1968-06

The Design of a Morse-To-Teletype Signal Converter Using Intergrated Micrologic Circuitry.

Powers, Byron Lippincott

Monterey, California. Naval Postgraduate School

UNITED STATES
NAVAL POSTGRADUATE SCHOOL

THESIS

THE DESIGN OF A

MORSE-TO-TELETYPE SIGNAL CONVERTER

USING INTEGRATED MICROLOGIC CIRCUITRY

by

Byron Lippincott Powers, Jr.

and

Fred Robert Scalf, Jr.

June 1968

[redacted]

THE DESIGN OF A

MORSE-TO-TELETYPE SIGNAL CONVERTER

USING INTEGRATED MICROLOGIC CIRCUITRY

by

Byron Lippincott Powers, Jr.
Lieutenant, United States Navy
B.S., Naval Academy, 1960

and

Fred Robert Scalf, Jr.
Lieutenant, United States Navy
B.S., Naval Academy, 1960

[redacted]

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1968

Signature of Authors Byron L. Powers Jr.

Fred R. Scalf Jr

Approved by James S. Demetry
Thesis Advisor

Charles H. Rothauge
Chairman, Department of Electrical Engineering

W. F. Koehler for R. F. Rinehart
Academic Dean

ABSTRACT

This paper investigates the use of a Morse to teletype signal converter with existing teletype equipment to continuously monitor and display received Morse signals. This device would greatly reduce operator fatigue and provide increased efficiency.

Important aspects of the converter design problem are presented and different approaches to the problems encountered in this design are developed. A micrologic digital design is presented and its operation discussed. Although not fully implemented, it is considered far superior to other methods of implementation. It will accept Morse keyed audio signals in the 300-3000 Hz range at keying speeds of 10 to 100 words per minute and convert them to teletype code.

CHAPTER I

INTRODUCTION

The existence of two-way radio teletype communications capabilities in large aircraft, combatant ships, and shore stations presents an excellent opportunity for the addition of equipment for the automatic reception of Morse code signals. In patrol aircraft, for example, under conditions not permitting radio teletype operations, a single operator may spend as much as 15 hours monitoring Morse circuits without relief. In naval vessels a requirement exists for monitoring several circuits, a formidable task, especially under "Port and Starboard" watch conditions.

In both cases, the required monitor time is far in excess of the time of maximum operator effectiveness. In order to reduce operator fatigue and provide increased efficiency, a need exists for a Morse-to-teletype signal converter which would continuously monitor and display received Morse code signals.

The need for such a converter is not, however, limited to the military. The Federal Aviation Administration, International Press Service, and amateur radio operators all have extensive Morse code networks.

The idea of a Morse-to-teletype signal converter is not a new one, although only limited amounts of information are available on such devices. The first notable effort in the design of such a converter was undertaken by Lincoln Laboratories at the Massachusetts Institute of Technology. The converter which resulted from this project was actually a transistorized special purpose computer, called MAUDE (Morse AUtomatic DEcoder), and was quite large. The decoding decision functions of MAUDE took place at a number of different levels based on its "knowledge", not only of the relative durations of dits and dahs, but also of the Morse code and even of certain elementary properties of language. [1] MAUDE was quite successful, even under noisy circuit conditions, when used with a computer program to detect and correct output errors. [2] MAUDE did, however, have several disadvantages other than size and complexity. The computer program designed for MAUDE required that the processed information contain only alphabetic characters and spaces; hyphens, apostrophes, and other punctuation and numerals had to be spelled out. Also, MAUDE would not process the "combination" characters so often used in Morse operations. A few of these characters are $\overline{AS}$, $\overline{AR}$, $\overline{SK}$, and $\overline{BT}$.

Little further mention is made of Morse-to-teletype converters for reception purposes in the literature. However, several companies that make such devices are listed in the 1968 Edition of <u>Electronic Buyer's Guide</u>.

CHAPTER II

INTERNATIONAL MORSE CODE CHARACTERISTICS

The International Morse Code is specified by from one to six combinations of short and long pulses. The longer pulse, called a "dah", is normally three times the duration of the "dit", the shorter pulse. To send a message in this code three space lengths are used to separate these combinations of pulses. A short space of the same duration as a "dit" separates the individual pulses within one character code. When the character is completed, a character space, of the duration equal to a "dah", is used to distinguish each character. The combinations of characters which formulate a word are followed by a word space that has a width of from five to seven times that of a "dit."

A "dit" or a short space is normally considered the reference unit. Similarly, a "dah" or character space is nominally three units and a word space of five to seven unit duration. From the table of Morse Code given in Figure 1, the average Morse character has a duration of 8.5 units. In order to calculate the actual time duration of a unit, the assumption is made that the average word transmitted will have five characters.

A sample calculation of the length of a "dit" at ten words per minute is: 10 words/minute x 1 minute/60 seconds x 5 characters/word x 8.5 units/character = 7.08 units/second.

Dit length = $^{1}/_{7.08}$ units/second = 141.2 ms.

Character	Morse	Character	Morse
A	.-	1	.----
B	-...	2	..---
C	-.-.	3	...--
D	-..	4	....-
E	.	5	.....
F	..-.	6	-....
G	--.	7	--...
H	....	8	---..
I	..	9	----.
J	.---	Ø	-----
K	-.-	.	.-.-.-
L	.-..	,	--..--
M	--	/	-..-.
N	-.	Paren	-.--.-
O	---	:	---...
P	.--.	;	-.-.-.
Q	--.-	-	-....-
R	.-.	"	.-..-.
S	...	'	.----.
T	-	Error	....
U	..-	?	..--..
V	...-	AS	.-...
W	.--	AR	.-.-.
X	-..-	BT	-...-
Y	-.--		
Z	--..		

FIGURE 1

THE INTERNATIONAL MORSE CODE

Representative figures of the dit and dah lengths at 10 and 100 words per minute are given to show the radical changes necessary in an automatic decoder at these rates.

	10 w.p.m.	100 w.p.m.
Dit length	141.2 ms.	14.12 ms.
Dah length	423.6 ms.	42.36 ms.

The transmission of Morse code is accomplished by keying an oscillator tuned to the desired transmission frequency or a subharmonic of this frequency. The transmitted pulses oscillate at this frequency for the duration of the closed key. The receiver translates this high frequency oscillation using a beat frequency oscillator or product detector down to the audio range of 300 - 3000 Hz for the operator's reception at his desired tone frequency. The incoming signal to the signal converter will be this pulse modulated audio signal as shown in Figure 2 at 10 words per minute.

There are several characters or combinations of characters which have special meaning in code transmission. These abbreviations and their meaning are given in the following table:

Symbol	Code	Meaning
K	-•-	Go ahead, over
$\overline{SK}$	•••-•-	End of message
$\overline{BT}$	-•••-	Break
$\overline{AS}$	•-•••	Wait
$\overline{AR}$	•-•-•	End of transmission

Symbol	Code	Meaning
EEEEEEEE	....	Error
R	.-.	Roger

As can be observed from the above codes, these combination characters are sent without a character space between each character and are therefore treated as one character. There are many other abbreviations used in code transmission, but these are not sent as one character and do not need special treatment in a decoding process.

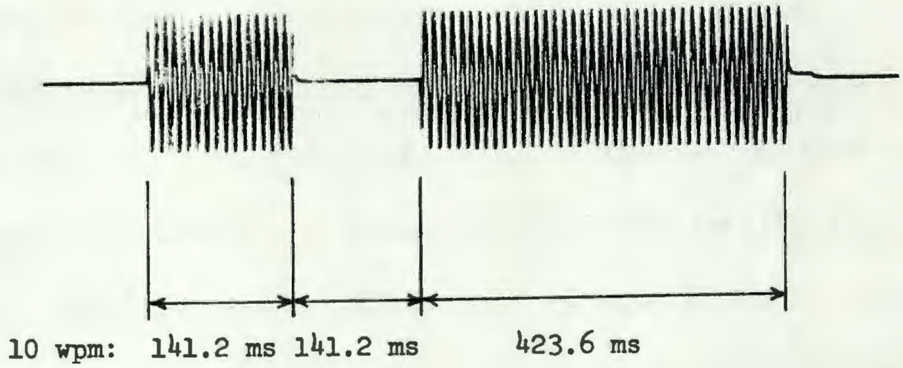


FIGURE 2

TYPICAL IDEAL CONVERTER INPUT SIGNAL CHARACTERISTICS

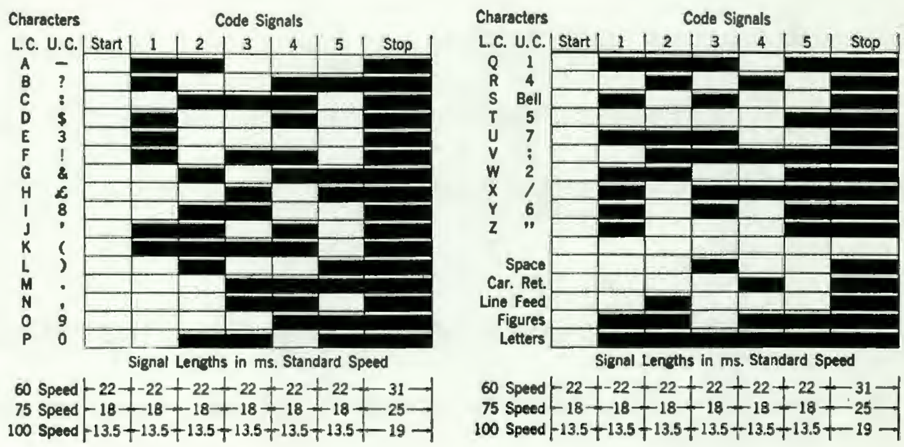


FIGURE 3

TELETYPEWRITER CODE

CHAPTER III

TELETYPE CODE CHARACTERISTICS

The 5-element binary code, sometimes called a 5-bit, 5-level, or 5-unit teletype code, is a binary permutation code in that we construct a table of five-digit binary numbers counting from 0 to 31 (binary 00000 to 11111), we find that there are 32 binary numbers. Further examination shows that all possible combinations of zeros and ones in a five-digit binary number are included.

These 32 permutations are sufficient to provide for the 26 letters, A through Z, of the alphabet; but not for the ten numerals, 0 through 9, as well; nor for a sufficient number of other symbols. The manner in which these additional characters are accommodated in teletypewriter codes is to use the case shift principle, whereby the different bit patterns each represent two characters, for example, a letter of the alphabet taken from the "letters" group or "case", and a numeral taken from the "figures case." With this method, as is used in teletype operation, it is necessary to indicate, by a "case shift" operation, whether a character in the letters case or one in the figures case is to be understood.

It is therefore necessary to use one of the bit patterns to command the printer to "shift" from one case to another, and a second bit pattern to convert to the former case. The control of case shift in the teletypewriter is thus similar in many ways to that of a typewriter in selecting upper or lower case letters or characters, but it reduces the total number of bit patterns available for the assignment of letters, numerals, symbols, and control functions from 32 to 30 in each of the two cases. This figure may be further reduced if certain control-function characters, such as LINE FEED and CARRIAGE RETURN also appear in both cases.

Since the coded character consists of a bit pattern containing several signal elements, it is necessary, when transmitting a number of characters in succession, to "frame" the bit patterns in some way in order to separate them so that they may be properly read at the receiver. The addition of two signal elements, a "start" element and a "stop" element, to the information or "code" elements of each coded character makes it possible to send characters in sequential mode at regular or irregular intervals on a "start-stop" basis. It is this terminology from which the name "5-Unit Start-Stop Teletypewriter Code" came.

The start element and the code elements generated are nominally of the same duration. The stop element has a nominal minimum duration, not necessarily the same as that of the start and code elements, which may be prolonged indefinitely when there is a quiescent interval between transmitted characters. The stop element is often greater than, but never less than, the code elements.

Reference to Figure 3 reveals that in the 5-element teletypewriter code the letters case is assigned entirely to the 26 letters of the alphabet, and a few machine functions. A message consisting of only letters, without punctuation signs or numerals, could, therefore, be transmitted without recourse to case shift action. Every time a single punctuation sign, numeral, or other symbol is introduced in the message, however, the "figures" shift character must be transmitted before the symbol and the "letters" shift character after it.

The actual durations of the elements for 60, 75, and 100 speed teletype transmission are shown in Figure 3. Note that the stop element is 1.42 units in length (the code element length considered as 1 unit), hence the code is known as a 7.42 unit Baudot Code. A rate of transmission of one pulse per second is defined as one "baud", therefore, for each of the three speeds, the rates are 45.45, 56.88, and 74.20 bauds.

CHAPTER IV

DESIGN CONSIDERATIONS

A. General

The design of a Morse-to-teletype signal converter must include careful consideration of several unique problems as well as some very common ones. A converter that can be used in airborne, shipboard, and shore configurations must be capable of operating from several voltage sources or must have available a variety of detachable power supplies operating from various voltages and/or frequencies. It should be easy to adjust, maintain and operate. The use of a very limited number of simple controls is the solution to adjustment and operation. Maintenance efforts could be reduced considerably by the use of plug-in circuit boards which could be removed from the unit and tested on simple test benches. The complete unit could be made easier to test by using digital circuitry with a central master clock from which all other clock frequencies are derived. In the test mode it would be possible to substitute a much slower clock for the master clock and insert a predetermined Morse test code at the input to test all converter functions in "slow-motion". This type of testing could be performed with a minimum amount of field test equipment.

If the converter is to be used in airborne installations, it must be small in size and light in weight. These requirements, when considered with the desirability of using digital circuitry, lead directly to the use of integrated circuitry in the design of the converter.

To be used with a variety of models of existing teletype equipment, it should have a variety of output options. For example, to be used with the older Model 15 and Model 19 teletype equipments, as well as the newer Navy TT-264 airborne teletypewriter and Marine AN/TGC-14, it should have a erial teletype output. On the other hand, operation with the more recent Model 28 equipments could be either in a serial or parallel output mode. Operation in the parallel output mode with Model 28 equipments would require the use of a parallel to serial converter. All units mentioned use the 5-unit start-stop code.

B. The Received Signal

The received Morse characters enter the receiver as a pulse modulated sinusoid, previously illustrated in Figure 2, plus noise. The noise usually consists of broad-band "white" noise with occasional noise bursts of considerably greater amplitude than the average r.m.s. noise level. Depending upon the quality of the receiver used and the band-width selected, some increase in the effective signal-to-noise ratio may be possible. The signal then appears at the audio output of the receiver with similar characteristics, but translated from radio frequencies to audio frequencies. These audio frequencies are band limited in a range specified by the receiver audio characteristics. Typical values of the upper and lower audio frequency limits are 300 to 3000 Hz and 300 to 2500 Hz.

This is the signal that must be decoded; that is, the pulse modulated signal must be detected while preserving the pulse lengths and minimizing signal distortion caused by noise. The problem is compounded by the fact that most Morse signaling employs on-off keying vice frequency shift keying as is prevalent in teletype communications. The effect of using on-off keying is to prevent the use of standard teletype converter input circuitry, e.g., high gain amplifiers followed by limiters, since this type of circuitry would cause severe noise distortion when the signal was not present due to the large amount of amplification. This is not as much of a problem in teletype because the receiver normally remains "quieted" since one or the other of the Mark or Space signals is always on.

This problem could be partially eliminated by the use of band pass and channel filters, but this technique would introduce other problems. Using narrow filters to increase the effective signal-to-noise ratio would require greatly increased receiver stability and much more accurate receiver tuning since the converter input would respond only to audio frequencies within the bandwidth of the filter.

C. Hand vs. Machine Keyed Code

The Morse character dit-dah-space relationships presented in Chapter II are not always realized in actual operations. Perfect character transmission is usually realizable only when the code characters have been generated by an automatic Morse machine, an example of which is the familiar Boehme Keyer, or by electronic keyers. These electronic keyers can be either of two types. The most common is capable of generating sequences of dits or dahs, depending upon which is selected by a paddle key. This type is capable of near perfect code when properly adjusted and used by a proficient operator. The other type of electronic keyer is still quite rare. It electronically generates the proper code character in response to keyboard character selection.

The major portion of transmitted code is hand sent on conventional hand keys because of their simplicity and low cost. A variation of the hand key called the semiautomatic key or "bug" is also in common use. A bug is more difficult to control and like the common type of electronic key is generally used by more experienced operators. The dahs are produced manually from one "on" position on the key, while the other "on" position produces a machine-like sequence of dits. Because of inherent imperfections in both types of hand keys and in the operators using them, the dit-dah-space relationships are not ideal. Commonly encountered distortions are decreases or increases in the dit-dah length ratios and sending speed variations. Another variation of significance is the use of different numbers of dits to specify the "error" (........) character. Although this character is specified to contain 8 dits, operators are commonly found to send 6 to 8 dits to specify it. Recognition of "error" must be made when any of these combinations is transmitted. One approach would be to consider any character with 6 dits or more as the error character.

The dit-dah-space variations present probably the most formidable deterrent to accurate recovery of the code characters from the received signal. Probably the most feasible solution would be to establish an arbitrary ratio criterion, such as 'dahs must be at least twice as long as dits'. Since the ideal ratio is 3 to 1, a 2 to 1 ratio would provide considerable leeway for individual sending characteristics.

Since speed changes, either random or slow gradual changes, are to be expected in hand-keyed code, the converter should be able to automatically adapt to these changes in order to properly analyze the incoming signal and convert it to the proper teletype character. Sending speeds could vary over a very wide range and, although most circuits operate in the 15 to 25 words per minute range, some operators send more slowly and it is possible to operate the automatic Morse machines at quite high speeds. A speed range of 10 to 50 words per minute might be considered minimal, with a range of 10 to 100 words per minute more desirable.

CHAPTER V

INTEGRATED CIRCUIT SUMMARY

There are many families of integrated circuits available, the most popular at this time being Resistor-Transistor Logic (RTL), Diode-Transistor Logic (DTL), and Transistor-Transistor Logic (TTL). Availability and economy dictates the use of RTL integrated circuits in this design. The design can be easily converted, if desired, to another micrologic family.

The temperature range of fifteen to fifty-five degrees centrigrade, very low propagation delays, and low power consumption are characteristics of the economy lines of integrated circuits. Requiring only interconnection for operation, these circuits are a set of compatible integrated logic building blocks which are well suited for this design. The elements are manufactured by a planar epitaxial process in which all the necessary transistors and resistors are diffused into a single silicon wafer. The individual RTL circuits within the logic blocks are interconnected by metal over oxide.

There are two logic levels in all elements of this device, the positive level "1", and the negative level "0". The actual voltages of these levels depends on the manufacturer and family of micrologic circuits.

The basic elements used in the construction of the signal converter are Inverters, two input NAND/NOR gates, three input NAND/NOR gates, four input NAND/NOR gates, and JK flip-flops. In logical operation the gates may be considered Negative-NAND or Positive-NOR depending on the desired logic. In the signal converter Positive logic is consistently used.

The operation of these essential elements is briefly described in the following paragraphs.

1) Inverter/Driver

The Inverter is used to reverse the logic of the incoming signal. If the input is "1", the output of the Inverter is a "0", and vice-versa. An Inverter may also be used as a Driver to increase fan-out, the number of other elements which may be connected to the output. The Driver can also be non-inverting and in this case, it increases the fan-out but does not change the incoming logic state.

2) NOR Gate

The purpose of a NOR gate is to enable and/or add two or more inputs. The output of this gate is always "0" unless all inputs to the gate are "0" and only then is the output "1", as shown in Figure 4.

3) JK Flip-Flop

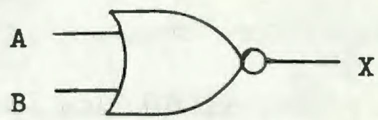
The primary applications of the JK Flip-Flop are in binary counters and shift registers. Whenever there is a transition from "1" to "0" (a falling transient), the state of the inputs is transferred to the outputs; otherwise the output states remain the same. A reset (sometimes called a preset) pulse always returns the outputs to the $\overset{0}{1}$ state. The flip-flop will not change from the $\overset{0}{1}$ state until the reset pulse is removed and another falling transient takes place. The shift register and normal JK Flip-Flop are designed in this manner and the truth table for this application is given in Figure 5.

When the JK Flip-Flop is used as a counter or it is desired that it shift states on all falling transients, the input leads A and B are set to "0" and the clock pulse input becomes the only input lead. The output then changes state on each falling transient.

Some versions of the JK Flip-Flop have both set and reset inputs. In this case, the set input is used to place the outputs in a $\genfrac{}{}{0pt}{}{1}{0}$ condition and reset as previously stated in the $\genfrac{}{}{0pt}{}{0}{1}$ state.

NOR Gate -

$X = \overline{A + B}$

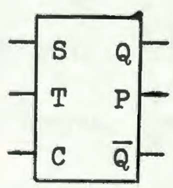


Truth Table

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

FIGURE 4

JK Flip Flop -



FUNCTIONS

SET	CLEAR t = n	OUTPUT t = n + 1
H	H	X
H	L	H
L	H	L
L	L	$\overline{X}$

H = High
L = Low
X is the output state at time n
A high on pin P will preset output to low

T is the clock or enable pulse.
It must be a high-to-low transition (falling transient).
The clock pulse is required for all functions except Preset.

FIGURE 5

CHAPTER VI

DESIGN DEVELOPMENT

A. General

After careful consideration of the factors discussed in Chapter IV, it was decided that a Morse-to-teletype converter design could be divided into separate sections, as shown in Figure 6, according to the particular functions required. Each section will first be discussed in general and then developed in more detail.

The purpose of the Input section is to detect the incoming pulse-modulated audio signal while discriminating against both "white" noise and random noise crashes. It must provide the Decision section with as accurate pulse length information as possible. This was accomplished by the formation of an output of d.c. pulses whose pulse durations were the same of those of the Morse characters.

The Decision section receives the noise-free pulse string and sorts the pulses according to their relative lengths. In addition, it must perform this same function with the incoming spaces since their lengths are also important in Morse signalling. There are two pulse lengths, the dit and dah, the three space lengths, the unit space, characters space, and word space, that must be obtained from this sorting process.

The Dit-Dah Storage section accepts these five outputs from the Decision section and uses the three space outputs to control the storage of the dits and dahs. Some type of register storage seemed most practical here. The only function of the unit space is to separate the

SECTION BLOCK DIAGRAM

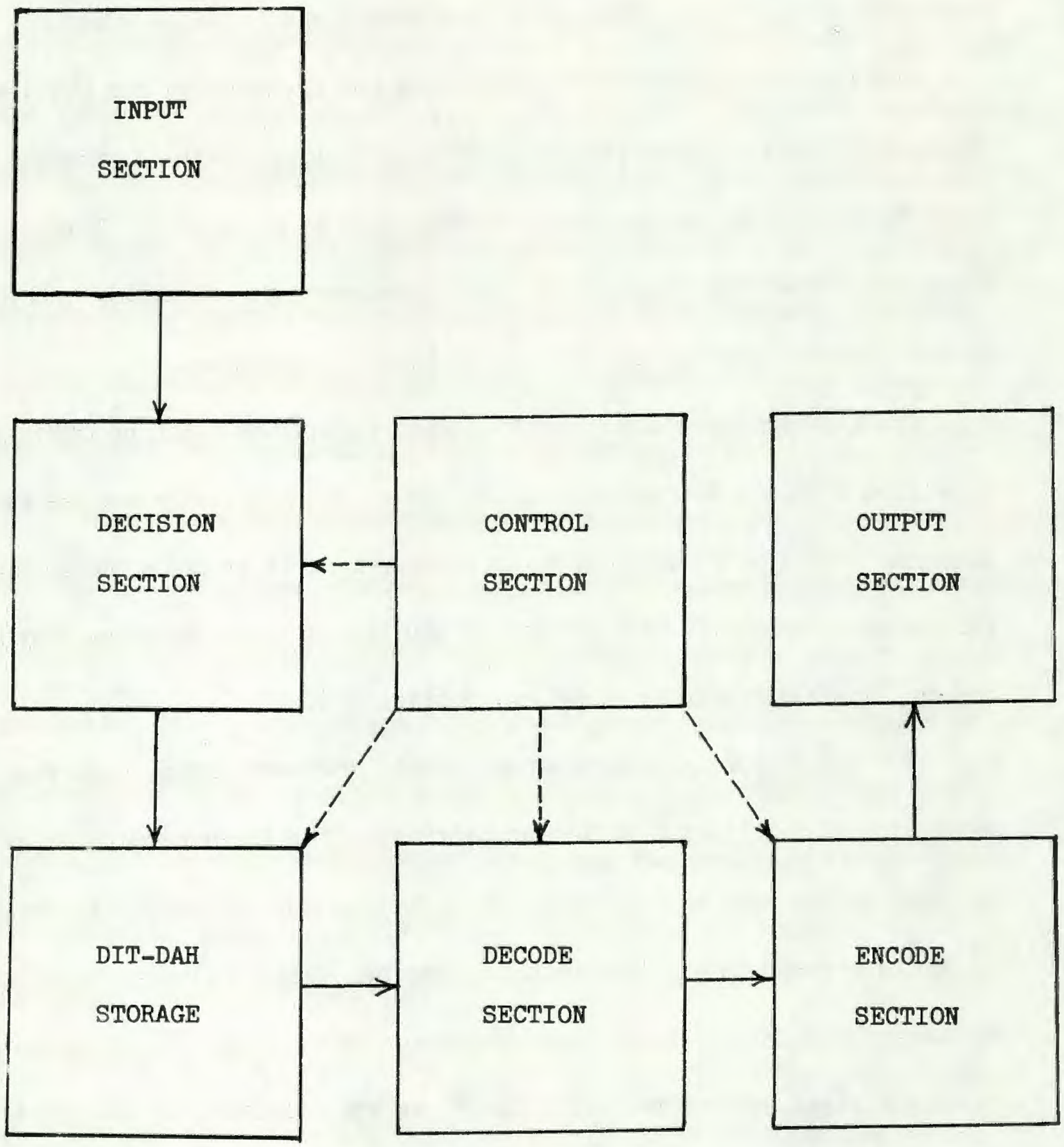


FIGURE 6

pulses. The character space and the word space are much more important, however, since they signify the end of a character or word and begin the decoding process. The word space must also generate a space in the teletype output after the stored character is printed.

The Decoding section must interpret the contents of the Dit-Dah Storage section and provide the appropriate signal to the Encoding section so that the proper teletype character is generated. It must be able to recognize all alpha-numeric characters, punctuation, and the special characters that are typical of Morse signalling.

The Encoding section generates the appropriate teletype character on signal from the Decoding section. It must have ready access to all teletype characters and must make them available as selected to the Output section which will provide proper timing and introduce carriage return, line feed, and appropriate carriage shifts.

This complete process must be carefully coordinated, with the exception of the Input and Output sections. The Input section must respond at the rate of the transmitting station and the output to the speed of the teletype printer. Because of possible speed differences in these peripheral functions, some type of storage or buffer is desirable within the converter. All intermediate functions are regulated by the Control register and must be performed at high speed to remain accurate pulse and space lengths.

B. The Input Section

As mentioned in Chapter IV. B., the use of amplifying-limiting stages as are common in teletype converter input circuitry is impractical under conditions of on-off keying vice frequency shift keying. If, however, a channel filter is placed prior to these stages in order to gain an effective increase in the signal-to-noise ratio, considerable improvement is realized. A circuit with a channel filter was quite successful in detecting and shaping the incoming signal, although it had several disadvantages. It provided insufficient discrimination against noise crashes and required a very selective and stable receiver. The circuit also required two additional supply voltages and interface circuitry for use with the integrated circuits that followed.

A digital envelope detection circuit was tried and, although more complex, was found to be much more effective. This input circuit is illustrated and described in Chapter VII. In this input circuit any pulse or noise crash of duration less than 5 milliseconds is not processed.

C. The Decision Section

The process of sorting the incoming pulse string into dits, dahs, unit spaces, character spaces, and word spaces presented many problems. The initial design used a ramp generator ("bootstrap" circuit) whose output voltage was a function both of the supply voltage and the duration of the incoming pulse. The supply voltage was varied to compensate for the Morse sending speed and ranged from -6 volts at

10 words per minute to -15 volts at 60 words per minute. By using a peak reading voltage indicator and a variation of the conventional series voltage regulator circuit, with sufficient control lag, to control this supply voltage, it is possible to regulate the maximum ramp voltage to the same level and, thereby, automatically compensate for speed changes.

The output of this ramp generator was fed to Schmidt trigger circuits with different triggering levels. It was found experimentally that triggering levels of 1.2 volts for dits and unit spaces, 2.4 for dahs and character spaces, and 4.0 volts for word spaces gave the arbitrary 2 to 1 dah-dit and 4 to 1 word space to unit space ratios desired. The 4.0 volt word space level gave better performance than a 4.8 volt level (4 times 1.2 volts) since the ramp was not perfectly linear at the higher voltages.

As was true of the input stage, this circuit required interfacing when used with integrated circuits and required additional supply voltages. The triggering levels of the Schmidt triggers were also quite critical due to loading problems and characteristics inherent in the design.

The all-integrated-circuit version of the decision section bears no resemblance to this linear design. It is discussed in Chapter VII.

D. Dit-Dah Storage Section

Once its component parts are known, there are two principal methods by which a character can be decoded. Although not true uses of the terms, they might be called "serial" and "parallel". The serial method requires no storage as such and involves the arrangement of the Morse characters in a progression of increasing numbers of dits and dahs with branches for variations. A "tree" is formed to represent the entire code, and a single character can make its way to the end of only one branch. This method was rejected on the grounds that the problems in gating, synchronizing, and in incompatibility with changes in code characters or with the number of characters used, would far outweigh any advantages it offered in theory.

The parallel method requires the storage of the dits and dahs in sequential order until a word space or character space is received, at which time the entire character is decoded. The longest Morse character, with the exception of the "error" character, is six dits and/or dahs in length. Since the "error" character is often transmitted with 6, 7, or 8 dits, the policy of permitting 6 or more dits to specify it will also permit storage of this character in a 6 bit register. To store a complete character and to avoid ambiguity, two six bit registers are required.

Once the character is stored in these two registers, referred to as the "dit register" and the "dah register", it may be decoded directly using gates or a decoding matrix. It may also be regenerated as a perfect Morse character, stored in a single register, and then decoded by either of the previously mentioned methods. Regeneration as a perfect code has the advantage of facilitating the use of a possible option of retransmission of the received character in perfect code; however, the addition of the extra circuitry required is not merited in a special purpose converter. It would, nevertheless, be appropriate at this point to clarify this single register perfect code theory. Each bit in the register would represent a unit length, that is, a dit or unit space would be represented by one bit, and a dah by three bits. The longest character (largest number of units) would determine the register size. An example of this character is zero (-----) and would appear in the register as 1110111011101110111, a total of nineteen bits.

E. Decoding Section

It is possible to translate directly from the Morse character code stored in the Dit and Dah registers directly to teletype code using AND/OR logic, but this would prevent the addition of the appropriate combination characters previously mentioned and would create some case determination problems. For these reasons, and for the purpose of reducing the amount of logic required for this character translation process, it was decided to decode (AND) and encode (OR) in separate steps. Also, the use of Veitch diagram techniques to minimize two-level (AND/OR) functions becomes an almost impossible task when applied to such large-scale problems.

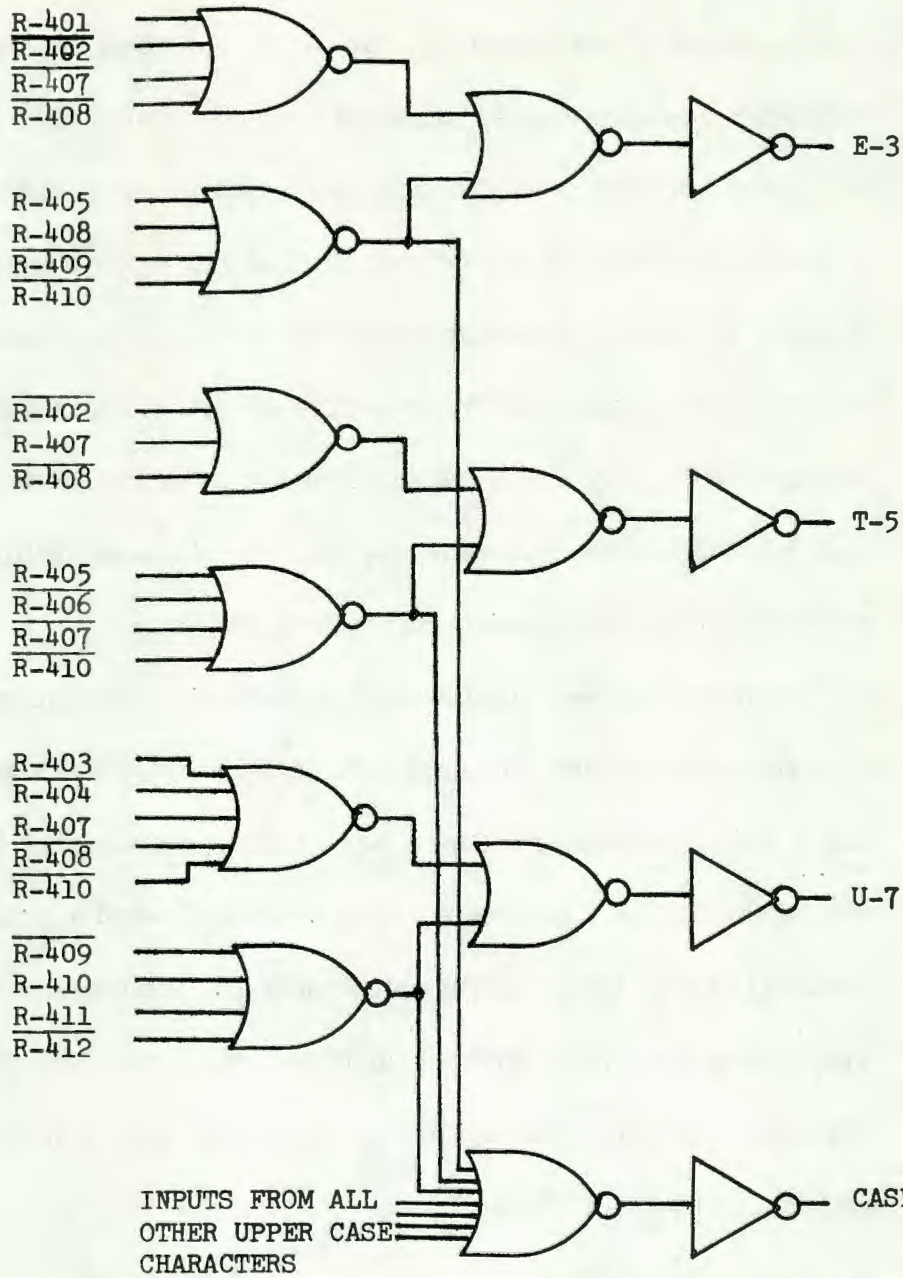
A technique was developed, however, to reduce the AND logic circuitry required for the decoding process. [3] The OR functions were then added, by inspection, in the encoding matrix discussed later in this section. A Fortran IV computer program was written for performing this reduction and is enclosed in Appendix II. The results of the minimization of the decoding logic AND functions required in this converter design are also included. A true minimum is not obtained since the solution depends upon the reduction sequence selected, but a savings of some 65 per cent in the number of gate inputs (number of diodes, if a diode decoding matrix is used) was realized with a random minimization sequence. Twenty-five other random sequences were selected with only a few per cent variation in the savings. A study is now in progress to determine the true significance of this reduction method and its relationship to the true minimum.

Using separate decoding and encoding steps, the output of the Decoding Section would consist of 51 terminals, each terminal representing a Morse character. Since many teletype character codes represent two of the Morse characters, it is possible to reduce the number of terminals to 27 by using OR gates on these related characters and specifying the case, whether upper or lower, of each character appearing at the output. One method of implementing the required decoding matrix is shown in Figure 7.

F. Encoding Section

The process of encoding the proper teletype character is easily implemented by using an encoding matrix. The matrix required is shown in Figure 8.

The outputs of the Decoding Section include not only simple alphanumeric characters, but also the case, upper or lower, of each



NOTE - ALL OTHER CHARACTERS ARE DECODED IN THIS SAME MANNER. THE MINIMIZED CODE USED TO DETERMINE GATE INPUT CONNECTIONS IS SHOWN IN APPENDIX II.

FIGURE 7

DECODE MATRIX CONFIGURATION

NOTE

NO CONNECTIONS ARE MADE TO THE "CASE" LINE IN THE ENCODING MATRIX SECTION. THE LINE PASSES FROM THE DECODING MATRIX THRU THE ENCODING MATRIX TO THE INPUTS OF R500C.

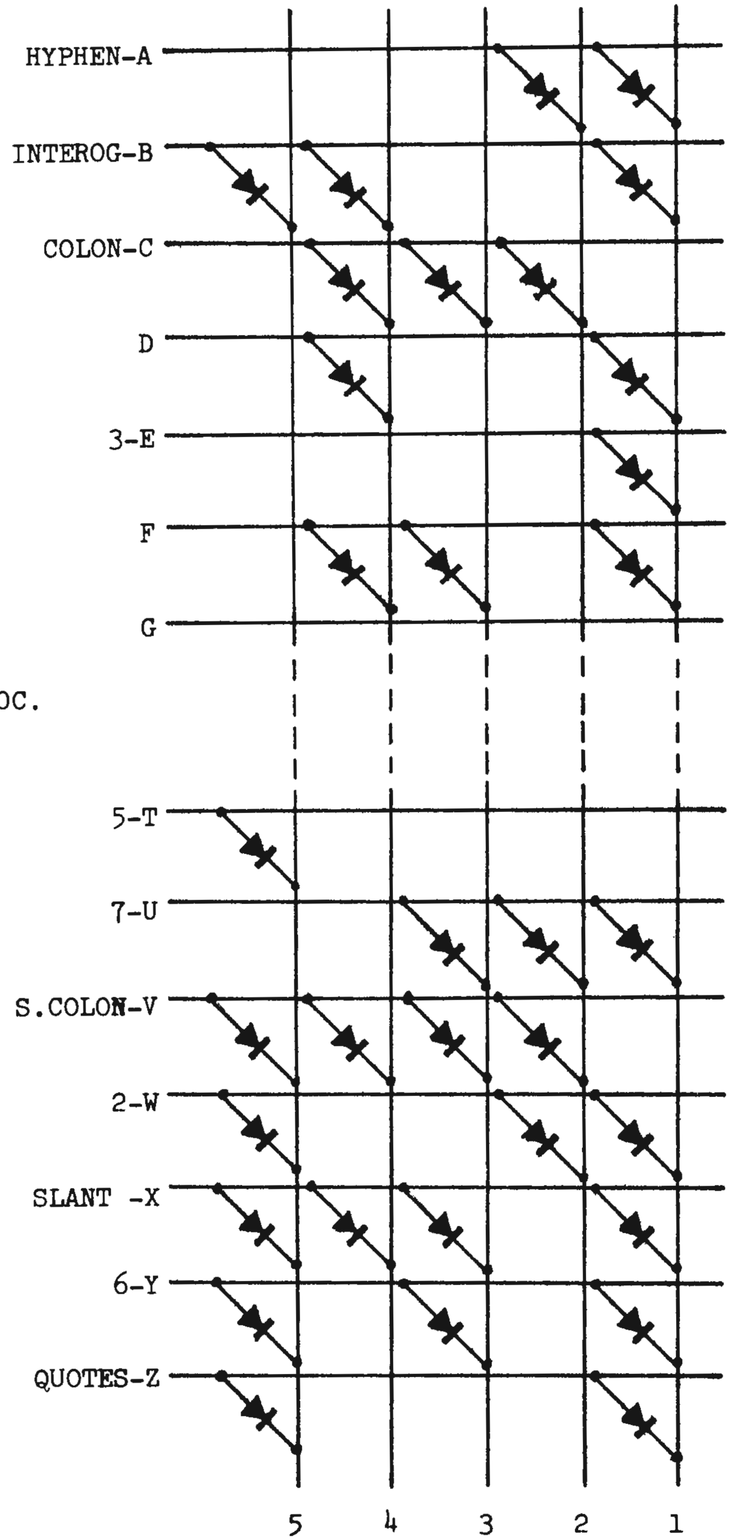


FIGURE 8

ENCODING MATRIX

character encoded and the previously mentioned "combination" characters. These characters require the generation of two teletype characters for a single Morse character received. An example would help clarify this problem.

The Morse code for the letter "A" is "·-", while that for "S" is "...". When these are transmitted as a single character, ".-...", the meaning is WAIT and is usually written $\overline{AS}$. When the following character space is reached, the Decoding Section will recognize this character and will energize the $\overline{AS}$ output to the encode section. Now the problem is that the Encoding Section must generate two complete characters, "A" and "S".

The Encoding Section should recognize that a combination character has been received and encode the first character, in this case "A". When this character has been received by the Output Section, the Encoding Section must generate the second character "S". The proper case designation must then accompany each character to the Output Section.

G. Output Section

This section must accept the encoded teletype characters, in parallel form, from the Encoding Section and control their distribution to the peripheral parallel-to-serial teletype code converter. While performing this task, however, provision must be made for the storage of characters received when the teletype print rate lags the Morse reception rate. Provision must also be made for the teletype case control and carriage return-line feed at the end of each printed line.

The output to the peripheral equipment consists of five parallel code lines and two signalling lines. The signalling lines are used to synchronize the output of the converter with the parallel-to-serial converter. One signal line indicated that a character is ready for printing and triggers the generation of the "Start" space. The other signal line receives an indication that the beginning of the "Stop" pulse has been reached, and caused the Output Section to make the next character available.

To provide storage for characters awaiting printing, a buffer section was provided which consists of a series of seven bit registers which parallel shift the characters from the Decoding Section to the teletype output code lines. Since the actual teletype character consists of only five information bits, the extra two bits are used to perform special functions required in character processing. One bit is useful at the output of the buffer for signalling that a character is ready for printing, and, within the buffer to signify the existence of a valid character in that buffer stage. This latter feature is very important since any character entered into the buffer will automatically check the next indicator bit and then shift to that stage if found empty. It will continue this process until a filled stage is found or the output is reached. As characters are printed and empty positions are made available, the characters move up to fill them.

The second extra bit is used to store the case information for the character contained in that position. At the output, the last printed character case is stored, the next character case sampled, and a decision made whether or not a case shift is needed. If required, the proper shift is encoded and read by the peripheral converter.

After each sixty-three characters are entered into the buffer, the carriage return-line feed control examines each of the next nine characters for a word space. If such a space is received, the teletype codes for carriage return and line feed are entered into the buffer. If none is received by character seventy-two, the entry is still made. In this manner, an effort is made to prevent division of a word at the end of a printed line.

CHAPTER VII

THE PROPOSED DESIGN

A. General

Reference should be made to the block diagram in Figure 9 for clarification of the converter components and sections described in this chapter. The detailed circuit diagram is given in Figure 12. Before proceeding with a more detailed description of the signal processing aspects of the converter design, it is worthwhile to discuss the general purpose of some of the major component sections not mentioned in previous chapters.

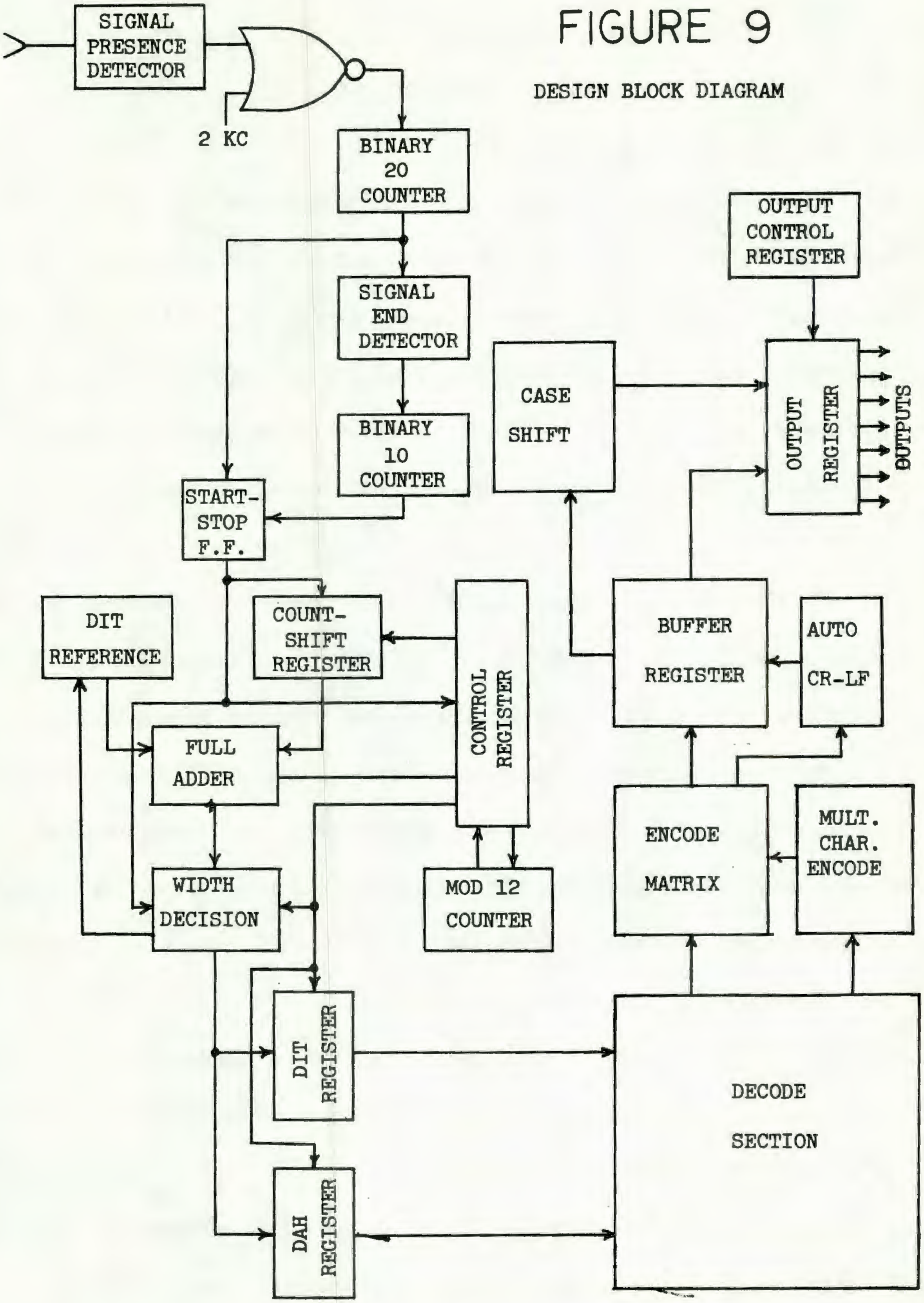
The Control Register provides master control for all functions between the Input Section and the Buffer input in the Output Section. These functions will be cumulatively referred to as the Processing Section of the converter. The Control Register is essentially a shift register, but is configured such that only a single "1" is contained in the register at any time. This "1" starts in the left-most bit ("ready" position) and shifts to the right as each step in the conversion process is completed.

The Reference Register is a uni-directional shift register and is used to store the reference count which is representative of a unit length at the code speed being processed.

The Count/Shift Register is a combination binary counter and shift register. It is used to count a reference clock when the converter is in the "ready" position and when gated by an incoming pulse or space.

FIGURE 9

DESIGN BLOCK DIAGRAM



The final count will be representative of the length of the subject incoming pulse (space) length and will be used for comparison with the count contained in the Reference Register.

There are actually two fixed clock rates in the converter, although one is derived from the other. The 1 mhz Master Clock is used for clocking all signal processing functions. A 2 KHz clock signal is derived from the Master Clock using a divide-by-500 decode divider. This clock signal is gated to the Count/Shift Register to provide the pulse (space) count described above. It is also gated to the Mod 10 counter in the input section. The purpose of this counter will be described later.

B. Description of Operation

1. Ready State

When power is first applied to the converter, it is necessary to press the Master Reset button which initializes all required functions. The Control Register is preset to the 10000000 or "ready state"; the Count/Shift Register, Dit-Dah Storage Registers, and all counters are "zeroed". The Reference Register is preset to the 000010111100 state (approximately 15 words per minute). The Paren Control Flip Flop is preset to LEFT PAREN position by depressing the Paren Reset button. The Master Reset button is also used to re-initialize these functions when a sudden Morse sending speed decrease of greater than 2 to 1 has taken place. The Output Section is not affected by the Master Reset in order that the information already processed and awaiting transfer to the teletype equipment will not be lost. Since the Output Section is not reset, a few random characters can be expected when the converter is first placed in operation. The Paren Reset Button can also be used to reset to LEFT PAREN when the printed parens do not correspond properly due to an error by the Morse operator. The 1 MHz and 2 KHz clock frequencies operate continuously when power is applied to the device. The Overflow Shut-Off flip flop holds the converter in the ready state awaiting the first pulse.

2. Dit Processing

Consider now that the first pulse modulated audio signal received is a dit. The Signal Presence Detector enables the input gate to the Binary 20 Counter allowing the 2 KHz to be counted for the duration of the incoming pulse plus 5 milliseconds. The extra 5 milliseconds is the result of the "one-shot" multivibrator action. The Binary 20 Counter counts 20 of the 2 KHz pulses (10 ms.) before setting the Start-Stop Flip Flops. This processing delay is provided to insure that an input pulse of at least 5 milliseconds duration is present, thereby eliminating noise crashes of shorter duration. The Start-Stop Flip Flop gate the 2 KHz clock signal to the Count/Shift Register and set the X_1 Flip Flop to "1" to indicate that a pulse rather than a space is being processed.

At the end of the incoming dit plus the five millisecond extension in the Signal Process Detector, the Binary 20 Counter is reset. This reset signal also starts the Binary 10 Counter counting at the 2 KHz rate. At the end of the 10 count (5 ms.), the Control Register is shifted to the 2nd position (01000000) and the count into the Count/Shift Register is stopped. The complete envelope detected signal has now been counted into the Count/Shift Register. In the process, however, the signal has been delayed 10 milliseconds. Refer to Figure 10 for a graphic representation of the time relationships involved.

At this time, all signal processing operations are performed at the 1 MHz clock speed to permit process completion before much of the succeeding space has been received.

Before the Control Register is shifted to the 3rd position (00100000), the least significant bit of the Count/Shift Register is checked and, if necessary, is set to zero. (The reason for this action is explained in Appendix I.) Position 3 of the Control Register enables the Full Adder to serially compare the count in the Count/Shift Register with that in the Reference Register. Since the count from the Count/Shift Register is taken from the second least significant bit, the comparison is actually to see if the incoming signal is twice as long as the reference signal, since this is the dit-dah decision criteria. Both registers also recirculate to preserve their individual counts for later use. The Mod 12 counter controls this operation, allowing only the 12 clock pulses necessary for comparison and recirculation. At the end of the 12th clock pulse, the Control Register is shifted to the 4th position (00010000). In this position the carry bit of the Full Adder is examined. If this bit is "1", the incoming signal is at least twice as long as the reference. Since an incoming dit has been assumed in this example, the carry bit should

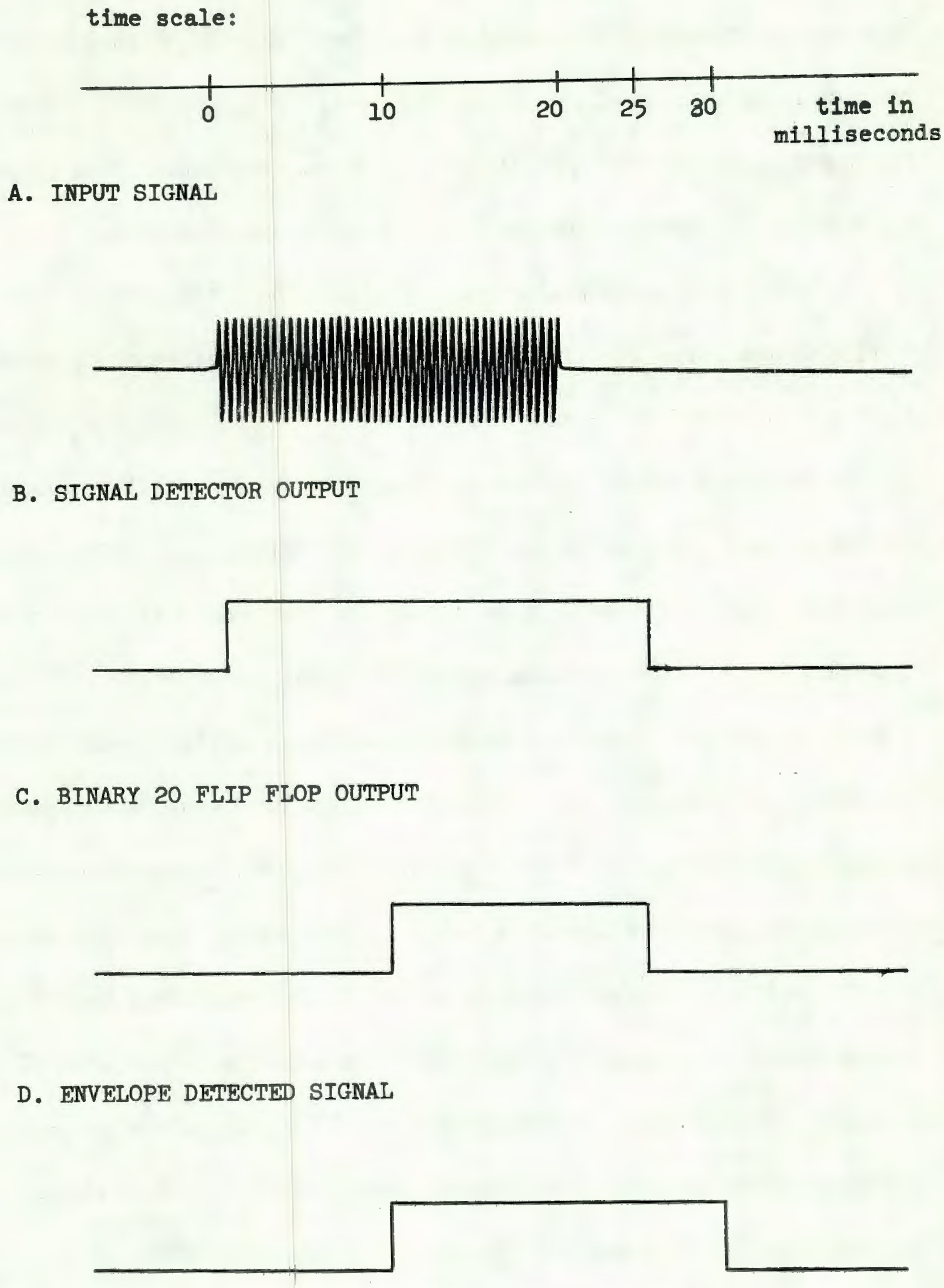


FIGURE 10

INPUT SIGNAL TIME RELATIONSHIPS

be "0". (As will be seen later, the Control Register positions 1 through 4 perform exactly the same functions regardless of the type of incoming signal.)

The Control Register 5th position (00001000), however, consists of three different branches dependent upon the type of incoming signal. The previously mentioned X_1 Flip Flop, which is "1" for a pulse and "0" for a space, and the carry bit of the Full Adder, provide the necessary data for the decision process. The following table presents a summary of the possible conditions in the decision process and their respective required actions.

INPUT	F.A. CARRY BIT	X_1	NEXT CONTROL REGISTER POSITION	ACTION REQUIRED
Dit	0	1	5X	Replace Reference with contents of Count/Shift register and put dit into Dit Register.
Unit Space	0	0	5X	Replace Reference with contents of Count/ Shift Register.
Dah	1	1	5Y	Put dah in Dah Register.
Character or word space	1	0	5Z	Zero 2nd bit of Count/ Shift Register, decode Dit-Dah Registers, re-examine the contents of the Count/Shift Register divided by 4.

Results at end of re-examination:

Character space	1	0	--	No action.
Word space	0	0	--	Encode word space and enter in output Buffer.

Considering still a dit input, the control register shifts from position 4 to position 5X (00001000) and transfers the latest count in the Count/Shift Register to the Reference Register, again using the Mod 12 counter for controlled operation. This process of replacing the dit reference with the latest dit count or, as will be seen later, with the unit space count performs the function of continuously updating the dit reference. This enables the device to follow speed changes with no loss of characters. At the end of the 12th bit, the control register shifts to position 6 (00000100) and places a "1" in the dit register and a "0" in the dah register.

The Count/Shift Register is reset to zero in position 7 (00000010) and the Control Register is reset to the ready state (10000000) at position 8. The Control Register in the ready position enables the Count/Shift Register to begin counting the space interval. This count will continue until an overflow condition (a full Count/Shift Register) is reached signifying a word space or a break in the incoming signal transmission, or until the next dit or dah is received signifying the end of the space interval.

3. Unit Space Processing

The processing procedure for a unit space is identical to that of a dit except the X_1 Flip Flop state, which stays "0" signifying a space. Also, no information is entered into the Dit and Dah Registers.

4. Dah Processing

Dah processing is the same as dit and unit space processing through Control Register position 4 (00010000), where the carry bit of the Full Adder is examined. Since a dah is greater than twice as long as a dit, the Full Adder output is a "1" and the Control Register is shifted from position 4 to position 5Y. A "1" is placed in the Dah Register and a "0" in the Dit Register. The Control Register shifts directly to position 7 and then on to 8 to complete the reset functions.

5. Character and Word Space Processing

The character or word space entrance to the converter leaves the X_1 Flip Flop in the "0" state and follows the same process until the decision is made in Control Register position 4. Since the Full Adder output is "1" and X_1 is "0", the Control Register is shifted from position 4 to 5Z. As this shift occurs, the decode signal is sent to the Dit-Dah Storage Registers, Decode Section, and Encode Section. The second least significant bit of the Count/Shift Register is zeroed in order to allow division by 4 and comparison with the reference count. This process is explained more fully in Appendix I.

The carry bit is examined again at the end of this comparison. A word space is encoded and transmitted to the output section if the Full

Adder carry bit is "0" If the carry bit is a "1" or if the word space transmission has been completed, the Control Register shifts directly to position 7 and then to position 8 to reset all functions.

6. Decoding the Dit and Dah Registers

When the decode signal is received at the Dit and Dah Registers, Decode and Encode sections, the Morse character stored in the Dit and Dah Registers is decoded, and its equivalent teletype character encoded and entered into the output buffer register. The teletype character is accompanied in the output buffer by a "1" in the indicator bit to show that it is a valid character to be transferred to the teletype equipment, and by a case bit which is "1" if upper case and "0" if lower case.

7. Multiple Character Processing

The multiple characters are processed in the same manner as the conventional characters up to the Encoding Section. Since it is necessary to print two characters when a multiple character is received, it is necessary to store the identity of the character and the decode pulse so that the second character can be processed. After the first of the two characters has been encoded, entered into the buffer, and has shifted out of the first buffer state, the second character is encoded and a new decode signal generated to enter it into the buffer. The control register is returned to the ready position after the first character has been encoded so that no delay in processing will occur when multiple characters are received.

8. Output Buffer Operation

As mentioned previously, characters awaiting printing are stored in a 7 bit parellel shift register in the output section. This register, referred to as the buffer, also stored the indicator bits and case bits for each stored character. Control and shifting within the buffer is independent of all processing section functions once the character has been entered into the first register position. The buffer does, however, make use of the 1 MHz clock for shifting. Shift control is governed only by the indicator bits. As each clock pulse occurs, each indicator bit is compared with the next indicator bit. If the next indicator bit is "0" (no character is stored in that register position), the previous character, its indicator and case bits are parallel shifted into the empty position. On each succeeding clock pulse, the character will continue to advance until a filled register position is encountered. Therefore, if an N bit buffer is used and contains no characters, any character entered into the buffer will be available for printing N microseconds after entry.

9. Case Control in the Output Section

When each character appears at the output of the buffer, the case bit is compared with the case bit of the previously printed character. If a difference in case exists, a case shift is necessary. The necessary case shift is determined, encoded, and made available to the teletype equipment. When the case shift has been completed, the character at the output of the buffer is then gated to the teletype. After the character is printed, the indicator bit is reset, the next character in the buffer moved up to the output, and the process is repeated.

10. Carriage Return and Line Feed Control

Since the desired teletype printed line length is 72 characters, a counter is provided to count the characters as they are entered into the buffer. Case shifts, carriage return, and line feed are not counted. When the count reaches 63, the word space indicator line is sampled continuously. If a word space is received before character 72, the teletype codes for carriage return and line feed are entered into the buffer. If the space has not been received by character 72, these function codes are entered anyway. This procedure is an effort to minimize the random division of words at the end of each printed line. The detailed circuit diagram of the carriage return and line feed control is shown in Figure 11.

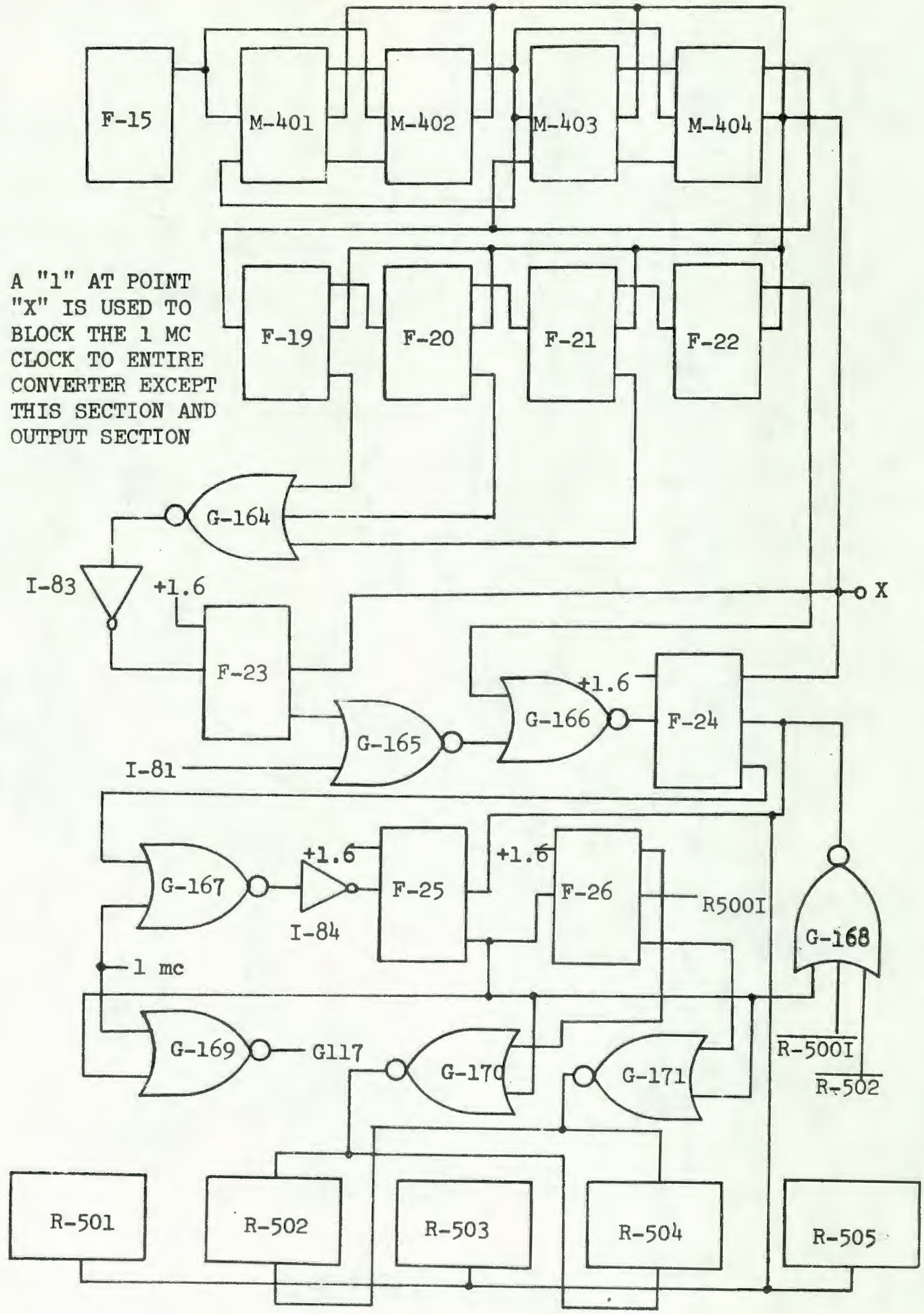


FIGURE 11

MODIFICATION FOR AUTOMATIC CARRIAGE RETURN AND LINE FEED

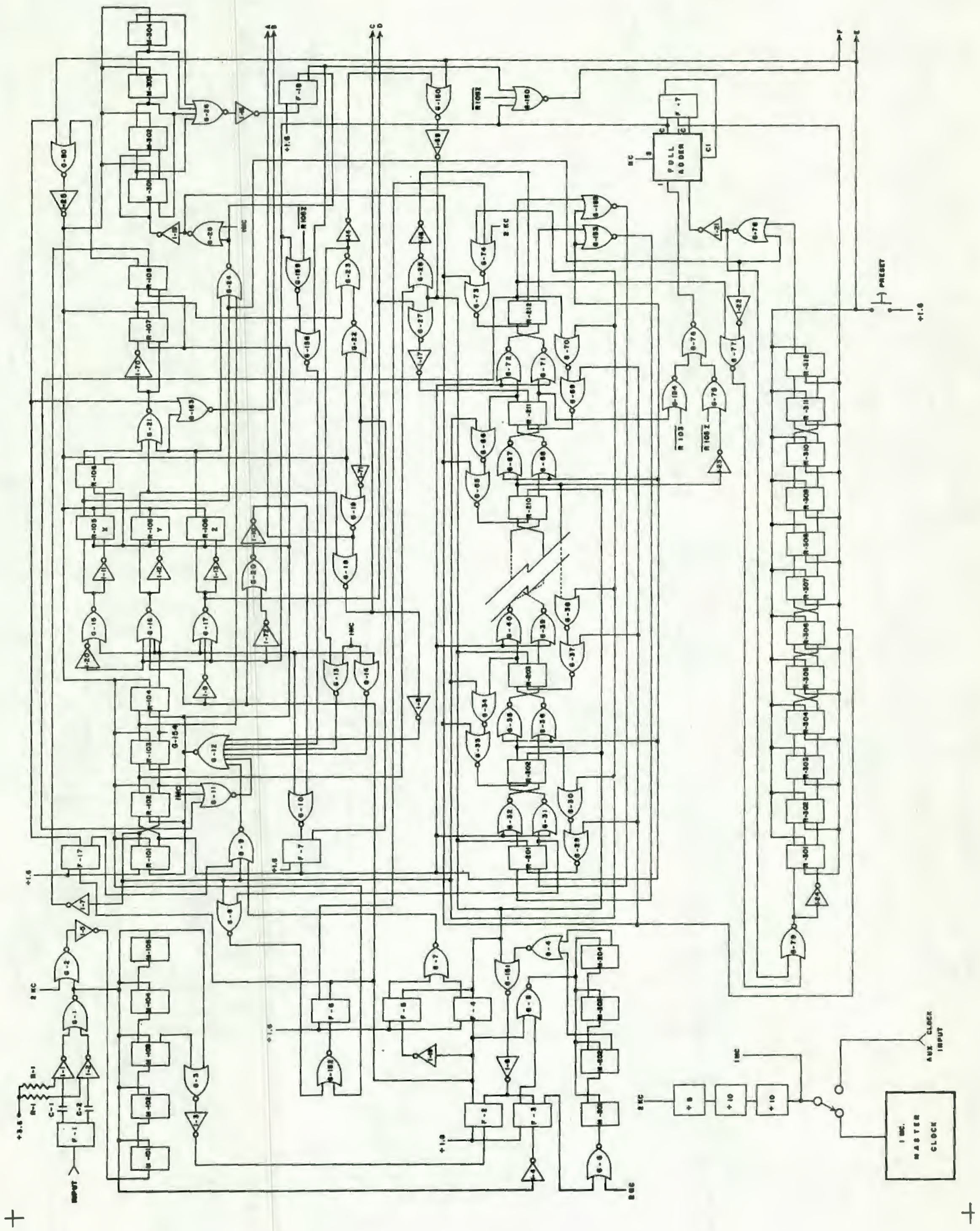


FIGURE 12A

DETAILED SCHEMATIC DIAGRAM

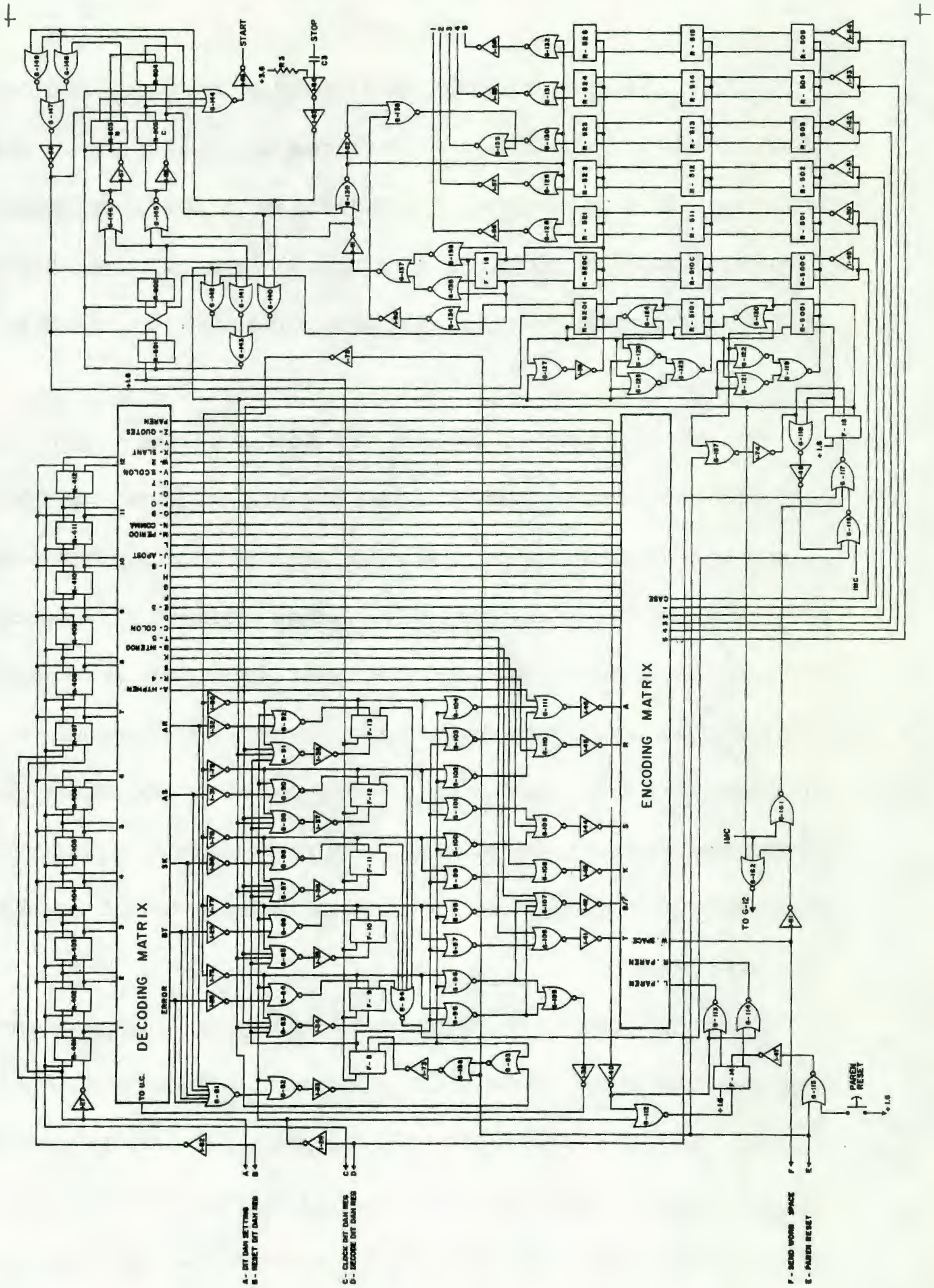


FIGURE 12B

DETAILED SCHEMATIC DIAGRAM

CHAPTER VIII

SUMMARY

Although not fully implemented at the time of writing, the results obtained thus far have been most encouraging. It is felt that some problems will be encountered, but that the basic design is sound. Due to the high speed of micrologic circuitry, there is no reason that the 100 words per minute maximum Morse sending speed could not be at least doubled.

The use of integrated circuitry has yielded a light weight, reliable, high speed converter. With a variety of power supplies, the converter should be useable in fixed, mobile, or airborne applications, both afloat and ashore as proposed in the initial design parameters. Automatic features of this design include: copying of Morse code at rates of 10 to 100 words per minute sent by machine, hand or other methods; decoding of combination characters used in proper operating procedures; sending carriage return and line feed; and sampling of operator's speed to follow his speed shifts losing only one character when a radical speed shift has taken place.

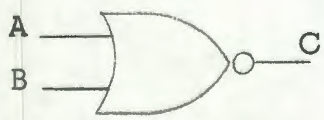
Estimated costs of the integrated circuitry for construction are approximately $1600.00 for RTL type circuits, $1740.00 for DTL type circuits, and $2270.00 for TTL type circuits. The final decision on which type of circuit to use should be based more on operating temperature range than on cost, choosing the minimum cost circuits meeting the required temperature specifications. An additional $200.00 would be required for circuit board fabrication, chassis work, and miscellaneous components. Over-all cost would be a function of the number of units being built and the additional design development costs that must be defrayed. It is felt, however, that this unit could be constructed, packaged, and used in airborne and other applications for a cost considerably lower than presently available converters.

APPENDIX I

By adding resistors, capacitors, or combining elements, many other functions can be performed by the essential elements described in Section V. Some of these special configurations that are used in the design of the signal converter are described below.

1) OR Gate

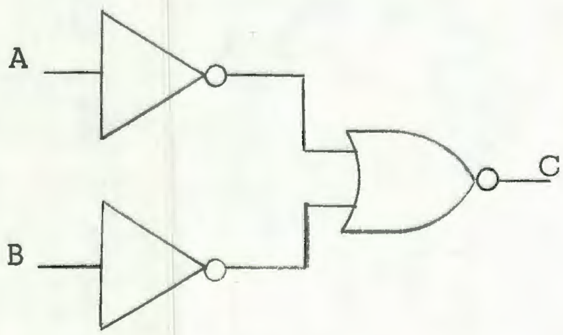
By combining a NOR gate and an Inverter, an OR gate can be formed, as shown in the truth table below.



A	B	C
1	0	1
0	1	1
1	1	1
0	0	0

2) AND Gate

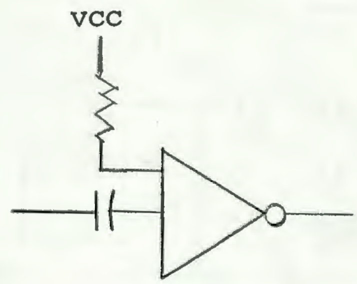
A NOR gate and two Inverters can be used to function as an AND gate, as shown below.



A	B	C
1	0	0
0	1	0
0	0	0
1	1	1

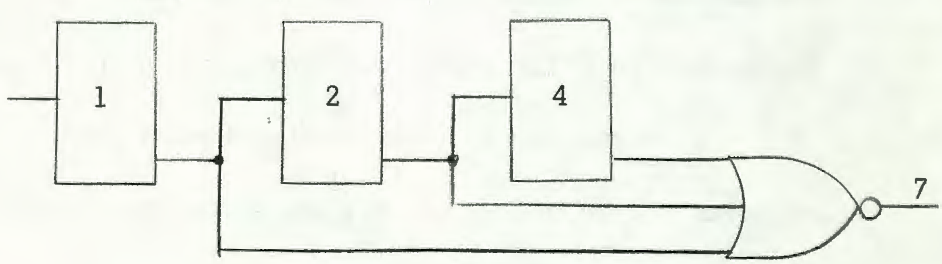
3) One Shot Multivibrator

A one shot multivibrator is a device which can be made from a resistor, a capacitor and an Inverter. Since it is capacitively coupled to the previous element, its output is normally in a "0" state. When a "1" is placed on its input, the output goes immediately to "1" and remains at "1" for a time period dependent on the values of the added resistor and capacitor. After this period, the output goes back to its "0" state until another transient on its input.

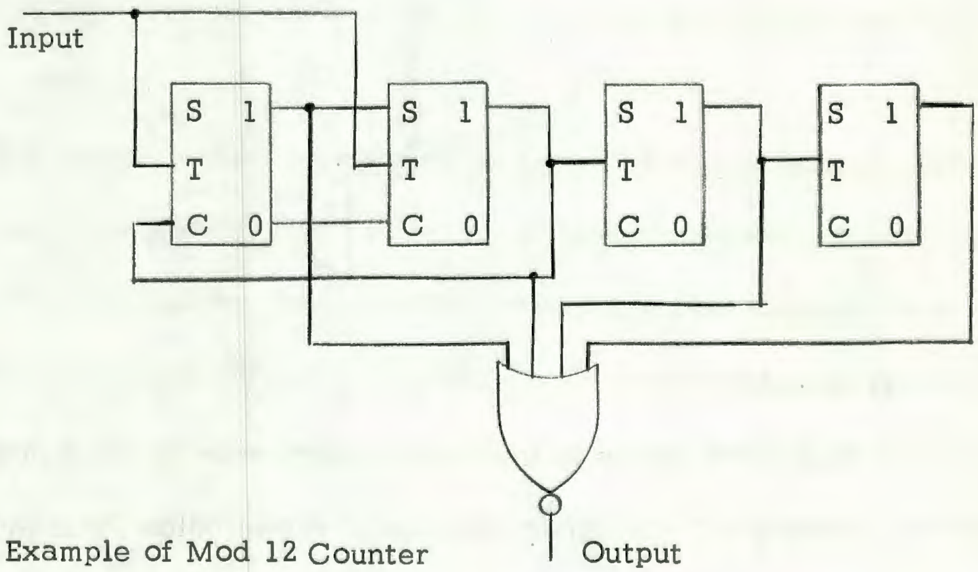


4) Counters

The JK Flip-Flop is the primary element in the design of most counters. The binary counter is shown below, and in this counter, the complement output of the previous stage is connected to the clock pulse input of the succeeding stage. The "0" outputs of the flip-flops are inputs to an NOR gate to provide the desired count. By using the "0" outputs of the flip-flops, the desired count is the output of a NOR gate.



A Mod X counter can be realized using JK Flip-Flops, but connection of inputs and outputs is different for each X count desired. The X count is the number reached before the count cycle is repeated; for example, a Mod 10 counter will accept 10 pulses and then repeat its steps. A Mod 12 counter is shown below.



Example of Mod 12 Counter

5) Shift Registers

There are many possible variations of shift registers using the JK Flip-Flop, dependent on the function desired. Some of these are serial input-parallel output, serial input-serial output, parallel input-serial output, and parallel input-parallel output. In the design of the signal converter, the serial input-parallel output shift register is used for the dit-dah register. The control register and reference register are serial input-serial output types, and the output buffer registers are parallel input-parallel output examples.

6) Count/Shift Registers

Two functions can be performed by one register by gating the inputs and outputs of each JK Flip-Flop. By gating the clock input to the flip-flop on the right end of the register and enabling the proper gates between each flip-flop, the register will count with the least significant bit remaining on the right side. This method of counting is performed in order to have the count in the proper order for shifting it out of the register to the right for serial addition or subtraction. This count left-shift right register is used to count the time duration of the incoming pulse and shift its count to a Full Adder for comparison with a stored reference. By taking the output from the complement side of the flip-flop, the complement of the incoming pulse count is obtained and the full adder actually performs a subtraction. The count can also be divided by two by taking the output from the second flip-flop from the right, but it is neccessary to "0" the output of the first flip-flop before performing the subtraction. These operations are performed to obtain an output of the Full Adder which equals reference minus incoming count divided by two. In the comparison of a character space and word space, a similar operation is performed by taking the output from the third flip-flop outputs to "0" to obtain an output of the Full Adder which equals reference minus incoming divided by four.

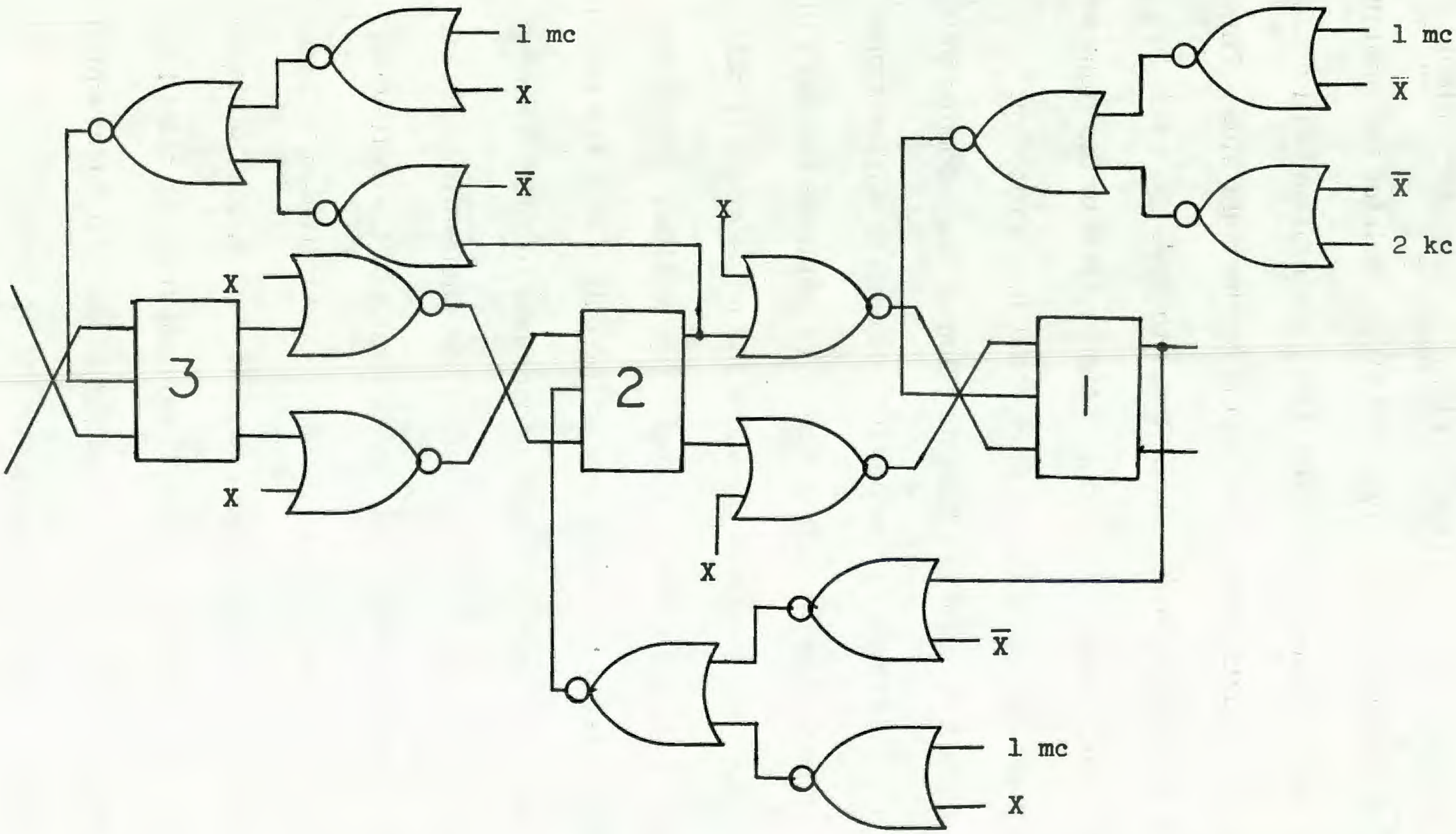


FIGURE 13

COUNT LEFT / SHIFT RIGHT REGISTER

```
      PROGRAM MINIMIZE
      DIMENSION IFORM(20),INPUT(70),INC(100)
      COMMON/MATRIX/IMATRX(70,100),JMATRX(70,100),KMATRX(1,100),KOLUMN,Y
     1,LMATRX(70,100)
      COMMON/TALEY/VAR,X,ICOUNT
      COMMON/TIM/MINS,MSECS
      DATA JUNK/'        '/,SYM1/'X       '/,SYM2/'0       '/,ZERO/'0       '/
      REAL*4  LMATRX,JMATRX,IMATRX,KMATRX
      INTEGER*4X,Y
      REAL*8VAR(100)
      SECS=ITIME(0)/100.
C
C
C   THIS PROGRAM IS DESIGNED TO MINIMIZE A DIODE DECODING MATRIX WITH X
C   PAIRS OF INPUTS AND Y OUTPUTS WITH ONLY ONE OUTPUT SPECIFIED BY A
C   PARTICULAR INPUT CODE. MAXIMUM NUMBER OF INPUT PAIRS IS 70 IN THIS
C   PROGRAM.
C
C
C   THE FIRST DATA CARD MUST CONTAIN AN INTEGER NUMBER RIGHT ADJUSTED
C   IN COLUMNS 1 THRU 4 SPECIFYING THE NUMBER OF INPUT PAIRS. THE MAXIMUM
C   NUMBER PERMITTED IN THIS PROGRAM IS 70.
C
C   THE SECOND DATA CARD MUST CONTAIN AN INTEGER NUMBER RIGHT ADJUSTED,
C   IN COLUMNS 1 THRU 4 SPECIFYING THE NUMBER OF DISCRETE OUTPUTS. THE
C   MAXIMUM NUMBER PERMITTED IN THIS PROGRAM IS 100.
C
C  (COMMENT - OBVIOUSLY A SIMPLE METHOD AS USED IN THIS PROGRAM COULD NOT
C   PERMIT FULL UTILIZATION OF ALL POSSIBLE DISCRETE OUTPUTS FOR UP TO
C   70 INPUT PAIRS DUE TO THE NUMBER FORMATS USED AND THE AMOUNT OF COMPUTER
C   MEMORY STORAGE AVAILABLE SINCE FOR 70 INPUT PAIRS, A TOTAL OF SOMETHING
C   LIKE 1,180,591,620,717,411,303,424 DISCRETE OUTPUTS ARE POSSIBLE.)
C
C   NOTE - THE MAXIMUM MATRIX SIZE IN THIS PROGRAM SHOULD NOT EXCEED 70 BY 100.
C   ALL SUCCEEDING CARDS ARE DATA CARDS, WHOSE CONTENT IS SPECIFIED BELOW.
C   THE DATA CARDS USED TO SPECIFY THE CODE OF EACH OUTPUT MUST BE IN
C   THE SAME FORM AS SHOWN AT THE END OF THE PROGRAM. THE OUTPUT NAMES
C   (ALPHANUMERIC CHARACTERS, SYMBOLS, AND WORDS IN THIS EXAMPLE) MUST
C   CONTAIN A MAXIMUM OF 8 CHARACTERS WHICH ARE ENTERED RIGHT ADJUSTED
C   IN COLUMNS 1 THRU 8 OF THE DATA CARD. THE INPUTS ARE ENTERED CONSECUTIVELY
C   IN COLUMNS 11 THRU 80, WITH THE ONE OR ZERO SPECIFIED FOR INPUT NUMBER
C   ONE IN COLUMN 11, FOR INPUT NUMBER 2 IN COLUMN 12, ETC.
C
C
C   IN THIS EXAMPLE THE INPUTS ARE READ FROM THE OUTPUTS OF A 19 BIT SHIFT
C   REGISTER WITH THE CHARACTER A BEING SPECIFIED IN BITS 1 THRU 19 IN THE
C   FOLLOWING MANNER......
C
```

```
C       A 111010000000000000
C
C
C
C  READ NUMBER OF INPUT PAIRS.
C
      READ(5,1)X
    1 FORMAT(1I4)
C
C  READ NUMBER OF DISCRETE OUTPUTS.
C
      READ(5,1)Y
C
C  READ IN DATA CARDS.
C
      DO 2 J=1,Y
    2 READ(5,3)VAR(J),(IMATRX(I,J),I=1,X)
    3 FORMAT(1A8,2X,70A1)
C
C  TEST EACH OUTPUT IN SEQUENCE AND MINIMIZE IT.
C
      DO 700 J=1,Y
      DO 4 M=1,X
      DO 4 N=1,Y
    4 JMATRX(M,N)=IMATRX(M,N)
      DO 800 I=1,X
      KOLUMN=I
      CALL MASK
C
C  CHECK TO SEE IF STILL DISCRETE.
C
      DO 5 N=1,Y
      IF(N.EQ.J)GO TO 5
    6 DO 7 K=1,X
      IF(JMATRX(K,J).NE.JMATRX(K,N))GO TO 5
    7 CONTINUE
      CALL REPLAC
      GO TO 800
    5 CONTINUE
  800 CONTINUE
      DO 8 L=1,X
    8 LMATRX(L,J)=JMATRX(L,J)
  700 CONTINUE
      WRITE(6,153)
  153 FORMAT(1H1//,T30,'THE ORIGINAL MATRIX FOLLOWS',//)
      WRITE(6,154)X
  154 FORMAT(T2, 'NUMBER OF INPUT PAIRS -',1I4)
      WRITE(6,155)Y
```

```
155 FORMAT(T2,'NUMBER OF DISCRETE OUTPUTS-',1I4,//)
    WRITE(6,156)
156 FORMAT(T2 ,'CHARACTER',T42,'INPUT NUMBER'//)
    INC(1)=1
    DO 9 I=2,100
  9 INC(I)=INC(I-1)+1
    NZ=X
    ISTART=1
    ISTOP=0
 10 IF(NZ.LT.26)GO TO 11
 12 ISTOP=ISTOP+25
    WRITE(6,157)(INC(I),I=ISTART,ISTOP)
157 FORMAT(7X,25I4,/)
    DO 42 J=1,Y
    WRITE(6,158)VAR(J),(IMATRX(I,J),I=ISTART,ISTOP)
158 FORMAT(  1A8,2X,25A4)
 42 CONTINUE
    ISTART=ISTART+25
    NZ=NZ-25
    WRITE(6,200)
200 FORMAT(1H1//,T2,'CHARACTER',T42,'INPUT NUMBER',//)
    GO TO 10
 11 ISTOP=X
    WRITE(6,157)(INC(I),I=ISTART,ISTOP)
    DO 43 J=1,Y
    WRITE(6,158)VAR(J),(IMATRX(I,J),I=ISTART,ISTOP)
 43 CONTINUE
    WRITE(6,160)
160 FORMAT(1H1//,T30,'THE MINIMIZED MATRIX FOLLOWS',//)
    ICOUNT=0
    WRITE(6,154)X
    WRITE(6,155)Y
    DO 333 J=1,Y
    DO 33 I=1,X
    IF(LMATRX(I,J)-3.)35,34,35
 34 LMATRX(I,J)=SYM1
    GO TO 33
 35 IF(LMATRX(I,J)-ZERO)300,301,300
301 LMATRX(I,J)=SYM2
300 ICOUNT=ICOUNT+1
 33 CONTINUE
333 CONTINUE
    WRITE(6,161)ICOUNT
161 FORMAT(T2,'TOTAL NUMBER OF DIODES REQUIRED -'1I6,//)
    WRITE(6,156)
    NZ=X
    ISTART=1
    ISTOP=0
```

```
  100 IF(NZ.LT.26)GO TO 111
  112 ISTOP=ISTOP+25
      WRITE(6,157)(INC(I),I=ISTART,ISTOP)
      DO 142 J=1,Y
      WRITE(6,258)VAR(J),(LMATRX(I,J),I=ISTART,ISTOP)
  258 FORMAT(1A8,2X,25A4)
  142 CONTINUE
      ISTART=ISTART+25
      NZ=NZ-25
      WRITE(6,200)
      GO TO 100
  111 ISTOP=X
      WRITE(6,157)(INC(I),I=ISTART,ISTOP)
      DO 143 J=1,Y
      WRITE(6,258)VAR(J),(LMATRX(I,J),I=ISTART,ISTOP)
  143 CONTINUE
      SECS=ITIME(0)/100.-SECS
      MINS=SECS/60.0
      MSECS= SECS-60*MINS
      WRITE(6,175)
  175 FORMAT(//,T2,'ALL INPUTS X-ED OUT ARE NOT REQUIRED FOR DISCRETE DE
     1FINITION.',/)
      WRITE(6,176)
  176 FORMAT(T2,'ANY OUTPUT DEFINED ONLY BY ZEROS SHOULD INCLUDE AT LEAS
     1T A',/,T2,'SINGLE ONE SINCE AN EMPTY REGISTER WOULD RESULT IN THIS
     2 CHARACTER.')
      CALL TALLEY
      RETURN
      END
      SUBROUTINE MASK
      COMMON/MATRIX/IMATRX(70,100),JMATRX(70,100),KMATRX(1,100),KOLUMN,Y
     1,LMATRX(70,100)
      REAL*4  LMATRX,JMATRX,IMATRX,KMATRX
      INTEGER*4X,Y
C  THIS SUBROUTINE MASKS THE INPUT SPECIFIED BY THE VALUE OF KOLUMN.
      DO 1 J=1,Y
      KMATRX(1,J)=JMATRX(KOLUMN,J)
      JMATRX(KOLUMN,J)=3
    1 CONTINUE
      RETURN
      END
      SUBROUTINE REPLAC
      COMMON/MATRIX/IMATRX(70,100),JMATRX(70,100),KMATRX(1,100),KOLUMN,Y
     1,LMATRX(70,100)
      REAL*4  LMATRX,JMATRX,IMATRX,KMATRX
      INTEGER*4X,Y
C  THIS SUBROUTINE REPLACES ALL VALUES OF MASKED INPUT IF TEST FAILS.
      DO 1 J=1,Y
```

```
    1 JMATRX(KOLUMN,J)=KMATRX(1,J)
      RETURN
      END
      SUBROUTINE TALLEY
      DIMENSION IDRECT(70),ICOMPL(70)
      COMMON/MATRIX/IMATRX(70,100),JMATRX(70,100),KMATRX(1,100),KOLUMN,Y
     1,LMATRX(70,100)
      COMMON/TALEY/VAR,X,ICOUNT
      COMMON/TIM/MINS,MSECS
      DATA JUNK/'    '/,SYM1/'X   '/,ONE/'1   '/,ZERO/'0   '/
      REAL*4  LMATRX,JMATRX,IMATRX,KMATRX
      REAL*8VAR(100)
      INTEGER*4X,Y
C   THIS SUBROUTINE COUNTS THE NUMBER OF CONNECTIONS TO EACH INPUT AND
C
C   LISTS THE INPUTS NECESSARY TO UNIQUELY SPECIFY EACH OUTPUT
C
      WRITE(6,1)X,Y
    1 FORMAT(1H1//,T30,'TABULATION OF RESULTS',//,T2,'NUMBER OF INPUT PA
     1IRS -'1I4,/,T2,'NUMBER OF DISCRETE OUTPUTS -',1I4,//)
      WRITE(6,2)ICOUNT
    2 FORMAT(T2,'TOTAL NUMBER OF DIODES REQUIRED -',1I6,//)
      ISAVE=X*Y-ICOUNT
      WRITE(6,13)ISAVE
   13 FORMAT(T2,'NUMBER OF DIODES SAVED -',1I5,//)
      WRITE(6,999)MINS,MSECS
  999 FORMAT(T2,'TIME REQUIRED FOR MINIMIZATION - ',I4,2X,7HMINUTES,2X,
     1I2,2X,7HSECONDS,//)
C
C   THE NOTATION '1C' DENOTES THE COMPLEMENT OF INPUT 1.
C
      WRITE(6,3)
    3 FORMAT(T3,'INPUT',T12,'NUMBER OF CONNECTIONS REQUIRED',/)
      DO 4 I=1,X
      NZEROS=0
      NONES=0
      DO 5 J=1,Y
      IF(LMATRX(I,J).EQ.SYM1)GO TO 5
      IF(LMATRX(I,J).EQ.ONE)GO TO 6
      NZEROS=NZEROS+1
      GO TO 5
    6 NONES=NONES+1
    5 CONTINUE
      WRITE(6,7)I,NONES,I,NZEROS
    7 FORMAT(1X,1I4,17X,1I4,/,3X,1I2,1HC,16X,1I4)
    4 CONTINUE
      WRITE(6,8)
    8 FORMAT(//,T3,'OUTPUT',T23,'INPUTS REQUIRED',/)
```

```
      DO 9 J=1,Y
      NUMD=0
      NUMC=0
      DO 20 K=1,70
      ICOMPL(K)=JUNK
   20 IDRECT(K)=JUNK
      DO 10 I=1,X
      IF(LMATRX(I,J).EQ.SYM1)GO TO 10
      IF(LMATRX(I,J).EQ.ONE)GO TO 11
      NUMC=NUMC+1
      ICOMPL(NUMC)=I
      GO TO 10
   11 NUMD=NUMD+1
      IDRECT(NUMD)=I
   10 CONTINUE
      IF(NUMD.EQ.0)GO TO 100
      WRITE(6,12)VAR(J),(IDRECT(L),L=1,NUMD)
   12 FORMAT(1A8,T10,'DIRECT          ',70(I2,1X))
      GO TO 102
  100 WRITE(6,101)VAR(J)
  101 FORMAT(1A8,T10,'DIRECT         NONE')
  102 IF(NUMC.EQ.0)GO TO 103
      WRITE(6,15) (ICOMPL(M),M=1,NUMC)
   15 FORMAT(T10,'COMPLEMENT    ',70(I2,1X),/)
      GO TO 9
  103 WRITE(6,104)
  104 FORMAT(T10,'COMPLEMENT   NONE')
    9 CONTINUE
      RETURN
      END
      DATA FOLLOWS
12
51
        A  010000100000
        B  111000000100
        C  101000010100
        D  110000001000
        E  100000000000
        F  101100010000
        G  100000011000
        H  111100000000
        I  110000000000
        J  000100111000
        K  010000101000
        L  110100001000
        M  000000110000
        N  100000010000
        O  000000111000
```

P	100100011000
Q	010000101100
R	101000010000
S	111000000000
T	000000100000
U	011000100000
V	011100100000
W	001000110000
X	011000100100
Y	001000110100
Z	110000001100
1	000010111100
2	000110111000
3	001110110000
4	011110100000
5	111110000000
6	111100000010
7	111000000110
8	110000001110
9	100000011110
0	000000111110
PERIOD	010101101010
COMMA	001100110011
SLANT	101100010010
INTEROG	110011001100
PARENS	010010101101
COLON	111000000111
S-COLON	101010010101
HYPHEN	011110100001
QUOTES	101101010010
APOST	100001011110
ERROR	111111000000
AS	111010000100
AR	101010010100
BT	011100100010
SK	010111101000

THE ORIGINAL MATRIX FOLLOWS

NUMBER OF INPUT PAIRS - 12
NUMBER OF DISCRETE OUTPUTS- 51

CHARACTER INPUT NUMBER

	1	2	3	4	5	6	7	8	9	10	11	12
A	0	1	0	0	0	0	1	0	0	0	0	0
B	1	1	1	0	0	0	0	0	0	1	0	0
C	1	0	1	0	0	0	0	1	0	1	0	0
D	1	1	0	0	0	0	0	0	1	0	0	0
E	1	0	0	0	0	0	0	0	0	0	0	0
F	1	0	1	1	0	0	0	1	0	0	0	0
G	1	0	0	0	0	0	0	1	1	0	0	0
H	1	1	1	1	0	0	0	0	0	0	0	0
I	1	1	0	0	0	0	0	0	0	0	0	0
J	0	0	0	1	0	0	1	1	1	0	0	0
K	0	1	0	0	0	0	1	0	1	0	0	0
L	1	1	0	1	0	0	0	0	1	0	0	0
M	0	0	0	0	0	0	1	1	0	0	0	0
N	1	0	0	0	0	0	0	1	0	0	0	0
O	0	0	0	0	0	0	1	1	1	0	0	0
P	1	0	0	1	0	0	0	1	1	0	0	0
Q	0	1	0	0	0	0	1	0	1	1	0	0
R	1	0	1	0	0	0	0	1	0	0	0	0
S	1	1	1	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	1	0	0	0	0	0
U	0	1	1	0	0	0	1	0	0	0	0	0
V	0	1	1	1	0	0	1	0	0	0	0	0
W	0	0	1	0	0	0	1	1	0	0	0	0
X	0	1	1	0	0	0	1	0	0	1	0	0
Y	0	0	1	0	0	0	1	1	0	1	0	0
Z	1	1	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	1	0	1	1	1	1	0	0
2	0	0	0	1	1	0	1	1	1	0	0	0
3	0	0	1	1	1	0	1	1	0	0	0	0
4	0	1	1	1	1	0	1	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	1	1	0
8	1	1	0	0	0	0	0	0	1	1	1	0
9	1	0	0	0	0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	1	1	1	1	1	0
PERIOD	0	1	0	1	0	1	1	0	1	0	1	0
COMMA	0	0	1	1	0	0	1	1	0	0	1	1
SLANT	1	0	1	1	0	0	0	1	0	0	1	0
INTEROG	1	1	0	0	1	1	0	0	1	1	0	0
PARENS	0	1	0	0	1	0	1	0	1	1	0	1
COLON	1	1	1	0	0	0	0	0	0	1	1	1
S-COLON	1	0	1	0	1	0	0	1	0	1	0	1
HYPHEN	0	1	1	1	1	0	1	0	0	0	0	1
QUOTES	1	0	1	1	0	1	0	1	0	0	1	0
APOST	1	0	0	0	0	1	0	1	1	1	1	0
ERROR	1	1	1	1	1	1	0	0	0	0	0	0
AS	1	1	1	0	1	0	0	0	0	1	0	0
AR	1	0	1	0	1	0	0	1	0	1	0	0
BT	0	1	1	1	0	0	1	0	0	0	1	0
SK	0	1	0	1	1	1	1	0	1	0	0	0

THE MINIMIZED MATRIX FOLLOWS

NUMBER OF INPUT PAIRS - 12
NUMBER OF DISCRETE OUTPUTS- 51

TOTAL NUMBER OF DIODES REQUIRED - 214

CHARACTER	INPUT NUMBER 1	2	3	4	5	6	7	8	9	10	11	12
A	X	1	0	X	X	X	1	X	0	X	X	X
B	X	X	X	X	0	X	0	0	0	1	0	X
C	X	X	X	X	0	X	0	1	X	1	0	X
D	X	X	X	0	X	X	0	0	1	0	X	X
E	X	0	X	X	X	X	0	0	X	X	X	X
F	X	X	X	1	X	X	0	1	0	X	0	X
G	X	X	X	0	X	X	0	1	1	X	0	X
H	X	X	X	1	0	X	0	0	0	X	0	X
I	X	1	0	X	X	X	0	X	0	X	X	X
J	X	X	X	1	0	X	1	X	1	X	0	X
K	X	X	X	X	X	0	1	0	1	0	X	X
L	X	X	X	1	X	X	0	0	1	X	X	X
M	X	X	0	X	X	X	1	1	0	X	X	X
N	X	X	0	X	X	X	0	1	0	X	X	X
O	X	X	X	0	X	X	1	1	1	0	X	X
P	X	X	X	1	X	X	0	1	1	X	X	X
Q	X	X	X	X	X	X	1	0	1	1	X	0
R	X	X	1	0	X	X	0	1	X	0	X	X
S	X	X	1	0	X	X	0	0	X	0	X	X
T	X	0	X	X	X	X	1	0	X	X	X	X
U	X	X	1	0	X	X	1	0	X	0	X	X
V	X	X	X	1	0	X	1	X	0	X	0	X
W	X	X	1	X	0	X	1	1	X	0	X	0
X	X	X	X	X	X	X	1	0	0	1	X	X
Y	X	X	X	X	X	X	1	1	0	1	X	X
Z	X	X	X	X	X	0	0	X	1	1	0	X
1	X	X	X	X	X	X	X	1	1	1	0	X
2	X	X	X	X	1	X	X	1	1	0	X	X
3	X	X	X	X	1	X	X	1	0	0	X	X
4	X	X	X	X	1	X	1	0	0	X	X	0
5	X	X	X	X	1	0	0	X	X	0	X	X
6	X	X	X	X	X	X	0	0	X	0	1	X
7	X	X	X	X	X	X	X	X	0	1	1	0
8	X	X	X	X	X	X	X	0	1	1	1	X
9	X	X	X	X	X	0	0	1	X	1	1	X
0	X	X	X	X	X	X	1	X	X	1	1	X
PERIOD	X	X	X	X	X	X	X	X	1	0	1	X
COMMA	X	X	X	X	X	X	X	X	X	0	1	1
SLANT	X	X	X	X	X	0	X	1	X	0	1	0
INTEROG	X	X	X	X	X	1	X	X	X	1	0	X
PARENS	X	X	X	X	X	X	X	X	1	X	X	1
COLON	X	X	X	X	X	X	X	X	X	1	1	1
S-COLON	X	X	X	X	X	X	X	X	0	1	0	1
HYPHEN	X	X	X	X	X	X	X	X	X	0	0	1
QUOTES	X	X	X	X	X	1	X	X	0	X	1	X
APOST	X	X	X	X	X	1	X	X	X	1	1	X
ERROR	X	X	X	X	X	1	X	X	0	X	0	X
AS	X	X	X	X	1	X	X	0	0	1	X	X
AR	X	X	X	X	1	X	X	1	0	1	X	0
BT	X	X	X	X	X	X	1	X	0	X	1	0
SK	X	X	X	X	X	1	X	X	1	0	0	X

ALL INPUTS X-ED OUT ARE NOT REQUIRED FOR DISCRETE DEFINITION.

ANY OUTPUT DEFINED ONLY BY ZEROS SHOULD INCLUDE AT LEAST A SINGLE ONE SINCE AN EMPTY REGISTER WOULD RESULT IN THIS CHARACTER.

TABULATION OF RESULTS

NUMBER OF INPUT PAIRS - 12
NUMBER OF DISCRETE OUTPUTS - 51

TOTAL NUMBER OF DIODES REQUIRED - 214

NUMBER OF DIODES SAVED - 398

TIME REQUIRED FOR MINIMIZATION - 0 MINUTES 5 SECONDS

INPUT	NUMBER OF CONNECTIONS REQUIRED
1	0
1C	0
2	2
2C	2
3	4
3C	4
4	6
4C	6
5	6
5C	6
6	5
6C	5
7	15
7C	17
8	16
8C	15
9	15
9C	19
10	17
10C	16
11	12
11C	14
12	5
12C	7

OUTPUT		INPUTS REQUIRED
A	DIRECT	2 7
	COMPLEMENT	3 9
B	DIRECT	10
	COMPLEMENT	5 7 8 9 11
C	DIRECT	8 10
	COMPLEMENT	5 7 11
D	DIRECT	9
	COMPLEMENT	4 7 8 10
E	DIRECT	NONE
	COMPLEMENT	2 7 8
F	DIRECT	4 8
	COMPLEMENT	7 9 11
G	DIRECT	8 9
	COMPLEMENT	4 7 11
H	DIRECT	4
	COMPLEMENT	5 7 8 9 11
I	DIRECT	2
	COMPLEMENT	3 7 9
J	DIRECT	4 7 9
	COMPLEMENT	5 11
K	DIRECT	7 9
	COMPLEMENT	6 8 10
L	DIRECT	4 9
	COMPLEMENT	7 8
M	DIRECT	7 8
	COMPLEMENT	3 9
N	DIRECT	8
	COMPLEMENT	3 7 9
O	DIRECT	7 8 9
	COMPLEMENT	4 10
P	DIRECT	4 8 9
	COMPLEMENT	7
Q	DIRECT	7 9 10
	COMPLEMENT	8 12
R	DIRECT	3 8
	COMPLEMENT	4 7 10
S	DIRECT	3
	COMPLEMENT	4 7 8 10
T	DIRECT	7
	COMPLEMENT	2 8
U	DIRECT	3 7
	COMPLEMENT	4 8 10
V	DIRECT	4 7
	COMPLEMENT	5 9 11
W	DIRECT	3 7 8
	COMPLEMENT	5 10 12
X	DIRECT	7 10
	COMPLEMENT	8 9
Y	DIRECT	7 8 10
	COMPLEMENT	9
Z	DIRECT	9 10
	COMPLEMENT	6 7 11
1	DIRECT	8 9 10
	COMPLEMENT	11
2	DIRECT	5 8 9
	COMPLEMENT	10
3	DIRECT	5 8
	COMPLEMENT	9 10
4	DIRECT	5 7
	COMPLEMENT	8 9 12
5	DIRECT	5
	COMPLEMENT	6 7 10
6	DIRECT	11
	COMPLEMENT	7 8 10
7	DIRECT	10 11
	COMPLEMENT	9 12

8	DIRECT	9 10 11
	COMPLEMENT	8
9	DIRECT	8 10 11
	COMPLEMENT	6 7
0	DIRECT	7 10 11
	COMPLEMENT	NONE
PERIOD	DIRECT	9 11
	COMPLEMENT	10
COMMA	DIRECT	11 12
	COMPLEMENT	10
SLANT	DIRECT	8 11
	COMPLEMENT	6 10 12
INTEROG	DIRECT	6 10
	COMPLEMENT	11
PARENS	DIRECT	9 12
	COMPLEMENT	NONE
COLON	DIRECT	10 11 12
	COMPLEMENT	NONE
S-COLON	DIRECT	10 12
	COMPLEMENT	9 11
HYPHEN	DIRECT	12
	COMPLEMENT	10 11
QUOTES	DIRECT	6 11
	COMPLEMENT	9
APOST	DIRECT	6 10 11
	COMPLEMENT	NONE
ERROR	DIRECT	6
	COMPLEMENT	9 11
AS	DIRECT	5 10
	COMPLEMENT	8 9
AR	DIRECT	5 8 10
	COMPLEMENT	9 12
BT	DIRECT	7 11
	COMPLEMENT	9 12
SK	DIRECT	6 9
	COMPLEMENT	10 11

BIBLIOGRAPHY

1. Gold, B. Machine recognition of hand-sent Morse Code. Institute of Radio Engineers, Transaction on Information Theory. March, 1959: 17.

2. Mc Elwain, D. K. and M. B. Evens. The Degarbler - a program for correcting machine read Morse Code. Information and Control, May, 1962: 368.

3. Scalf, F. R., B. L. Powers and J. S. Demetry. Simplification of two-level logic by decoding and encoding. Institute of Electrical and Electronics Engineers, Transactions on Education. March, 1968.

4. Kretzman, B. H. The New RTTY Handbook. Cowan, 1962.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, California 93940	2
3.	Professor J. S. Demetry Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	LT B. L. Powers, Jr., USN 300 N. River Drive Pennsville, New Jersey	1
5.	LT F. R. Scalf, Jr., USN 1417 N. Fourth Avenue Knoxville, Tennessee	1

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

THE DESIGN OF A MORSE-TO-TELETYPE SIGNAL CONVERTER USING INTEGRATED MICROLOGIC CIRCUITRY

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Thesis

5. AUTHOR(S) *(Last name, first name, initial)*

POWERS, Byron L., Jr., LT, USN, and SCALF, Fred R., Jr., LT, USN

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1968	72	4

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

[redacted]

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

This paper investigates the use of a Morse to teletype signal converter with existing teletype equipment to continuously monitor and display received Morse signals. This devide would greatly reduce operator fatigue and provide increased efficiency.

Important aspects of the converter design problem are presented and different approaches to the problems encountered in this design are developed. A micrologic digital design is presented and its operation discussed. Although not fully implemented, it is considered far superior to other methods of implementation. It will accept Morse keyed audio signals in the 300-3000 hz. range at keying speeds of 10 to 100 words per minute and convert them to teletype code.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
MORSE CODE TELETYPE SIGNAL CONVERSION CODE CONVERSION						

[redacted]