct and $M^{2^n-1}=I_n$.

Given a primitive polynomial

$$f(\lambda)=\lambda^n+\lambda^{n-1}c_{n-1}+\ldots+\lambda c_1+1 \tag{A-22}$$

we can immediately devise a matrix whose characteristic polynomial will be $f(\lambda)$. This matrix is termed a companion matrix and is defined as

$$M_c = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ & & & c_1 \\ & & & c_2 \\ & I_{n-1} & & \vdots \\ & & & \\ & & & c_{n-1} \end{pmatrix} . \tag{A-23}$$

Observe that M_c is merely the state transition matrix for the most common realization of an m-sequence generator shown in Figure A-1. If we denote the content of the n-stage shift register's stage i at time t as b_i^t and let $\underline{b}^t=(b_{n-1}^t, b_{n-2}^t, \ldots b_0^t)$, then

$$\underline{b}^{t+1}=\underline{b}^t M_c. \tag{A-24}$$

We must provide a note of caution. It has become traditional to list primitive polynomials as polynomials in x, i.e., f(x). To create a matrix of the form (A-23) which will implement the sequence corresponding to these polynomials we must make the transformation $f(x)\rightarrow\lambda^n f(\frac{1}{\lambda})$ to convert f(x) to the polynomial of form (A-22), see (Golomb, 1967, p. 35). Thus, if we wish to construct the companion matrix for the primitive polynomial x^3+x^2+1 we would use $\lambda^3(\frac{1}{\lambda^3}+\frac{1}{\lambda^2}+1)=\lambda^3+\lambda+1$ as the characteristic polynomial.

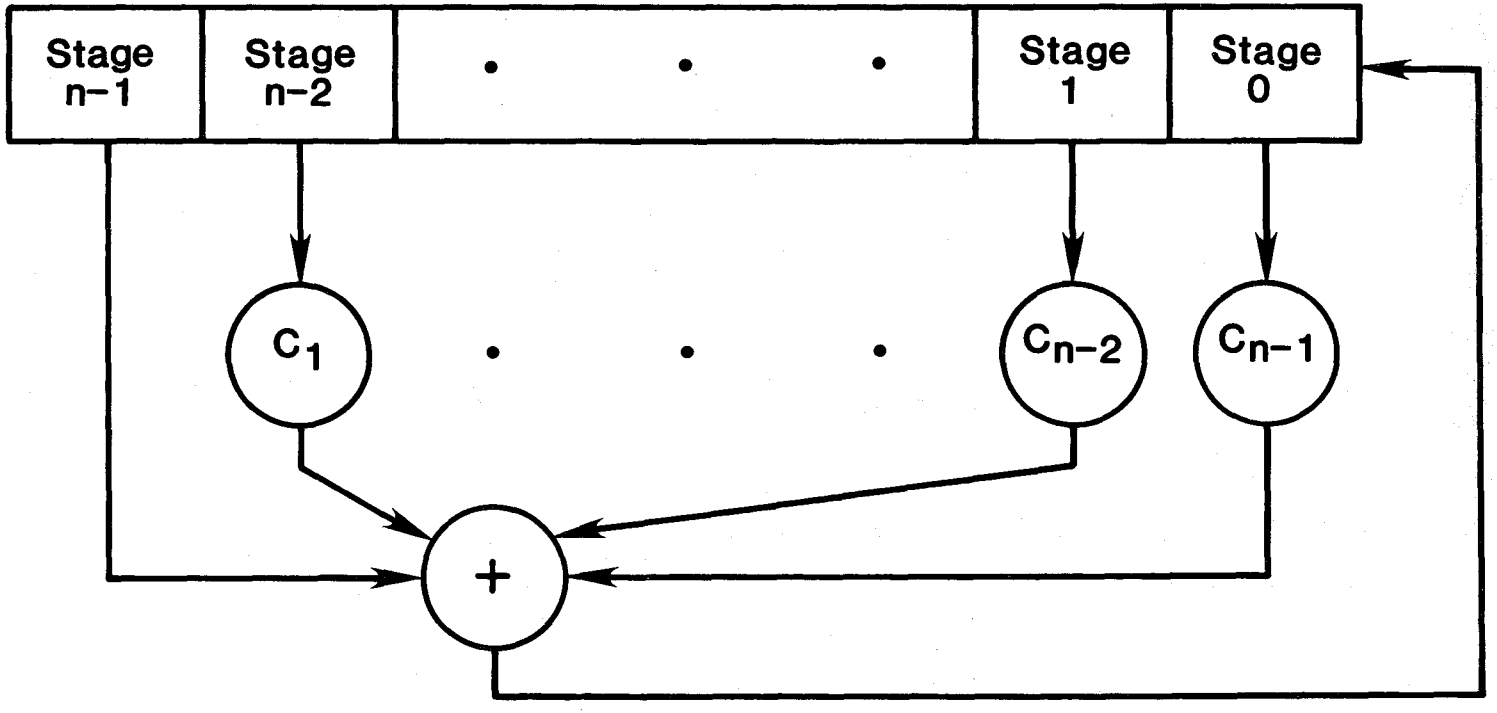


Figure A-1. The companion matrix structure.

It is at this juncture that the pure mathematician loses interest for, as he correctly asserts, there is only one finite field of 2^n elements and the set of elements

$$\{I_n, M_c, M_c^2, \ldots M_c^{2^n-2}\} \tag{A-25}$$

is "as good as" any other choice as all finite fields of the same order are isomorphic. See, for example (MacDuffie, 1940, p. 180). But the isomorphisms of the Galois field can be of significant practical importance and should not be dismissed as mere mathematical curiosities. The following from Berlekamp (1968, p. 104) captures the essence of this thought:

> "From an engineering standpoint, it is misleading to overstress the uniqueness of $GF(p^k)$, for this field may have many different representations.... The design and cost of circuitry to perform calculations in $GF(p^k)$ depend critically on the representation. For this reason, some engineers prefer to think of different representations as different fields. This viewpoint is particularly justified in solutions where the cost of transforming from one representation to another is large."

Well, just how many field representations are there and how can they help us? Consider the general 3X3 matrix:

$$E = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{pmatrix} . \tag{A-26}$$

If we take the determinant of $E+\lambda I$ we obtain:

$$\lambda^3+\lambda^2(e_{11}+e_{22}+e_{33})+\lambda(e_{11}e_{22}+e_{11}e_{33}+e_{22}e_{33}+e_{23}e_{32}+e_{12}e_{21}+e_{13}e_{31})$$
$$+(e_{11}e_{22}e_{33}+e_{11}e_{23}e_{32}+e_{12}e_{21}e_{33}+e_{12}e_{23}e_{31}+e_{13}e_{21}e_{32}+e_{13}e_{22}e_{31}) . \tag{A-27}$$

There are two primitive polynomials of degree 3, viz

$$\lambda^3+\lambda+1, \ \lambda^3+\lambda^2+1 . \tag{A-28}$$

Direct solution of the $\{e_{ij}\}$ for these cases yielding the polynomials in (A-28) uncovers no fewer than 48 distinct matrices which are arrayed in eight fields each with 2^3 members (each field contains I_3 and 0_3, the multiplicative and

additive identities, respectively, which do not, of course, exhibit a primitive characteristic polynomial). The eight fields are shown in Figure A-2.

The counting problem has been solved for general n. Reiner (1961) has derived an expression for the number of matrices that exhibit a particular characteristic polynomial. Following a little manipulation of Reiner's result, we find that the number of nxn matrices that possess a specific primitive characteristic polynomial is

$$\frac{R(n)}{2^n-1} \tag{A-29}$$

where R(n) is the number of regular or non-singular nxn matrices, i.e., the number of matrices whose determinant is unity. Explicitly,

$$R(n)=2^{n^2}\left(\frac{1}{2}\right)\left(\frac{3}{4}\right)\left(\frac{7}{8}\right)\ldots\frac{2^n-1}{2^n} \quad . \tag{A-30}$$

Recalling that there are $\frac{\phi(2^n-1)}{n}$ primitive polynomials of degree n, we find that the number of matrices that can serve as finite field generators, or equivalently, "wiring schematics" for m-sequence generators, is

$$c(n) = \frac{\phi(2^n-1)}{n} \cdot \frac{R(n)}{2^n-1} \quad . \tag{A-31}$$

Note that for Mersenne primes (p and 2^p-1 both prime), $c(p)\rightarrow \frac{R(p)}{p}$. Table A-1 demonstrates just how very rich the potential architectural schematisms are.

3.2 An Example and Its Analysis Via Matrices

As an example of a specific architecture, different from the companion matrix structure of Figure A-1, let us consider and analyze the following machine:

a) There are n flip-flop memory elements. Their states at time t are denoted by $b^t_{n-1}\ldots b^t_0$.

b) Their states at time t+1 are derived as follows. We add b^t_0 to b^t_{n-1}. The result is b^{t+1}_{n-1}. We then add b^{t+1}_{n-1} to b^t_{n-2}. The result is b^{t+1}_{n-2}. We add b^{t+1}_{n-2} to b^t_{n-3}. The result is b^{t+1}_{n-3}. We proceed in this fashion until we have attained b^{t+1}_0 by adding b^{t+1}_1 to b^t_0.

	FIELD A	FIELD B	FIELD C	FIELD D	FIELD E	FIELD F	FIELD G	FIELD H
0_3	000 000 000	000 000 000	000 000 000	000 000 000	000 000 000	000 000 000	000 000 000	000 000 000
M	111 110 100	111 110 011	111 101 100	111 101 011	111 100 110	111 100 101	111 011 110	111 011 101
M^2	101 001 111	010 001 101	110 011 111	001 100 110	101 111 011	110 111 010	010 101 100	001 110 010
M^3	011 100 101	110 011 100	010 001 110	011 111 010	001 101 010	011 110 100	011 001 111	101 100 011
M^4	010 111 011	001 101 111	101 100 010	110 001 101	110 001 100	001 011 111	101 110 010	010 111 110
M^5	110 101 010	011 100 010	011 111 101	010 011 100	011 110 111	101 001 110	001 100 011	011 001 100
M^6	001 011 110	101 111 110	001 110 011	101 110 111	010 011 101	010 101 011	110 111 101	110 101 111
$M^7=I_3$	100 010 001	100 010 001	100 010 001	100 010 001	100 010 001	100 010 001	100 010 001	100 010 001

Figure A-2. The eight field representations.

Table A-1. Architectural 'Richness'

n	$\frac{\phi(2^n-1)}{n}$	$\frac{R(n)}{2^n-1}$	c(n)
2	1	2	2
3	2	24	48
4	2	1,344	2,688
5	6	322,560	1,935,360

A little thought will convince that

$$\underline{b}^{t+1}=(b^t_{n-1},b^t_{n-2},\ldots,b^t_0) = \underline{b}^t M \tag{A-32}$$

where

$$M = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 1 \\ 0 & 1 & 1 & \ldots & 1 & 1 & 1 \\ 0 & 0 & 1 & \ldots & 1 & 1 & 1 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \ldots & 0 & 1 & 1 \\ 1 & 1 & 1 & \ldots & 1 & 1 & 0 \end{pmatrix} . \tag{A-33}$$

To find that characteristic polynomial of M given in (A-33) we write the determinant

$$f_n(\lambda)= \begin{vmatrix} \lambda+1 & 1 & 1 & \ldots & 1 & 1 & 1 \\ 0 & \lambda+1 & 1 & \ldots & 1 & 1 & 1 \\ 0 & 0 & \lambda+1 & \ldots & 1 & 1 & 1 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \ldots & 0 & \lambda+1 & 1 \\ 1 & 1 & 1 & \ldots & 1 & 1 & \lambda \end{vmatrix} . \tag{A-34}$$

Expanding (A-34) by evaluating the minors specified by the two non-zero entries of the first column and using simple matrix algebra we obtain the following recursion:

$$f_n(\lambda)=(\lambda+1)f_{n-1}(\lambda)+\lambda^{n-2} . \tag{A-35}$$

By direct computation we find that

$$f_2(\lambda)=\lambda^2+\lambda+1 \tag{A-36}$$

and then by recursive computation we find

$$\begin{aligned} f_3(\lambda)&=\lambda^3+\lambda+1 \\ f_4(\lambda)&=\lambda^4+\lambda^3+1 . \end{aligned} \tag{A-37}$$

Thus we know that the finite state linearly sequential machine defined by (A-32) and (A-33) will exhibit a maximum length cycle of states if and only if (A-35) is primitive.

Let us examine the succession of states for one of the primitive polynomials, $f_4(\lambda)=\lambda^4+\lambda^3+1$. We start the machine in the all ones state and observe the following state progression:

1 1 1 1
0 1 0 1
1 0 0 1
0 0 0 1*
1 1 1 0
1 0 1 1
0 0 1 0*
0 0 1 1
1 1 0 1
0 1 1 0
0 1 0 0*
0 1 1 1
1 0 1 0
1 1 0 0
1 0 0 0*

The unity density states are starred and we note that their positions seem to be at approximately equidistant spacings throughout the cycle. Will this be true in general for those cases in which $f_n(\lambda)$ is primitive, or is it merely fortuitous for this case?

First, let us define the density one or unit weight vectors as

$$u_i=(00\ldots010\ldots0) \tag{A-38}$$

where the single 1 is in the position i and $1\leq i\leq n$. Second, consider the immediate successor states of the unit weight vectors. On multiplying u_n by M we obtain:

$$u_nM=u_1+u_2+\ldots+u_{n-1} \tag{A-39}$$

where the sums in (A-39) are vector (modulo two sums, component by component). Similarly, multiplying M by u_1 yields:

$$u_1M=u_1+u_2+\ldots+u_{n-1}+u_n \tag{A-40}$$

Summing (A-39) and (A-40) we get:

$$u_nM+u_1M=u_n \tag{A-41}$$

In a manner similar to the above we also derive the equation:

$$u_1M+u_2M=u_1 \quad . \tag{A-42}$$

Now suppose that u_n and u_1 are on the same cycle, which, of course, they will be if $f_n(\lambda)$ is primitive. There then exists an integer d such that

$$u_1=u_nM^d \tag{A-43}$$

Now, if we add (A-41) to (A-42) we obtain:

$$u_nM+u_2M=u_n+u_1 \tag{A-44}$$

Because the inverse of M exists we can convert (A-44) to

$$u_2+u_n+u_nM^{-1}=u_1M^{-1} \tag{A-45}$$

On substituting (A-43) into (A-45) we get

$$u_2+u_n+u_nM^{-1}=u_nM^{d-1} \tag{A-46}$$

Post-multiplying (A-41) by M^{-1} we find that

$$u_n+u_nM^{-1}=u_1 \tag{A-47}$$

Substituing (A-47) into (A-46) we get:

$$u_2+u_1=u_nM^{d-1} \tag{A-48}$$

Substituting (A-43) for u_1 in (A-48) we get:

$$u_2=u_nM^d+u_nM^{d-1}=(u_n+u_nM)M^{d-1} \tag{A-49}$$

Using (A-41) we can rewrite (A-49) as:

$$u_2=(u_1M)M^{d-1}=u_1M^d \tag{A-50}$$

Using (A-43) we immediately rewrite (A-50) as:

$$u_2=u_nM^{2d} \tag{A-51}$$

Equation (A-51) demonstrates that u_2 is the same distance from u_1 as u_1 is from u_n. For our example, we note that this distance is 11 steps. We can easily extend the above argument for all u_i up to i=n-1, i.e., if $u_1=u_nM^d$, then $u_2=u_nM^d$, $u_3=u_2M^d,\ldots,u_{n-1}=u_{n-2}M^d$. The final unit weight vector, u_n, however, is found to satisfy

$$u_n=u_{n-1}M^{d+1} \tag{A-52}$$

Thus, we have found that for the particular sequential machine defined by (A-32) and (A-33), the unit weight vectors are distributed at approximately equal distances around the cycle. This fact can be deduced without the matrix oriented argument presented but is intended as an example of how matrix arguments can be simply and efficaciously used.

3.3 Special Purpose Architectures

As we have seen in Figure A-1, the companion matrix shift register realization is the simplest realization of an m-sequence generator. It may not be the 'best,' however; that all depends on what constitutes value to the designer or implementer. As an example, let us assume that we wish to realize an m-sequence of period 255. As we have noted previously, Swan's criterion states that there are no primitive trinomials of degree n where n is divisible by 8. Thus an implementation of the form shown in Figure A-1 will require more than one two-input exclusive-or logic gate. As an example, let us choose the primitive pentanomial $x^8+x^6+x^5+x^3+1$. It may be implemented as shown in Figure A-3. Note that this implementation requires three modulo-two adders. More important, the adders are layered to a depth of two. The exclusive-or boolean function is not threshold realizable and, consequently, is often the time-limiting basic element in a logic family. How then can we obviate this annoying layering of relatively slow logic?

The answer can often be found through special architectures. Scholefield (1960) considers a variety of interconnected, parallel-clocked structures. For example, he shows that each stage of the structure shown in Figure A-4 exhibits the recursion specified by the polynomial $x^{p+q}+x^{p+v}+x^{q+u}+x^{u+v}+1$. By properly selecting the tetrad (p,q,u,v) it is possible to synthesize some polynomials in many ways with this particular structure. For our example, we can effect the realization of the pentanomial $x^8+x^6+x^5+x^3+1$ with (p,q,u,v)=(5,3,2,1) as shown in Figure A-5. The realization presented in Figure A-5 has two advantages over the realization presented in Figure A-3. First, there is one less exclusive-or gate required. Second, and perhaps more important, there is no layering of exclusive-or gates.

3.4 A Curious Architectural Property

Recall that the Cayley-Hamilton theorem (A-19) requires that each element of M, the finite field generator matrix, exhibit the same recursion as the matrix

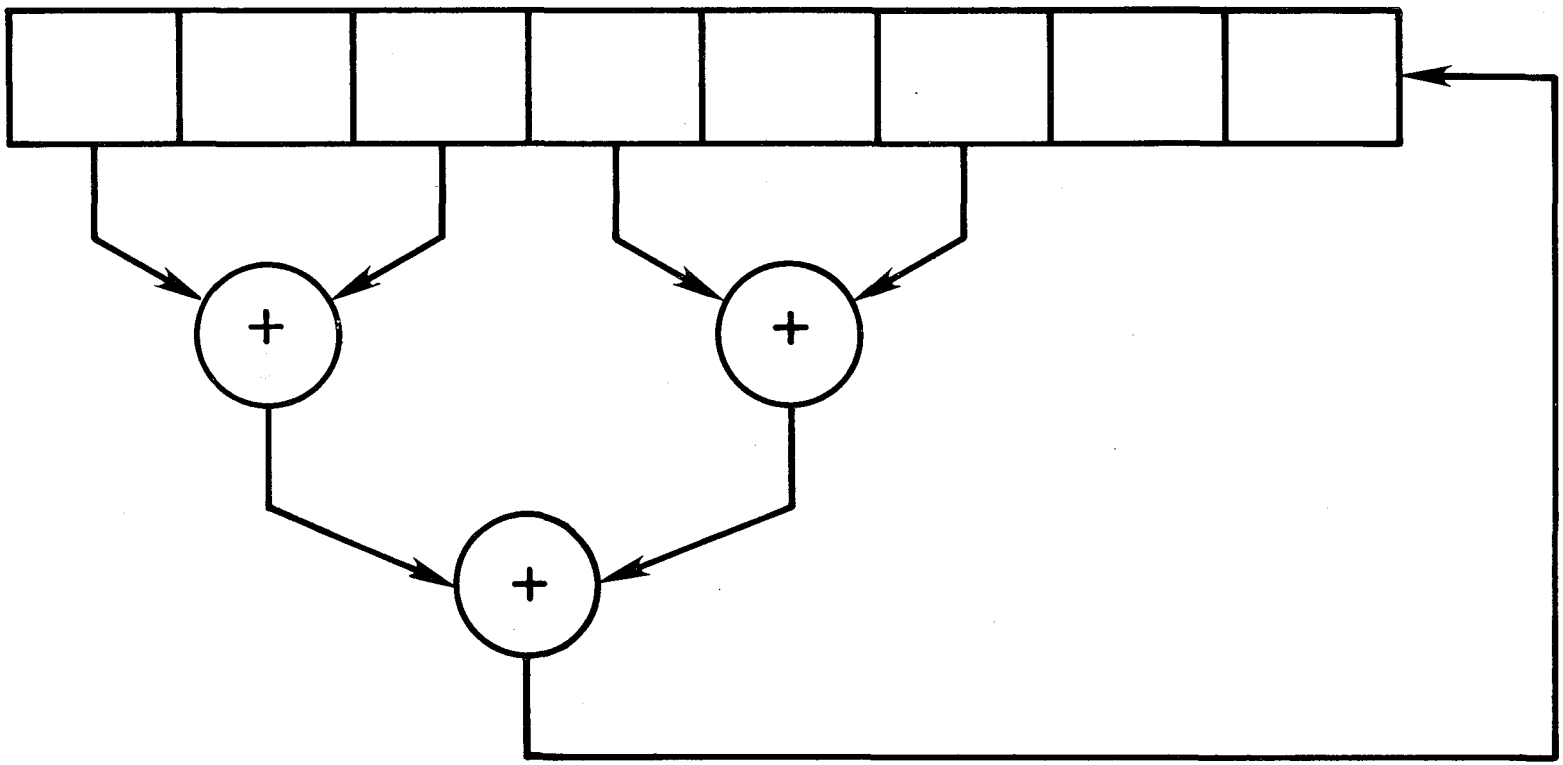


Figure A-3. Implementation of $x^8+x^6+x^5+x^3+1$.

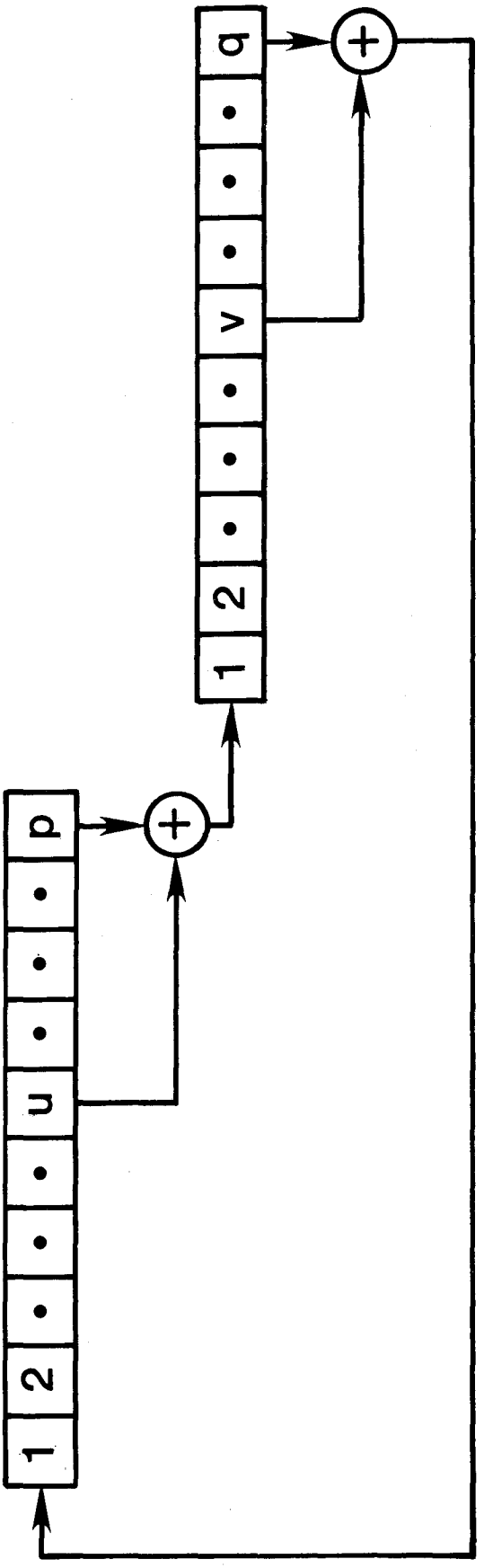


Figure A-4. Parallel clocked structure.

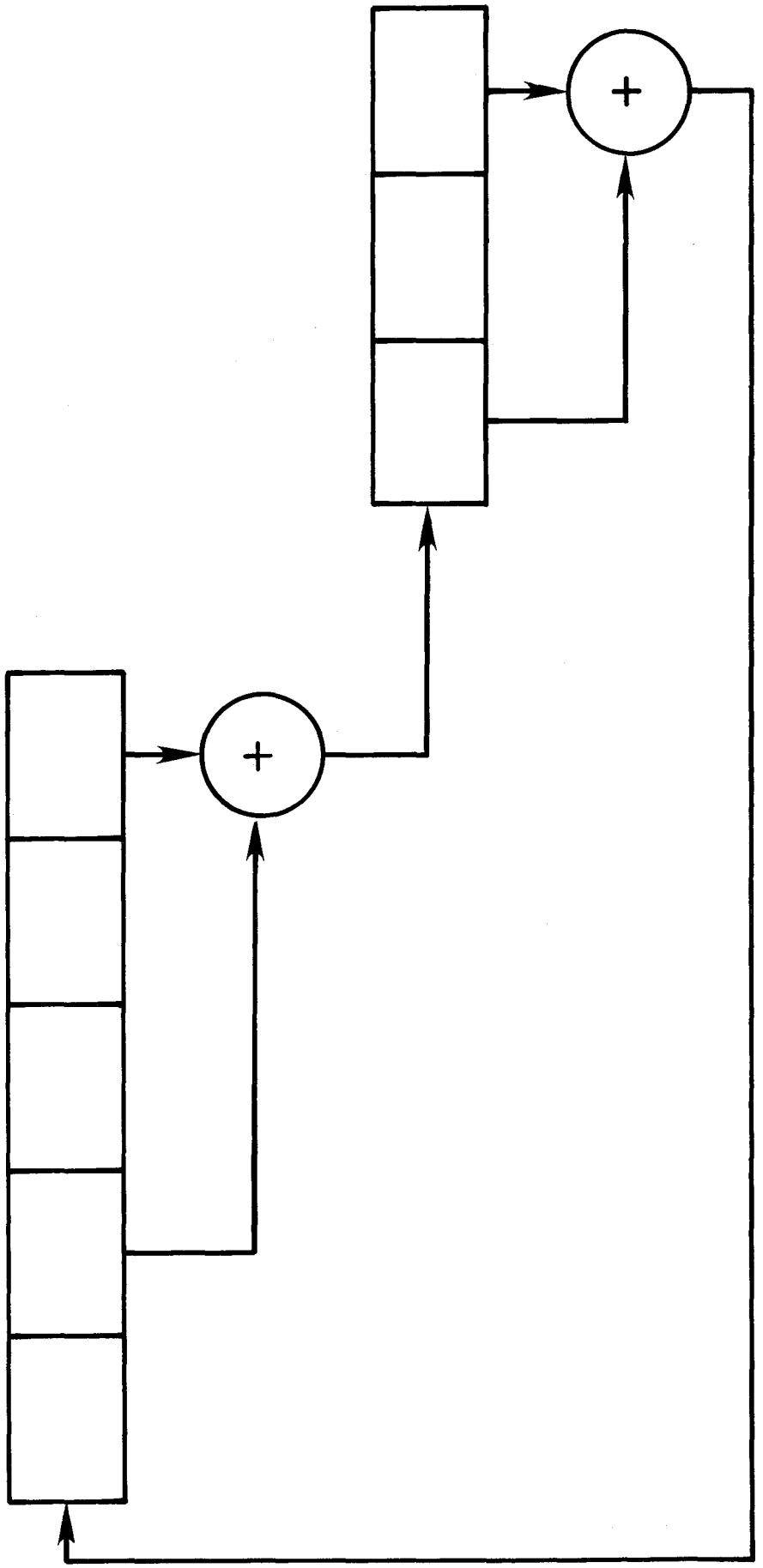


Figure A-5. Realization of $x^8+x^6+x^5+x^3+1$ by parallel clocked structure.

as a whole as its successive powers are computed. In later sections, it will become clear that the n sequences that describe any row or column of successive powers of M will be independent of each other, i.e., no term-by-term sums of up to n-1 of any of the row or column sequences will yield the nth remaining row or column sequence. These facts immediately yield the following theorem: Every power of M is expressible as the following matrix, i.e., the following form is invariant over exponentiation:

$$\begin{pmatrix} m_1 & m_2 & \cdots & m_n \\ L_{2,1}(\underline{m}) & L_{2,2}(\underline{m}) & \cdots & L_{2,n}(\underline{m}) \\ \vdots & \vdots & & \vdots \\ L_{n-1,1}(\underline{m}) & L_{n-1,2}(\underline{m}) & \cdots & L_{n-1,n}(\underline{m}) \end{pmatrix} . \quad \text{(A-53)}$$

The $m_i \varepsilon \{0,1\}$. The $L_{i,j}(\underline{m})$ are linear combinations of $m_1, m_2, \ldots m_n$.

As an example, consider FIELD B of Figure A-2. By solving some elementary linear equations, the general form (A-53) of the matrices in the field is easily shown to be

$$\begin{pmatrix} m_1 & m_2 & m_3 \\ m_3 & m_1 & m_2+m_3 \\ m_2+m_3 & m_3 & m_1+m_2+m_3 \end{pmatrix} . \quad \text{(A-54)}$$

Notice that all of the FIELD B matrices can be derived by letting (m_1, m_2, m_3) assume all possible $2^3=8$ binary triples.

4. m-SEQUENCE MANIPULATIONS

4.1 The 'Shift and Add' Property

One of the most celebrated properties of an m-sequence is the so-called "shift and add property." It deserves study not only because it is of theoretical interest but also because it lies at the heart of special architectural techniques. Essentially the shift and add property states that if an m-sequence is added, term-by-term, to a shift or phase of itself, the resulting sequence will be the same m-sequence but at yet another shift of itself. For example, consider the sequence generated by the primitive polynomial x^3+x^2+1: {1001011}. Let us delay the sequence by one clocktime: {1100101}. Adding these two sequences, we obtain: {0101110} which is the first sequence delayed by five clocktimes.

The proof of the shift and add property is simple and it is instructive to show it by two different methods, via algebra and via matrices. First, consider that the linear recurrence generating the m-sequence is the same as used in (A-13), viz,

$$s(t)=\sum_{i=1}^{n} \alpha_i s(t-i) \tag{A-55}$$

Let us create a set of 2^n elements whose members are the set of sequences:

$$\begin{aligned}
&s(1),s(2),\ldots s(2^n-1)\\
&s(2),s(3),\ldots s(1)\\
&s(3),s(4),\ldots s(2)\\
&\quad\vdots\\
&s(2^n-1),s(1),\ldots s(2^n-2)\\
&0,0,\ldots\ldots\ldots 0
\end{aligned} \tag{A-56}$$

Each of the 2^n sequences in (A-56) satisfies (A-55), and because (A-55) generates an m-sequence, we know that all n-tuples excepting the all zero n-tuple exist somewhere in the sequence $s(1),s(2),\ldots s(2^n-1)$. We realize, then, that all possible n-tuples, including the all zero n-tuple, exist as the first n bits of one of the sequences in (A-56). Thus we claim that all possible sequences or solutions to (A-55) are present in the set (A-56). These sequences or solutions are rotations or phases of each other. Now consider the term-by-term sum of any two of the sequences in (A-56). Let these sequences be denoted by $\{a_1,a_2,\ldots a_{2^n-1}\}$ and $\{b_1,b_2,\ldots b_{2^n-1}\}$. Because $a(t)=\sum_{i=1}^{n} \alpha_i a(t-i)$ and $b(t)=\sum_{i=1}^{n} \alpha_i b(t-i)$ linearity assures that the term-by-term sum sequence $\{a_1+b_1,a_2+b_2,\ldots a_{2^n-1}+b_{2^n-1}\}$ also satisfies the recursion and is therefore contained in (A-56) thus proving the shift and add property. Furthermore, it is clear that the set of sequences in (A-56) forms an abelian group under the operation of term-by-term addition. This proof and observation is given by Golomb (1967, pp. 44-45).

Weathers (1972, pp. 13, 15) has given a matrix proof. Assume that two m-sequence generators have the same state transition matrix, M. Assume that one machine is started with the initial state $\underline{b}^0$ and that the other machine started at d clock times away from $\underline{b}^0$, i.e., at state $\underline{b}^0M^d$. If an observer were to sum the contents of identical stages at each clock time, he would observe the sequence

$$\underline{b}^0+\underline{b}^0M^d,\ \underline{b}^0M+\underline{b}^0M^{d+1},\ \underline{b}^0M^2+\underline{b}^0M^{d+2},\ldots \tag{A-57}$$

which we can rewrite as

$$b^0(I+M^d),\ b^0(I+M^d)M,\ b^0(I+M^d)M^2,\ldots \tag{A-58}$$

But, as Weathers points out, (A-58) is equivalent to starting the machine at the state $\underline{\beta}^0=\underline{b}^0(I+M^d)$ and observing the sequence $\underline{\beta}^0$, $\underline{\beta}^0M$, $\underline{\beta}^0M^2$,...which is either the all zero sequence or a rotation or phase shift of the m-sequence.

4.2 Phase Shifts and the Delay Operator Calculus

Consider again an m-sequence realized by the companion matrix of Figure A-1. For specifics, let us consider the shift register arrangement of Figure A-6 corresponding to the primitive trinomial x^3+x^2+1. We have labelled the stages 0, 1, and 2. Consider that the observed m-sequence is taken from stage 0. Let D be the unit delay operator applied to a sequence. If the register of Figure A-6 is started with all ones, the following m-sequence is observed

$$S_0=\{1001011\}. \tag{A-59}$$

As stage 1, the following sequence is observed

$$S_1=\{1100101\} \tag{A-60}$$

which is D operating on (A-59). At stage 2 we observe

$$S_2=\{1110010\} \tag{A-61}$$

which is D^2 operating on (A-59). Consider, now, that we have a set of three switches $\{s_0,s_1,s_2\}$ as shown in Figure A-7. Table A-2 shows the sequences which are produced at point Σ according to the eight possible switch configurations. Note that the set of eight sequences of Table A-2 contain all seven phase shifts of the m-sequence produced by the primitive polynomial x^3+x^2+1 and the all-zero sequence. This set is of the form (A-56). The noteworthy point here is that all phase shifts are present and can be generated by summing, modulo two, various combinations of the three stages of the machine shown in Figure A-6. That this is true in general, that any phase of a 2^n-1 bit m-sequence can be formed by a linear combination of stages of the n-stage companion matrix shift register realization, is an amazing and useful property. [Tsao (1964) gives an excellent, elementary argument proving this.] Also implied is uniqueness, i.e., no two linear combinations of stages

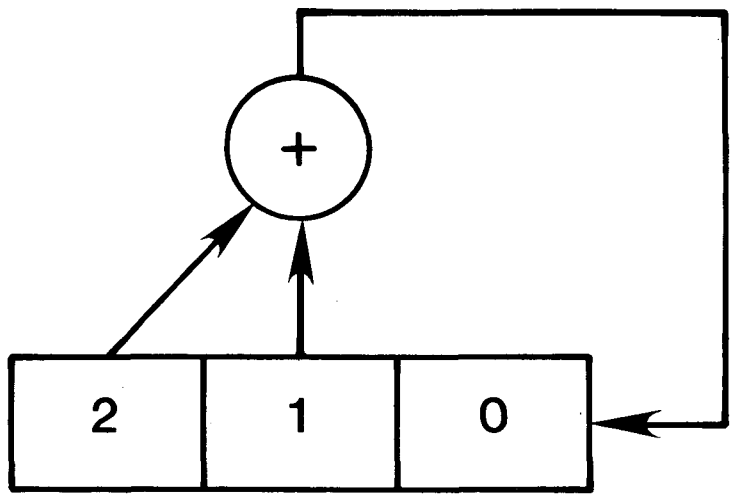


Figure A-6. Realization of x^3+x^2+1.

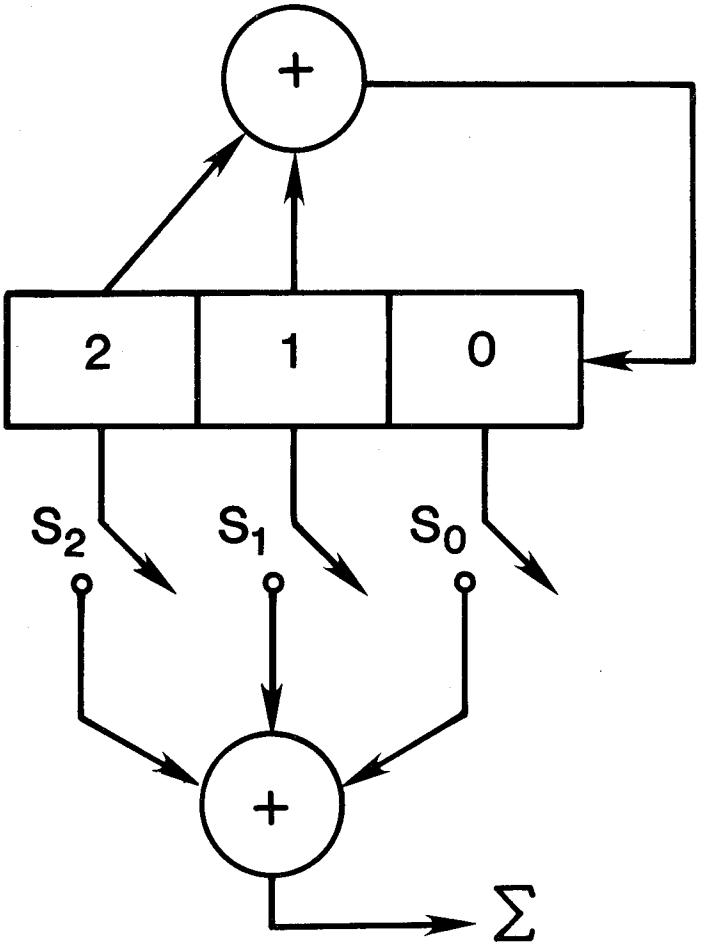


Figure A-7. Phase switchable m-sequence generator.

Table A-2. All Phase Shifts of x^3+x^2+1

S_0	S_1	S_2	Σ
0	0	0	0 0 0 0 0 0 0
0	0	1	1 1 1 0 0 1 0
0	1	0	1 1 0 0 1 0 1
0	1	1	0 0 1 0 1 1 1
1	0	0	1 0 0 1 0 1 1
1	0	1	0 1 1 1 0 0 1
1	1	0	0 1 0 1 1 1 0
1	1	1	1 0 1 1 1 0 0

will produce the same phase. This follows immediately from the 'pigeon-hole principle' (Schubeinfachprinzip) as there are $\sum_{i=0}^{n} \binom{n}{i} = 2^n$ possible linear combinations and 2^n possible phases including, of course, the all-zero sequence.

We can now move naturally into the delay operator calculus. Observe that the machine in Figure A-8 obeys the primitive polynomial x^4+x^3+1. Observe that it can be viewed in terms of the delay operator notation by observing that

$$D^3\sigma+D^2\sigma=D^{-1}\sigma \tag{A-62}$$

where σ is the m-sequence observed at stage 0. Rewriting (A-62) as

$$(D^4+D^3+1)\sigma=0 \tag{A-63}$$

we see that the polynomial in x converts directly to a polynomial in D and this is true in general.

Davies (1965) presents the following algorithm to derive the numbers of the stages which must be added together to achieve a phase delay of d steps:

a) Divide the degree n primitive polynomial, written left-to-right with terms of decreasing powers of D, into D^d.

b) Continue division until the remainder consists of powers of D all of which are less than n.

c) The powers of D in the remainder correspond to those stages that are to be summed modulo two.

For example, let us say that we wish to generate the m-sequence according to the primitive polynomial x^4+x^3+1 and that we simultaneously wish to generate the sequence delayed by six steps. Following Davies' algorithm we proceed as follows:

$$\begin{array}{r|l} D^4+D^3+1 & D^2+D+1 \\ \hline & D^6 \\ & D^6+D^5 \quad +D^2 \\ \hline & D^5 \quad +D^2 \\ & D^5+D^4 \quad +D \\ \hline & D^4 \quad +D^2+D \\ & D^4+D^3 \quad +1 \\ \hline & D^3+D^2+D+1 \end{array}$$

The machine depicted in Figure A-8 will produce an m-sequence at point Σ that is delayed by six steps from the sequence observed at stage 0.

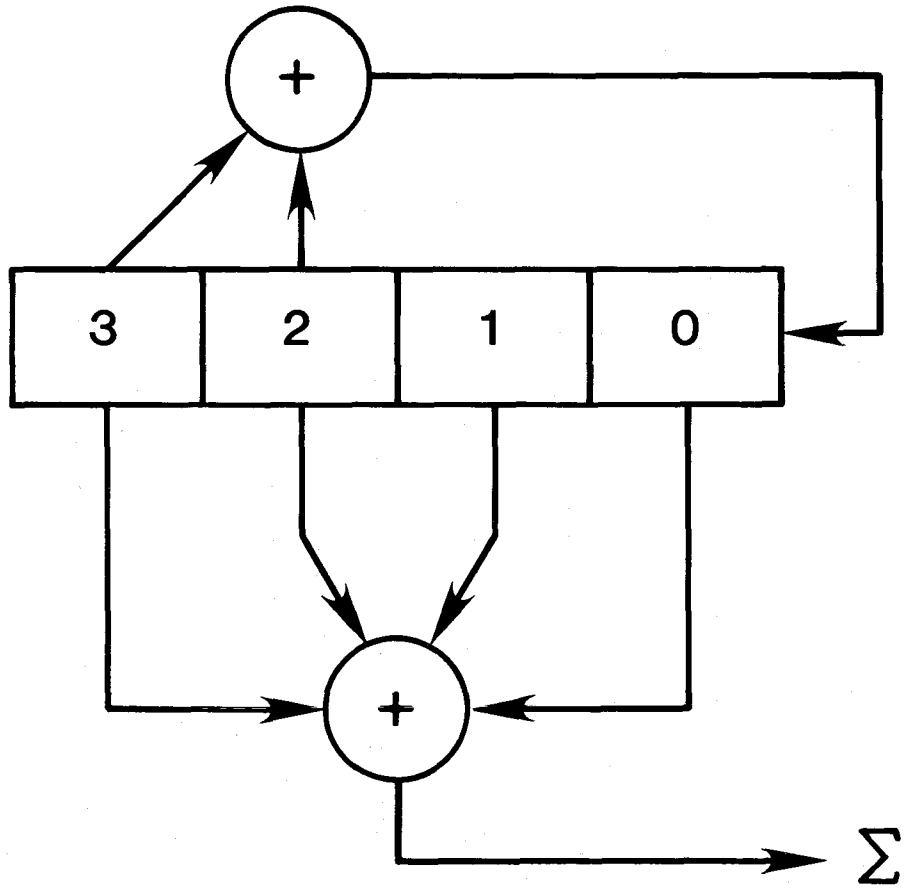


Figure A-8. A phase shift of x^4+x^3+1.

Douce (1968) and Davies (1968) worked the inverse problem, i.e., given the m-sequence generator polynomial and the stages of the companion matrix that are summed, determine the phase delay of the sum. This algorithm is also very simple:

a) Divide the degree n primitive polynomial, written left-to-right with terms of <u>increasing</u> powers of D, into the stage specifying powers of D, also written as a polynomial is ascending powers (this polynomial will be of degree <n).

b) Continue division until a one term remainder is obtained.

c) The exponent of D of the remainder above is the phase delay.

Using our previous example, we perform the following division:

$$
\begin{array}{r|l}
1+D^3+D^4 & 1+D+D^2 \\
\cline{2-2}
1+D+D^2+D^3 & \\
\underline{1\quad\quad +D^3+D^4} & \\
D+D^2\quad +D^4 & \\
\underline{D\quad\quad +D^4+D^5} & \\
D^2\quad\quad +D^5 & \\
\underline{D^2\quad\quad +D^5+D^6} & \\
D^6 &
\end{array}
$$

The first single-term remainder encountered is D^6 and our delay is therefore 6.

The above algorithms depend on the simple relation as expressed by Davies (1965):

$$D^d = f(D) \cdot q(D) + r(D) \qquad \text{(A-64)}$$

where f(D) is the primitive polynomial generating the m-sequence and q(D) and r(D) are the quotient and remainder polynomials, respectively. The 2^n possible residues, or equivalence classes that obtain upon the divsion correspond to either the all-zero sequence or the 2^n-1 possible phase delays.

Gardiner (1965) has devised, and Davis (1966) has further generalized, a sequential circuit that derives the numbers of the stages which must be added together to achieve a given phase shift delay.

4.3 Large Phase Shifts and the Art of Exponentiation

The preceding material is valuable in that it constructively demonstrates the calculability of the relation and inverse relation between phase shift and stage connections needed to be summed to achieve the phase shift. For large

phase shifts, however, Davies' method (1965) is not recommended as the computation time required to determine the stages to be summed grows linearly with the magnitude of the phase shift. Other proferred methods, such as (Ireland and Marshall, 1968a, 1968b, 1976) and (Latawiec, 1974, 1975, 1976) are complex and are not presented by the authors as efficient for handling very large phase delays. Only recently has a series of papers dealt with the problem of the computational complexity of calculating the stages required to achieve very large phase delays.

To review, what we are trying to determine is D^d modulo the primitive polynomial generating the m-sequence. Given d, how many multiplications are required to obtain D^d? The answer in general, as far as the authors are aware, is unknown. Knuth (1969, pp. 401+) considers the problem at length and presents the algorithm, slightly modified by the authors shown in Figure A-9 which he terms the 'binary algorithm,' for accomplishing the exponentiation $Y=D^d$. The binary algorithm requires $\lfloor \log_2 d \rfloor + s(d) - 1$ multiplications where s(d) is the number of ones in d's binary representation. As stated, the algorithm is not necessarily the 'cheapest' in terms of multiplications required for general d. Knuth cites d=15 as the smallest d for which there is a less costly procedure. The binary algorithm forms d^{15} from d with six multiplications. Let us, however, calculate d^{15} with only five multiplications as follows:

START:	$\delta \leftarrow D$	
	$\delta \leftarrow \delta^2$	(1 multiplication)
	$\delta \leftarrow \delta \cdot D$	(1 multiplication)
	$\gamma \leftarrow \delta$	(save D^3)
	$\delta \leftarrow \delta^2$	(1 multiplication)
	$\delta \leftarrow \delta^2$	(1 multiplication)
	$\delta \leftarrow \delta \cdot \gamma$	(1 multiplication)
END:	$\delta = D^{15}$	

Although the binary algorithm may not always be the cheapest in terms of multiplications required, it is easily programmed and its performance, in general, is quite good. Also, one must bear in mind that algorithms should not always be evaluated and selected by counting just one cost item, multiplications in this case. Total algorithmic complexity, and "convenience," depends upon many ancillary considerations such as storage requirements, indexing, sorting, and other housekeeping tasks.

The binary algorithm is quite useful for our task and is at the heart of a paper by VanLuyn (1978). VanLuyn's algorithm determines the stages to be summed

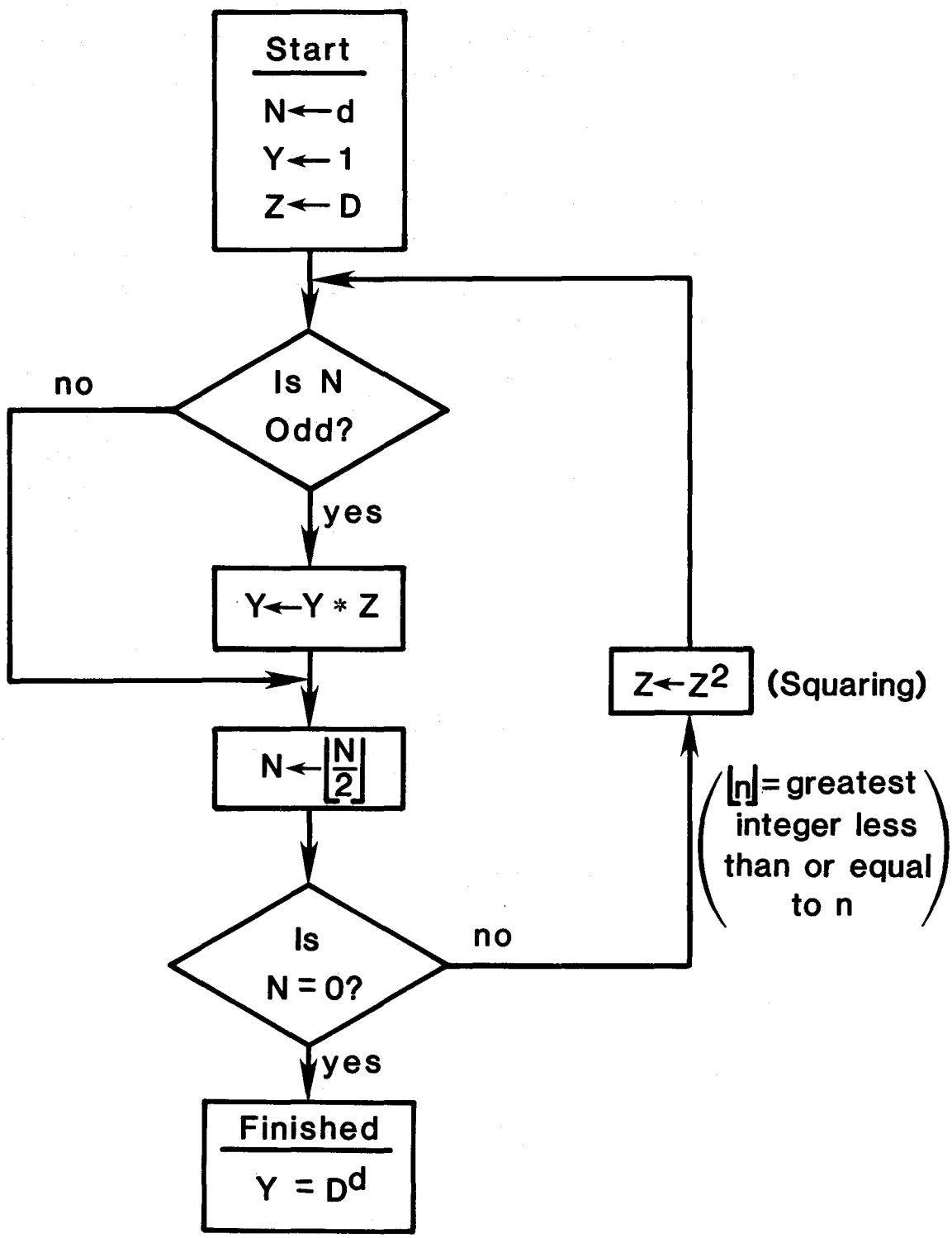


Figure A-9. The 'binary algorithm'.

to produce a phase shift of d steps. For consistency, we modify VanLuyn's algorithm slightly in order to comport with the definitions and architectures implied by the previous figures. The algorithm is motivated by the following: Assume that d is expressed in binary form as $d=a_0 2^0+a_1 2^1+a_2 2^2+\ldots+a_s 2^s$; then we have

$$D^d=D^{a_0+a_1 2^1+a_2 2^2+\ldots+a_s 2^s}=D^{a_0}(D^{a_1}(\ldots(D^{a_s})^2\ldots)^2)^2. \qquad \text{(A-65)}$$

But (A-65) is the recursive form of the binary algorithm. The embellishment needed is to reduce powers of D that equal or exceed n following each squaring and this can be done using Davies' (1965) long division method. VanLuyn's algorithm is then as shown in Figure A-10.

As an example of VanLuyn's algorithm, consider that we wish to determine the setting of the switches s_0,s_1,s_2,s_3,s_4 of the machine depicted in Figure A-11 so that the sequence at point Σ is delayed by 21 steps with respect to the sequence observed at stage 0. The machine of Figure A-11 obeys the recursion D^5+D^2+1 and the delay $s=2^0+2^2+2^4$, hence $a_0=1$, $a_1=0$, $a_2=1$, $a_3=0$, $a_4=1$. The algorithm proceeds as follows:

START: $r(d)=1$

$a_4=1$: $r(D)=D^{a_4}\cdot(1)^2+D$

$a_3=0$: $r(D)=D^{a_3}\cdot(D)^2=D^2$

$a_2=0$: $r(D)=D^{a_2}\cdot(D^2)^2=D^5$

$$\begin{array}{r|l} & 1 \\ \hline D^5+D^2+1 & D^5 \\ & \underline{D^5+\quad +D^2\quad +1} \\ & \qquad D^2\quad +1 = r(D) \end{array}$$

$a_1=0$: $r(D)=D^{a_1}\cdot(D^2+1)^2=D^4+1$

$a_0=1$: $r(D)=D^{a_0}\cdot(D^4+1)^2=D^9+D$

$$\begin{array}{r|l} & D^4 \quad +D \\ \hline D^5+D^2+1 & D^9 \qquad\qquad +D \\ & \underline{D^9 \quad +D^6 \quad +D^4} \\ & \qquad D^6 \quad +D^4 \quad +D \\ & \underline{D^6 \quad +D^3 \quad +D} \\ & \qquad D^4+D^3 \qquad =r(D) \end{array}$$

FINISHED:

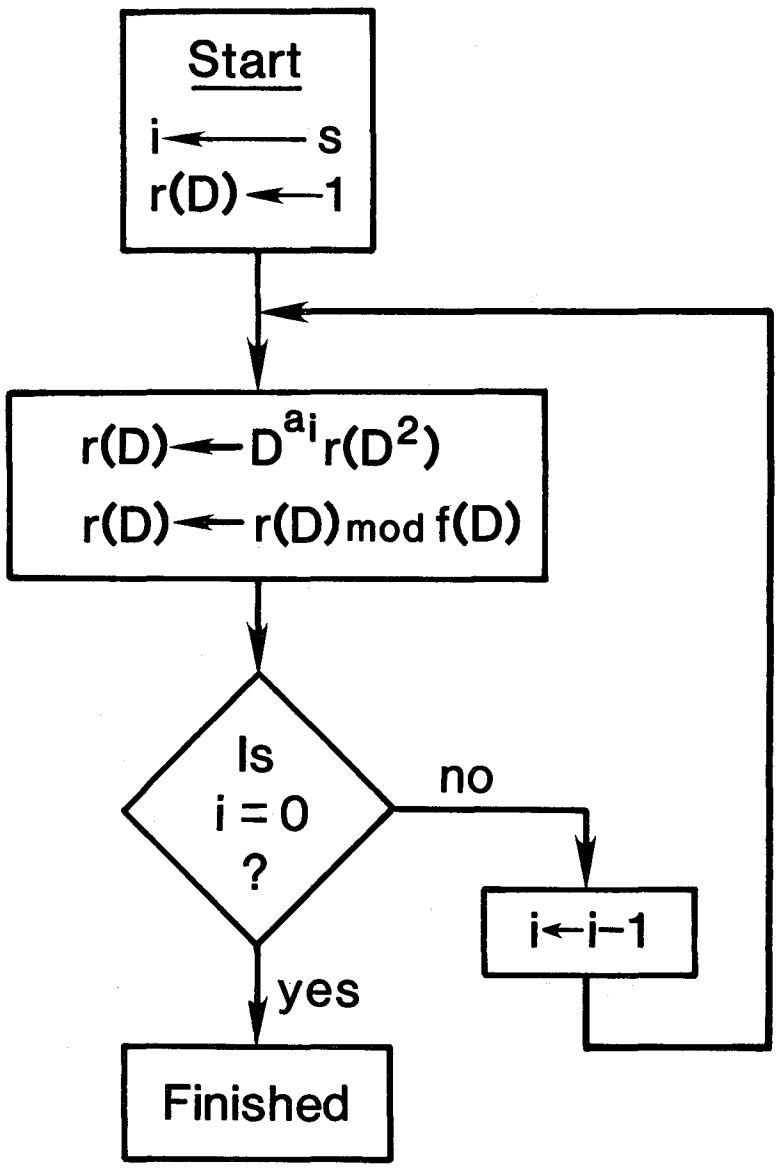


Figure A-10. VanLuyn's algorithm (modified).

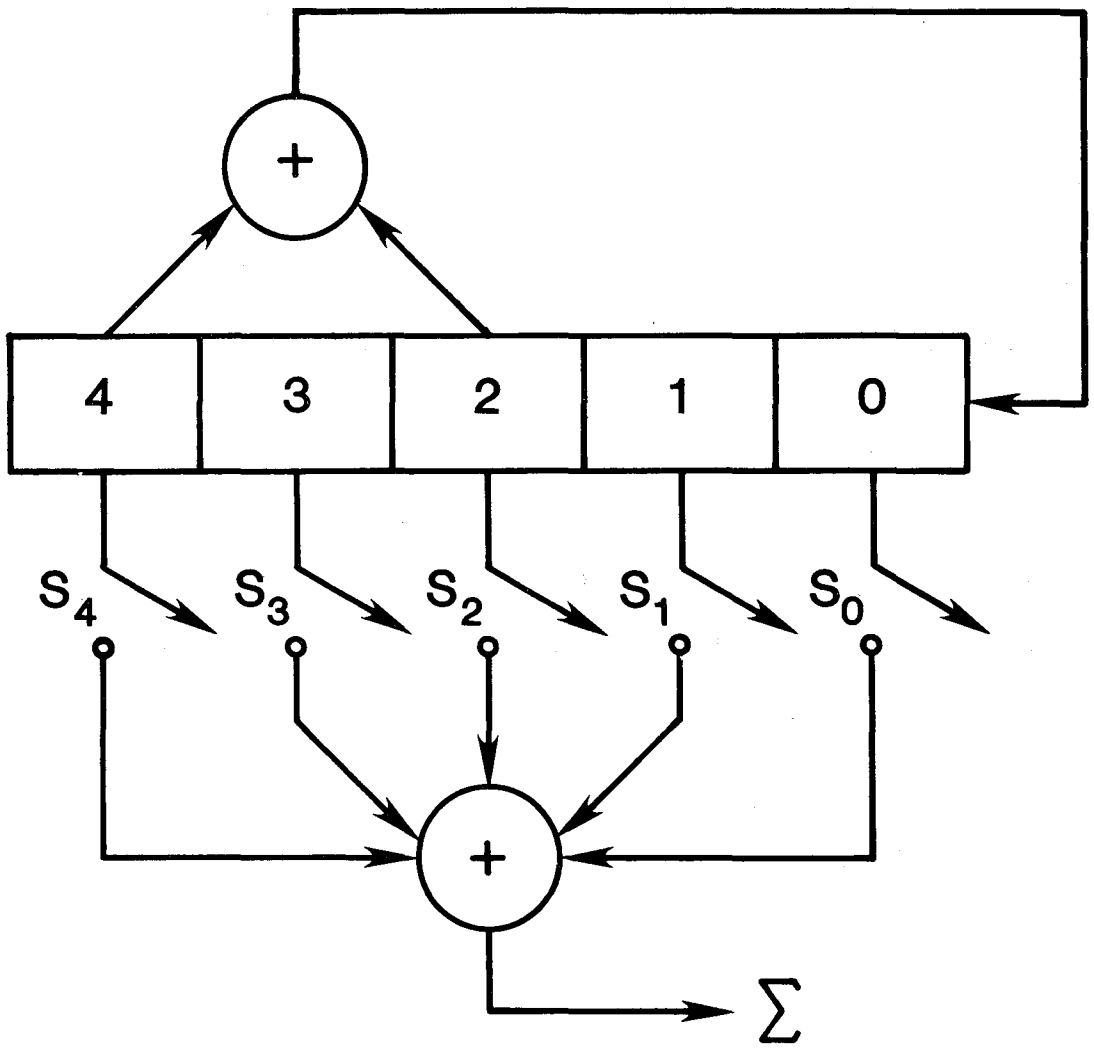


Figure A-11. Phase switchable generator for x^5+x^2+1.

Thus we see that if switches 3 and 4 are closed, the sequence at point Σ will be delayed by 21 steps with respect to the sequence observed at stage 0. The reader is encouraged to try Davies' method (1965) on the above example for two reasons: first to become convinced that the same results are achieved, and second to gain an appreciation of the computational advantage that is provided by Vanluyn's algorithm, log d versus d.

For the sake of completeness and history it should be noted that Roberts et al. (1965) developed a better than linear d method but did not put it into an easily manipulable form. Other authors have developed algorithms for special cases. Miller et al. (1977) have devised what is probably best captioned as a "coalescing pyramid" algorithmic structure that is efficacious when dealing with primitive trinomials. Hershey (1980) presented some timesaving shortcuts for those concerned with primitive trinomials of the form x^n+x+1 where n is a Mersenne prime. Finally, Yiu (1980) has also developed a fast algorithm that is similar to VanLuyn's.

4.4 Decimation

Decimation, the creation of a new sequence by the periodic sampling of an old, is a process of both great theoretical and practical import. Consider the m-sequence generated by the trinomial x^3+x+1:

1001110... (A-66)

Taking every second bit from (A-66) and starting from the zeroth, we get the sequence:

1010011... (A-67)

But (A-67) is merely a phase shift of (A-66) and therefore displays the same primitive trinomial. If, however, we take every third bit from (A-66), again starting from the zeroth, we obtain:

1100101..., (A-68)

which is not a phase shift of (A-66) but an entirely new m-sequence. Indeed, it is the m-sequence generated by the other primitive trinomial of degree 3, x^3+x^2+1.

4.5 Decimation by a Power of Two

Other experimentation would soon lead us to the hypothesis that decimating an m-sequence by a power of two always gives rise to the same m-sequence at some phase shift from the old. That this hypothesis is indeed true is a remarkable

property of m-sequences. The validity of this theorem is quickly demonstrated by using abstract algebra. The following is, however, an excellent matrix-oriented proof by Weathers (1972, pp. 12-13). Consider that an m-sequence generator is started at state $\underline{b}^0$ and we consider every other state, i.e., decimation by two. We observe

$$\underline{b}^0 \quad , \underline{b}^0 M^2 \quad , \underline{b}^0 M^4 \quad , \underline{b}^0 M^6 \quad , \ldots \tag{A-69}$$

But (A-69) is equivalent to the undecimated stepping from $\underline{b}^0$ of a machine whose transition matrix is M^2 vice M. Following this observation, consider now that the characteristic equation is

$$M^n + M^{n-\alpha_1} + M^{n-\alpha_2} + \ldots + I = 0 \tag{A-70}$$

We square (A-70), note that all the crossterms disappear under modulo-two arithmetic, and we have

$$M^{2n} + M^{2(n-\alpha_1)} + M^{2(n-\alpha_2)} + \ldots + I = 0 \tag{A-71}$$

Rewriting (A-71) as

$$(M^2)^n + (M^2)^{n-\alpha_1} + (M^2)^{n-\alpha_2} + \ldots + I = 0 \tag{A-72}$$

we see that M^2 satisfies the characteristic equation (A-70) and thus the decimated sequence (A-69) is the original, undecimated m-sequence merely shifted in phase.

An m-sequence is a bit like the proverbial and material ring. It has no beginning, no end. A natural benchmark does exist however. It is called the "characteristic sequence." Recall that an m-sequence is changed in phase only, when decimated by two. It turns out that there is a phase of the m-sequence that is left invariant upon decimation by two. This phase is the characteristic sequence and was discovered by Gold (1966). Gold's straightforward rules that construct the characteristic sequence for the nth degree primitive polynomial f=f(x) is to perform the divisions:

$$\frac{\frac{d}{dx}(xf)}{f} \qquad \text{if n is odd} \tag{A-73}$$

$$\frac{\frac{d}{dx}(xf)}{f} + 1 \qquad \text{if n is even} \tag{A-74}$$

For example, consider the trinomial x^3+x^2+1 which gives the sequence presented in (A-68).

As n is odd, we apply the formula (A-73) and we see that

$$\frac{\frac{d}{dx}(xf)}{f} = \frac{\frac{d}{dx}(x+x^3+x^4)}{1+x^2+x^3} = \frac{1+x^2}{1+x^2+x^3} \tag{A-75}$$

We now perform the division specified in (A-75)

$$\begin{array}{r|l}
1+x^2+x^3 & 1 + x^3 + x^5+x^6+\ldots \\
\hline
1+x^2 & \\
\hline
1+x^2+x^3 & \\
\hline
x^3 & \\
\hline
x^3+x^5+x^6 & \\
\hline
x^5+x^6 & \\
\hline
x^5+x^7+x^8 & \\
\hline
x^6+x^7+x^8 & \\
\hline
x^6+x^8+x^9 & \\
\hline
x^7+x^9 &
\end{array} \tag{A-76}$$

We obtain the characteristic sequence by sequentially reading the coefficients of $1,x,x^2,x^3,\ldots$ of the quotient. Thus, the characteristic sequence, or phase, of the primitive polynomial x^3+x^2+1 is 1001011... . Note that the characteristic sequence starts with zero for n even and one for n odd.

So enticing is this beautiful benchmark it has been independently discovered by Weinrichter and Surböck (1976) in an interesting and instructive way and will probably be rediscovered by other researchers in the future.

Finally, Arazi (1977) has developed the mathematical machinery to compute the initial setting or tuple of an m-sequence for its power of two decimation to achieve a desired phase shift. Arazi cites the characteristic sequence as the special case of zero phase shift.

4.6 General Decimation

If an m-sequence produced by an nth degree primitive polynomial is properly decimated, that is, if the period of decimation is relatively prime to 2^n-1, then another m-sequence will be produced. This new m-sequence will be described by the same primitive polynomial if and only if the decimation is a power of two. For decimation other than a power of two, there is no easy "paper-and-pencil" method of determining the polynomial with a few exceptions. The most famous of these exceptions is the "cubic transformation" introduced by Marsh (1957). This transformation is that produced by decimating by three, or, as Golomb (1967, p. 79)

calls it, "tertiation." Golomb (1969, pp. 363-366) has prepared a lucid algorithm for implementing Marsh's transformation. Golomb's algorithm proceeds as follows:

a) Create three "bins" A, B and C. We will place, but not reduce modulo-three, all numbers that we encounter that are 0 modulo-three into bin A. Into bin B we will place all numbers that are 1 modulo-three. Finally, we will place all numbers that are 2 modulo-three into bin C.

b) The first set of numbers to be sorted and placed into the bins is the exponents of x in f(x), the primitive polynomial.

c) For all distinct pairs (α_1,α_2) of exponents in f(x) that are in the same bin (which is in reality a residue class) we form $(2\alpha_1+\alpha_2)/3$ and $(2\alpha_2+\alpha_1)/3$ and place the results into the appropriate bin.

d) For all distinct tuples $(\alpha_A,\alpha_B,\alpha_C)$ where α_A is an exponent of x of f(x) which has been placed into bin A, α_B is an exponent of x of f(x) which has been placed into bin B, etc., we compute $(\alpha_A+\alpha_B+\alpha_C)/3$ and place it into the appropriate bin.

e) The occupants of the bins are now examined. If a particular occupant is present an odd number of times it is copied onto a list, L, otherwise it is discarded.

f) The new polynomial, the polynomial that describes the decimated-by-three recursion, is formed by summing together the terms consisting of an x raised to each of the powers in the list, L.

An example is in order. Consider the primitive pentanomial

$$f(x)=x^5+x^4+x^2+x+1 \tag{A-77}$$

It produces the sequence

$$1111011001110000110101001000 1011 \tag{A-78}$$

Decimating the sequence (A-78) by three, we obtain the sequence

$$1001011001111100011011101010000 \tag{A-79}$$

Let us use Golomb's method to discover the primitive polynomial which (A-79) obeys:

	bin A	bin B	bin C
STEP (b)	0	1,4	2,5
STEP (c)	3,3	4	2
STEP (d)	3	1	2,2

$L=\{0,3,5\}$

Thus the polynomial that generates (A-79) is $g(x)=x^5+x^3+1$. The reader should note that in order for decimation-by-three to yield a primitive polynomial, n, the degree of the polynomial generating the original sequence, must be odd as 2^n-1, where n is even and greater than 0, is divisible by three and the decimation would therefore be improper.

Golomb (1969, pp. 366-369) generalizes the cubic transformation to a general kth power transformation but deriving the generator polynomial for the decimated sequence quickly becomes overly involved with increasing k. An easy way to derive the generating polynomial of a decimated sequence is to solve a set of simultaneous equations derived as follows. We know that if an m-sequence, that is generated by an nth degree primitive polynomial, is properly decimated, the resulting sequence will also be an m-sequence generated by an nth degree polynomial. Thus, all we need to do in order to uncover the polynomial is to produce a sequence of bits of the decimated recursion

$$\{d(i)\}\ ,\quad i=0,1,2,... \tag{A-80}$$

and then solve for the α_i's from n-1 independent equations of the form

$$d(n+i)=\alpha_1 d(n+i-1)+...+\alpha_{n-1}d(i+1)+d(i) \tag{A-81}$$

This method, for a different application, was suggested by Meyer and Tuchman (1972). As an example, let us apply the method to the sequence given in (A-79). The first four 6-tuples are

$$\begin{matrix}100101\\001011\\010110\\101100\end{matrix} \tag{A-82}$$

From the four 6-tuples of (A-82) we immediately derive the four equations

$$
\begin{aligned}
1 \quad &+\alpha_2 \quad &&= 1 \\
\alpha_1 \quad &+\alpha_3 \quad &&= 1 \\
\alpha_1 &+\alpha_2 \quad +\alpha_4 &&= 0 \\
1 \quad &+\alpha_2 +\alpha_3 \quad &&= 0
\end{aligned}
\tag{A-83}
$$

The solution of the equations in (A-83) yields $\alpha_1=\alpha_2=\alpha_4=0$ and $\alpha_3=1$. Thus, the primitive polynomial which generates the sequence in (A-79) is x^5+x^3+1 as previously determined by the cubic transformation.

4.7 Inverse Decimation

The inverse problem of determining the undecimated sequence given just a short segment of the decimated sequence is considered by Arazi (1977). The following problem development and solution is different from Arazi's but more in consonance with our overall development. We assume that every kth bit is taken from the zeroth stage of a shift register which is implementing an nth degree primitive polynomial by the companion matrix architecture. The contents of the shift register, $\underline{b}^t$, change according to ($t \geq 1$)

$$\underline{b}^{t+1}=\underline{b}^t M \tag{A-84}$$

$$a^t=\underline{b}^t \underline{C} \tag{A-85}$$

where

$$C=\mathrm{col}[1,0,0\ldots,0] \tag{A-86}$$

and a^t corresponds to the output bit at time t. From the hypothesis, we assume that we know

$$a^t,\ a^{t+k},\ a^{t+2k},\ \ldots,\ a^{t+(n-1)k} \tag{A-87}$$

and we need to compute

$$\underline{b}^1=[a^1,\ a^2,\ \ldots,\ a^n] \tag{A-88}$$

from (A-87). In control theory, this is usually referred to as the observability problem (Luenberger, pp. 285-287).

From (A-84) and (A-85), we have

$$a^t = \underline{b}^1 M^t \underline{C} \tag{A-89}$$

and

$$[a^t,\ a^{t+k},\ \ldots,\ a^{t+(n-1)k}]=\underline{b}^1\ M^{t-1}[\underline{C}\ M^k\ \underline{C}\ \ldots\ M^{(n-1)k}\underline{C}] \qquad \text{(A-90)}$$

since it is known that the solution exists and is unique, it is clear that

$$\underline{b}^1=[a^t,\ a^{t+k},\ldots,a^{t+(n-1)k}]\ [M^{t-1}\ \underline{C}\ M^{t+k-1}\underline{C}\ \ldots\ M^{t+(n-1)k-1}\underline{C}]^{-1} \qquad \text{(A-91)}$$

5. GENERATION OF HIGH-SPEED M-SEQUENCES

The ability to generate high-speed m-sequences is important to a number of disciplines, especially direct sequence spread spectrum systems and precise ranging systems. The maximum rate at which the companion matrix realization of an m-sequence generator, Figure A-1, can be run is determined by two parameters, the propagation delay of a shift register stage and the computation time required by the modulo-two adder.

An "electronic" approach to speed increase was proffered by Harvey (1974) for certain trinomial feedback functions in which an analog delay line was substituted for the zeroth stage, which must be untapped for feedback to the modulo-two adder. The delay of the analog line is chosen to be the difference of the clocking period and the computation time of the modulo-two adder. The "before-and-after" diagrams illustrating Harvey's method for implementing the m-sequence specified by x^4+x^3+1 are shown in Figures A-12a and A-12b. Clearly, Harvey's method also applies to the substitution of more than one shift register stage by analog elements.

Ball et al. (1975) extended the above method and showed that for trinomials of the form $x^n+x^{n-1}+1$, no shift register stages are needed at all as is shown in Figure A-13. The delays indicated in the figure are determined by

$$\begin{aligned} D_1&=T \\ D_2&=(n-1)T-D_g \end{aligned} \qquad \text{(A-92)}$$

where T is the bit period, n is the degree of the trinomial ($x^n+x^{n-1}+1$) and D_g is the computation time of the modulo-two adder. Ball, et al., noted that the configuration shown in Figure A-13 will run free and is, in their words, "a special case of the delay-line oscillator."

Another approach to high-speed generation was offered by Lempel and Eastman (1971). Unlike the methods just reviewed, Lempel and Eastman's method depends on a mathematical, vice electronic, basis to achieve high-speed generation. The method depends upon a curious aspect of decimated m-sequences. Essentially, what

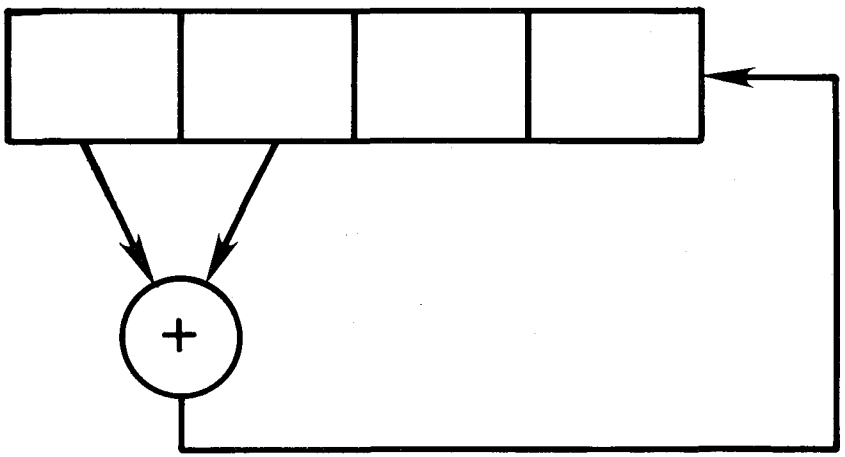


Figure A-12a. Companion matrix realization of x^4+x^3+1.

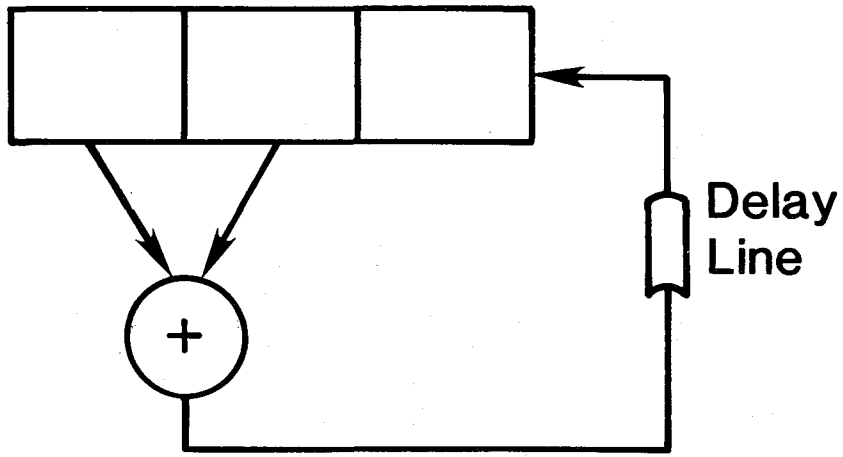


Figure A-12b. Delay line realization of x^4+x^3+1.

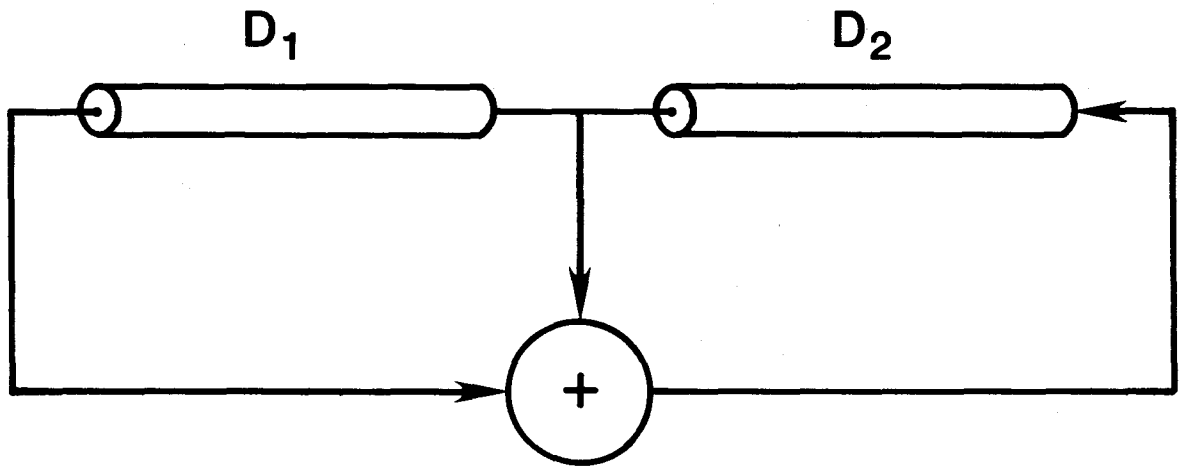


Figure A-13. 'Analog' (delay line) shift register.

Lempel and Eastman proposed was the creation of k shift registers each of which would be driven by the feedback polynomial corresponding to the kth decimation of the sequence {s(i)}, i=0,1,2,... which is generated by the nth degree polynomial that it is desired to implement at high speed. (The decimations need not be proper, that is, k and 2^n-1 need not be relatively prime.) What is done is to step, in succession, each of the k registers and modulo-two add together the last stage from each. The registers are started with initial conditions so that register j exhibits the decimated sequence {s(ik+j)}, i=0,1,2,... . The sequence produced by this addition exhibits the desired polynomial and runs at a rate k times the rate of the individual shift registers. As an example, consider that we wish to synthesize the sequence produced by the primitive trinomial x^3+x+1 at a rate two times faster than we shift our registers. For this example, k=2, the decimation is a proper one and, additionally, the polynomial driving our two registers must also be x^3+x+1 as the decimation is a power of two. (Recall that this was shown in the previous section.) Thus, the diagram shown in Figure A-14 will implement the recursion at twice the clock rate. If the top register is started with 010 and the bottom with 111, the machine will progress as follows:

CLOCK:	10101010101010
TOP REGISTER STAGE 2:	01100001111110
BOTTOM REGISTER STAGE 2:	11111100110000
MOD-TWO ADDER OUTPUT:	10011101001110

The three implementations so far reviewed assume that the limiting parameter is the propagation delay of the shift register stage. Often this is not the case but it is rather the computation time required by the modulo-two adder that caps the operating speed. One solution against this inherently different problem was given by Quan (1974) who proposed a modification of Lempel and Eastman's method. He suggested time division multiplexing the decimated sequences instead of adding them modulo-two. Figure A-15 depicts the scheme shown in Figure A-14 recast under Quan's modification. The initial conditions of the registers are the same as for Eastman and Lempel's version.

One final method for generating high-speed m-sequences is that proposed by Warlick and Hershey (1980). These authors first developed a special architecture for m-sequence generators based on primitive trinomials

$$x^n+x^a+1 \qquad \text{(A-93)}$$

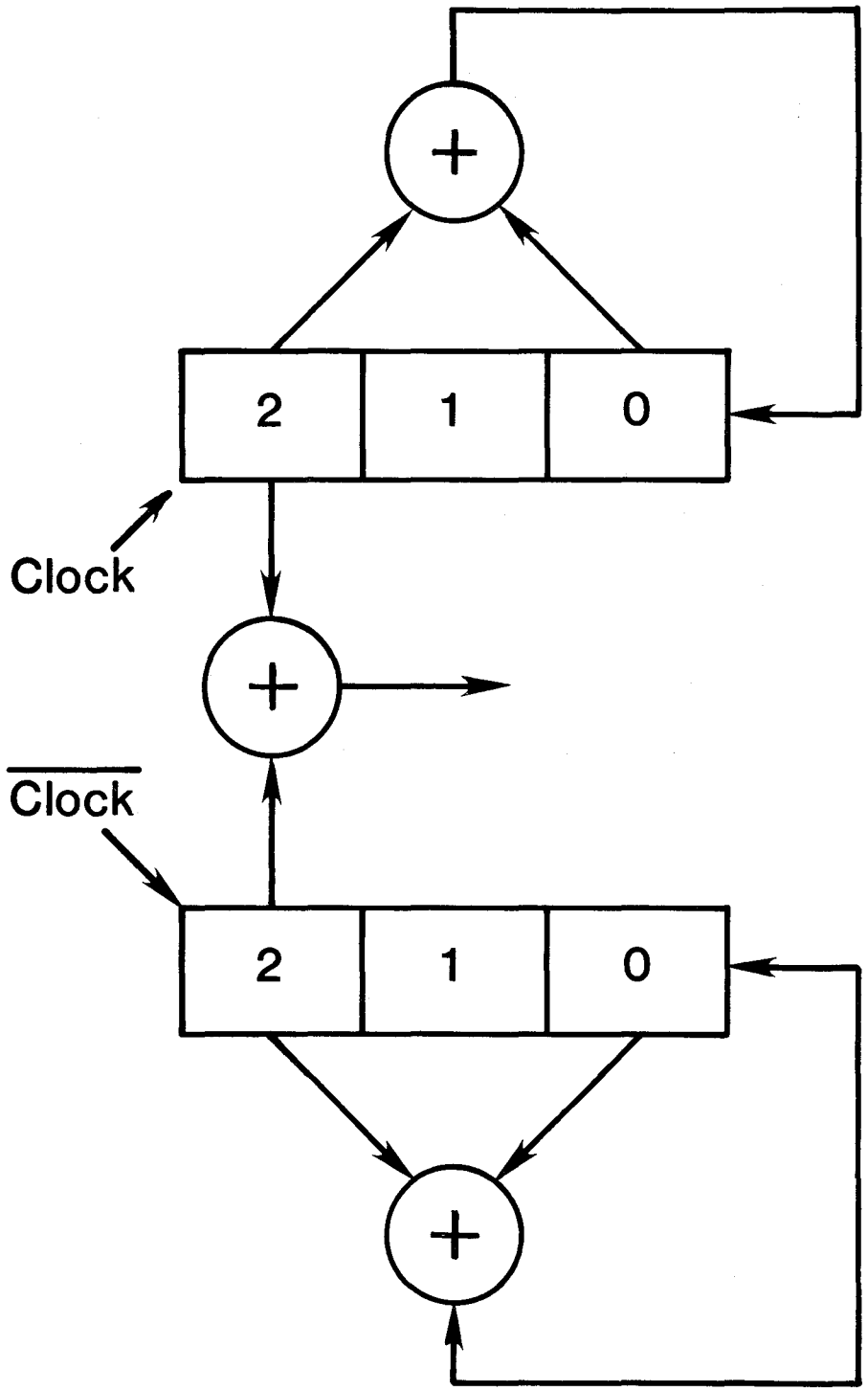


Figure A-14. High speed m-sequence generation by decimation and modulo-two addition.

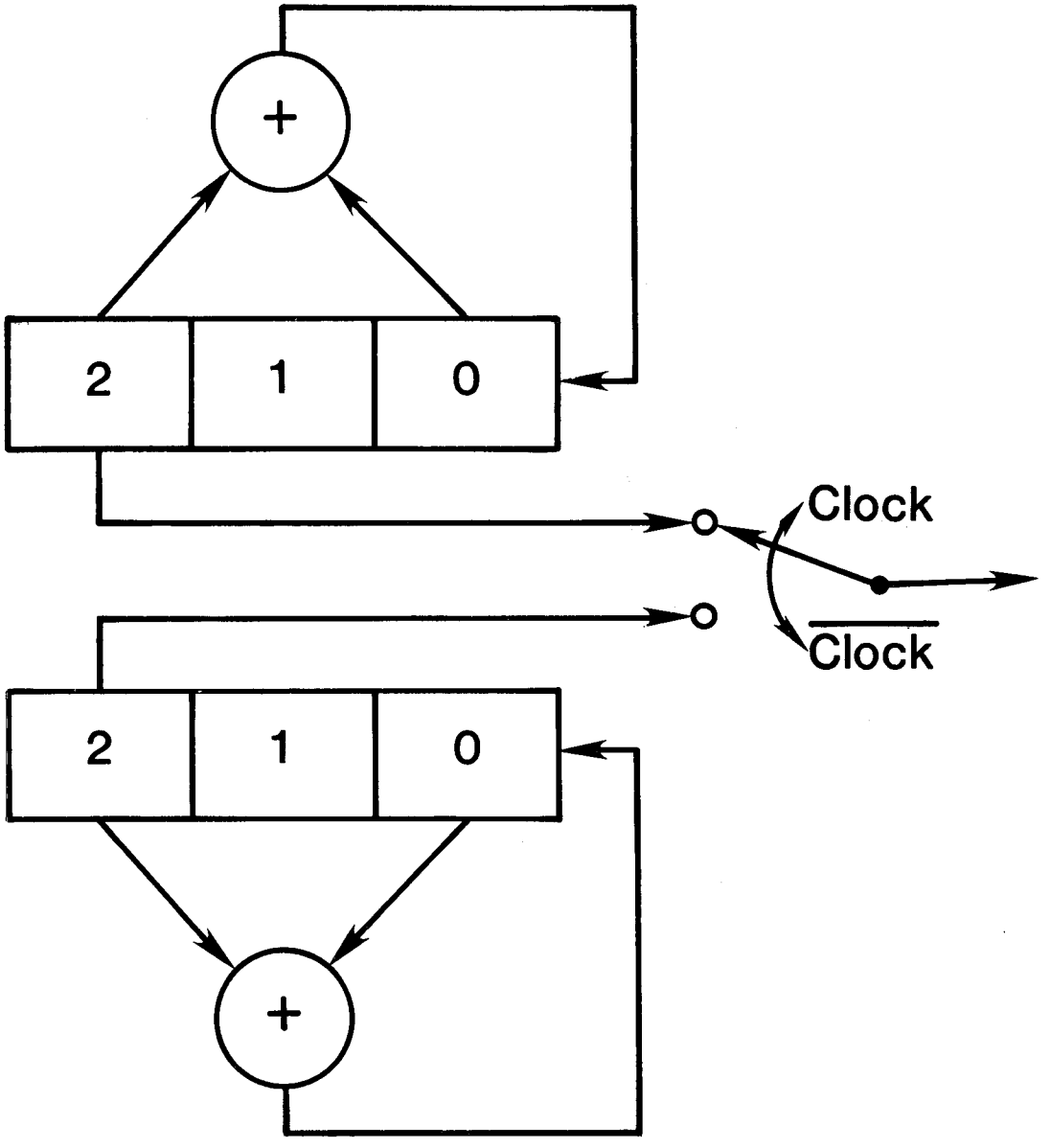


Figure A-15. High speed m-sequence generation by decimation and multiplexing.

They found that the general architecture depicted in Figure A-16, which they termed the "vanestream structure," (The repeated register blocks with feedback are termed the "vanes.") will progress through a maximum length cycle of $2^{v\ell+1}-1$ states for a surprisingly rich set of triples (v,ℓ,t) which they catalog, along with the primitive trinomials (A-93) upon which the structures are based, for $v\ell \leq 99$. The architecture of Figure A-16 is then coupled with a parallel in (broadside load), serial out multiplexer as shown in Figure A-17 to form what they term the "WINDMILL m-sequence generator." (The name "WINDMILL" derives from a predecessor sequential machine of theoretical interest only.) The slower logic vanestream generator is stepped at a rate R. After each step, the contents of the v stages are copied into the v high-speed shift register stages indicated and the v high-speed shift register is shifted v times in the direction indicated. The high-speed stream exhibits the m-sequence specified by the trinomial

$$x^n+x^{n-a}+1 \qquad \text{(A-94)}$$

(not x^n+x^a+1 as erroneously reported in the paper).

The WINDMILL thus achieves a speed ratio increase of v to attain a high-speed rate Rv. Warlick and Hershey briefly considered the electronic architectural implications of realizing a WINDMILL and concluded that the WINDMILL offers power advantages for certain hybrid MOS layouts and its periodic substructures are especially amenable to LSI blacksmithery.

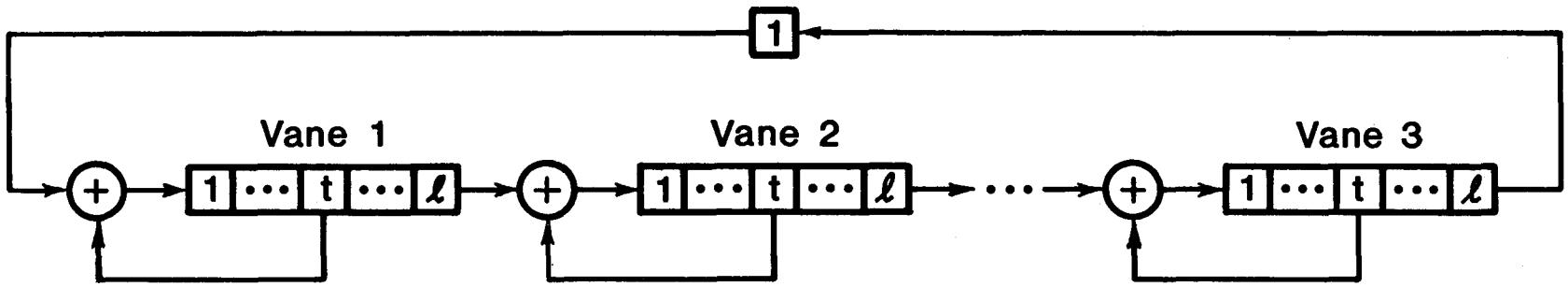


Figure A-16. The vanestream structure.

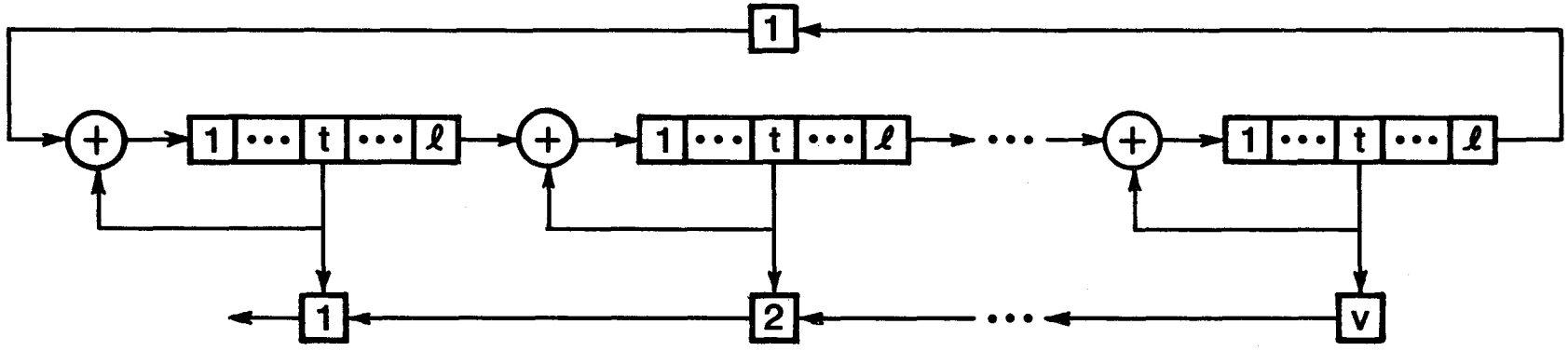


Figure A-17. The WINDMILL high-speed m-sequence generator.

6. REFERENCES: APPENDIX A

Arazi, B. (1977), Decimation of m-sequence leading to any desired phase shift, Electron. Letters 13, No. 7, pp. 213-215, March.

Ball, J., A. Spittle, and H. Liu (1975), High-speed m-sequence generation: A further note, Electron. Letters 11, No. 5, pp. 107-108, March.

Berlekamp, E. (1967), Factoring polynomials over finite fields, BSTJ, pp. 1853-1859, October.

Berlekamp, E. (1968), Algebraic Coding Theory (McGraw-Hill Book Co., New York, NY).

Berlekamp, E. (1970), Factoring polynomials over large finite fields, Mathematics of Computation 24, No. 111, pp. 713-735, July.

Birdsall, T., and M. Ristenbatt (1958), Introduction to linear shift-register generated sequences, Technical Report No. 90, Electronic Defense Group, Dept. of Electrical Engineering, The University of Michigan Research Institute, Ann Arbor, MI. (Work performed for the Signal Corps, Dept. of the Army, Control No. DA-36-039 sc-63203, Task Order No. EDG-3, Project 2262.)

Birkhoff, G., and S. MacLane (1964), A Survey of Modern Algebra, Revised Edition (MacMillan Company).

Cantor, D., and H. Zassenhaus (1981), A new algorithm for factoring polynomials over finite fields, Mathematics of Computation 36, No. 154, pp. 587-592, April.

Charney, H., and C. Mengani (1961), Generation of linear binary sequences, RCA Rev., pp. 420-430, September.

Davies, A. (1965), Delayed versions of maximal-length linear binary sequences, Electron. Letters 1, No. 3, pp. 61-62, May.

Davies, A. (1968), Calculations relating to delayed m-sequences, Electron. Letters 4, No. 14, pp. 291-292, July.

Davis, W. (1966), Automatic delay changing facility for delayed m-sequences, Proc. IEEE 54, pp. 913-914, June.

deVisme, G. (1971), Binary Sequences (The English Universities Press Ltd.).

Douce, J. (1968), Delayed versions of m-sequences, Electron. Letters 4, No. 12, p. 254, June.

Gardiner, A. (1965), Logic P.R.B.S. delay calculator and delayed-version generator with automatic delay-changing facility, Electron. Letters 1, No. 5, pp. 123-125, July.

Gold, R. (1966), Characteristic linear sequences and their coset functions, Journal SIAM 14, No. 5, pp. 980-985, September.

Golomb, S. (1967), Shift Register Sequences (Holden-Day, Inc.).

Golomb, S. (1969), Irreducible polynomials, synchronization codes, primitive necklaces, and the cyclotomic algebra, Chapter 21, Combinatorial Mathematics and Its Applications (The University of North Carolina Press), pp. 358-370. (Proceedings of the conference held at the University of North Carolina, Chapel Hill, NC, April.)

Golomb, S. (1980), On the classification of balanced binary sequences of period 2^n-1, IEEE Trans. Inform. Theory 26, No. 6, pp. 730-732, November.

Harvey, J. (1974), High-speed m-sequence generation, Electron. Letters 10, No. 23, pp. 480-481, November.

Hershey, J. (1980), Implementation of MITRE public key cryptographic system, Electron. Letters 16, No. 24, pp. 930-931, November.

Ireland, B., and J. Marshall (1968a), Matrix method to determine shift-register connections for delayed pseudorandom binary sequences, Electron. Letters 4, No. 15, pp. 309-310, July.

Ireland, B., and J. Marshall (1968b), Matrix method to determine shift-register connections for delayed pseudorandom binary sequences, Electron. Letters 4, No. 21, pp. 467-468, October.

Ireland, B., and J. Marshall (1976), New method of generating shifted linear pseudorandom binary sequences, Proc. Inst. Elec. Engrs. (London) 123, p. 182, February.

Knuth, D. (1969), The art of computer programming, Seminumerical Algorithms, (Addison-Wesley), Vol. 2, pp. 381-396.

Latawiec, K. (1974), New method of generation of shifted linear pseudorandom binary sequences, Proc. Inst. Elec. Engrs. (London) 121, No. 8, pp. 905-906, May.

Latawiec, K. (1975), New method of generation of shifted linear pseudorandom binary sequences, Proc. Inst. Elec. Engrs. (London) 122, No. 4, p. 448, April.

Latawiec, K. (1976), New method of generation of shifted linear pseudorandom binary sequences, Proc. Inst. Elec. Engrs. (London) 123, No. 2, p. 182.

Laxton, R., and J. Anderson (1972), Linear recurrences and maximal length sequences, The Mathematical Gazette LVI, No. 398, pp. 299-309, December.

Lempel, A., and W. Eastman (1971), High speed generation of maximal length sequences, IEEE Trans. Computers, pp. 227-229, February.

Luenberger, D. (1979), Introduction to Dynamic Theory -- Theory, Models and Applications (Wiley), pp. 285-289.

MacDuffie, C. (1940), An Introduction to Abstract Algebra (John Wiley & Sons, Inc.).

Marsh, R. (1957), Table of irreducible polynomials over GF(2) through degree 19, U.S. Department of Commerce, Office of Technical Services, Washington, DC.

Meyer, C., and W. Tuchman (1972), Pseudorandom codes can be cracked, Electronic Design 20, No. 23, pp. 74-76, November.

Meyer, F. (1976), Gigabit/s m-sequence generation, Electron. Letters 12, No. 14, p. 353, July.

Miller, A., A. Brown, and P. Mars (1977), A simple technique for the determination of delayed maximal length linear binary sequences, IEEE Trans. Computers C-26, No. 8, pp. 808-811, August.

Moenck, R. (1977), On the efficiency of algorithms for polynomial factoring, Mathematics of Computation 31, No. 137, pp. 235-250, January.

Perlis, S. (1952), Theory of Matrices (Addison-Wesley).

Quan, A. (1974), A note on high-speed generation of maximal length sequences, IEEE Trans. Computers, pp. 201-203, February.

Reiner, I. (1961), On the number of matrices with given characteristic polynomial, Illinois Journal of Mathematics 5, pp. 324-329.

Roberts, P., A. Davies, and Tsao (1965), Discussion on generation of delayed replicas of maximal-length binary sequences, Proc. Inst. Elec. Engrs. (London) 112, No. 4, pp. 702-704, April.

Rodemich, E., and H. Rumsey (1968), Primitive trinomials of high degree, Mathematics of Computation 22, pp. 863-865.

Sarwate, D., and M. Pursley (1980a), Crosscorrelation properties of pseudorandom and related sequences, Proc. IEEE 68, No. 5, pp. 593-619, May.

Sarwate, D., and M. Pursley (1980b), Correction of "Crosscorrelation properties of pseudorandom and related sequences," Proc. IEEE 68, No. 12, p. 1554, December.

Scholefield, P. (1960), Shift registers generating maximum-length sequences, Electronic Technol., pp. 389-394, October.

Stahnke, W. (1973), Primitive binary polynomials, Mathematics of Computation 27, No. 124, pp. 977-980, October.

Swan, R. (1962), Factorization of polynomials over finite fields, Pacific Journal of Mathematics 12, pp. 1099-1106.

Tsao, S. (1964), Generation of delayed replicas of maximal-length linear binary sequences, Proc. Inst. Elec. Engrs. (London) 111, No. 11, pp. 1803-1806, November.

VanLuyn, A. (1978), Shift-register connections for delayed versions of m-sequences, Electron. Letters 14, No. 22, pp. 713-715, October.

Warlick, W., and J. Hershey (1980), High-speed m-sequence generators, IEEE Trans. on Computers C-29, No. 5, pp. 398-400, May.

Watson, E. (1962), Primitive polynomials (Mod 2), Mathematics of Computation 16, No. 79, pp. 368-369, July.

Weathers, G. (1972), Statistical properties of filtered pseudo-random digital sequences, Sperry Rand Corporation, Technical Report No. SP-275-0599, January. (Prepared for NASA George C. Marshall Space Flight Center, Huntsville, AL.)

Weinrichter, H., and F. Surböck (1976), Phase normalized m-sequences with the inphase decimination property {m(k)} = {m(2k)}, Electron. Letters 12, No. 22, pp. 590-591, October.

Yiu, K. (1980), A simple method for the determination of feedback, shift register connections for delayed maximal-length sequences, Proc. IEEE 68, No. 4, pp. 537-538, April.

Zierler, N. (1959), Linear recurring sequences, Journal SIAM 7, No. 1, pp. 31-48, March.

Zierler, N., and J. Brillhart (1968), On primitive trinomials (Mod 2), Information and Control 13, pp. 541-554.

Zierler, N., and J. Brillhart (1969), On primitive trinomials (Mod 2), II, Information and Control 14, pp. 566-569.

Zierler, N. (1969), Primitive trinomials whose degree is a Mersenne exponent, Information and Control 15, pp. 67-69.

Zierler, N. (1970), On $x^n + x + 1$ over GF(2), Information and Control 16, pp. 502-505.

APPENDIX B: A CURSORY LOOK AT SYNCHRONIZATION FOLLOWING CLOCK RECOVERY

1. INTRODUCTION

Synchronization is usually performed to establish either epoch or phase. Epoch synchronization refers to those techniques that seek to establish agreement on the epoch or occurrence of a particular instant of time. Phase synchronization comprises those processes that endeavor to determine the phase of a cyclic (steady state deterministic) digital process. We will consider a variety of epoch and phase synchronization methods, starting with epoch synchronization followed by the phase synchronization genres.

1.1 Epoch Synchronization

Epoch synchronization is a process by which a transmitter communicates a reference time mark to a receiver. This is usually done in the time domain by sending a carefully constructed sequence. The sequence is such that it possesses an autocorrelation that has low sidelobes thus allowing a large peak to sidelobe ratio when passed through a matched filter. There is some confusion in the literature that centers on sequence design according to the presence or absence of bit sense or "bit ambiguity." This difficulty is explored.

1.1.1 Autocorrelation

The autocorrelation function most commonly used for sequence design is as follows. Let the sequence be denoted by s(1), s(2), ... s(n) where $s_i \epsilon \{0,1\}$. We compute the autocorrelation, r(k), by counting the agreements, A, between s(i) and s(i+k) over the range $0 \leq i \leq n-k$ and subtracting the disagreements over the same range. From this definition it is clear that r(k)=r(-k). As an example, let us determine the autocorrelation of the sequence

0000011010111001 (B-1)

We calculate r(1) by lining up the slipped sequence with the unslipped sequence as follows

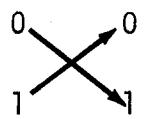
```
0000011010111001
 0000011010111001
 ----------------
 AAAADADDDDAADAD
```

A indicates an AGREEMENT:
D a DISAGREEMENT

from the above we see that r(1)=8-7=1. Figure B-1 depicts r(k) for k=0, $\pm$1, $\pm$2, ... $\pm$15. Note that max(A-D) for k$\neq$0 is 1. Note further that at k=8, A-D=-6. This is a consequence of designing the sequence (B-1) in order to achieve

$$\min_{k\neq 0} (\max (A-D)) \tag{B-2}$$

What (B-2) implies is that bit 'sense' is known, i.e., the receiver knows the received bits exactly and not within the



(B-3)

ambiguity of some differentially coded systems. This is a reasonable and practical assumption for many communications circuits and signaling architectures.

1.1.2 Bit Sense Known

Sequences, or as they are often called, 'unique words,' which are designed under (B-2) may be used in various telemetry schemes, see Maury et al. (1964), and for Time Division Multiple Access (TDMA) satellite communications. See, for example Sekimoto (1968), Gabard (1968), Schrempp and Sekimoto (1968), and Nuspl et al. (1977). A common implementation that can be used for synchronization using sequences wherein bit sense is known, is a three step process (Schrempp and Sekimoto, 1968):

I. Allow carrier recovery by sending the receiver a stream of all zeros

II. Provide bit timing recovery by transmitting a stream of zero/one alternations

III. Mark epoch by sending a unique word.

As an example, let us assume we have carrier and clock and let us examine a few cases where we identify the epoch with the unique word specified in (B-1). We assume, for this case, that we are provided with a zero/one quantized bit stream and that we are passing this bit stream through a matched filter or replica of the unique word (B-1). (This is not the optimum method of detection, incidentally, but it is easy to realize in hardware.) Our experiment is depicted in Figure B-2. The bitstream that is passed into the Bitstream Analysis Window is a stream of ones and zeros taken from a balanced Bernoulli source which is a source of zeros and ones such that the probability of a one at time t equals the probability of a

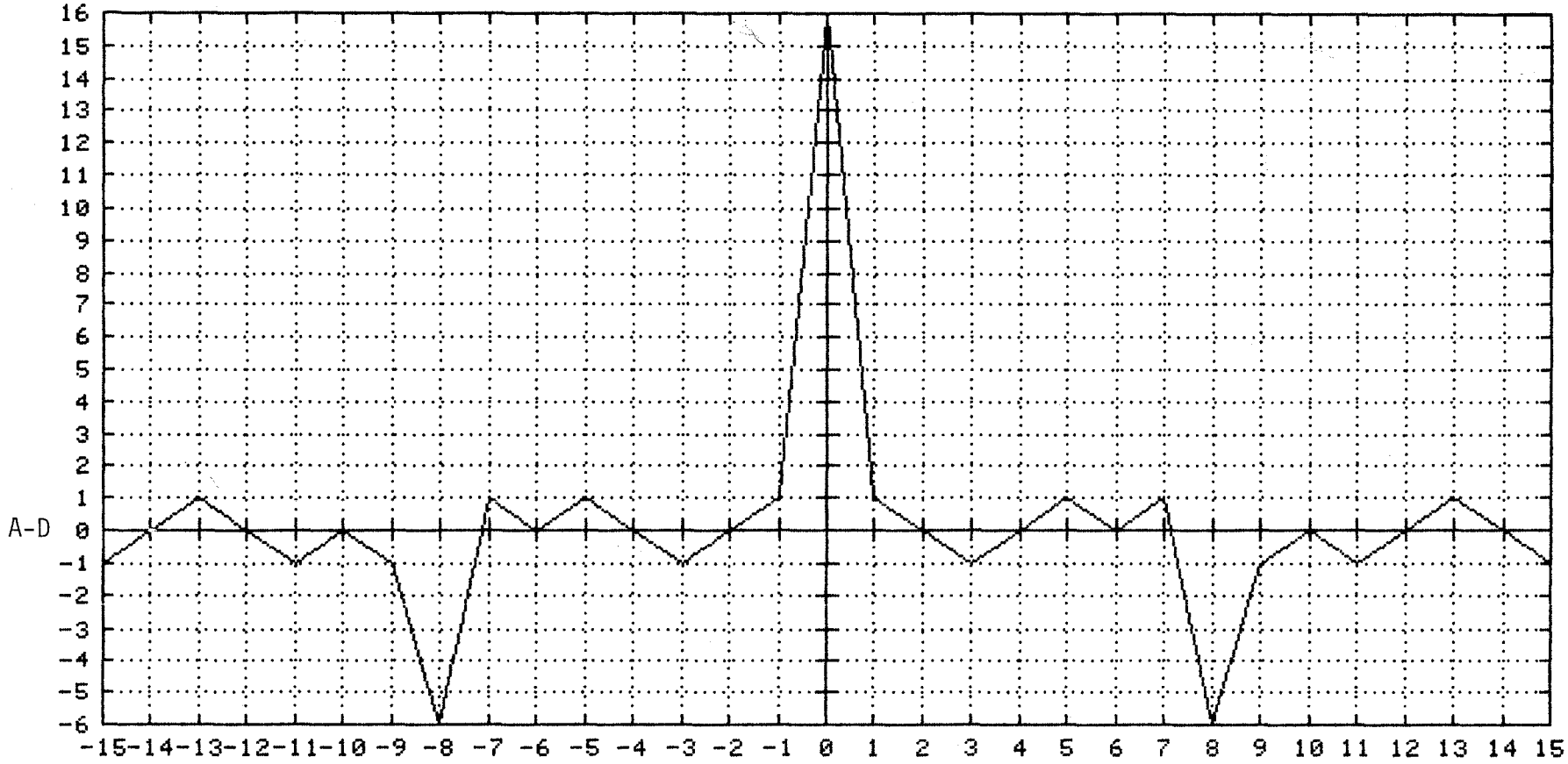


Figure B-1. Autocorrelation.

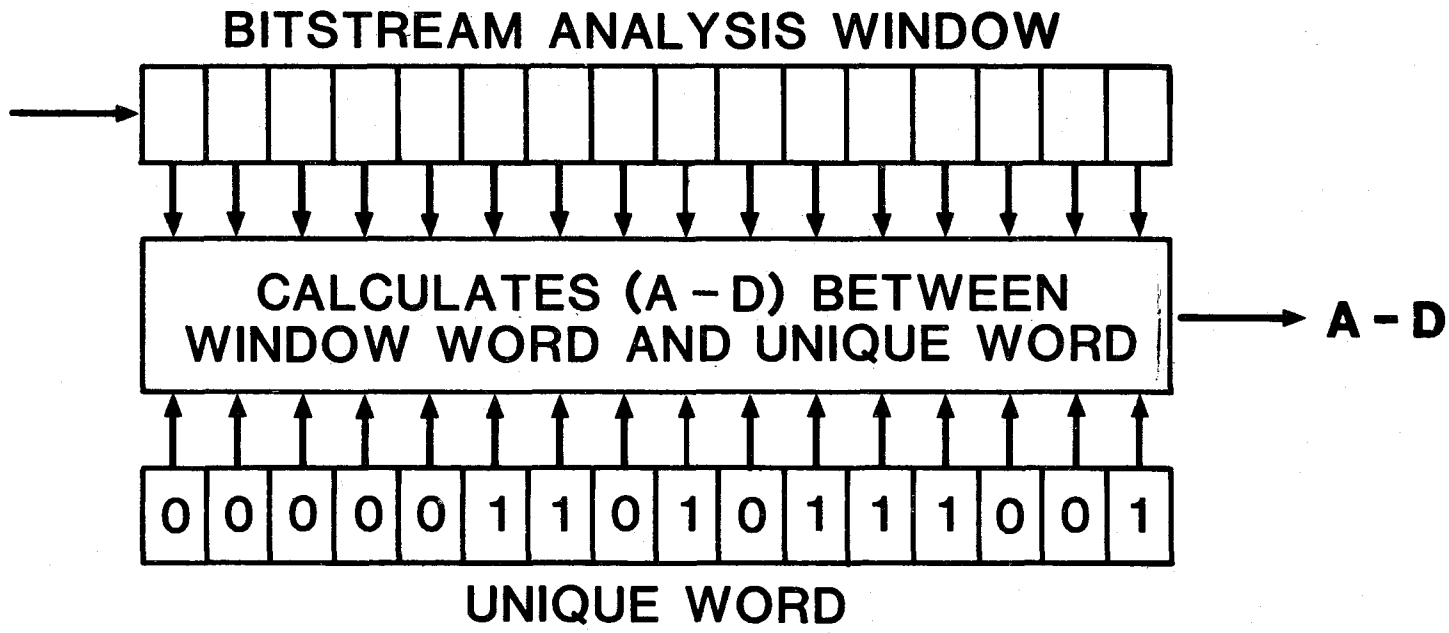


Figure B-2. A matched filter.

zero at time t equals one-half; further, the bit at time t is independent of all bits preceding it. At t=30, the unique word (B-1) is sent and thus at t=45, the unique word is lined up with its replica in the filter. At this point, if the unique word has not been corrupted by the channel, it will completely agree with its stored replica and cause a large pulse. Figure B-3 depicts the action showing A-D as a function of time. Time is assumed to flow left to right.

As Figure B-3 depicts, the epoch is clearly defined by the A-D pulse of height 16 at t=45. (The correlation is not observed for the first 15 clock times as the window is being initially filled during this time.) To detect the epoch, then, we would set a threshold on the A-D waveform and make the rule that the epoch is declared when the A-D waveform meets or exceeds the threshold. This is obviously doable and effective if the unique word is received, detected and quantized without errors. Such is not the case in general, however. Figure B-4 depicts the action wherein the unique word is passed through a binary symmetric channel (BSC) with $p = \frac{1}{16}$. (A BSC with parameter p is a simple channel model that dictates that the bit transmitted at time t is received correctly with probability 1-p and is independent of previous channel errors.) In this case, the unique word suffered a single bit error causing A to decrease by one at epoch and D to increase by one causing A-D to drop from 16 to 14. For this case we would have correctly detected the epoch only if we had set our threshold in the range $8<\text{threshold}\leq 14$.

Figure B-5 repeats the above experiment with a BSC with $p = \frac{1}{4}$. In this example the unique word suffered 4 errors and the A-D waveform exhibited a peak value of only 8 at the correct epoch, t=45. Note however that a randomly occurring peak of height 8 also occurred at t=27. For this case it would have been impossible to select a threshold which would have correctly identified the epoch.

So far we have learned that we must be careful in choosing the structure of our unique word and we must be careful to correctly set the epoch detection threshold. There is another consideration that is worthwhile and this refers to 'look time.' The look time is defined as the amount of time a matched filter will be allowed to search for an epoch. The random local maxima that occur may cause a false epoch determination. The probability of a false epoch determination varies in a direct relationship to the look time. If a long look time is required, it may be necessary to select a high threshold. Doing so may require the user to pick a longer unique word because a higher threshold may allow for fewer channel errors. As an example, Figure B-6 depicts the action of

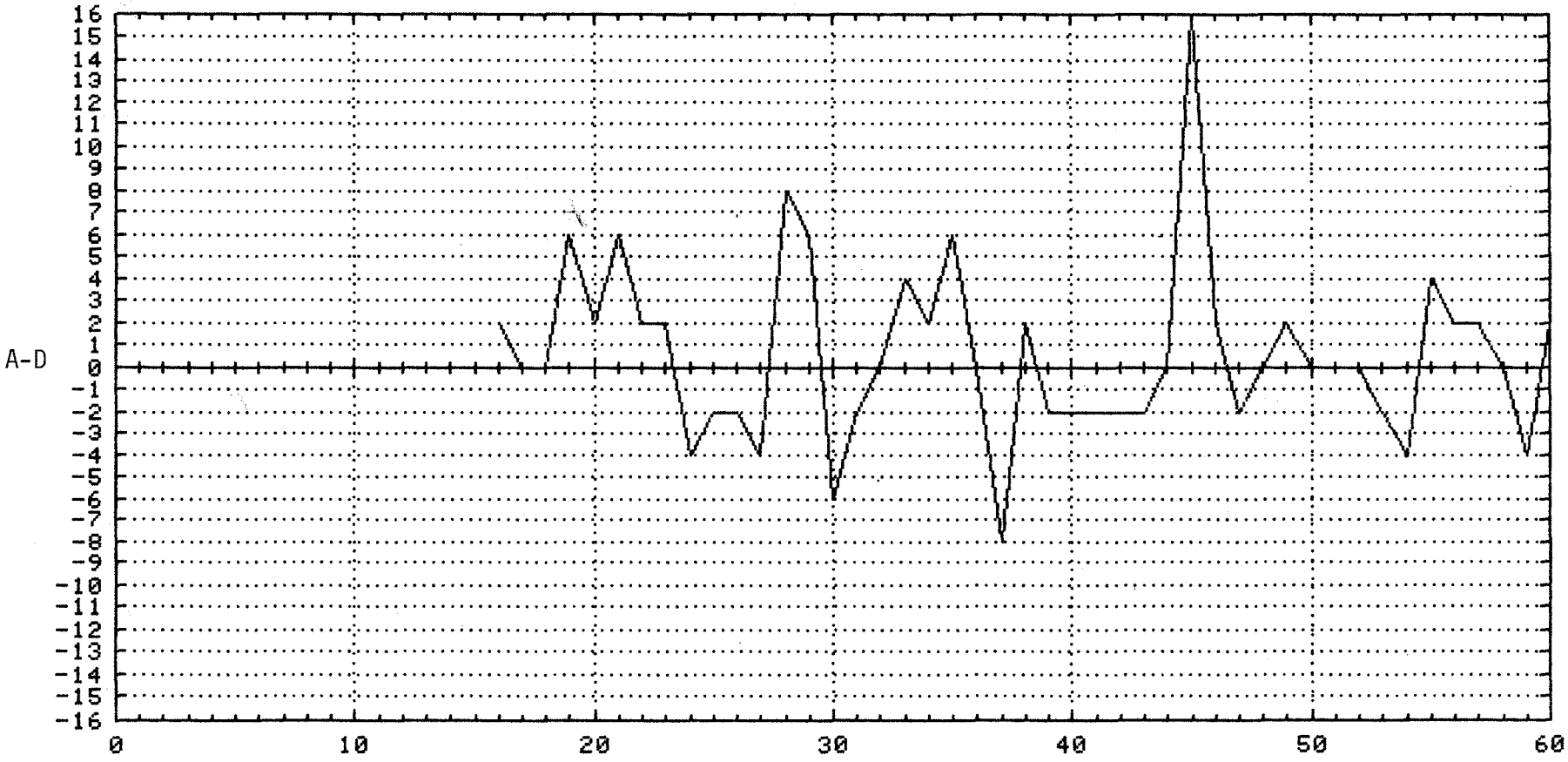


Figure B-3. Matched filter action.

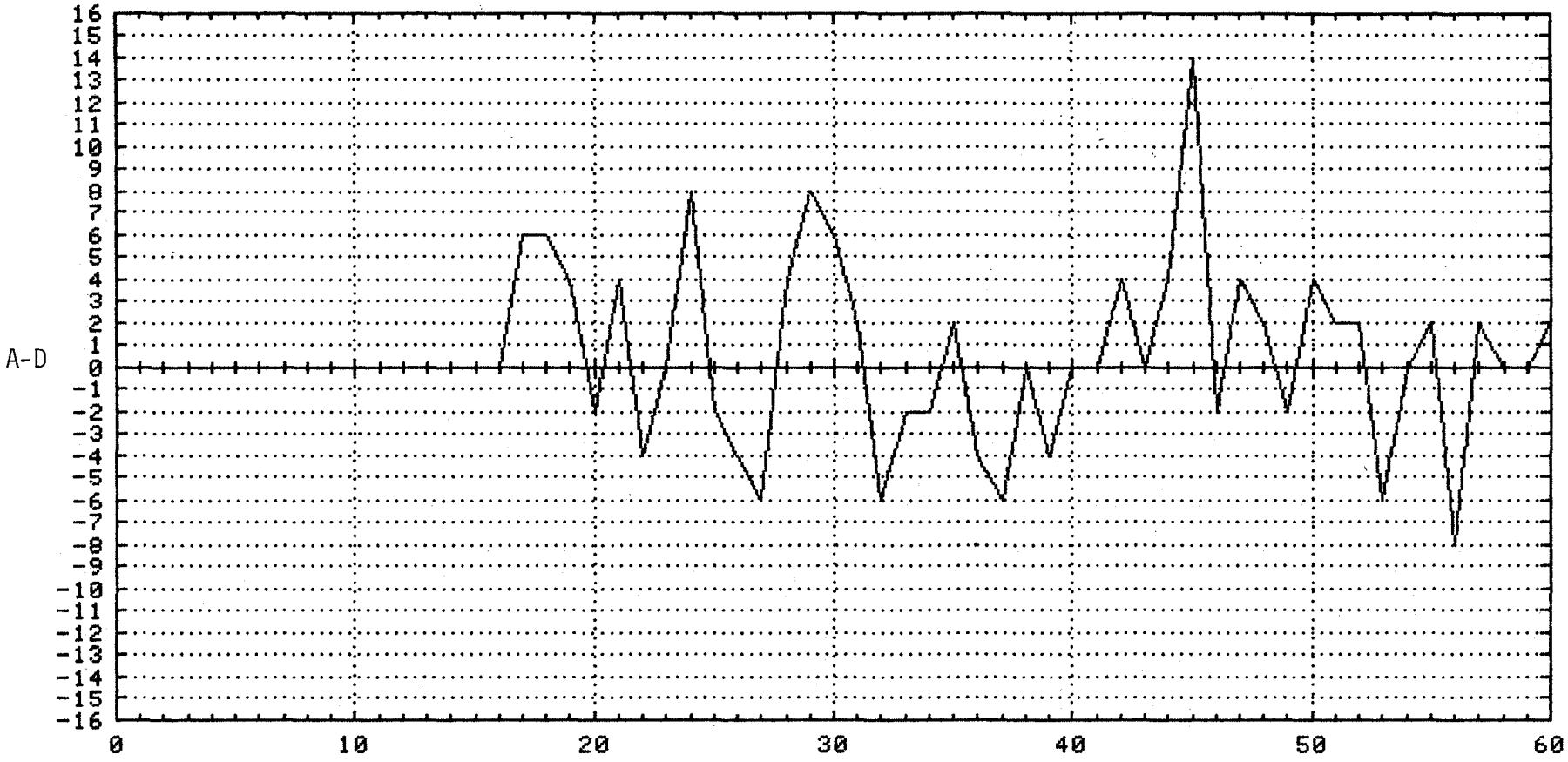


Figure B-4. Matched filter action.

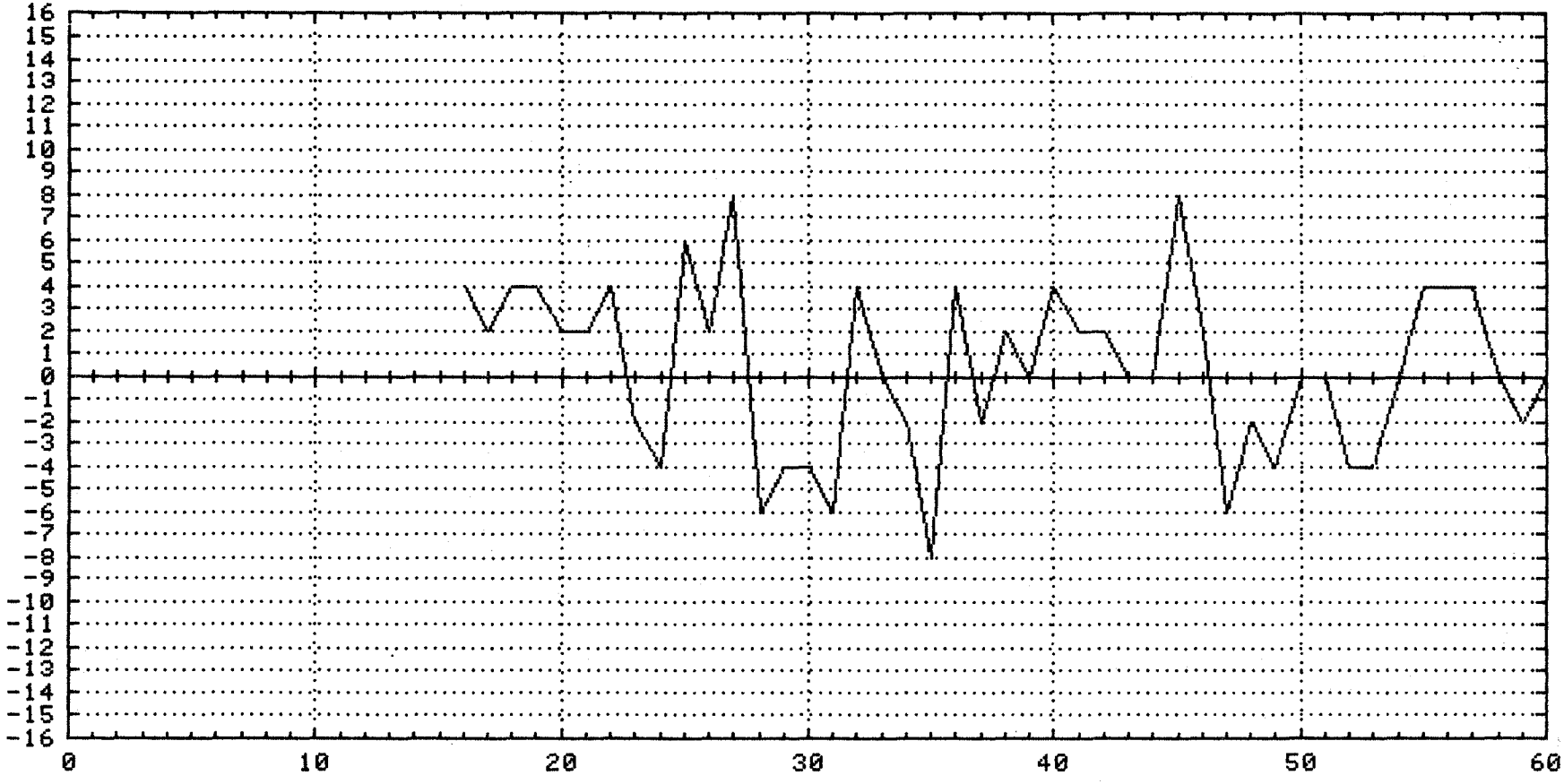


Figure B-5. Matched filter action.

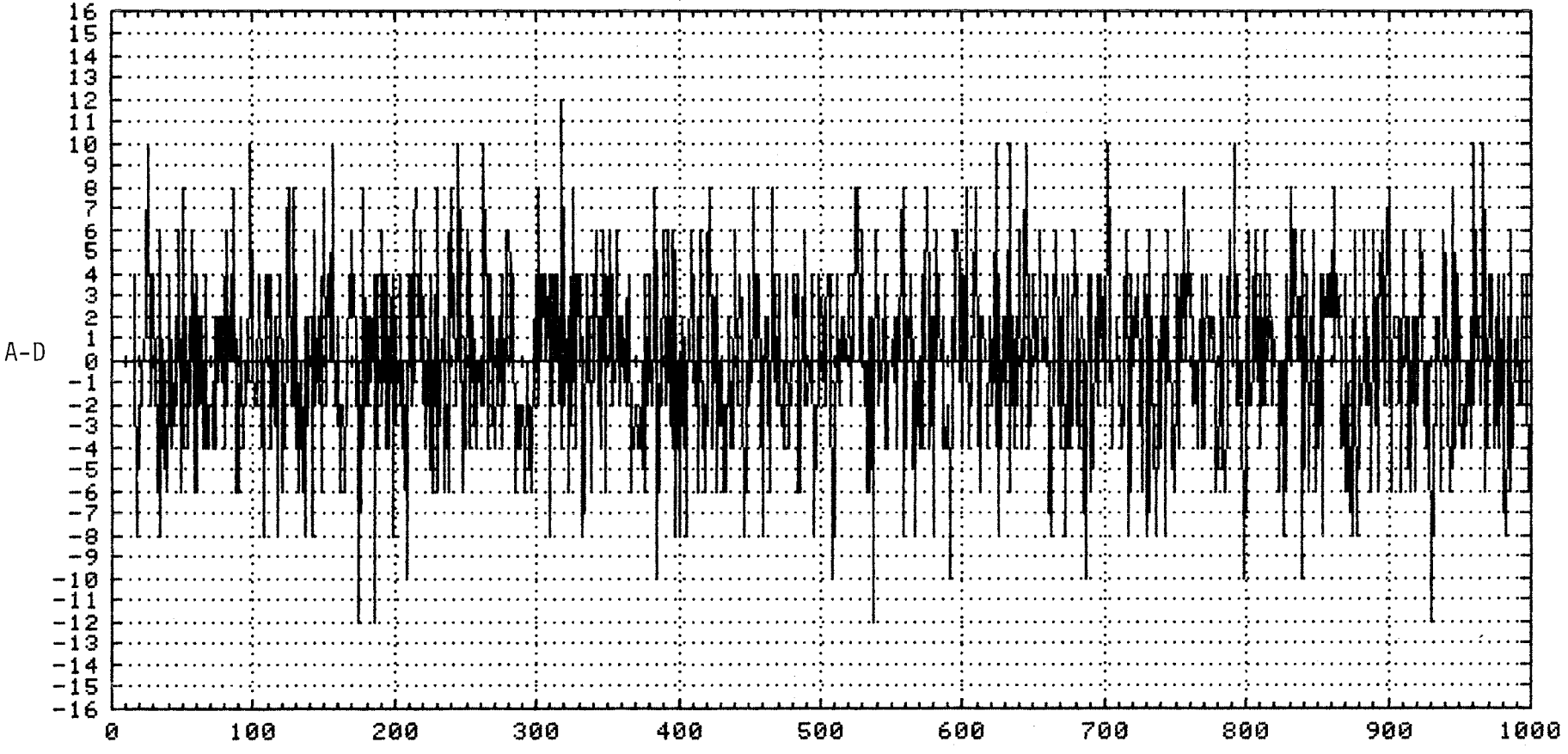


Figure B-6. Matched filter action on random input bitstream.

the matched filter of Figure B-2 for an input bitstream produced by a balanced Bernoulli source of 1000 bits. Note the many relatively large maxima that occur randomly.

Table B-1 lists examples of unique words meeting condition (B-2) for lengths 3-29 bits.

1.1.3 Bit Sense Unknown

In many cases bit sense is unknown and unique word sequences are then designed by modifying (B-2) to the following

$$\min \left(\max_{k \neq 0} (|A\text{-}D|) \right) \tag{B-4}$$

By way of introduction, let us examine a 16 bit sequence designed according to (B-4).

$$0000011001101011 \tag{B-5}$$

Figure B-7 depicts r(k) for this word for k=0, ±1, ±2, ... ±15. Note that for this word, the best that can be achieved under condition (B-4) is $|r(k)| \leq 2$ in contrast to $r(k) \leq 1$ for the word (B-1) which was designed under condition (B-2).

The great majority of research on unique words has addressed the case where bit sense in unknown. Probably this was a result of the great influence that the radar field has had on special sequence development. A PSK modulated radar return is a good example of a case in which one would instinctively choose a pulse compression code or unique word designed under (B-4).

The most famous of the sequences designed under (B-4) are the Barker codes. These sequences are those designed under (B-4) for which

$$\max_{k \neq 0} (|A\text{-}D|) = 1 \tag{B-6}$$

Some of the sequences that meet (B-6) were introduced by R. H. Barker (1953) and bear his name. No Barker sequences beyond length 13 are known but it is known that if any do indeed exist, then they must have a length that is an even square. All possibilities up to $78^2=6084$ have been eliminated. (See Petit, 1967.)

Lindner (1975a) has compiled an exhaustive listing of all binary sequences or unique words that meet (B-4) for lengths 3-40 bits. Table B-2 lists one of these sequences for each word length for n=3 to n=40 bits along with the $\max_{k \neq 0}(|A\text{-}D|)$ value achievable for n (Lindner, 1975b).

Table B-1. Examples of Unique Words When Bit Sense is Known

LENGTH	MAX ACF	SEQUENCE
3	0	001*
4	1	0001*
5	1	00010*
6	1	000110
7	0	0001101
8	1	00001101
9	1	000011010
10	1	0001110010
11	0	00011101101*
12	1	000010110011
13	1	0000011001010*
14	1	00011011110010
15	2	000000110010100
16	1	0000011010111001
17	1	00000101100111010
18	1	000010101101100111
19	2	0000001011001110010
20	1	00000101110100111001
21	1	000000111001101101010
22	1	000100011111001101101O
23	2	00000011100101011011001
24	1	000000111001110101011O110
25	1	0000001101101110001101010
26	2	00000001101010110011110010
27	2	000000011011001111001010100
28	2	0000000110110110001110101011
29	1	00000010110010011100111101010

*Indicates Barker code

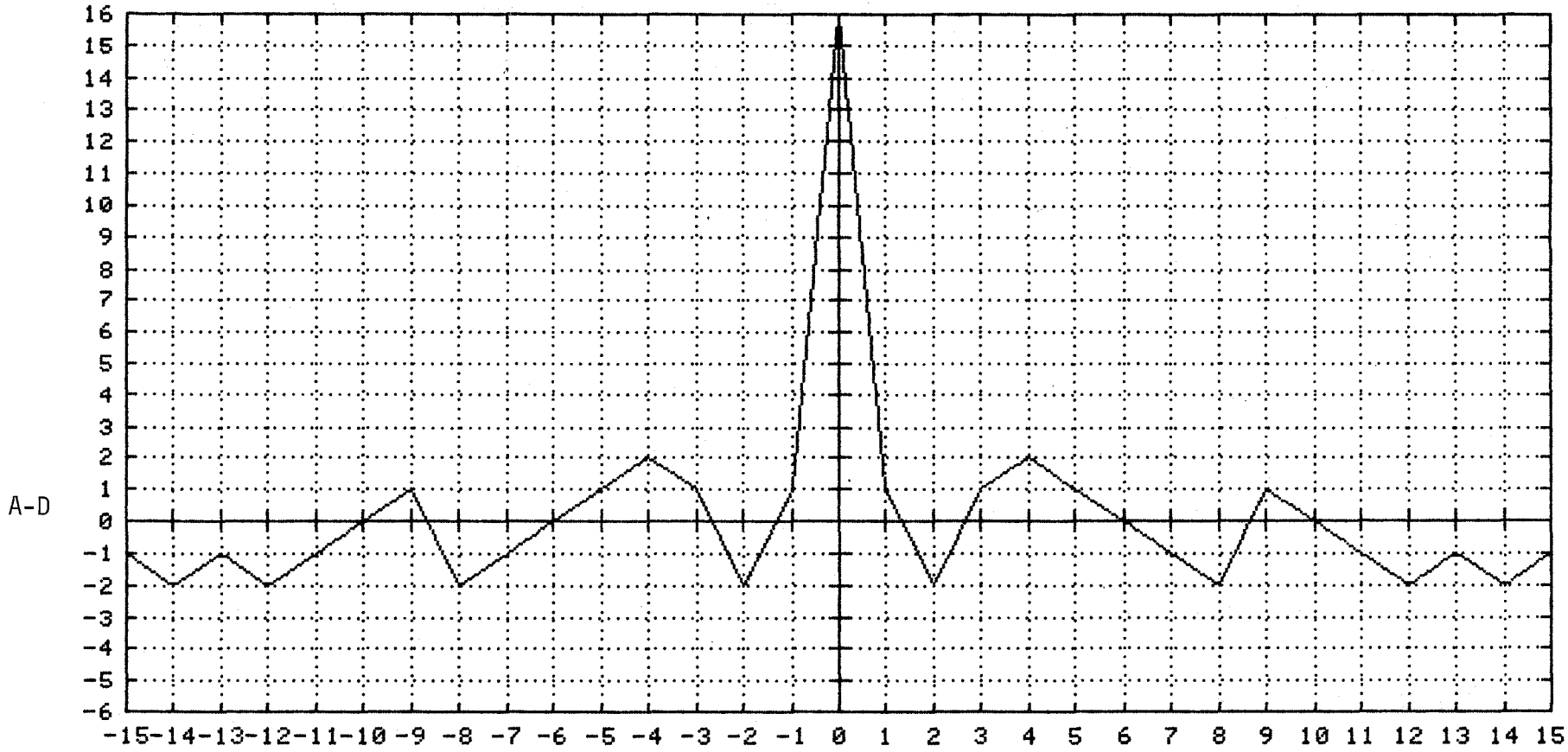


Figure B-7. Autocorrelation.

Table B-2. Examples of Unique Words When Bit Sense is Unknown

LENGTH	MAX ABS ACF	SEQUENCE
3	1	001*
4	1	0001*
5	1	00010*
6	2	000010
7	1	0001101*
8	2	00001011
9	2	000001101
10	2	0000011010
11	1	00011101101*
12	2	000001010011
13	1	0000011001010*
14	2	00000011001010
15	2	000000110010100
16	2	0000011001101011
17	2	00001100100101011
18	2	000001011010001100
19	2	0000111000100010010
20	2	00000100011101001011
21	2	000000101110100111001
22	3	0000000011100100110101
23	3	00000000111001010110010
24	3	000000001110010101100100
25	2	0001100011111101010110110
26	3	00000000110100110101001110
27	3	000000001100111010001011010
28	2	0001100011111101010110110110
29	3	00000000110110010111000110101
30	3	000000001110001101010011011001
31	3	0000000011100010101001011011001
32	3	00000000101101010011011001110001

*Indicates Barker Code

Table B-2 concl'd. Examples of Unique Words When Bit Sense is Unknown

LENGTH	MAX ABS ACF	SEQUENCE
33	3	000000000111100101101010100110011
34	3	0000000001111001011010101001100110
35	3	0000000001111000110011010010010101O
36	3	000000001110100011100110011010010100
37	3	0000000010110110010001101010001110001
38	3	00000000111100001101001010101001100110
39	3	000000010011111000110110010100111001010
40	3	0000000101010110100100111100011011001100

Many efforts have been made to create longer sequences from the Barker sequences that exhibit good values for the autocorrelation. Klyuyev and Silkov (1976), for example, proposed the construction

$$\alpha\alpha\alpha\bar{\alpha} \tag{B-7}$$

where α is a Barker sequence and the superbar indicates complementation. Constructions such as (B-7) can be viewed as examples of Kronecker constructions which are produced as follows (see Stiffler, 1971, or Turyn, 1968):

a) Let s(1), s(2), ... s(ℓ) and t(1), t(2), ... t(m) be two sequences with r(k) and r'(k) the autocorrelations of s() and t() respectively.

b) Form the ℓm long sequence s(1)+t(1), s(1)+t(2), ... s(1)+t(m), s(2)+t(1), ... s(ℓ)+t(m). (B-8)

Unfortunately, the Kronecker constructed sequence (B-8) possesses an autocorrelation function r''(k) such that

$$\max r''(k) \geq (\max r(k))(\max r'(k)) \tag{B-9}$$

and thus can never lead to a better normalized autocorrelation function than either of its component sequences.

A very interesting result due to Moser and Moon and cited on p. 198 of Turyn (1968) is that if a sequence is generated at random (from a balanced Bernoulli source) then one can expect max (|A-D|) to be on the order of the square root of the length of the sequence. Keeping this thought in mind, let us examine a technique that has often been suggested to finding a long sequence with good autocorrelation behavior. This technique is to generate an m-sequence and then pick the best phase to minimize the maximum value of |A-D|. The author has done this with the 63-bit m-sequence generated according to x^6+x^5+1. The best phase (one of several, actually) is

$$101101110110011010101111110000010000110001010011110100011100100 \tag{B-10}$$

Figure B-8 depicts the autocorrelation of this sequence. Note that the maximum absolute value is 6 which is a bit better than the square root of 63.

1.1.4 Concluding Remarks

The reader will note that we have not fully examined the behavior of epoch synchronization under channel noise nor have we even delved into the simple statistics

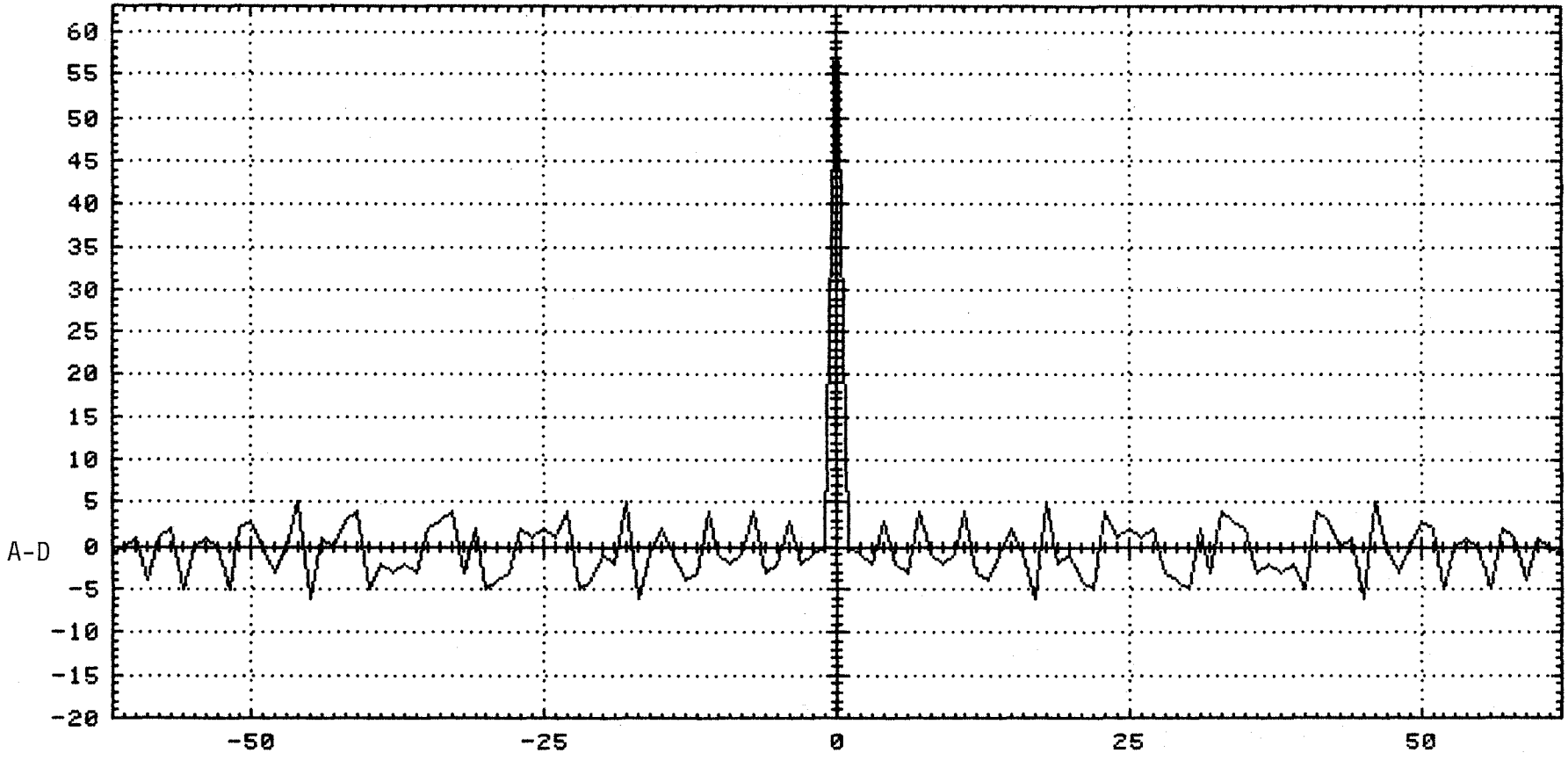


Figure B-8. Autocorrelation.

relevant to choosing an appropriate threshold. The reasons for these omissions are twofold:

a) Epoch synchronization can be implemented by first hard quantizing the incoming data stream into zeros and ones. This hard quantization method may be practical but it is certainly not optimal.

b) The channel noise processes will heavily influence the choice of parameters for the unique word once the matched filter or unique word detection system has been agreed upon. It is only then that the necessary statistics for describing the behavior of the unique word detection system should be derived.

One final observation is that unique word detection is a classical TYPE I/TYPE II decision process as diagramed in Figure B-9. (See Kreyszig, 1967, p. 802.) In picking the unique word, the designer must assess appropriate costs for the TYPE I/TYPE II errors. For example, for an Amateur Radio transmission to be occasionally lost (TYPE II errors) may be far less annoying (costly) than the occurrence of frequent false synchronizations (TYPE I errors).

1.2 Phase Synchronization

Figure B-10 depicts the generic diagram for the phase synchronization process. The top box represents a cyclic digital process, in effect a repetitive sequence of binary n-tuples. The middle box is a time invariant (fixed) mapping from the binary n-tuples to a single binary unit. The sequence of bits thus produced constitutes the sequence. The period of the sequence is, of course, upperbounded by the cycle length of the cyclic digital process.

1.2.1 m-Sequence Synchronization

This phase synchronization process uses a shift register of length n with a primitive polynomial for feedback as the cyclic digital process of Figure B-10, i.e., an m-sequence generator. The combinatorial logic is simply a single tapped stage of the shift register. The sequence is an m-sequence. This is depicted in Figure B-11 to show comportment to the canonical structure of Figure B-10.

As we have seen in the previous appendix, an m-sequence generated by a primitive polynomial of degree n has 2^n-1 distinct phases. To determine the phase and thereby establish synchronization it is necessary to know without error n consecutive bits and the polynomial that generates the m-sequence. If the signal-to-noise ratio is low, this may not be possible by direct demodulation.

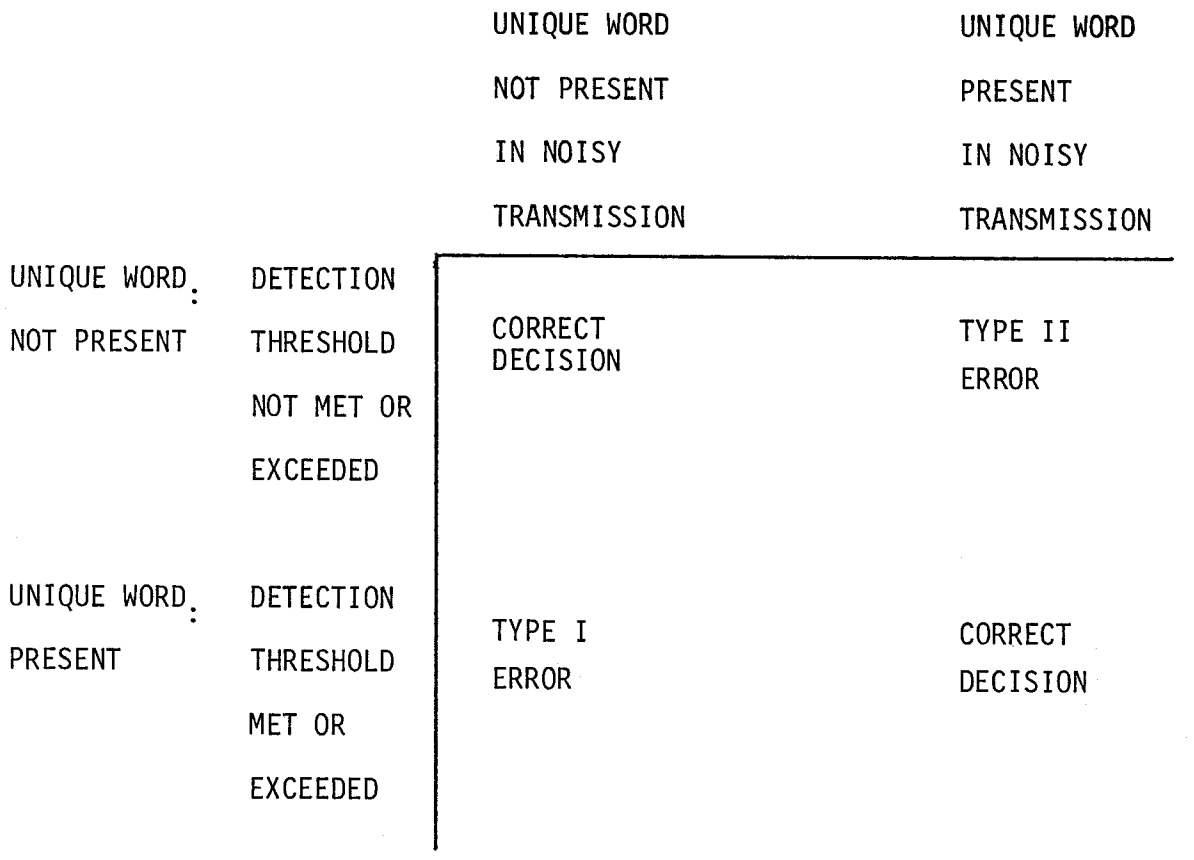


Figure B-9. Type I/Type II errors.

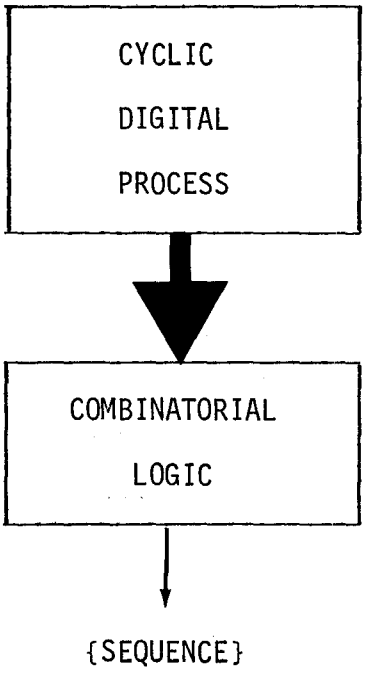


Figure B-10. Phase synchronization, generic diagram.

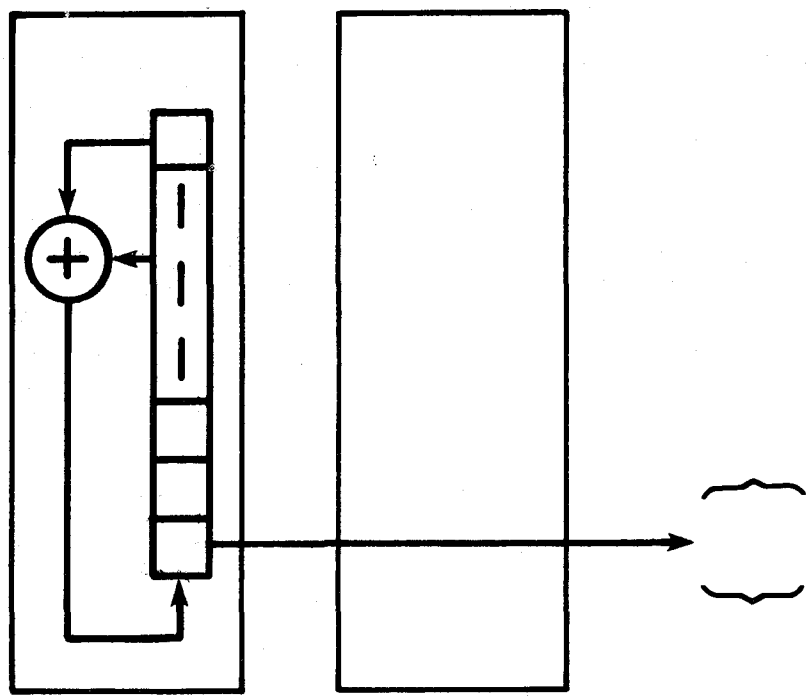


Figure B-11. m-sequence generator.

Consequently we may have to rely on methods that work with more than n bits in making a decision.

The following sections deal with the three possibilities denoted by X's:

	DEMODULATED BITS ERRORLESS	SOME DEMODULATED BITS IN ERROR
PRIMITIVE POLYNOMIAL OF DEGREE n KNOWN	X	X
PRIMITIVE POLYNOMIAL OF DEGREE n UNKNOWN	X	

Polynomial known: errorless reception

This is the simplest of the four possible cases and also the least likely as the result of a spread spectrum system is to spread the signal energy so that the individual chip possesses a low signal-to-noise investment thus making errorless acquisition of n consecutive bits very unlikely.

Assuming, however, that n consecutive bits are obtained without error, then the future behavior of the m-sequence is determined and synchronization is established.

Polynomial known: errors in reception

This is the most common case and actually consists of two subcases. First, if 2^n-1 is sufficiently small that one can expect to see all phases of the m-sequence during synchronization, then one need only create a matched filter to look for m consecutive bits of the m-sequence where m>n for moderate signal-to-noise ratios to m>>n for very low signal-to-noise ratios. When the m-sequence generates the m-bits, the matched filter will detect this epoch and phase synchronization will be accomplished.

If 2^n-1 is not sufficiently small that one can expect to see all of the m-sequence phases during the synchronization period, then the above method is not guaranteed and we must have a better than random estimate of the phase of the m-sequence before we attempt synchronization. Our job, then, is to search through a limited number of phase candidates and winnow the correct one.

The way this is usually done is by means of a serial search procedure implemented via what has become known as a "sliding correlator." The sliding correlator integrates the product of the received sequence against what is presumed to be the correct phase of the m-sequence. If the phase is correct, the integral will show a large departure from the mean value. As an example, consider that we are trying to synchronize with the m-sequence generated by x^6+x^5+1 and let us assume that our estimate of the phase is 4 clock times ahead of the true phase. For this experiment we correlate 6 bits at a time. Our rule for synchronization will be that we have achieved synchronization only if all 6 bits agree. Our procedure will be to test 6 bits. If all 6 agree, we declare that we are in synchrony; if all 6 do not agree, we retard our reference m-sequence by one clock time and compare (integrate) for another 6 bits. This process is depicted in Figure B-12.

The number of bits to be crosscorrelated against the reference m-sequence (6 in this example) will depend upon the type of noise, the appropriate signal-to-noise ratio, the decision threshold and the look time (in this case, the time allowed to achieve synchronization. Clearly, the sliding correlator can require an enormous amount of search time if the number of phase candidates is very large or if the signal-to-noise is very low. Dixon (1976, pp. 181-183) discusses implementation of the sliding correlator by sliding the reference m-sequence in a continuous fashion (i.e., not discrete retardation per our example). Braun (1982) provides an excellent relevant theoretical analysis. He also points out that schemes such as that used in our example are "single dwell time" procedures and he considers a multiple dwell time approach that dynamically changes the width of the correlation window, i.e., the number of bits correlated for each decision. This concept of sequential hypothesis testing can bring far greater efficiency to our search. For background on sequential testing the reader is reffered to Posner and Rumsey (1966).

Polynomial unknown: errorless reception

Yarlagadda and Hershey (1982) have proposed that Gold's characteristic sequence (see Appendix A) be used as a benchmark for m-sequence synchronization. They found that there is a curious crosscorrelation between the truncated Rademacher sequences and an m-sequence at its characteristic sequence phase.

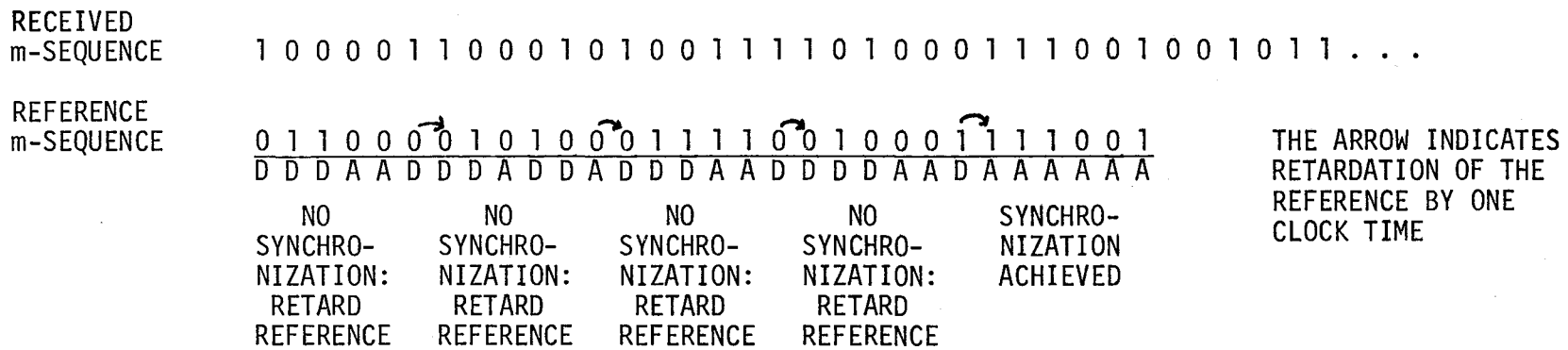


Figure B-12. Sliding correlator.

The Rademacher sequences are those sequences that describe the bit sequences of a normal binary counter started from zero. For example, consider the 3-bit binary counter below:

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

The first Rademacher sequence is the sequence exhibited by the counter's first bit, viz:

$$0\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \tag{B-11}$$

the second rademacher sequence is

$$0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \tag{B-12}$$

and the third is

$$0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \tag{B-13}$$

We drop the last bits of the Rademacher sequences to form sequences of length 2^n-1 where n is the number of bits or stages of the counter. These sequences are termed the truncated Rademacher sequences. We form a matrix R which is composed or partitioned of the n truncated Rademacher sequences of length 2^n-1. For our example:

$$R = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{B-14}$$

Now consider any m-sequence of period 7. Let us arbitrarily select the m-sequence generated by the primitive polynomial x^3+x+1:

$$1\ 0\ 0\ 1\ 1\ 1\ 0 \tag{B-15}$$

We now construct the $(2^n-1)x(2^n-1)$ matrix, S, of all phases of (B-15)

$$S = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad \text{(B-16)}$$

We now multiply R and S using conventional matrix multiplication, i.e., we do not modularly reduce the row-column dot products:

$$RS = \begin{bmatrix} 2 & 1 & 3 & 1 & 2 & 1 & 2 \\ 1 & 3 & 2 & 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 & 1 & 2 & 3 \end{bmatrix} \qquad \text{(B-17)}$$

Notice that the fourth column's entries are all equal. This will be so only of the characteristic sequence of the m-sequence which for our polynomial is:

$$1\ 1\ 1\ 0\ 1\ 0\ 0 \qquad \text{(B-18)}$$

A bank of n-correlators will thus be able to determine the epoch at which an m-sequence, <u>regardless of the generating polynomial</u>, passes through its characteristic sequence phase. Figures B-13 through B-17 show the output of the crosscorrelation

$$\sum_{i=0}^{30} r_k(i)s(i+j) \qquad \text{(B-19)}$$

where $r_k(i)$ is the ith bit of the kth Rademacher sequence and $\{s()\}$ is the m-sequence generated by x^5+x^3+1. The m-sequence is

$$\begin{array}{cccccccccccccccccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 \end{array} \qquad \text{(B-20)}$$

The phases of the m-sequence are arbitrarily assigned by the smaller subscript numbers. The abscissa of B-13 through B-18 are the phase numbers. Figure B-18 is an overlay of Figures B-13 through B-17. This figures allows the reader to spot the point at which the crosscorrelations are all equal and thereby identifies the phase (23) at which the characteristic sequence begins.

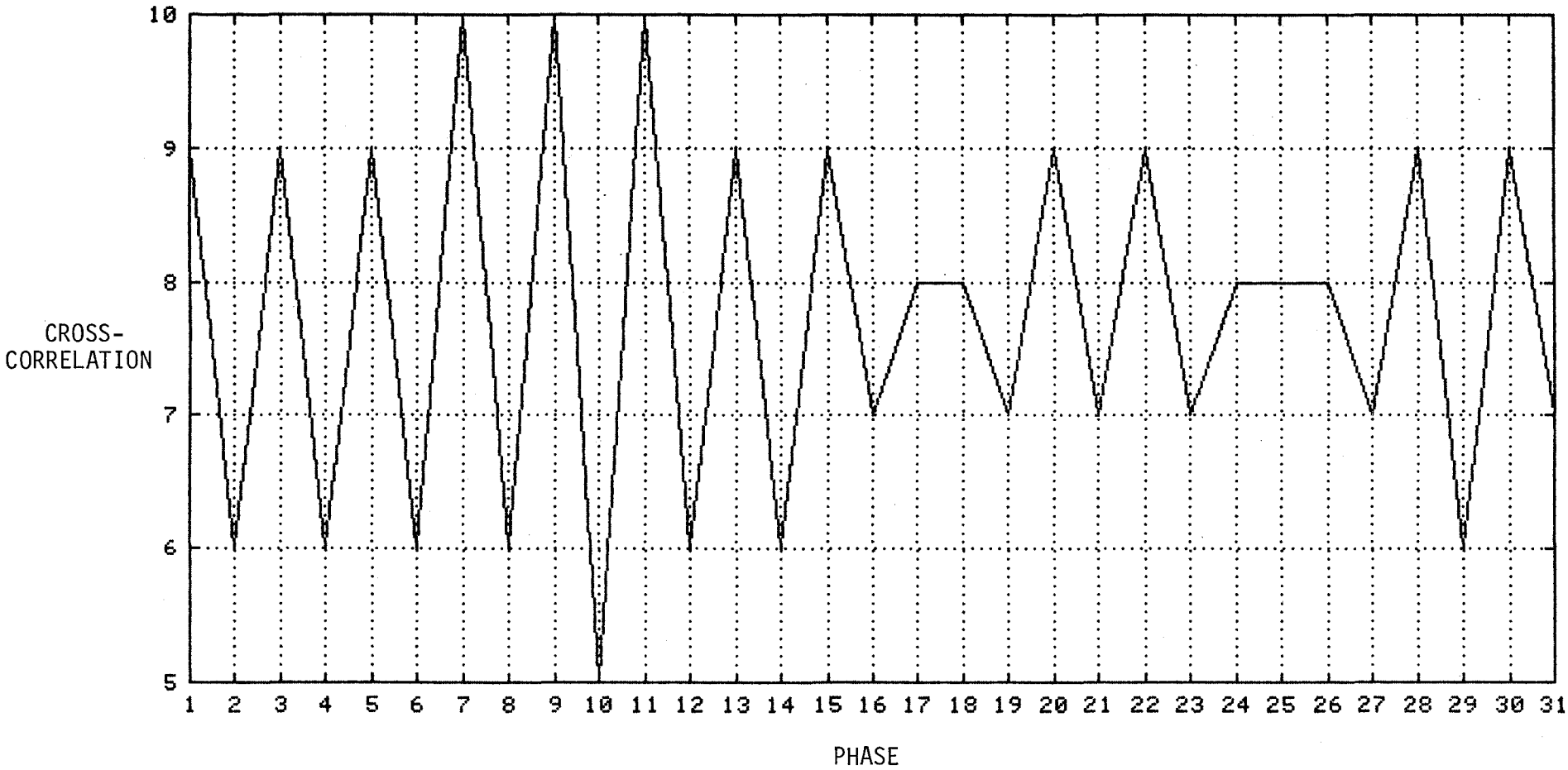


Figure B-13. Crosscorrelation of first Rademacher sequence.

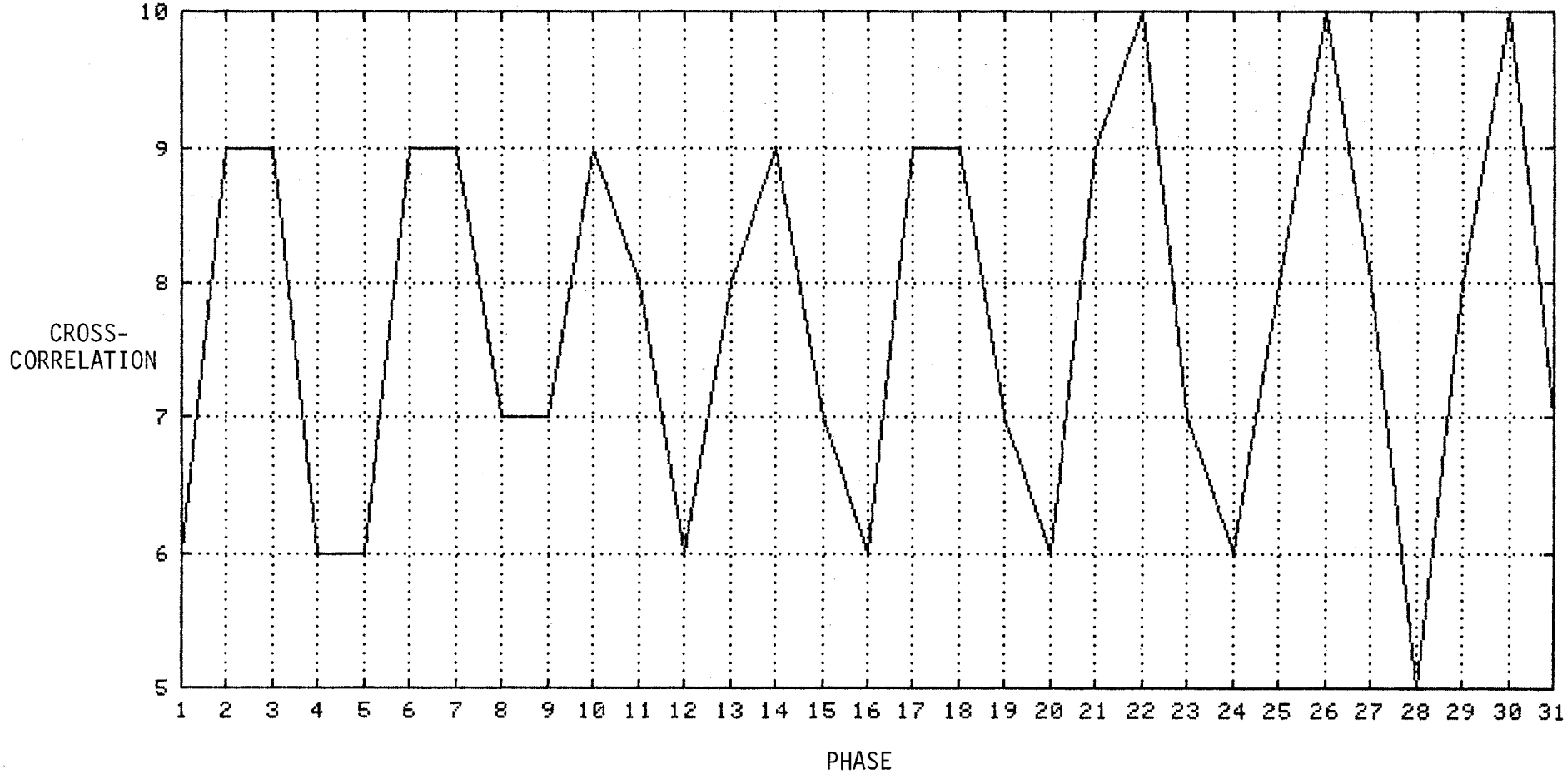


Figure B-14. Crosscorrelation of second Rademacher sequence.

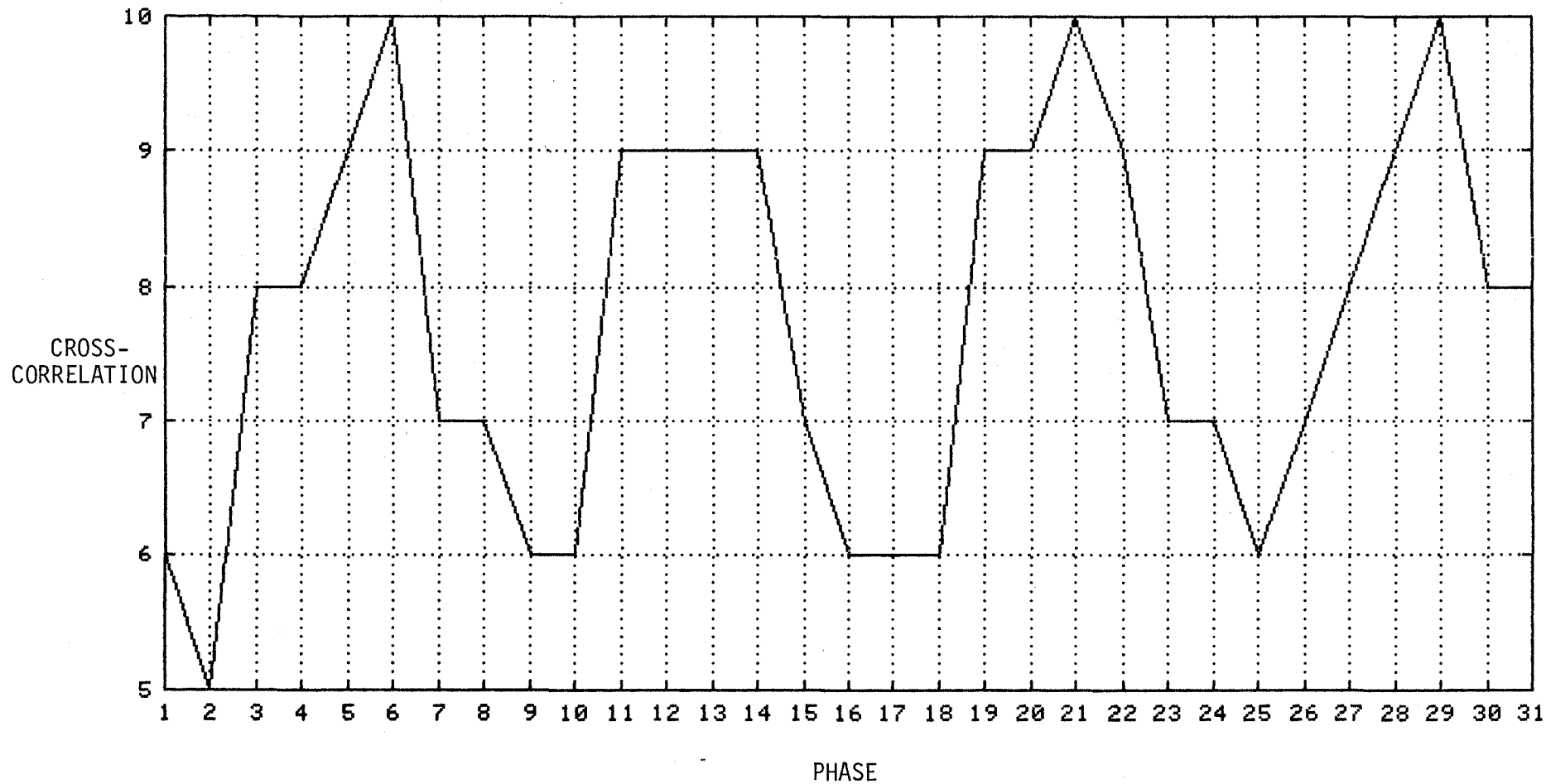


Figure B-15. Crosscorrelation of third Rademacher sequence.

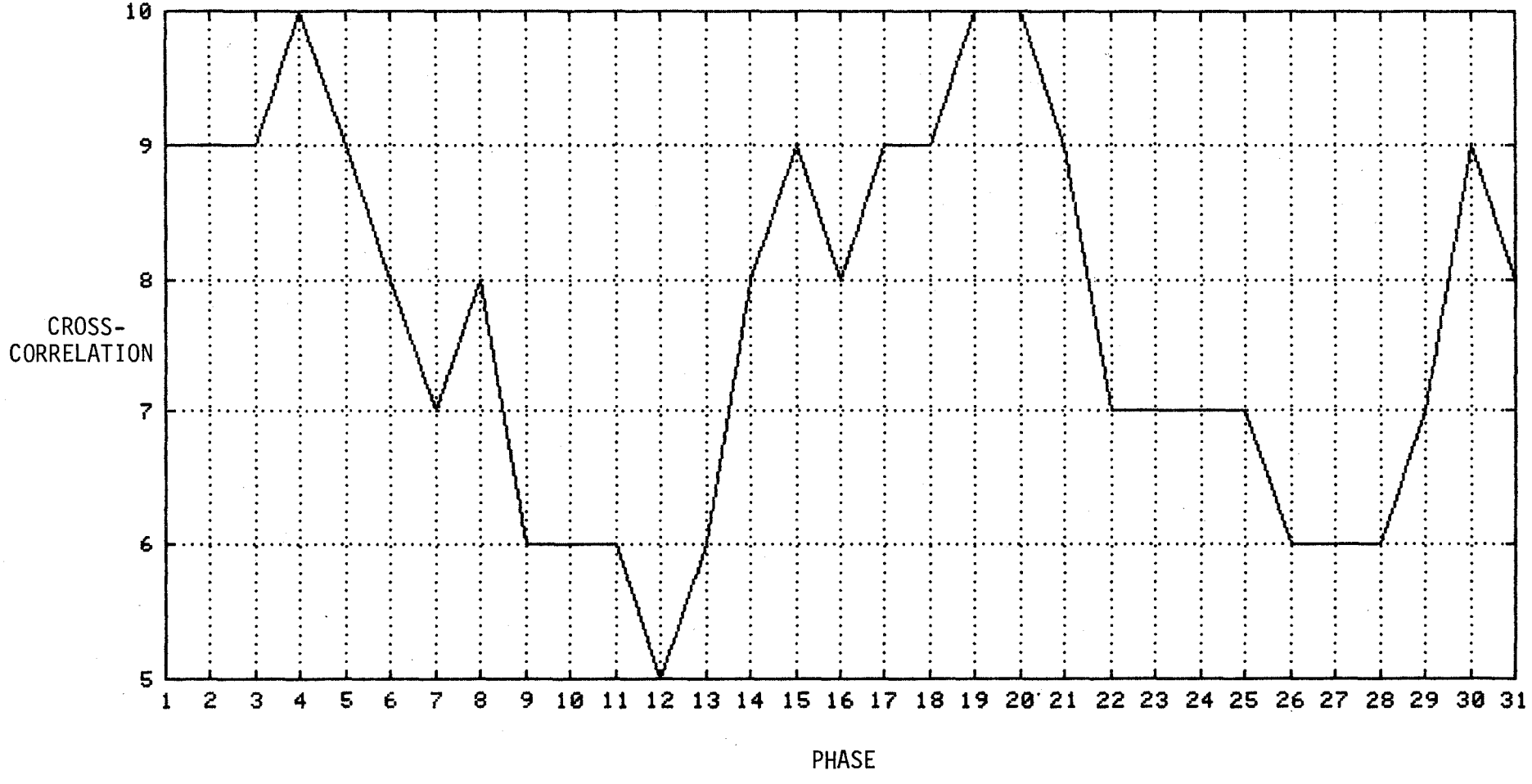


Figure B-16. Crosscorrelation of fourth Rademacher sequence.

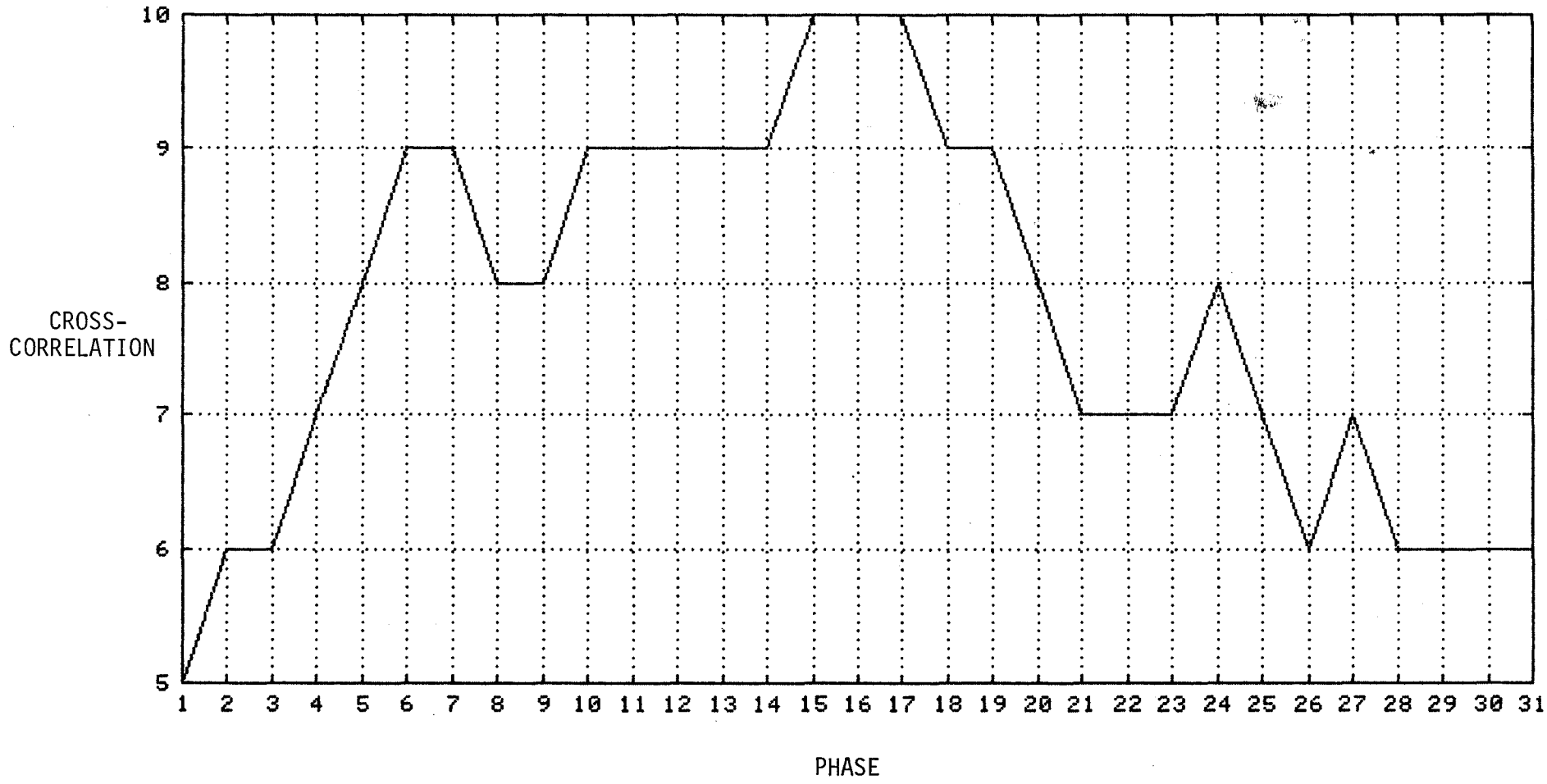


Figure B-17. Croscorrelation of fifth Rademacher sequence.

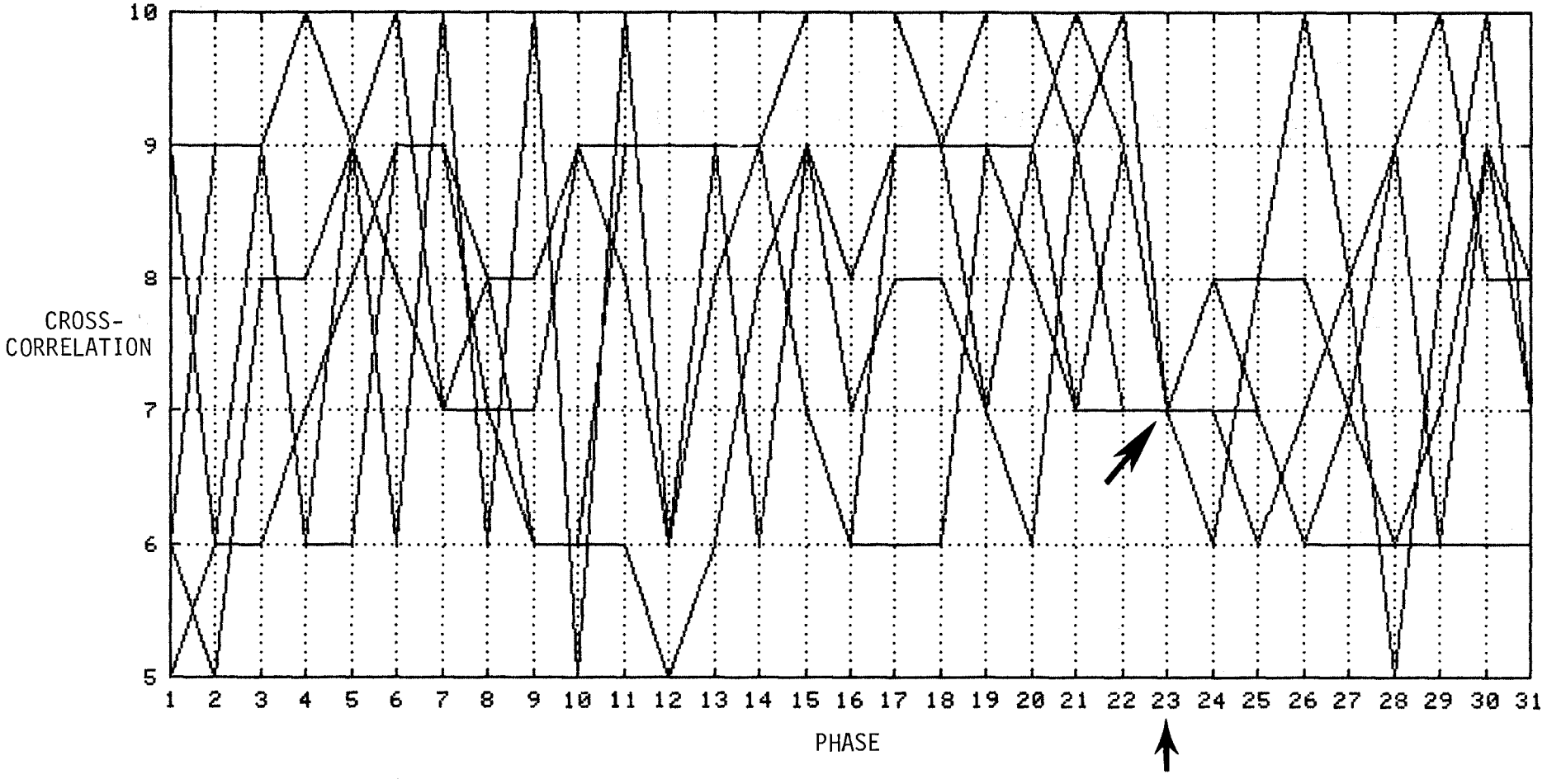


Figure B-18. Coalescence of crosscorrelations identifying the characteristic sequence.

1.2.2 Rapid Acquisition Sequences

This synchronization process was introduced by Stiffler (1968). It uses a normal binary counter of n stages as the cyclic digital process. The counter is started at zero and incremented by one every clock time. When the count has reached 2^n-1, the next incrementation causes the counter to be reduced modulo 2^n and have all zeros in its stages. The contents of the counter stages $x_1, x_2, \ldots x_n$ (x_1 is the least significant stage) are input to the following combinatorial logic,

$$f(x_1,x_2, \ldots x_n) = \begin{cases} 0 \text{ if } \sum_{i=1}^{n} x_i \leq [\frac{n}{2}] \\ 1 \text{ otherwise} \end{cases} \qquad \text{(B-21)}$$

where [ ] is the greatest integer function. The sequence produced, the output of the boolean function specified in (B-21), is termed the Rapid Acquisition Sequence (RAS) of length 2^n. Figure B-19 depicts this configuration in terms of our canonical model (Figure B-10).

Two items are worth mentioning. First, the function f() of (B-21) is a non-linear boolean function that is threshold realizable. Because it is threshold realizable, it can be implemented in very fast hardware. Second, the sequence described by the successive contents of x_i is the ith Rademacher sequence.

It is easy to verify that the RAS of length 8 is

00010111 (B-22)

Consider the full period crosscorrelation of (B-22) against the first Rademacher sequence. Because the first Rademacher sequence has only two distinct phases, i.e., the "normal" phase

01010101 (B-23a)

and the other phase

10101010 (B-23b)

it is clear that the crosscorrelation will exhibit only two values. By direct computation

00010111
01010101
ADAAAADA A-D=6-2=4

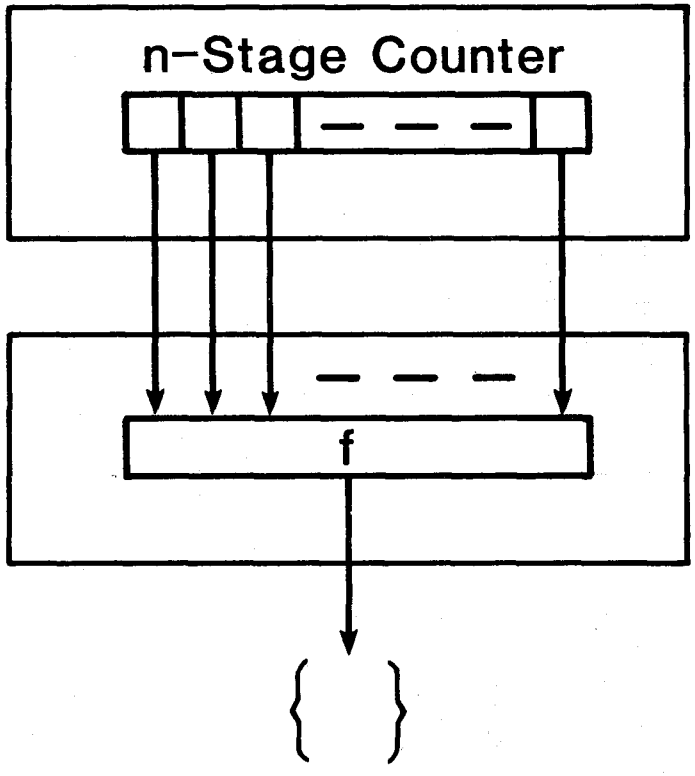


Figure B-19. Rapid Acquisition Sequence (RAS) generator.

and

$$\begin{array}{ll} 00010111 & \\ \underline{10101010} & \\ \text{DADDDDAD} & \text{A-D=-4} \end{array}$$

Thus the crosscorrelation is 4 for the Rademacher phase of (B-23a), the "normal" phase and -4 for the phase of (B-23b). The synchronization procedure begins to emerge. The first step is to crosscorrelate the first Rademacher sequence of length 2^n over a full period of the 2^n long RAS. This first step will give us one of two equally probable answers and this 'bit' of information resolves the phase of the RAS within modulo 2. Once the phase has been resolved modulo 2, the second Rademacher sequence is crosscorrelated in both of its possible phases. (There are, of course, four distinct phases of the second Rademacher sequence, but we consider only the two phases that "survive" the first test, i.e., only two of the four phases will be consonant with the determined phase of the first Rademacher sequence.) This computation tells us the phase of the RAS modulo 4. Thus at each stage we gain a bit of information and sequentially recover the counter's stage sequences. Stiffler (1968) showed that the (normalized, i.e., A-D divided by the sequence length) crosscorrelation of the kth 2^n long Rademacher sequences with the 2^n long RAS is

$$\frac{1}{2^{n-1}} \binom{n-1}{\frac{n-1}{2}} \qquad n \text{ odd} \qquad \text{(B-24a)}$$

and

$$\frac{1}{2^n} \binom{n}{\frac{n}{2}} \qquad n \text{ even} \qquad \text{(B-24b)}$$

if the k-1 phases have been correctly resolved. Note that (B-24a,b) are <u>independent</u> of k. Using Stirling's formula, Stiffler approximates (B-24a,b) by

$$(2/\pi)^{1/2} \; n^{-1/2} \qquad \text{(B-25)}$$

A graph of (B-24a,b) and the approximation (B-25) is given in Figure (B-20).

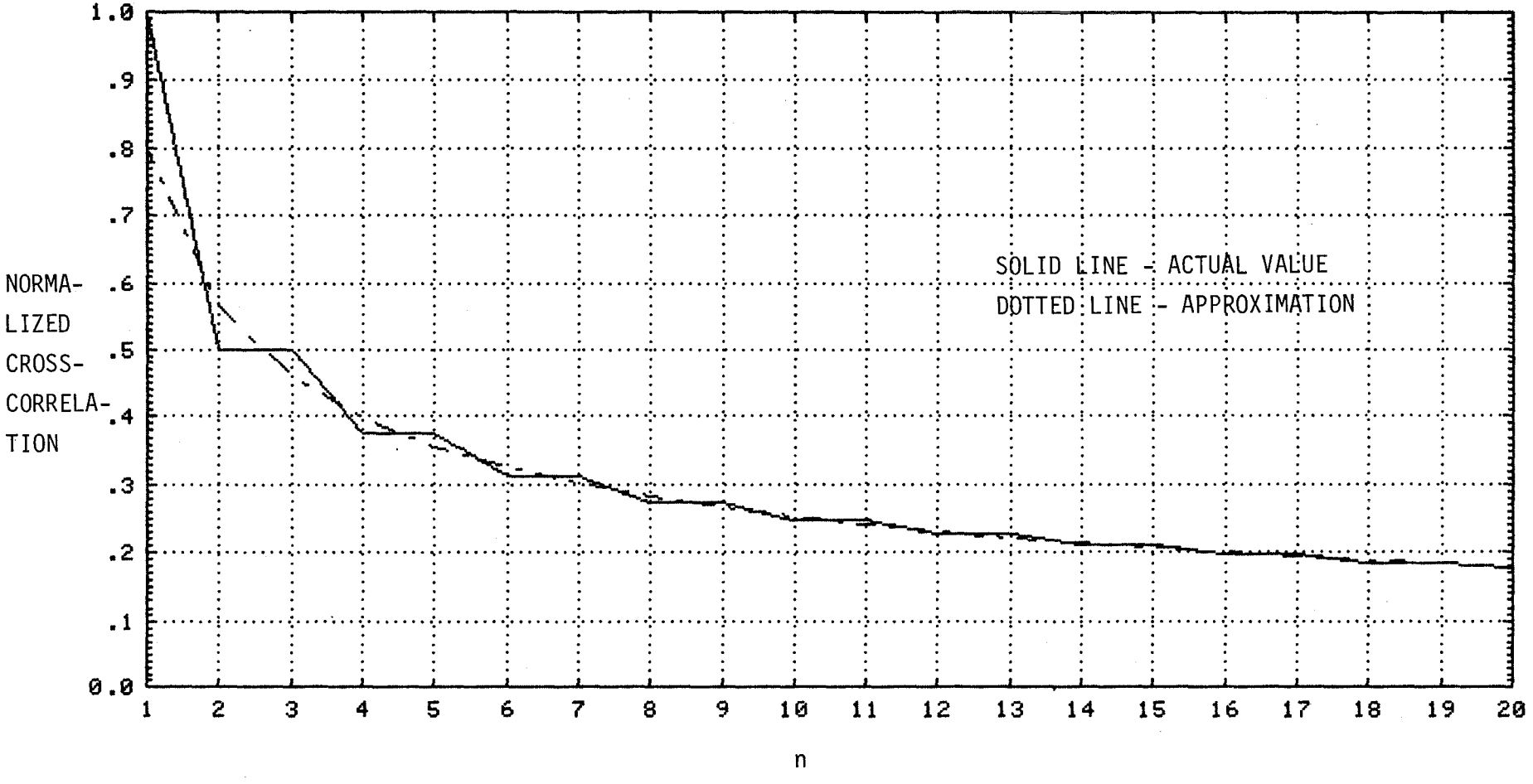


Figure B-20. Normalized crosscorrelation of the 2^n-long RAS.

As an example, consider that we wish to recover the phase of the 8-long RAS. We assume errorless reception for the example and that we are looking at the phase

1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 . . .

The first step is to compute the crosscorrelation of the first Rademacher sequence over a full period:

```
1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 . . .
0 1 0 1 0 1 0 1
D A D D A D D D                          A-D=-4
```

The crosscorrelation is negative and therefore the first stage of the counter is out of phase with the normal phase of the first Rademacher sequence. The next step is to delay one bit and thus bring us into phase with the first Rademacher sequence component of the RAS and then crosscorrelate the second Rademacher sequence with the RAS:

```
1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 . . .
0 1 0 1 0 1 0 1   0 0 1 1 0 0 1 1
                  D D D D A D D A                    A-D=-4
```

The crosscorrelation is again negative so we bring our second Rademacher sequence into phase by delaying it two bits and then crosscorrelate the third Rademacher sequence with the RAS:

```
1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 0 0 0 1 0
0 1 0 1 0 1 0 1   0 0 1 1 0 0 1 1   0 0 0 0 1 1 1 1
                                    A A A D D A A A          A-D=+4
```

The crosscorrelation is positive so here we find that our third Rademacher sequence is in phase and we are synchronized. Thus, a 2^n-long sequence can be synchronized with n decisions. A decision need not require a full period as per the example. We also need not "waste" time by delaying the crosscorrelations as shown, we could instead have advanced the Rademacher sequences. We chose not to do these things in order to enhance the clarity of the presentation. Stiffler (1968) also examines the question of how many bits need be crosscorrelated when operating in Additive White Gaussian Noise.

Now consider the RAS of length 32:

00000001000101110001011101111111 (B-26)

Notice that there are runs of bits (stretches of identical bits) up to length 7 in (B-26). Ipatov et al. (1975) noted that the RAS of length 2^n(n odd) will typically display runs of lengths up to $2^{(n+1)/2}-1$. Ipatov et al. noted that these long run lengths could have deleterious effects on some systems that depend upon transitions in the data for clock recovery. Ipatov et al. proposed a modification to the RAS (MRAS) that limits the maximum length of the runs to 2 bits without affecting the equal crosscorrelation property (B-24a). Ipatov et al.'s method is to change the combinatorial logic function from (B-21) to

$$f'(x_1,x_2,\ldots x_n) = r_1(x_1) + g(x_2,x_3,\ldots x_n) \tag{B-27}$$

where the addition is modulo 2 and

$$g(x_2,x_3,\ldots x_n) = \begin{cases} 0 \text{ if } \sum_{i=2}^{n} x_i \leq \frac{n-1}{2} \\ 1 \text{ otherwise} \end{cases} \tag{B-28}$$

The MRAS of length 32 (n=5) is then

01010101010101100101011001101010

1.2.3 The Thue-Morse Sequence

The Thue-Morse sequence (TMS) is that sequence produced by exclusive-oring or adding modulo two the contents of all the stages of an infinitely long binary counter started at zero and allowed to count indefinitely. The first few terms of the TMS are seen to be 0110100110... from the following:

COUNTER	TMS
0	0
1	1
10	1
11	0
100	1
101	0
110	0
111	1
1000	1
1001	0
.	.
.	.
.	.

Figure (B-21) casts the TMS in terms of the canonical model (Figure B-10).

Hershey (1979) reported that the TMS (a) never exhibits runs greater than length 2 (b) never repeats and (c) is related to the coefficients of $\{x^i\}$ in the expansion of the infinite product

$$\prod_{i=0}^{\infty} (1-x^{2^i})$$

In his paper Hershey (1979) showed that the TMS could serve as a comma-free code to synchronize binary counters. Hershey and Lawrence (1981) suggested a follow-on method which is more amenable to implementation in the types of communications systems discussed in this report.

A statistical property of the TMS

The TMS exhibits the following statistical property that is key to the method. Consider the two cases in which the counter's first m bits are either 011···1 or 111···1 (rightmost bit is least significant).

1) For the former case, the probability that the TMS will change at the next count is one if m is odd and zero is m is even. This is because the next count will result in the first m bits becoming 100···0 which represents a change in the number of ones in the counter from m - 1 to 1; clearly, the sum of ones modulo 2 is unchanged if and only if m is even.

2) For the latter case, the next count brings the first m bits to all zero and a carry propagates into the higher bits of the counter. The probability that the carry will stop at the m + 1st, m + 3rd, m + 5th, etc. position is 1/2 + 1/8 + 1/32 + ··· = 2/3. (The countersize is assumed large.) If the carry propagates in this manner, there will be a unit change, modulo 2, in the density of ones in the counter above the first m bits. In the lower part of the counter, the first m bits, the counter experiences a unit change in the density of ones modulo two if and only if m is odd. Combining these two effects, we note then that the probability that the TMS will change at the next count is one-third if m is odd and two-thirds if m is even.

Use of the TMS for synchronization

The goal is to resolve the first s stages of the TMS transmitter's counter by using the above statistical property. The method is best and most easily

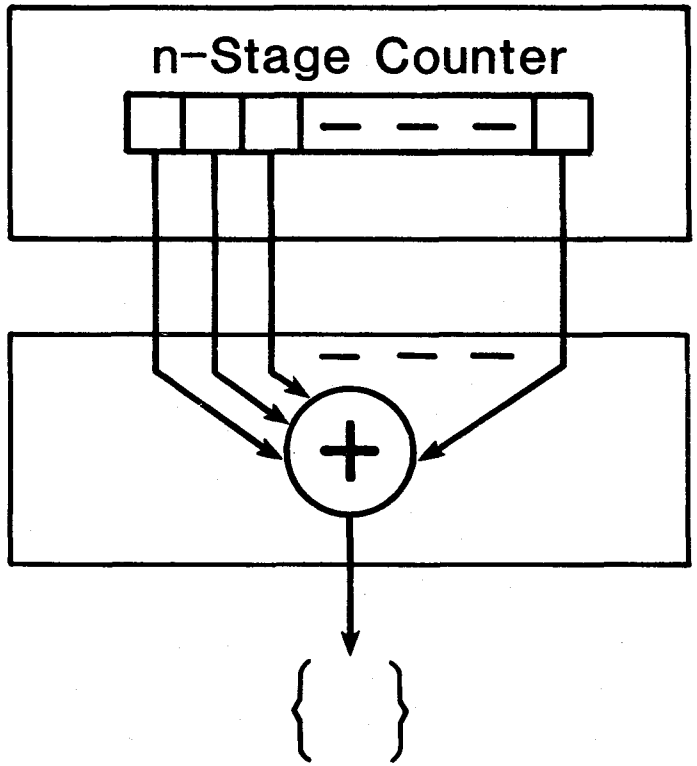


Figure B-21. The Thue-Morse (TM) sequence generator.

presented as an example which can be extended in an obvious fashion. Consider the following 30-bit segment from somewhere in the TMS:

101101001100101101001011001101

(time order is left to right). We will examine this segment in light of the above statistical property. We first make two counts, called "A" and "B" counts, of bit reversals or changes in the TMS. The B counts are one bit out of phase with the A counts. Starting at the beginning of our 30-bit segment, arbitrarily make only two counts each for A and B as shown in the top block of Figure B-22.

Note that the TMS changed once during the A "windows" and twice during the B windows. From the TMS statistical property we conclude that the counter generating the TMS segment had a zero as its first bit every time a B window was begun. We have thus resolved, or phased, the first bit sequence from the transmitter's counter. Using this information, we can now proceed to resolve the second bit time sequence of the transmitter's counter.

To do this, we choose our sampling windows to begin at those times when the first bit in the transmitter's TMS counter is a one. What we are trying to determine is which of the sampling windows, A or B, is "seeing" the first two bits in the transmitter's TMS counter chance from 01 to 10. From the middle block of Figure B-22, we note that the TMS changed once during the B windows and did not change during the A windows. Thus, we know that the counter generating the TMS segment had a zero as its second bit and a one as its first bit every time an A window was begun.

Going one step further, we can resolve the third bit. As we already know how the first two bits are progressing, we shall now sample the TMS whenever the first two bits in the TMS counter are both one as shown in the bottom block of Figure B-22. This will allow us to find the 011 to 100 transitions and thus resolve the third bit.

From the bottom block of Figure B-22, we note that the TMS changed once during the A windows and twice during the B windows. Thus, we know that the first three bits in the TMS counter were 011 every time a B window was begun. Other bits can be resolved in a similar manner.

The reader should note that in order to make n "A-B" counts to statistically resolve the sth bit of the TMS counter (n may have to be large under very noisy

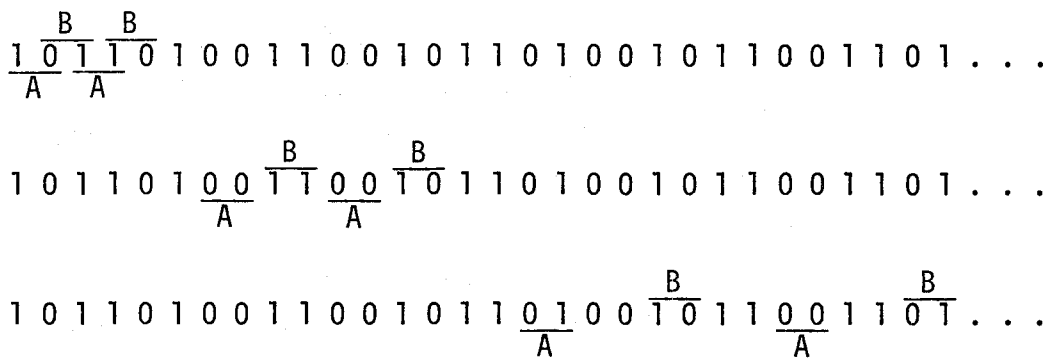


Figure B-22. A-B counts to determine: first bit of TMS counter (top block); second bit of TMS counter (middle block); and third bit of TMS counter (bottom block).

conditions) requires a TMS segment on the order of $n \cdot 2^s$ bits as each "A-B" count requires approximately 2^s bits. Thus, the number of bits required to sequentially resolve the first m bits using n hard decisions per bit is approximately $n(2 + 2^2 + 2^3 + \ldots + 2^m)$ plus a few "overhead" bits. The total is on the order of $n \cdot 2^{m+1}$ bits. As in the case of the RAS, we have recovered the same number of bits as decisions made.

1.2.4 Concluding Remarks

We have attempted to present a cursory look at phase synchronization. As in the portion that dealt with epoch synchronization, we have not examined seriously the behavior of the systems considered within a noisy environment. Nor have we attempted to be complete. Some extremely important systems such as Titsworth's ranging (JPL) codes (1964) and Bluestein's pseudorandom-interleaving scheme (1968) have been omitted as the supporting mathematics necessary to appreciate them would require an inordinate amount of space.

2. REFERENCES: APPENDIX B

Barker, R. (1953), Group synchronizing of binary digital systems, Communication Theory, Willis Jackson (Ed.), (Academic Press), Chapter 19, pp. 273-287.

Bluestein, L. (1968), Interleaving of pseudorandom sequences for synchronization, IEEE Trans. Aerospace Electron. Systems AES-4, No. 4, pp. 551-556, July.

Braun, W. (1982), Performance analysis for the expanding search PN acquisition algorithm, IEEE Trans. Commun. COM-30, No. 3, pp. 424-435, March.

Dixon, R. (1976), Spread Spectrum Systems (Wiley).

Gabard, O. (1968), Design of a satellite time-division multiple-access burst synchronizer, IEEE Trans. Commun. Technol. COM-16, No. 4, August.

Hershey, J. (1979), Comma-free synchronization of binary counters, IEEE Trans. Inform. Theory IT-25, No. 6, pp. 724-725, November.

Hershey, J., and W. Lawrence (1981), Counter synchronization using the Thue-Morse sequence and PSK, IEEE Trans. Commun. COM-29, No. 1, pp. 79-80, January.

Ipatov, V., Yu. Kolomenskiy, and P. Sharanov (1975), On modified rapid search sequences, Radio Engineering and Electronic Physics 20, pp. 135-136, September.

Klyuyev, L., and N. Silkov (1976), Periodic sequences synthesized from Barker sequences, Telecommunications and Radio Engineering 30, No. 4, pp. 128-129.

Kreyszig, E. (1967), Advanced Engineering Mathematics (John Wiley and Sons).

Lindner, J. (1975a), Binary sequences up to length 40 with best possible autocorrelation function, Part 1: Complete Tables, Institute für elektrische Nachrichten-technik der rheinisch-westfälischen Technischen Hochschule Aachen, Internal Report, September.

Lindner, J. (1975b), Binary sequences up to length 40 with best possible autocorrelation function, Electron. Letters 11, No. 21, p. 50, October.

Maury, J., and F. Styles (1964), Development of optimum frame synchronization codes for Goddard Space Flight Center PCM Telemetry Standards, Proc. of 1964 National Telemetering Conference, Los Angeles, CA, June 2-4, 1964.

Nuspl, P., K. Brown, W. Steenaart, and B. Ghicopoulos (1977), Synchronization methods for TDMA, Proc. IEEE 65, No. 3, March.

Petit, R. (1967), Pulse sequences with good autocorrelation properties, Microwave J., pp. 63-67, February.

Posner, E., and H. Rumsey, Jr. (1966), Continuous sequential decision in the presence of a finite number of hypotheses, IEEE Trans. Inform. Theory IT-12, No. 2, pp. 248-255, April.

Schrempp, W., and T. Sekimoto (1968), Unique word detection in digital burst communications, IEEE Trans. Commun. Technol. COM-16, No. 4, August.

Sekimoto, T., and J. Puente (1968), A satellite time-division multiple-access experiment, IEEE Trans. Commun. Technol. COM-16, No. 4, August.

Stiffler, J. (1968), Rapid acquisition sequences, IEEE Trans. Inform. Theory IT-14, No. 2, March.

Stiffler, J. (1971), Theory of Synchronous Communications (Prentice-Hall).

Titsworth, R. (1964), Optimal ranging codes, IEEE Trans. Space Electron. Telemetry SET-10, pp. 19-30, March.

Turyn, R. (1968), Sequences with small correlation in Error Correcting Codes, H. Mann (Ed.), (John Wiley and Sons, Inc.), pp. 195-228.

Yarlagadda, R., and J. Hershey (1982), Benchmark synchronization of m-sequences, Electron. Letters 18, No. 2, pp. 68-69, January.

APPENDIX C: SPECTRAL SHAPING

1. INTRODUCTION

We may, of course, wish to shape our output spectrum for many reasons. One immediately obvious motivation is to promote "spectral disjointedness" with concentrated groupings of narrowband communications. One way is to filter the IF or rf. This is a kind of "brute force" approach. Not only is it often difficult to do but it can lead to nonlinearities and also a loss of signal-to-noise ratio. It may, in the end, be necessary to do this but it may first be worthwhile to consider that spectrum shaping is the result of a multidimensional process. Consider Figure C-1. We know that what gives us our characteristic sinc-squared envelope is the rectangular pulse we use in modulation following the OUTPUT. If we were to change the modulating waveform, we would change the spectrum. Much work has been done along these lines. Glance (1971) has studied arbitrary pulse shapes. Holmes (1982) devotes an early portion of his book to some special pulse and process spectra. Mavraganis (1979) has investigated pulse shapes other than the "conventional" rectangular one specifically for spread spectrum communications.

Unfortunately, the beauty and simplicity of many of the DS modulation schemes' implementation depend, inherently, on using the common rectangular pulses. All is not necessarily lost, however, as there is still the possibility of varying the statistics of the CODE by "Markov filtering."

2. MARKOV FILTERING

Consider again the CODE module of Figure C-1. Until now we have considered the CODE as a balanced Bernoulli source. Suppose that we now modify the source in the following way. We pass the bits from the balanced Bernoulli source through an n-stage shift register as shown in Figure C-2. We define the state of the shift register at time t as the n-tuple $(b_1^t, b_2^t, \ldots b_n^t)$ where b_i^t is the bit (contents) of stage i at time t.

There are, of course, 2^n possible states and the progression through the states is described by a DeBruijn diagram. For example, if n=3 the flow would be as shown in Figure C-3. We now restrict the flow by not allowing some of the states and investigate what this can do for us. The example we shall use is based on Figure C-2 with n=3. The restriction is that we do not allow either the 000 or 111 tuples to occur. In other words, if the register contains 100 or 011 at time t, then no matter what bit the balanced Bernoulli source produces at time t+1, the bit is set to a 1 or 0, respectively, and the states at time t+1

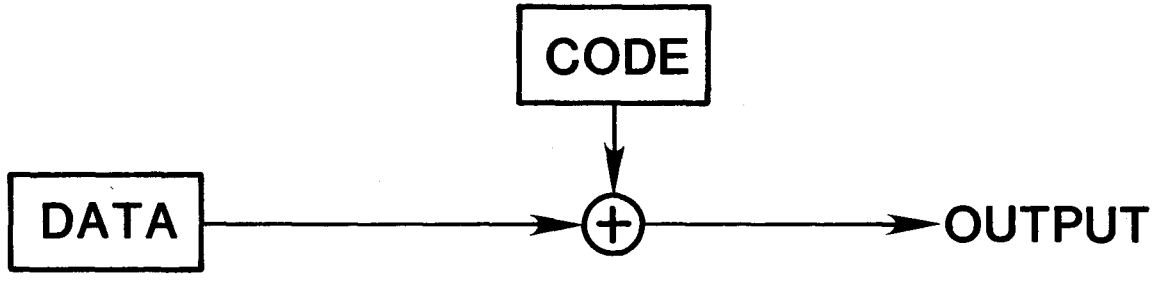


Figure C-1. The basic DS spread spectrum system.

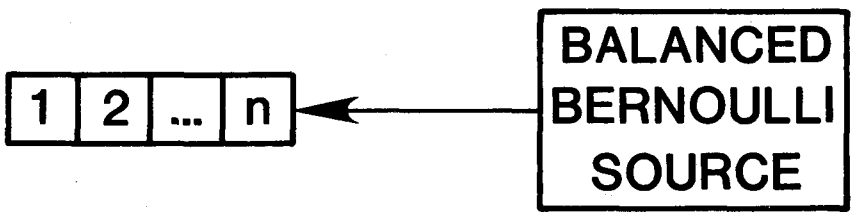


Figure C-2. The Markov filter window.

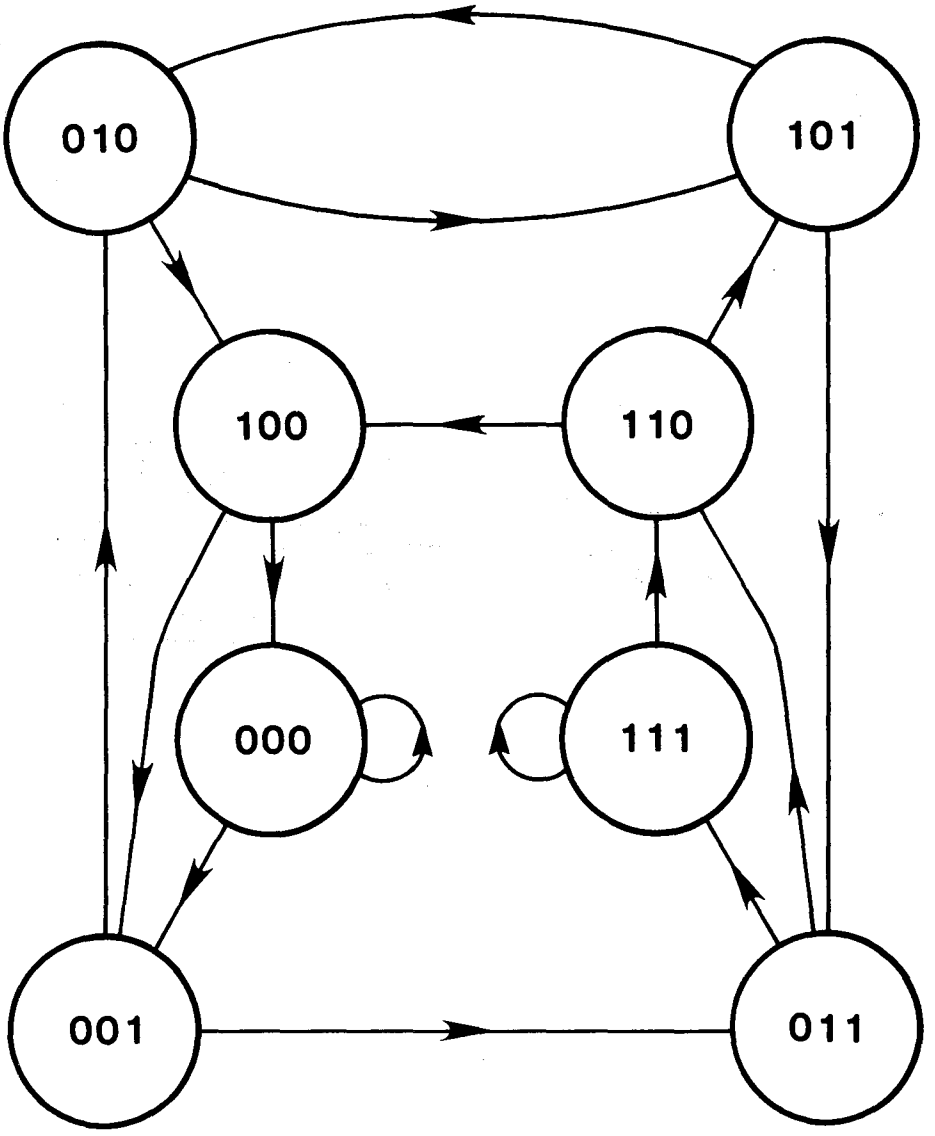


Figure C-3. The DeBruijn diagram.

must be 001 or 110, respectively. The restricted process is depicted in Figure C-4. The numbers aside the transition arrows are the transition probabilities. The encircled numbers inside the state circles are arbitrarily assigned state numbers. The 3 bits are the contents of the three shift register stages. The plus or minus ones are the output values of the states. (We are considering plus and minus ones vice zeros and ones.)

This excision of states changes the spectrum of the CODE stream (the stream of plus and minus ones, the successive outputs of the states). Sittler (1956) has done an excellent job of presenting the relevant mathematics and we will make extensive use of his work.

To analyze our example, we first form the matrix $Q=(q_{ij})$ in which q_{ij} is the Markov probability of a transition from state i to state j. By inspection we see that

$$Q = \begin{matrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{matrix} \qquad \text{(C-1)}$$

Sittler shows that the (right-half plane) power spectral density, $\Phi^{+}(Z)$, is produced by computing

$$\Phi^{+}(Z)=\sum_{i,j} a_i a_j p_i(\infty) P_{ij}(Z) \qquad \text{(C-2)}$$

where the $\{p_i(\infty)\}$ are the steady state probabilities of the Markov process, the $\{a_i\}$ the output values of the states, and the P_{ij} are the elements of the matrix

$$(I-ZQ)^{-1} \qquad \text{(C-3)}$$

Computing (C-3) for our case we find that the inverse matrix is

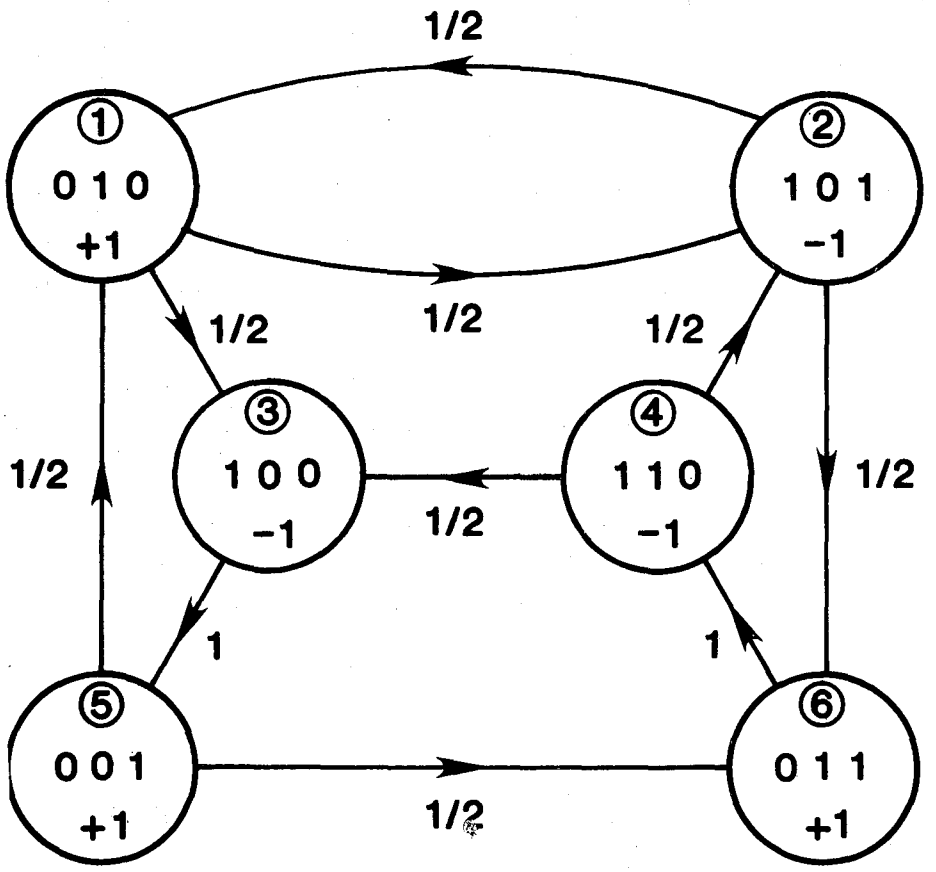


Figure C-4. The Markov process resulting from an excision of states in the DeBruijn diagram.

$$\begin{matrix}
1-\frac{Z^3}{4}-\frac{Z^4}{4} & \frac{Z}{2} & \frac{Z}{2} & \frac{Z^3}{4}+\frac{Z^4}{4} & \frac{Z^2}{2} & \frac{Z^2}{4}+\frac{Z^3}{4} \\
\frac{Z}{2} & 1-\frac{Z^3}{4}-\frac{Z^4}{4} & \frac{Z^2}{4}+\frac{Z^3}{4} & \frac{Z^2}{2} & \frac{Z^3}{4}+\frac{Z^4}{4} & \frac{Z}{2} \\
\frac{Z^2}{2} & \frac{Z^3}{4}+\frac{Z^4}{4} & 1-\frac{Z^2}{4}-\frac{Z^3}{4} & \frac{Z^3}{2} & Z-\frac{Z^3}{4}-\frac{Z^4}{4} & \frac{Z^2}{2} \\
\frac{Z^2}{4}+\frac{Z^3}{4} & \frac{Z}{2} & \frac{Z}{2} & 1-\frac{Z^2}{4}-\frac{Z^3}{4} & \frac{Z^2}{2} & \frac{Z^2}{4}+\frac{Z^3}{4} \\
\frac{Z}{2} & \frac{Z^2}{4}+\frac{Z^3}{4} & \frac{Z^2}{4}+\frac{Z^3}{4} & \frac{Z^2}{2} & 1-\frac{Z^2}{4}-\frac{Z^3}{4} & \frac{Z}{2} \\
\frac{Z^3}{4}+\frac{Z^4}{4} & \frac{Z^2}{2} & \frac{Z^2}{2} & Z-\frac{Z^3}{4}-\frac{Z^4}{4} & \frac{Z^3}{2} & 1-\frac{Z^2}{4}-\frac{Z^3}{4}
\end{matrix} \qquad \text{(C-4)}$$

Substituting the values of (C-4) into (C-2) and computing the right half phase power spectral density, we obtain

$$\Phi^{+}(Z) = \frac{1-\frac{Z}{3}-\frac{7}{12}Z^2-\frac{Z^3}{12}}{1-\frac{Z^2}{4}-\frac{Z^3}{2}-\frac{Z^4}{4}} \qquad \text{(C-5)}$$

Converting (C-5) to trigonometric functions by noting that $\Phi(\omega)=\Phi^{+}(Z)+\Phi^{+}(Z^{-1})$ we obtain:

$$\Phi(\omega) = \frac{342+24\cos(\omega T)-150\cos(2\omega T)-144\cos(3\omega T)-72\cos(4\omega T)}{198+72\cos(\omega T)-54\cos(2\omega T)-144\cos(3\omega T)-72\cos(4\omega T)} \qquad \text{(C-6)}$$

Figure (C-5) displays $\Phi(\omega)$ as specified by (C-6) with T set (arbitrarily) to unity. As Sittler points out, $\Phi(Z)$ also obeys the Wiener-Lee relation

$$\Phi_y(Z) = H(Z)H(Z^{-1})\Phi_x(Z) \qquad \text{(C-7)}$$

where $H(Z)H(Z^{-1})$ is the square of the transfer function of the sampled data filter. If we were to choose a rectangle of unit height and duration T as our impulse response, then we will multiply (C-6) by the 'traditional' since-squared envelope.

3. CONCLUSION

We have chosen a very elementary example and achieved a dramatic change in the shape of the power spectral density. There is great potential for further

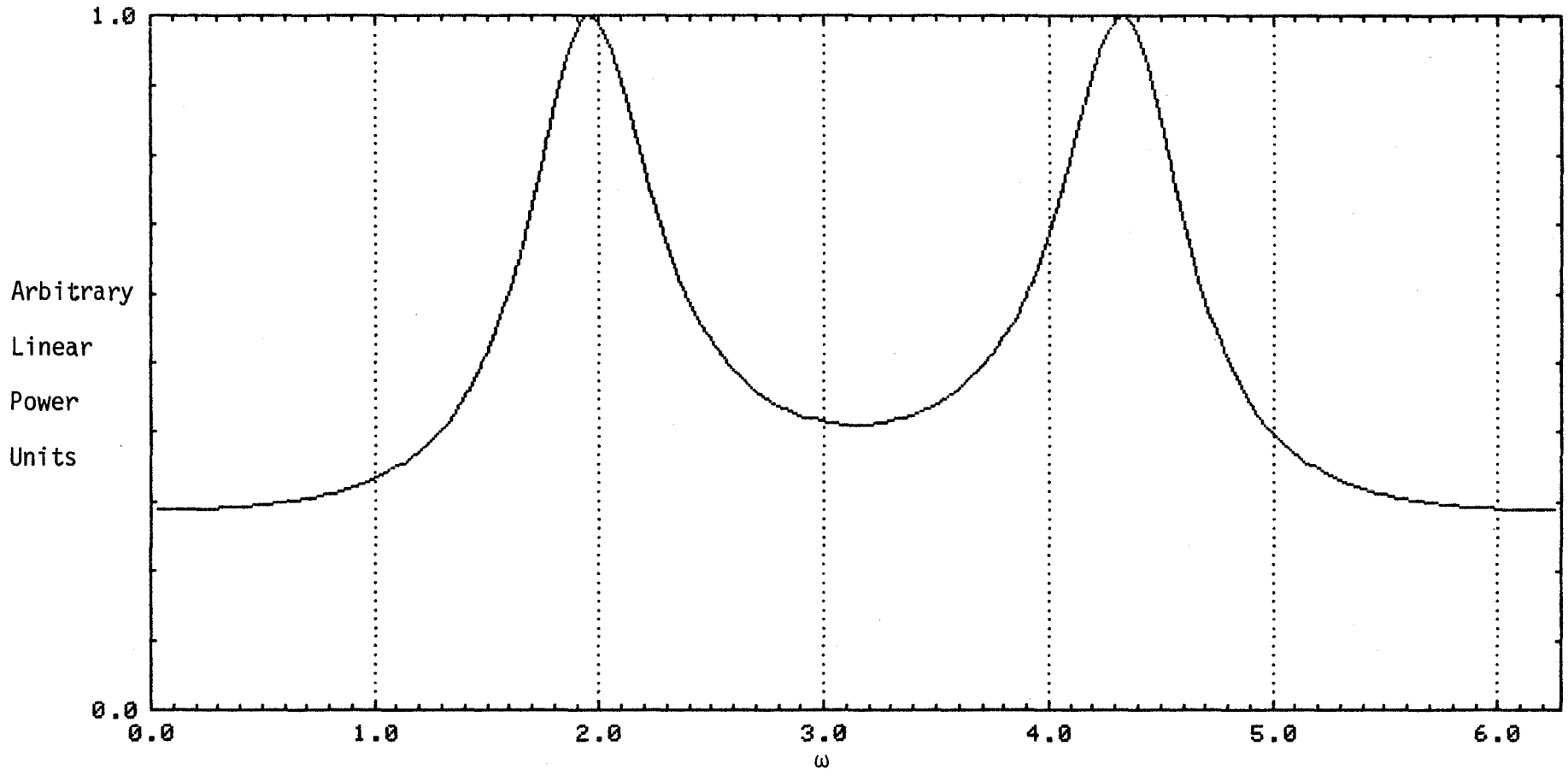


Figure C-5. Power spectral density.

innovation with this technique. One need not excise states but rather merely reduce and leave nonzero the transition probabilities to selected states. There should be sufficient latitude to markedly vary the shape of the spectrum. Markov filtering will, of course, affect the auto- and crosscorrelation of the CODE family members and this and a host of other questions need a lot of further study.

4. REFERENCES: APPENDIX C

Glance, B. (1971), Power spectra of multilevel digital phase-modulated signals, BSTJ 50, No. 9, pp. 2857-2878.

Holmes, J. (1982), Coherent Spread Spectrum Systems (John Wiley & Sons).

Mavraganis, P. (1979), Techniques and Benefits of Shaping the Pulses of Binary Sequences with Application to Spread Radio Communications, Engineer's thesis: Naval Postgraduate School, Monterey, CA.

Sittler, R. (1956), Systems analysis of discrete Markov processes, IRE Trans. on Circuit Theory, pp. 257-266.

APPENDIX D: PROPOSED CHANGES TO THE FCC'S RULES AND REGULATIONS

Part 2 of the FCC's Rules and Regulations govern frequency allocations and radio treaty matters. Section 2.106 is the Table of Freqency Allocations.

presently (as of the July 1981 edition), the first US Footnote reads:

In the bands 26.96-27.23, 50-54, and 144-148 MHz pulsed emissions are prohibited.

delete this footnote and all references to it in the Table of Frequency Allocations.

Part 97 of the FCC's Rules and Regulations govern the ARS. Parts proposed for change and the proposed changes are as follows:

1. Section 97.3, Definitions.

add new paragraph:

(aa)* Spread spectrum techniques. Any of a number of modulation schemes in which, (1) the transmitted radio frequency bandwidth is much greater than the bandwidth or rate of the information being sent and, (2) some function other than the information being sent is employed to determine the resulting modulated radio frequency bandwidth.

2. Section 97.7, Privileges of operator licenses.

presently paragraph (a) begins as follows:

(a) Amateur Extra Class and Advanced Class. All authorized amateur privileges including exclusive frequency operating authority in accordance with the following table:

change to:

(a) Amateur Extra and Advanced Class. All authorized amateur privileges including exclusive use of spread spectrum techniques and exclusive frequency operating authority in accordance with the following table:

3. Section 97.7, Privileges of operator licenses.

presently paragraph (d) is as follows:

(d) Technician Class. All authorized amateur privileges on the frequencies 50.0 MHz and above. Technician Class licenses also convey the full privileges of Novice Class licenses.

*May have to be paragraph (bb) as (aa) used for Amateur Code Credit Certificate in Part 97 of the Commission's Rules current as of 10/1/81.

change to:

(d) Technician Class. All authorized amateur privileges, except spread spectrum techniques, on the frequencies 50.0 MHz and above. Technician Class licenses also convey the full privileges of Novice Class licenses.

4. Section 97.61, Authorized frequencies and emissions.

add footnote number 1 to the 50-54 MHz, 144-148 MHz, and 220-225 MHz frequency bands as follows:

(a) The following frequency bands and associated emissions are available to amateur radio stations for amateur radio operation, other than repeater operation and auxiliary operation, subject to the limitations of section 97.65 and paragraph (b) of this section:

Frequency band	Emissions	Limitations (See paragraph (b))
*	* * *	*
50.0-54.0[1]	A1	
50.1-54.0	A2, A3, A4, A5, F1, F2, F3, F5	
51.0-54.0	A0	
144-148[1]	A1	
144.1-148.0	A0, A2, A3, A4, A5, F0, F1, F2, F3, F5	
220-225[1]	A0, A1, A2, A3, A4, A5, F0, F1, F2, F3, F4, F5	
*	* * *	*

[1]Spread spectrum techniques for domestic communications only are authorized in this band.

5. Section 97.73, Purity of emissions.

redesignate paragraph (d) as paragraph (e)

presently paragraph (c) is as follows:

(c) Paragraphs (a) and (b) of this section notwithstanding, all spurious emissions or radiation from an amateur transmitter, transceiver, or external radio frequency power amplifier shall be reduced or eliminated in accordance with good engineering practice.

revise paragraph (c) and redesignate it as paragraph (d) to read as follows:
(d) Paragraphs (a), (b), and (c) of this section notwithstanding, all spurious emissions or radiation from an amateur transmitter, transceiver, or external radio frequency power amplifier shall be reduced or eliminated in accordance with good engineering practice.

add a new paragraph (c) to read as follows:
(c) The limitations specified in paragraph (b) of this section shall also apply to spread spectrum modulated signals except for this purpose, "carrier frequency" is defined as the center frequency of the transmitted signal, and "mean power of the fundamental" is defined as the total emitted power.

6. Section 97.84, Station identification.
add a new paragraph (h) reading as follows:
(h) When an amateur radio station is modulated using spread spectrum techniques, identification in telegraphy shall be given on the center frequency of the transmission. Additionally, this identification shall include a statement indicating that the station is transmitting a spread spectrum signal and the upper and lower frequency limits of that signal.

7. Section 97.103, Station log requirements.
presently paragraph (g) reads as follows:
(g) Notwithstanding the provisions of section 97.105, the log entries required by paragraphs (c), (d), and (f) of this section shall be retained in the station log as long as the information contained in those entries is accurate.

change paragraph (g) and redesignate it as paragraph (h) reading as follows:
Notwithstanding the provisions of section 97.105, the log entries required by paragraphs (c), (d), (e), (f), and (g) of this section shall be retained in the station log as long as the information contained in those entries is accurate.

add a new paragraph (g) as follows:
(g) In addition to the other information required by this section, the log of a station modulated with spread spectrum techniques shall contain

information sufficiently detailed for another party to demodulate the signal. This information shall include at least the following:

(1) A technical description of the transmitted signal. If the signal is modeled after a published article, a copy of the article will be adequate.
(2) The dates that the signal format is changed. Changing the center frequency of the signal does not constitute a change in signal format.
(3) The chip rate (rate of frequency change), if applicable.
(4) The code rate if applicable.
(5) The method of achieving synchronization.
(6) The center frequency and the frequency band over which the signal is spread.

8. Section 97.117, Codes and ciphers prohibited.
presently, section reads as follows:
The transmission by radio of messages in codes or ciphers in domestic and international communications to or between amateur stations is prohibited. All communications regardless of type of emission employed shall be in plain language except that generally recognized abbreviations established by regulation or custom and usage are permissible as are any other abbreviations or signals where the intent is not to obscure the meaning but only to facilitate communications.

replace entire section with the following:
(a) The transmission by radio of messages in codes or ciphers in domestic and international communications to or between amateur stations is prohibited. All communications regardless of type of emission employed shall be in plain language except that generally recognized abbreviations established by regulation or custom and usage are permissible as are any other abbreviations or signals where the intent is not to obscure the meaning but only to facilitate communications.
(b) Spread spectrum transmissions between amateur stations of different countries are prohibited. However, for the purpose of the spread spectrum transmissions authorized between domestic stations in Sections 97.7 and 97.61, pseudorandom sequences may be used to generate the transmitted signal provided the following conditions are met:

(1) The sequence must be the output of a binary linear feedback shift register.

(2) Only the following shift register connections may be used:

Number of Stages in Shift Register	Taps Used in Feedback
7	[7,1]
13	[13,4,3,1]
19	[19,5,2,1]

(The numbers in brackets indicate which binary stages are combined with modulo-2 addition to form the input to the shift register in stage 1. The output is taken from the highest numbered stage.)

(3) For direct sequence modulation the successive bits of the highest stage of the shift register must be used directly to modulate the signal. No alteration or other data may be used for the direct sequence modulation. For frequency hop modulation, successive regular segments of the shift register sequence must be used to specify the next frequency, and no alteration or other data may be used for frequency selection.

(4) The shift register(s) may not be reset other than by its feedback during an individual transmission.

9. Section 97.131, Restricted operation.

redesignate paragraph (b) as paragraph (c)

add a new paragraph (b) as follows:

(b) If the operation of an amateur station using spread spectrum techniques causes interference to other licensed stations, the Commission's local Engineer in Charge may impose conditions necessary to resolve the interference, including termination of operation, on the offending station.

FORM **NTIA-29**
(4-80)

U.S. DEPARTMENT OF COMMERCE
NAT'L. TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION NO.	2. Gov't Accession No.	3. Recipient's Accession No.
	NTIA Report 82-111		

4. TITLE AND SUBTITLE
PROPOSED DIRECT SEQUENCE SPREAD SPECTRUM VOICE TECHNIQUES FOR THE AMATEUR RADIO SERVICE

5. Publication Date
November 1982

6. Performing Organization Code
NTIA/ITS

7. AUTHOR(S)
J. E. Hershey

8. PERFORMING ORGANIZATION NAME AND ADDRESS
U. S. Department of Commerce
National Telecomm. and Information Administration
Institute for Telecommunication Sciences
325 Broadway, Boulder, CO 80303

9. Project/Task/Work Unit No.
910 4142

10. Contract/Grant No.

11. Sponsoring Organization Name and Address
U.S. Department of Commerce
National Telecomm. and Information Administration
Institute for Telecommunication Sciences
325 Broadway, Boulder, CO 80303

12. Type of Report and Period Covered
NTIA Report

13.

14. SUPPLEMENTARY NOTES

15. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

General Docket 81-414, Notice of Inquiry and Proposed Rulemaking, proposes allowing the Amateur Radio Service to use spread spectrum techniques in three bands. This report reviews the Docket's proposals and the public's reaction, reviews direct sequence spread spectrum techniques, and proposes (for purposes of further discussion) a direct sequence spread spectrum system suitable for voice communications.

16. Key Words *(Alphabetical order, separated by semicolons)*
Amateur Radio Service; direct sequence spread spectrum; General Docket 81-414; spread spectrum

17. AVAILABILITY STATEMENT

☒ UNLIMITED

☐ FOR OFFICIAL DISTRIBUTION.

18. Security Class. *(This report)*
Unclassified

19. Security Class. *(This page)*
Unclassified

20. Number of pages
151

21. Price:

☆U.S. GOVERNMENT PRINTING OFFICE:1983—677-278 / 76