

April 15, 1969

J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 1 of 26

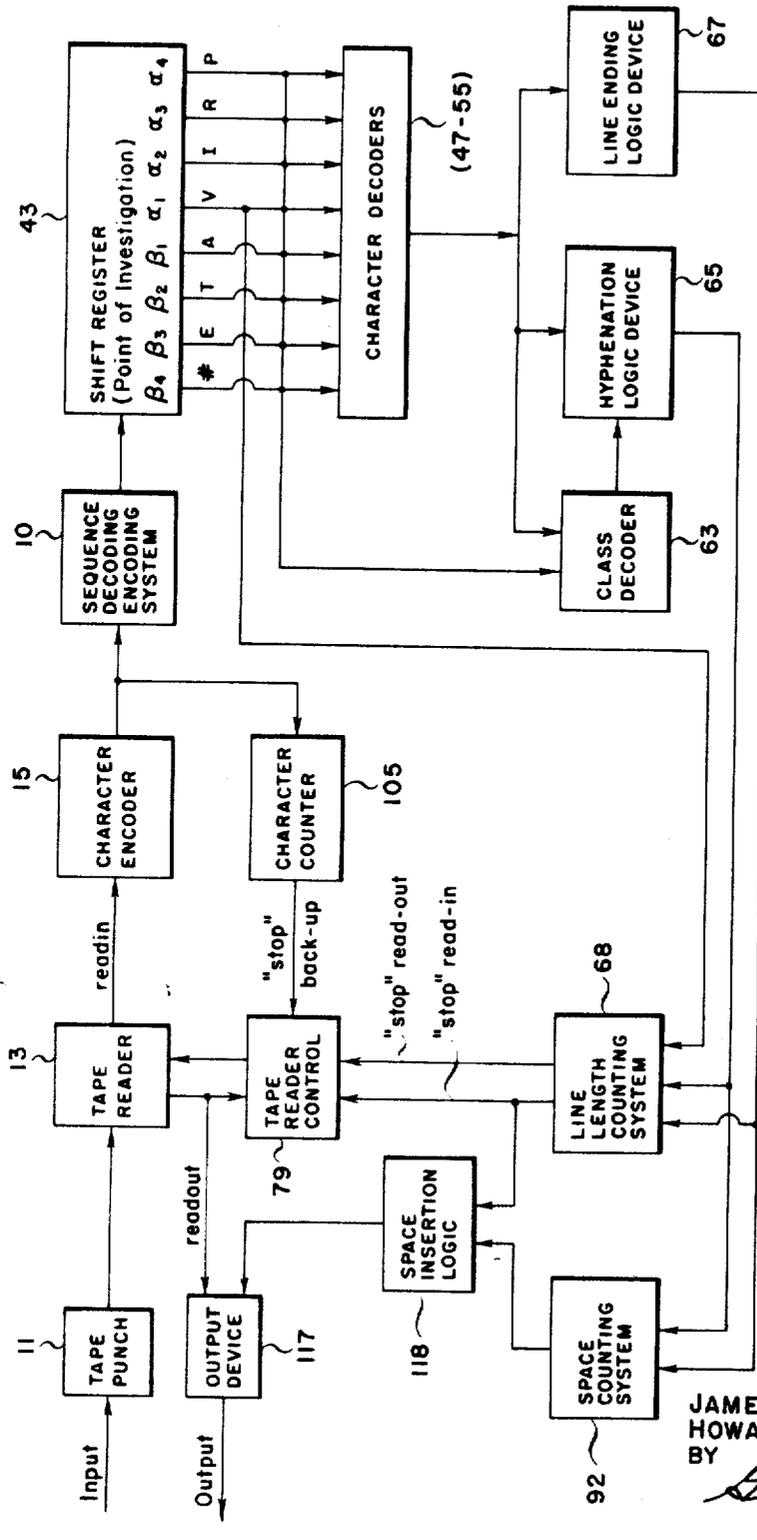


FIG. 1

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY

*George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 2 of 26

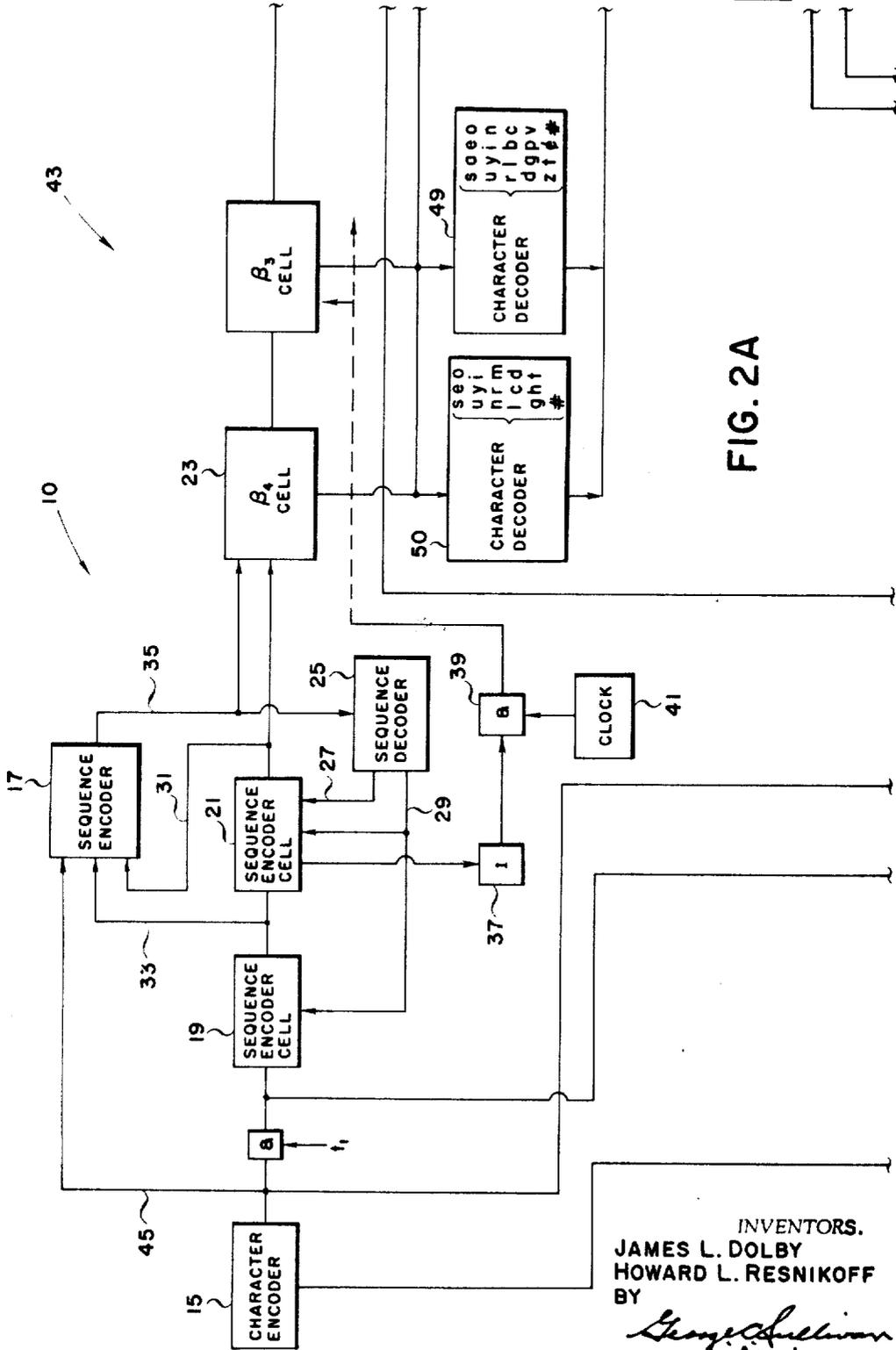


FIG. 2A

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY

*George Sullivan*  
Agent

April 15, 1969

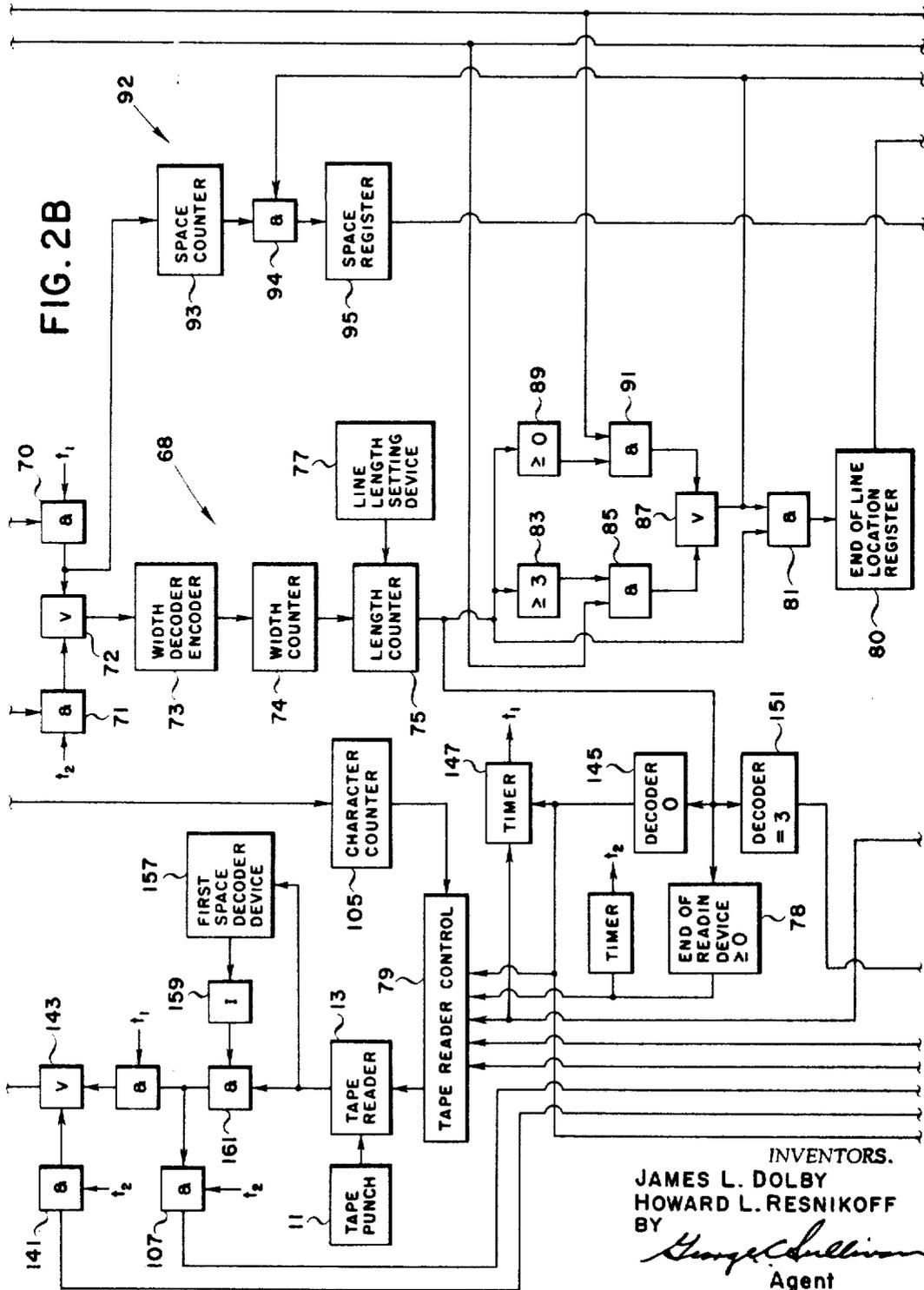
J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 3 of 26



INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

BY *George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 4 of 26

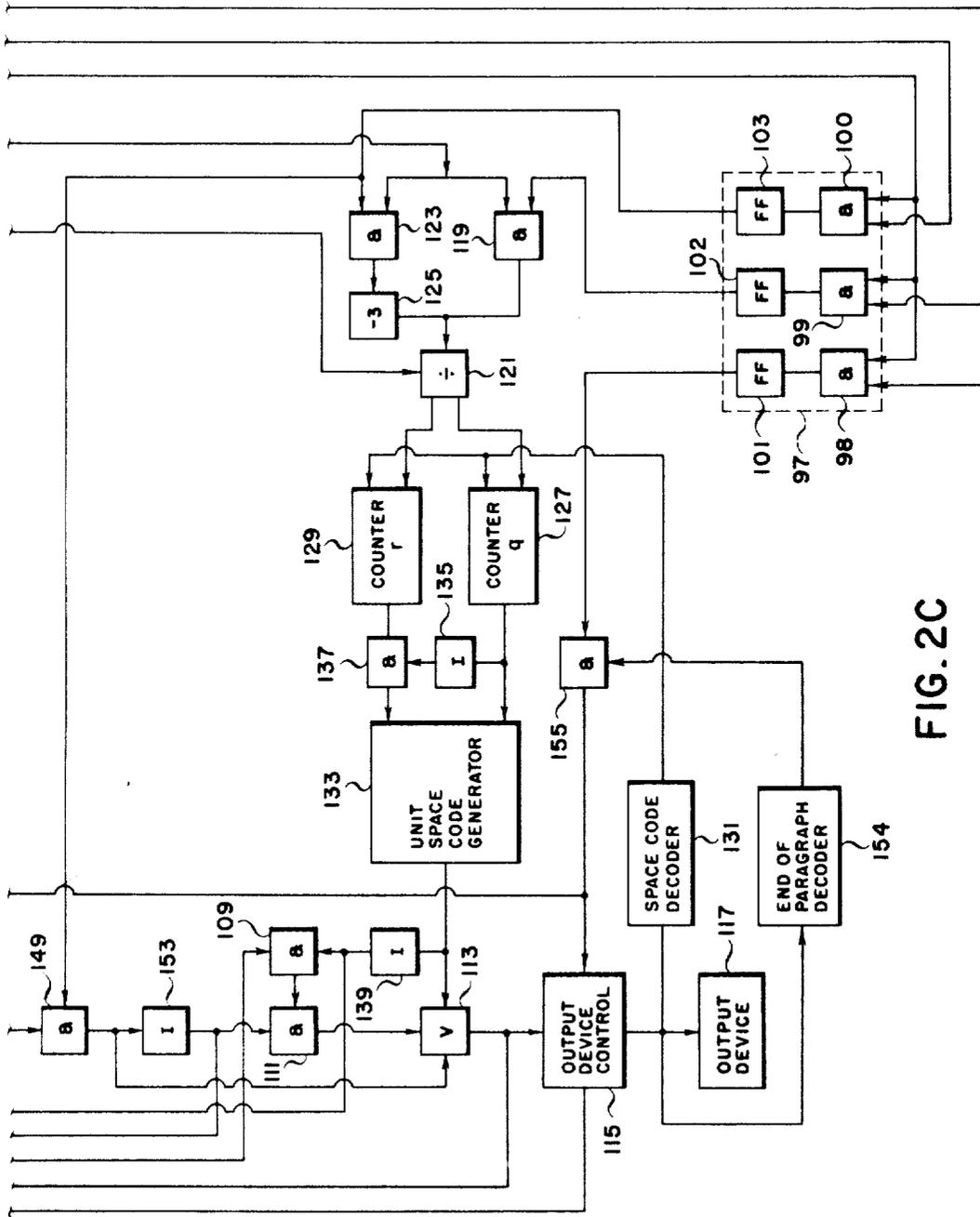


FIG. 2C

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY  
*George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 5 of 26

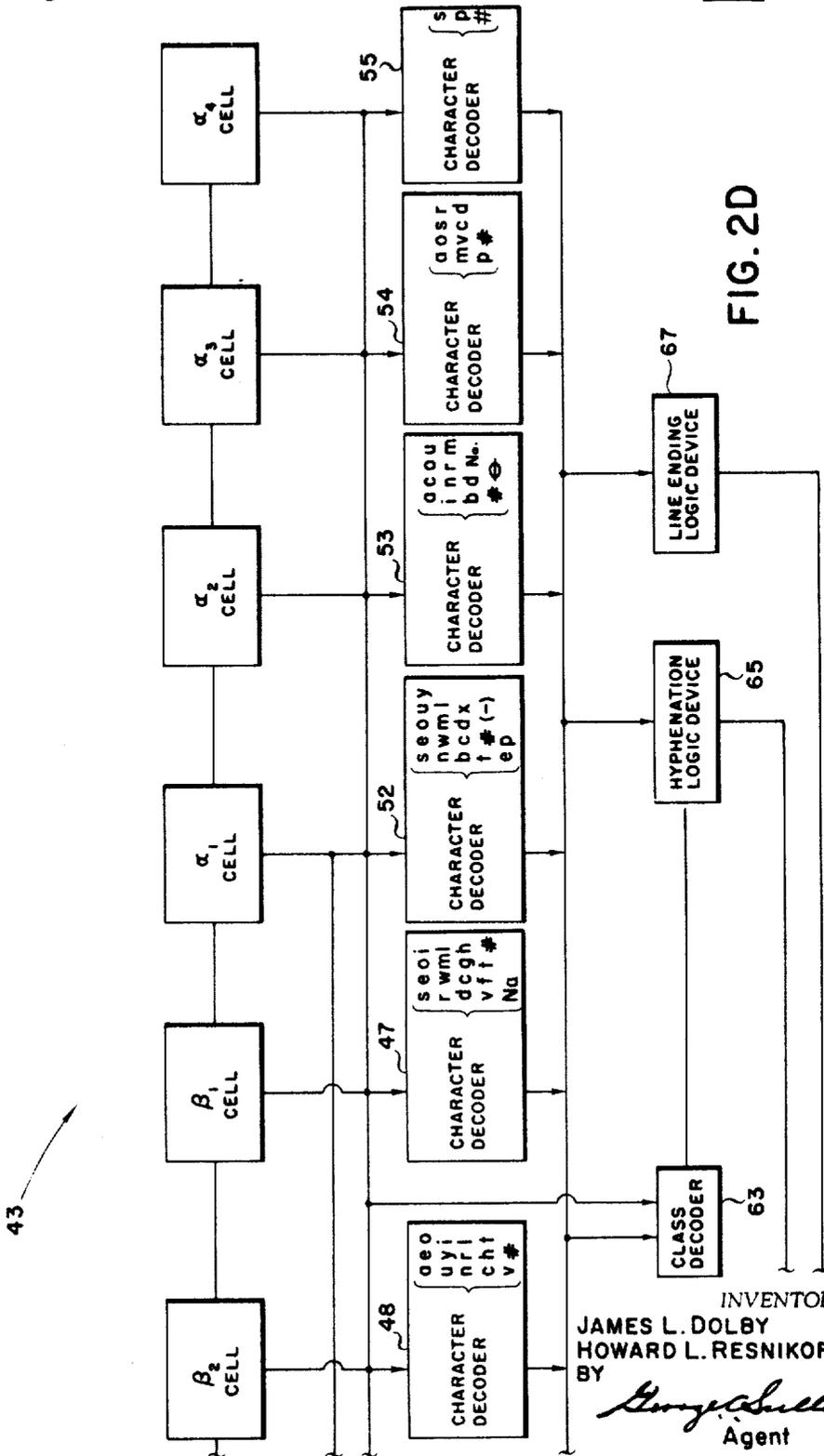


FIG. 2D

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

BY *George J. Sullivan*  
Agent

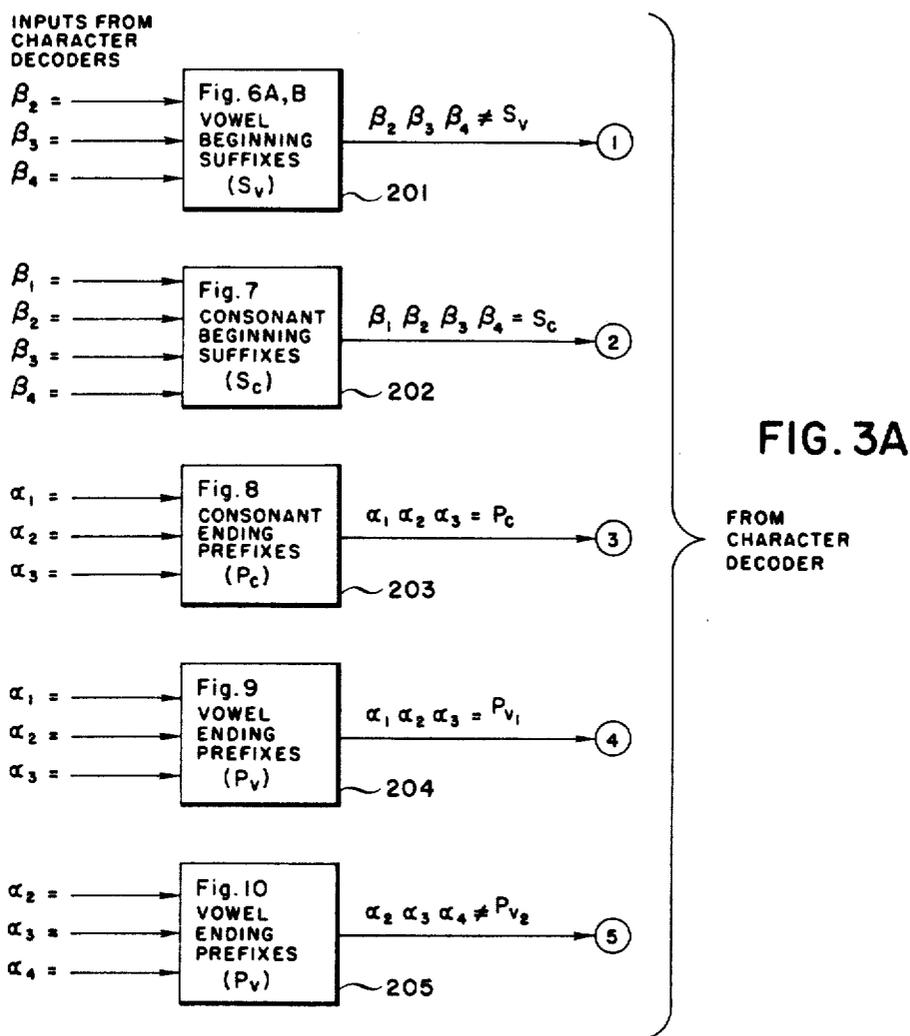
April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 6 of 26



INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY

*George Sullivan*  
Agent

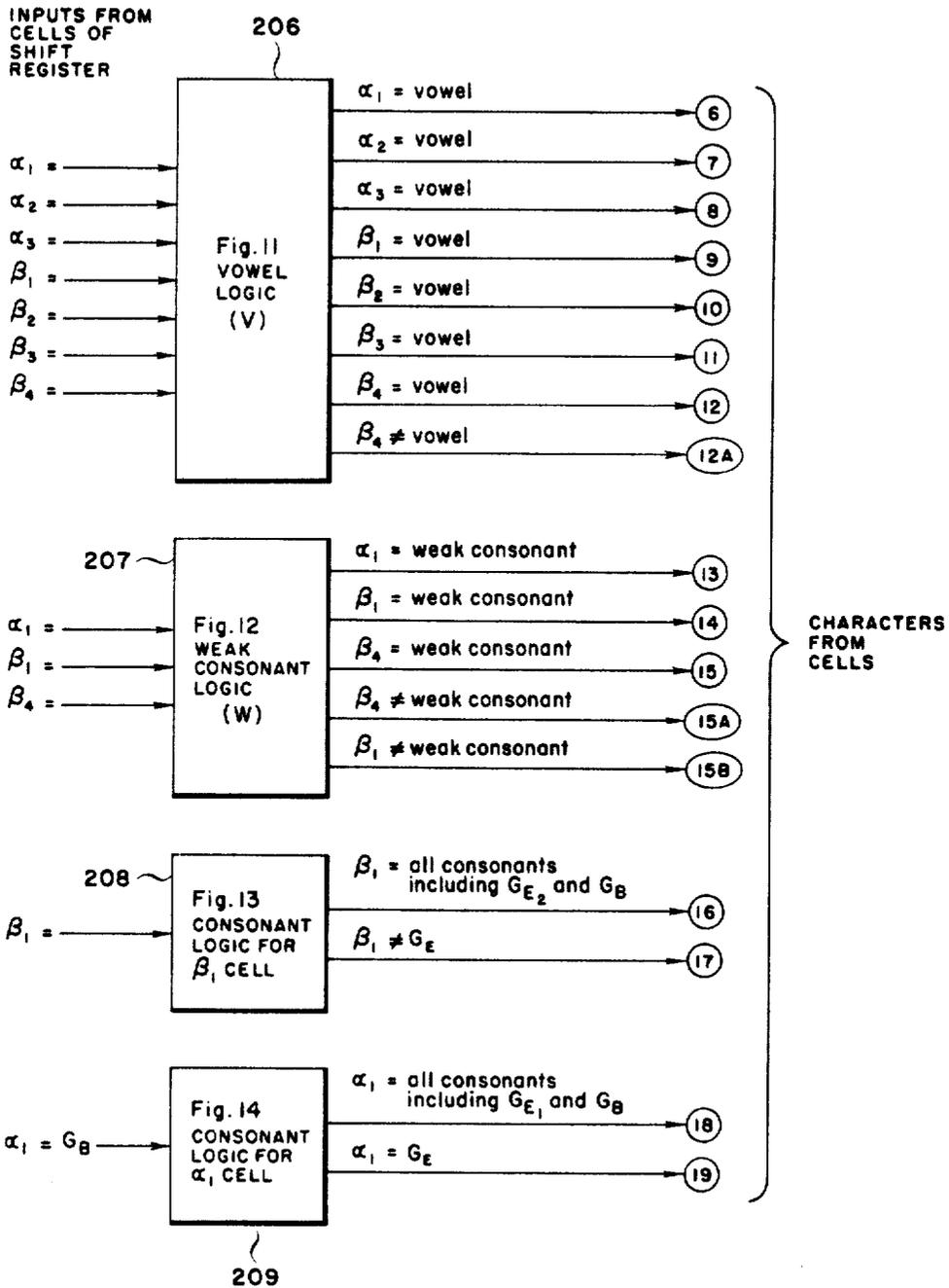


FIG. 3B

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY

*George Sullivan*  
Agent

April 15, 1969

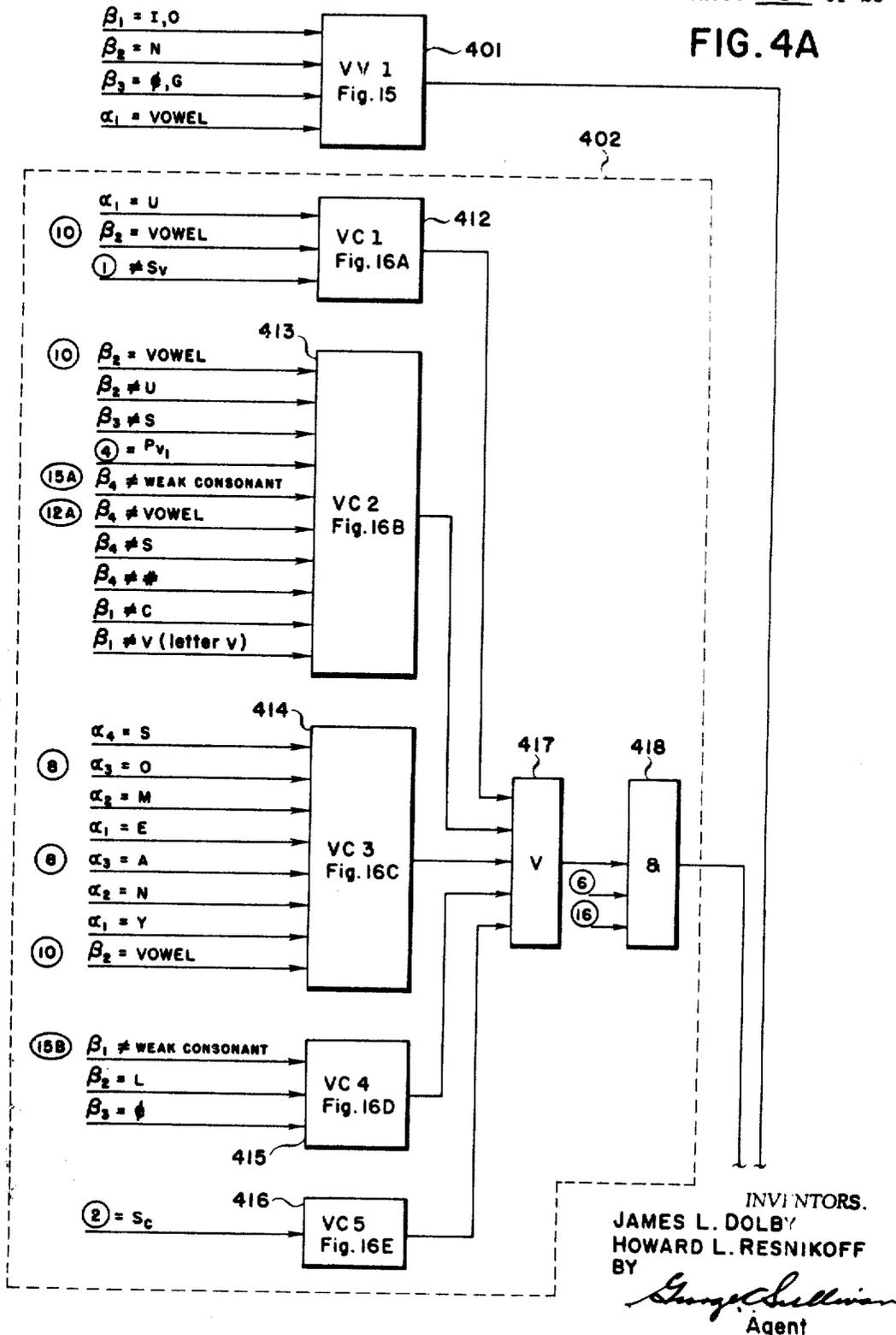
J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 8 of 26

FIG. 4A



April 15, 1969

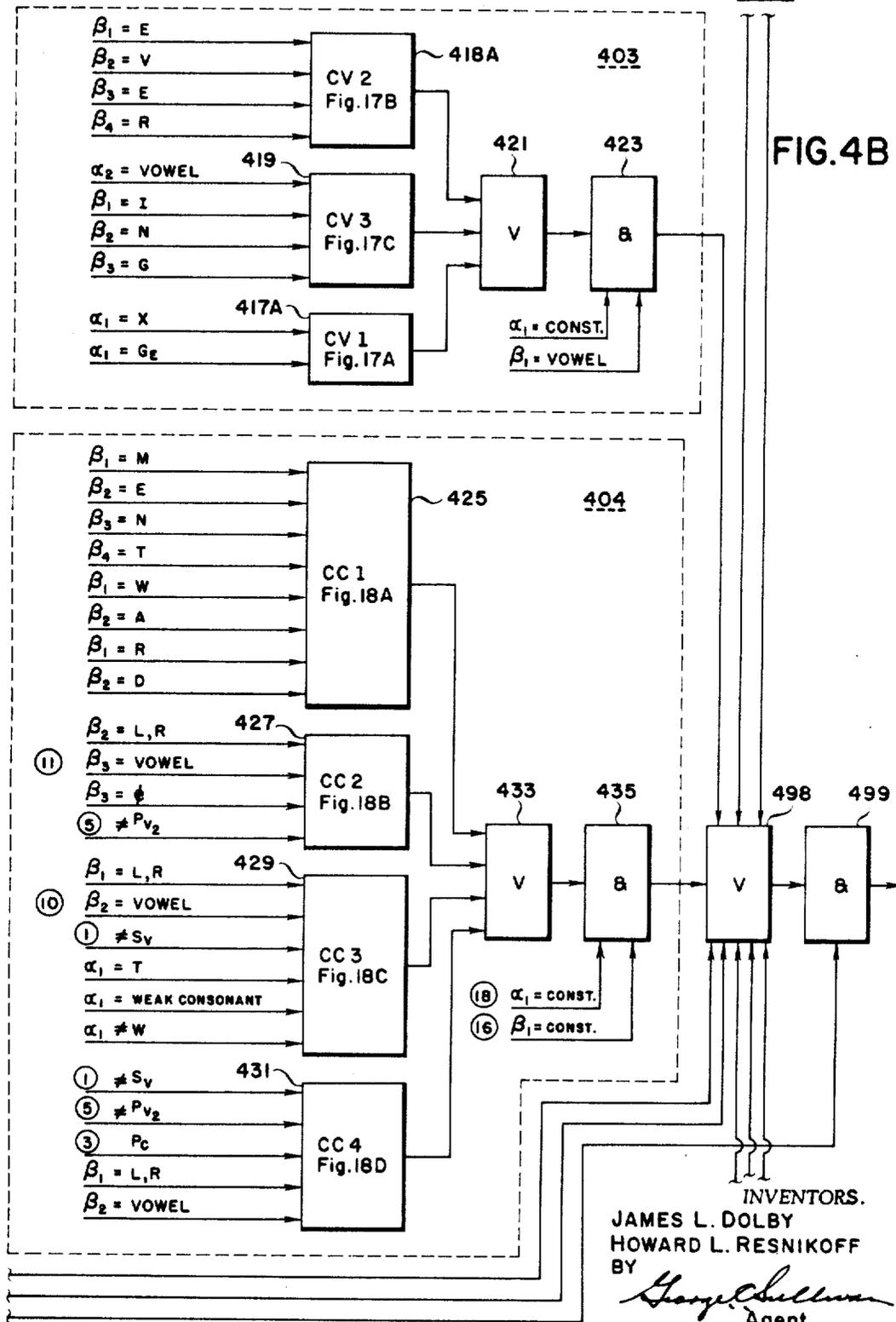
J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 9 of 26



INVENTORS.  
 JAMES L. DOLBY  
 HOWARD L. RESNIKOFF  
 BY *George S. Sullivan*  
 Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 10 of 26

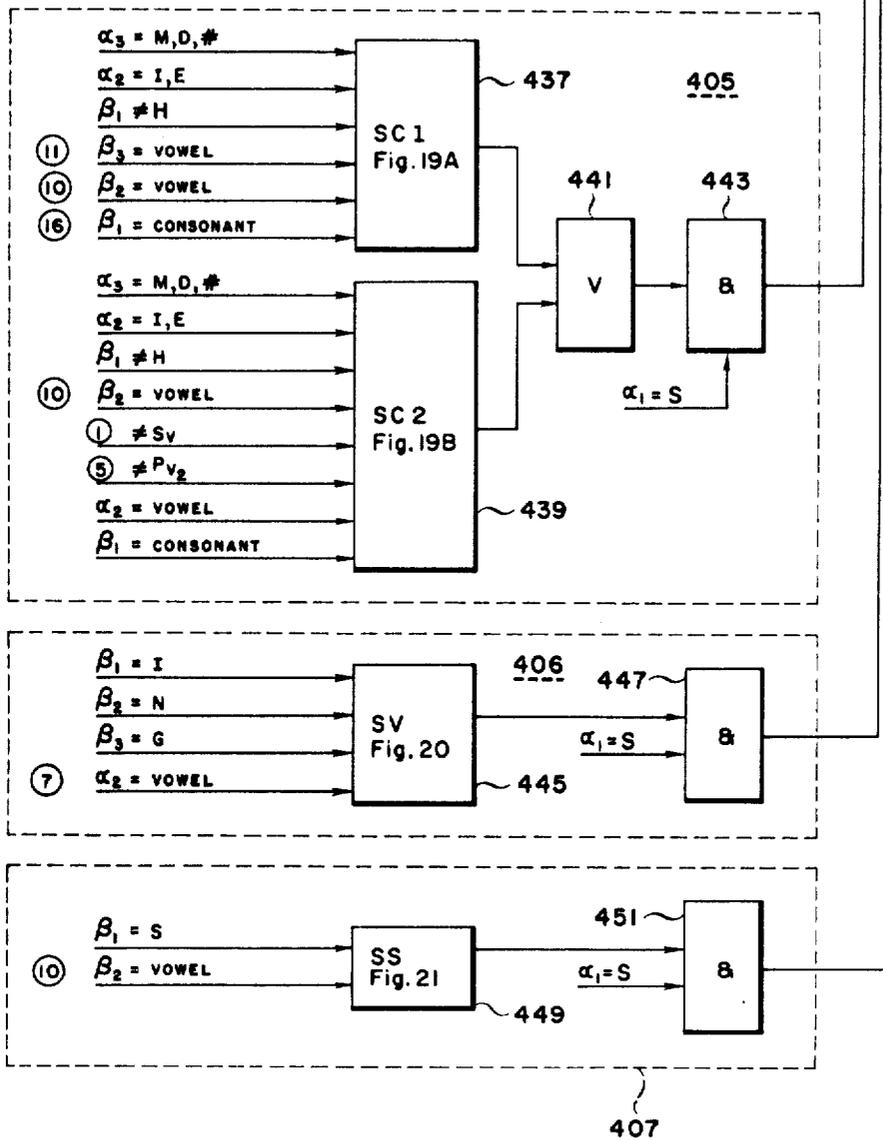


FIG. 4C

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY

*George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 11 of 26

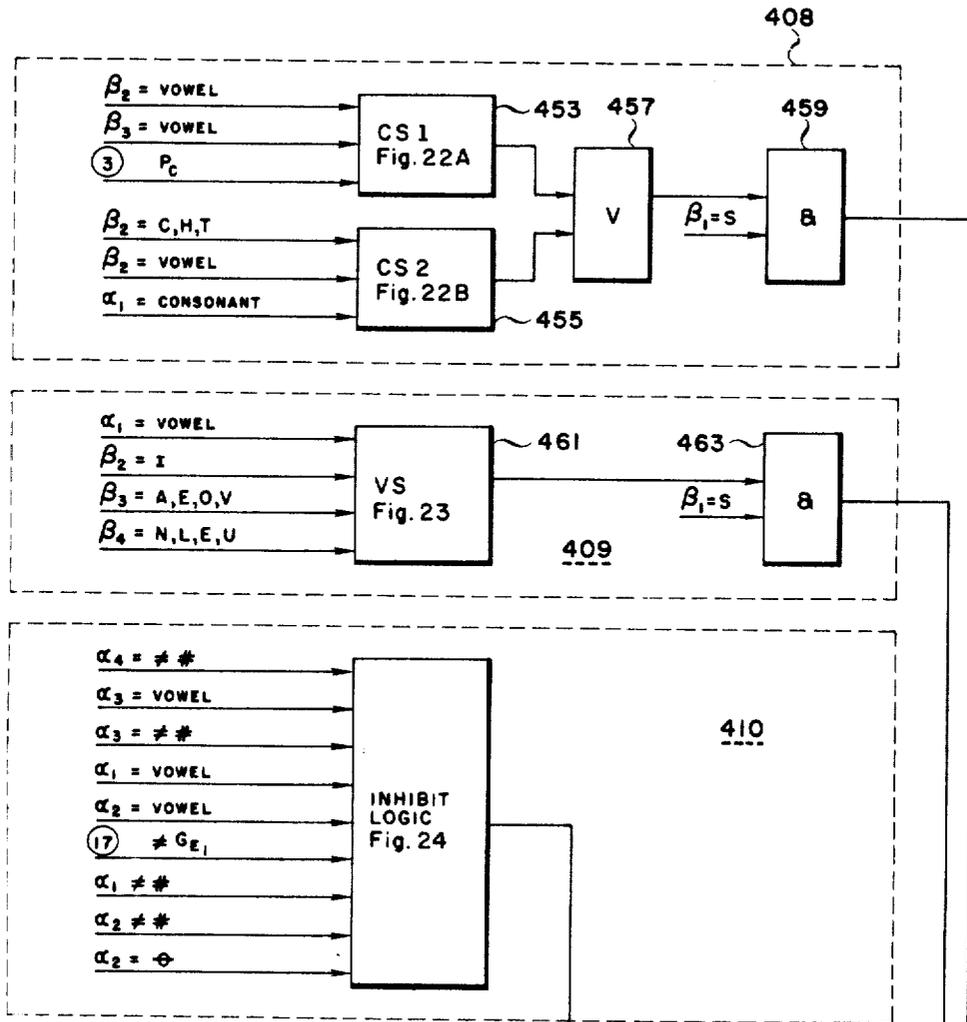


FIG. 4D

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY

*George Sullivan*  
Agent

TABLE I

TABLE OF CONSONANTS (C) — C = S + W + G <sub>b</sub> + G <sub>e</sub> (SEE TABLE II)		INTERNAL STATE OF FLIP-FLOPS OF CELLS IN SHIFT REGISTER							COMMENTS		
CHARACTER TYPES	CHARACTERS	WIDTH OF CHARACTER		X <sub>6</sub>	X <sub>5</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>		X <sub>1</sub>	X <sub>0</sub>
		LOWER CASE	UPPER CASE	UPPER AND LOWER CASE STATE							
STRONG CONSONANTS (S)	ƒ	3	4	1	0	0	0	1	0	1	The Consonant class (C) include Strong Consonants (S), Weak Consonants (W), Type "b" letter groups (G <sub>b</sub> ) and Type "e" letter groups (G <sub>e</sub> ). Table I includes S, W and G <sub>b</sub> groups which have the common code X <sub>5</sub> =1. This table and code excludes Type "e" letter groups (G <sub>e</sub> ) (see Table II), and the character "S" which involve separate logic networks.
	B	3	4	1	0	0	0	1	1	0	
	C	3	4	1	0	0	0	1	1	1	
	D	3	4	1	0	1	0	0	0	0	
	G	3	4	1	0	1	0	0	0	1	
	H	3	4	1	0	1	0	1	0	1	
	K	3	4	1	0	1	0	1	0	1	
	P	3	4	1	0	1	1	1	0	0	
	Q	3	4	1	0	1	1	1	0	1	
	V	3	4	1	0	1	1	1	1	0	
	X	3	4	1	0	1	1	1	1	1	
	Z	3	4	1	1	0	0	0	0	0	
	N	3	3	1	1	1	0	0	0	1	
	J	2	3	1	1	1	0	0	1	0	
	F	2	4	1	1	1	0	0	1	1	
	T	2	4	1	1	1	0	1	0	0	
WEAK CONSONANTS (W)	N	3	4	1	0	0	0	0	0	0	For the Weak Consonants (W) and Strong Consonants (S) the code represented by bits X <sub>0</sub> , X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub> , X <sub>4</sub> denote the particular character and width.
	R	3	4	1	0	0	0	0	1	1	
	W	4	5	1	0	0	0	1	0	0	
	M	5	5	1	0	0	0	1	1	1	
	L	2	4	1	0	0	1	0	1	0	
TYPE "b" LETTER GROUPS (G <sub>b</sub> )	G <sub>b</sub>	5		1	1	0	1	0	1	1	For the Type "b" letter group (G <sub>b</sub> ) the code represented by bits X <sub>0</sub> , X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub> , X <sub>4</sub> are generated by sequence encoding-decoding system (10) and represent only the width of letter groups (G <sub>b</sub> ) taking into account the upper and lower case condition. See Table of widths (Fig. 5C).
	G <sub>b</sub>	6		1	1	0	1	1	0	0	
	G <sub>b</sub>	7		1	1	0	1	1	1	1	
	G <sub>b</sub>	8		1	1	1	0	0	0	0	
	G <sub>b</sub>	9		1	1	1	1	0	0	1	

FIG. 5A

INVENTORS

JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Chellman*  
Agent

TABLE II

CHARACTER TYPES	CHAR-ACTERS	WIDTH OF CHARACTER LOWER UPPER CASE	INTERNAL STATE OF FLIP-FLOPS OF CELLS IN SHIFT REGISTER							COMMENTS
			X <sub>6</sub>	X <sub>5</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	
VOWELS (V)	A	3 4	0	0	0	0	0	0	0	The vowels in Vowel class (V) have the common code X <sub>3</sub> =0, X <sub>4</sub> =0 and X <sub>5</sub> =0. The code represented by bits X <sub>0</sub> , X <sub>1</sub> and X <sub>2</sub> denote the particular vowel. These codes are generated by character encoder (15) and are transmitted through system (10) to shift register (43).
	E	3 4	0	0	0	0	0	0	0	
	φ	3 4	0	0	0	0	0	0	0	
	U	3 4	0	0	0	0	0	0	0	
	Y	3 4	0	0	0	0	0	0	0	
	I	2 2	0	0	0	0	0	0	0	
SPACE OR PUNCTUATION (#)	SPACE (S)	2	0	1	0	0	0	0	0	This Space and Punctuation class (#) have the common code X <sub>3</sub> =0, X <sub>4</sub> =1 and X <sub>5</sub> =0. The code represented by bits X, X and X denote the width and type of character (except for groups P <sub>2</sub> , P <sub>3</sub> and P <sub>4</sub> ). These codes are generated by character encoder (15) and are transmitted through system (10) to shift register (43).
	'('):= P <sub>2</sub>	2	0	1	0	0	0	0	0	
	HYPHEN (H)	3	0	1	0	0	0	0	0	
	/_#X <sub>2</sub> P <sub>3</sub>	3	0	1	0	0	0	0	0	
	@%&=P <sub>4</sub>	4	0	1	0	0	0	0	0	
	TAB (T)	8	0	1	0	0	0	0	0	
	END OF PARA:(E/P) THIN SPACE	0	0	1	0	0	0	0	0	
TYPE "e" LETTER GROUPS (G <sub>e</sub> )	G <sub>e</sub> (H)	6	0	1	1	0	0	0	0	The Type "e" letter groups (G <sub>e</sub> ) have the common code X <sub>3</sub> =1, X <sub>4</sub> =1 and X <sub>5</sub> =0. The code represented by bits X <sub>0</sub> , X <sub>1</sub> and X <sub>2</sub> are generated by sequence encoding-decoding system (10) and represent the width of letter groups (G <sub>e</sub> ) taking into account the upper and lower case conditions. See Table of Widths (Fig.5C).
	G <sub>e</sub> (H)	7	0	1	1	0	0	0	0	
	G <sub>e</sub> (H)	8	0	1	1	0	0	0	0	
	G <sub>e</sub> (H)	9	0	1	1	0	0	0	0	
	G <sub>e</sub> (H)	10	0	1	1	0	0	0	0	
	G <sub>e</sub> (H)	11	0	1	1	0	0	0	0	
(S)	S	3 3	0	0	0	1	0	0	0	"S" is used in a special set of rules and is not considered either as a consonant or vowel. This code is generated by character encoder (15) and is transmitted through system (10) to shift register (43).

FIG-5B

INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George J. Sullivan*  
Agent



April 15, 1969

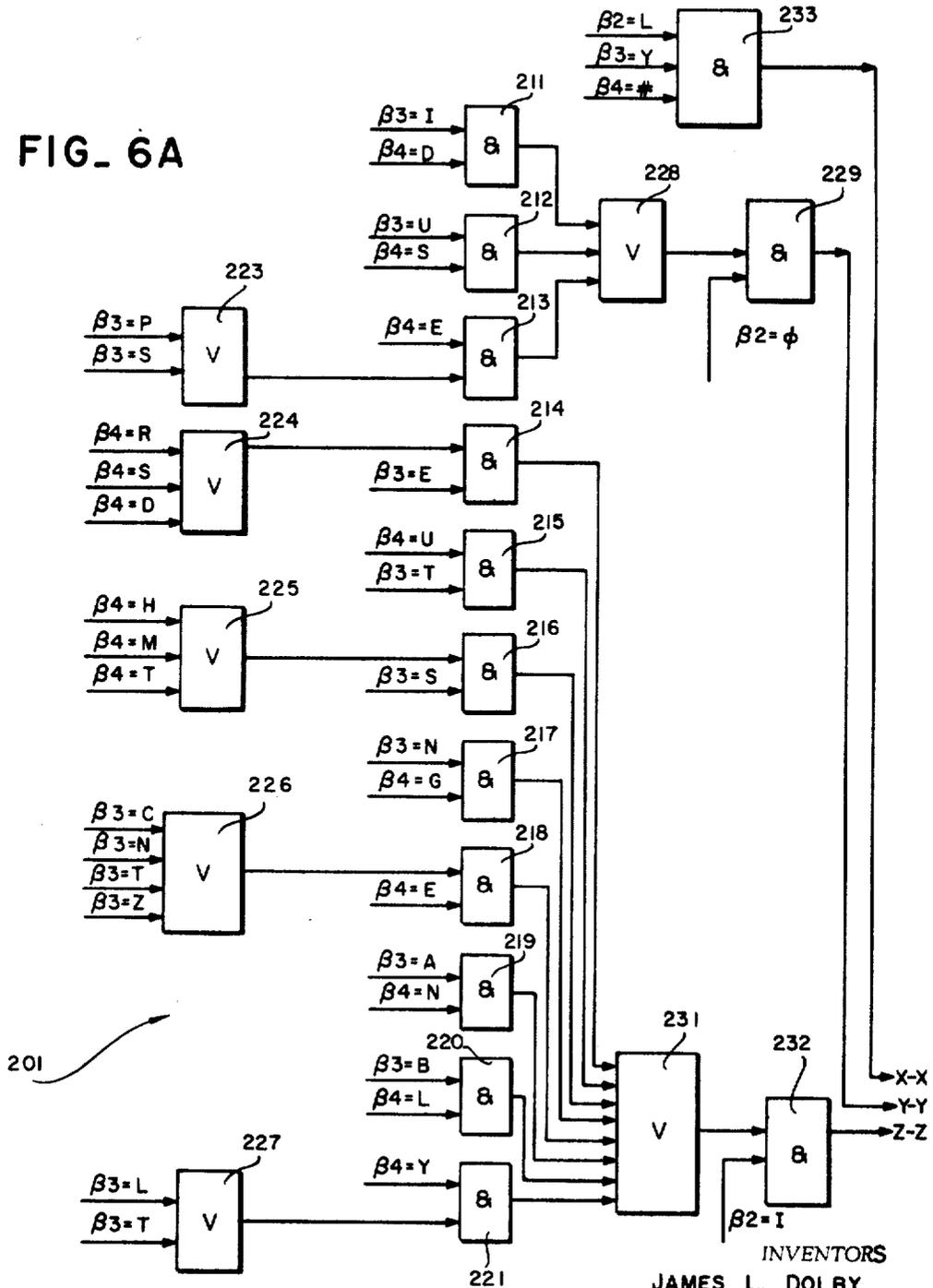
J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 15 of 26

FIG. 6A



INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *Henry Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL

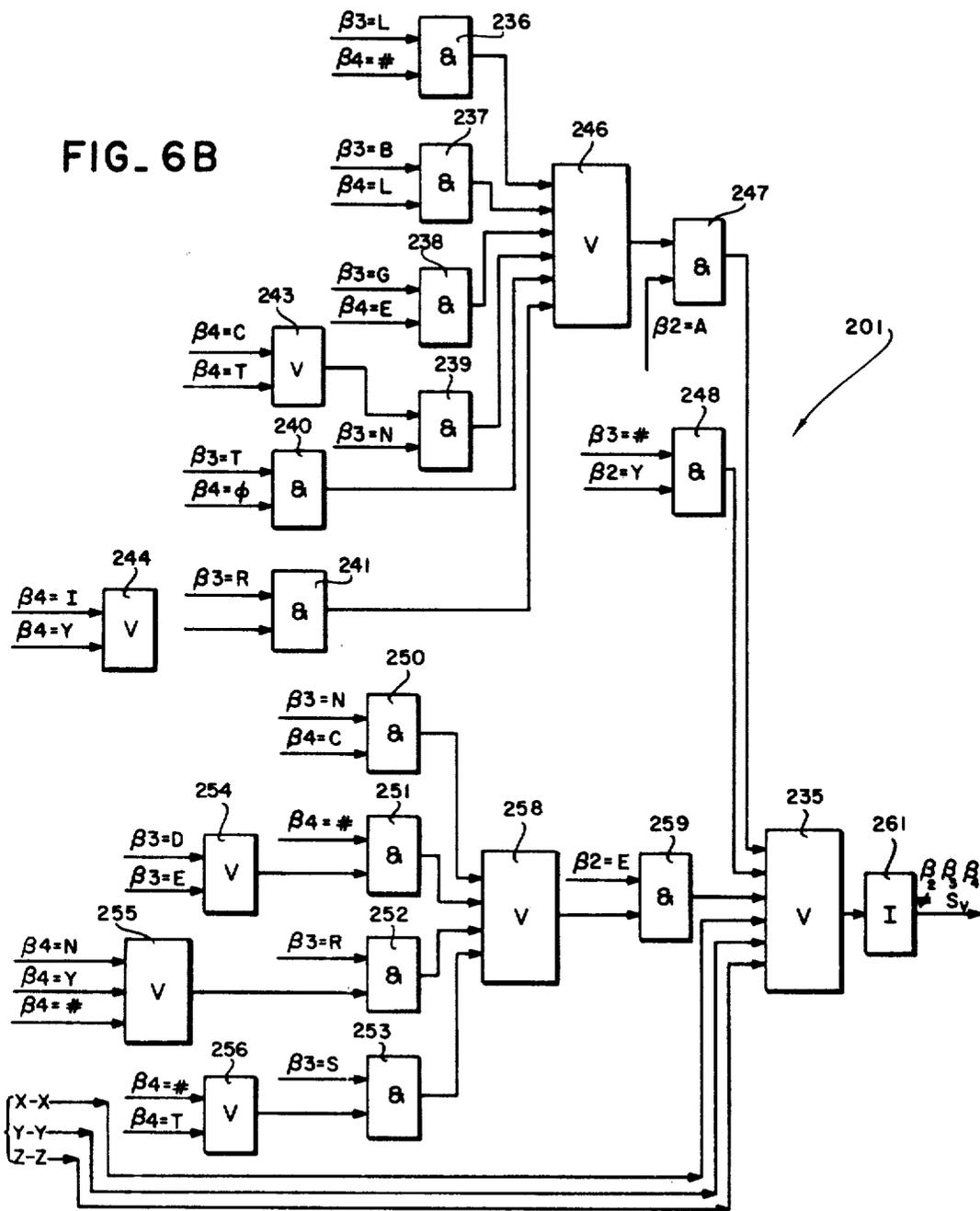
3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 16 of 26

FIG. 6B



INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 17 of 26

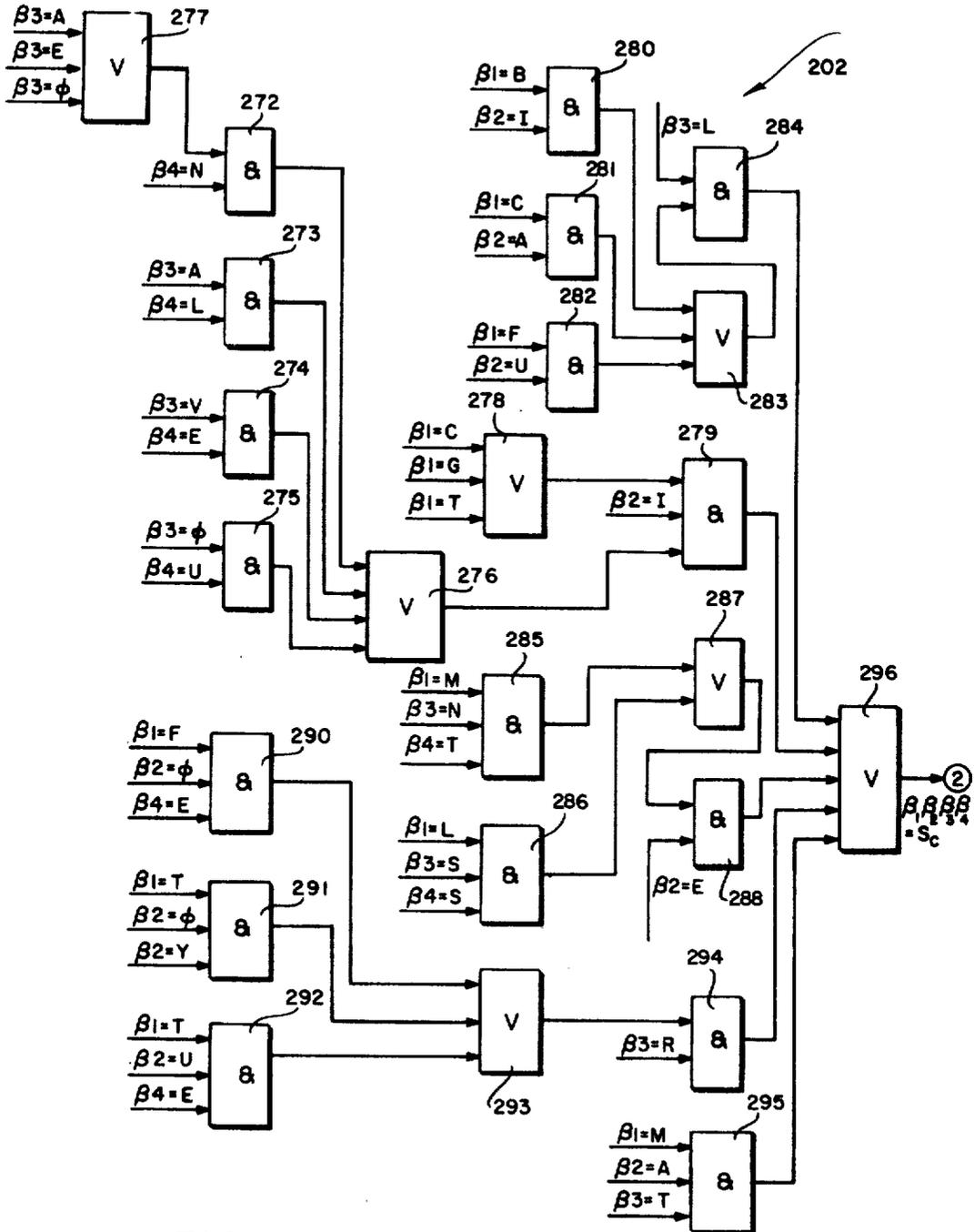


FIG. 7

INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

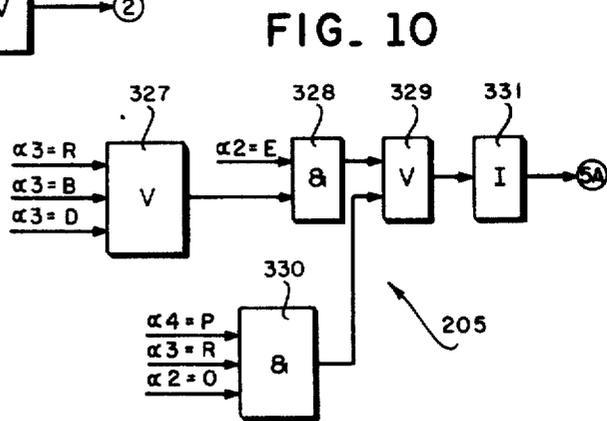
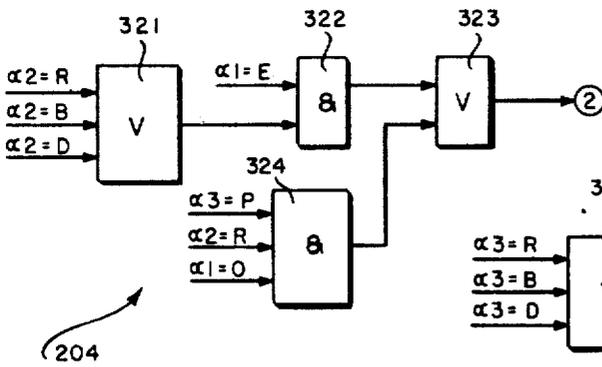
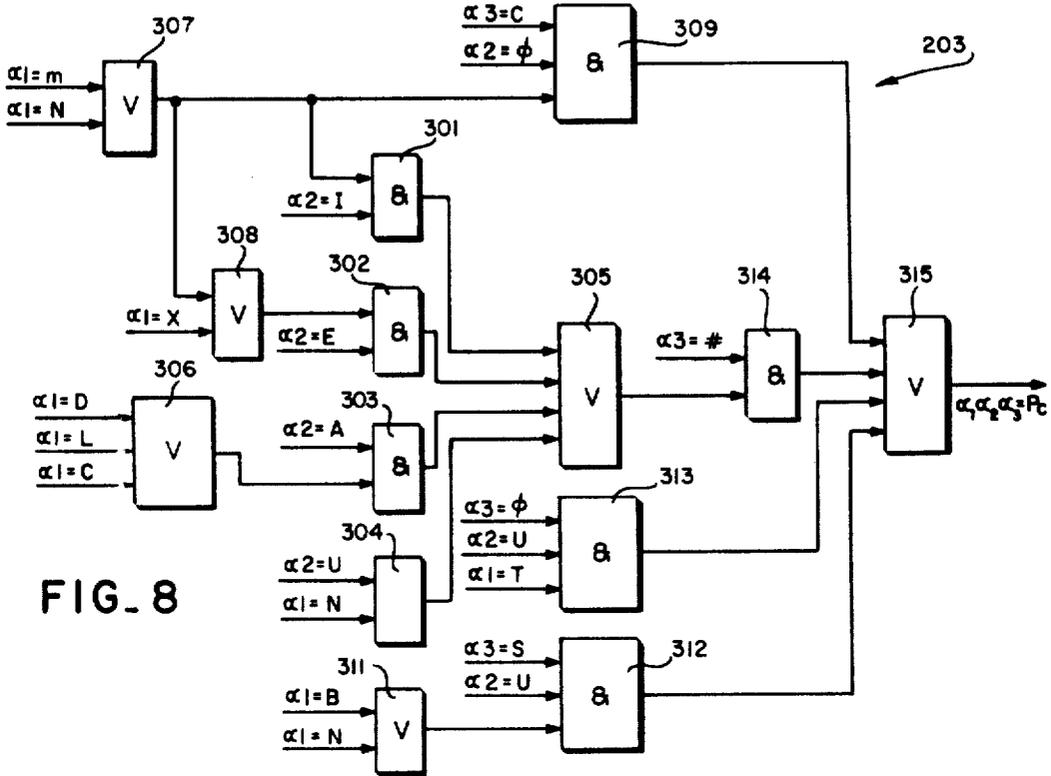
J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 18 of 26



INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *[Signature]*  
Agent

FIG. 12

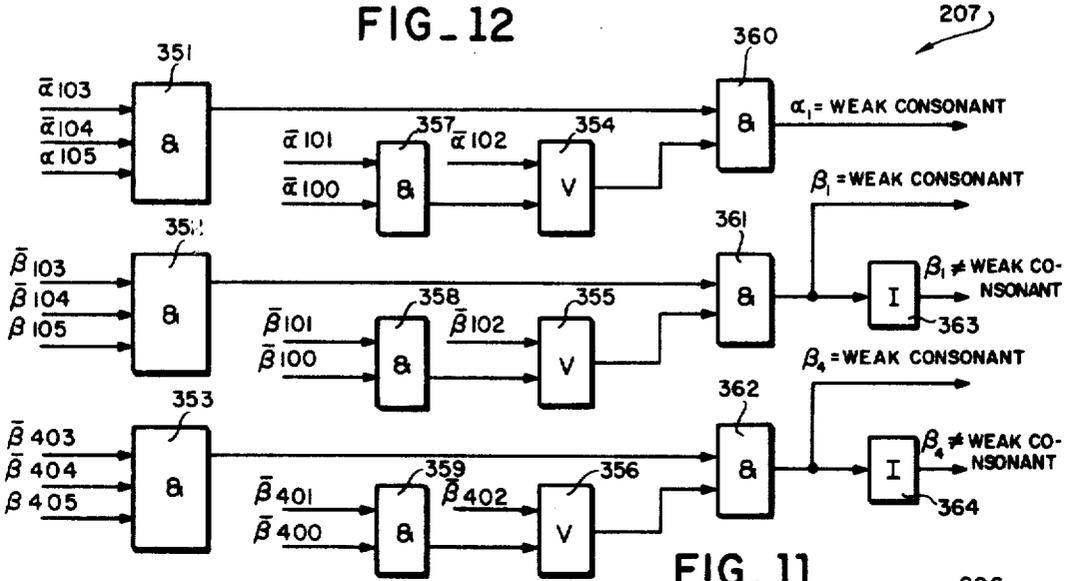


FIG. 11

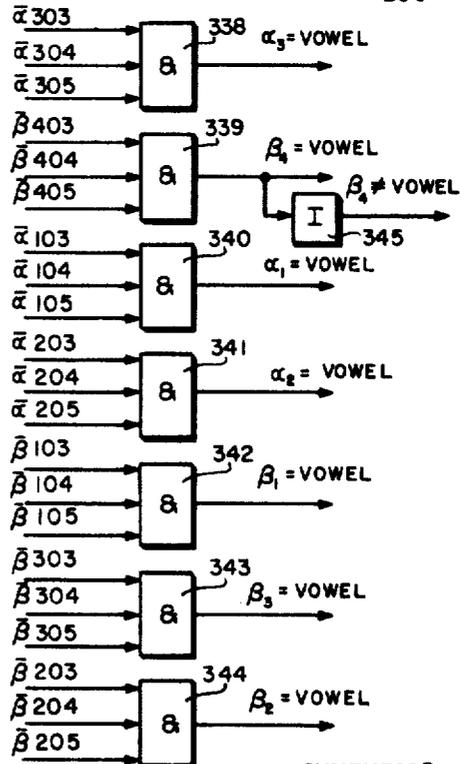


FIG. 14

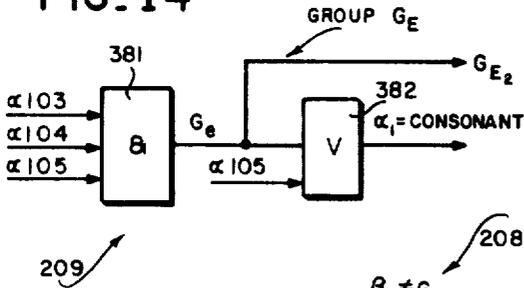
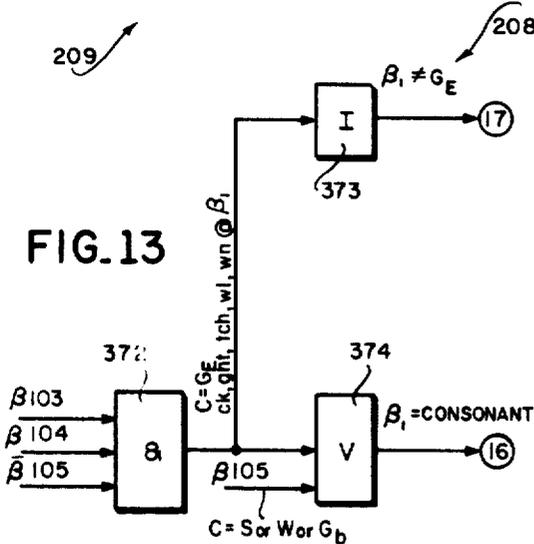


FIG. 13



INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

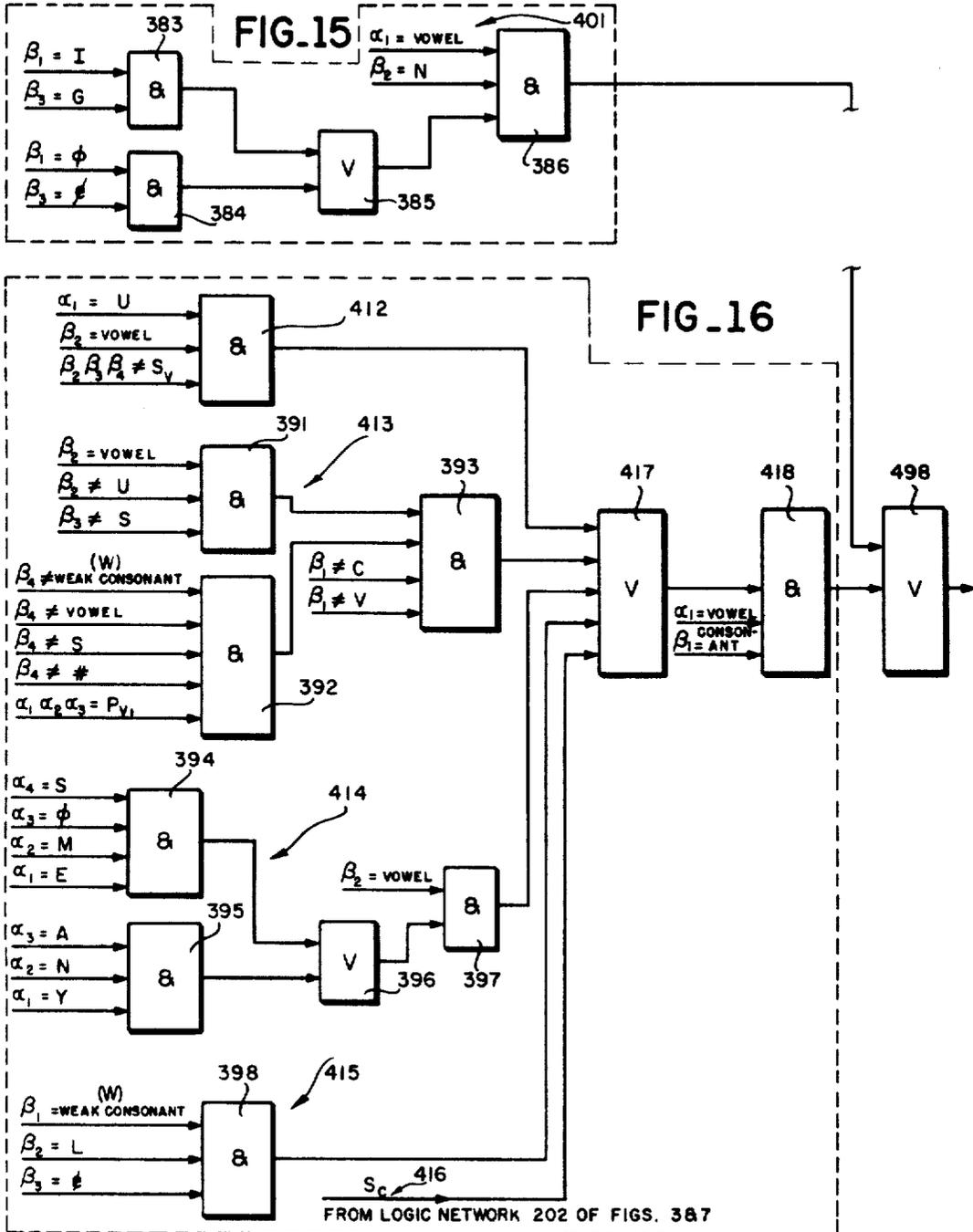
J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 20 of 26



INVENTORS

JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

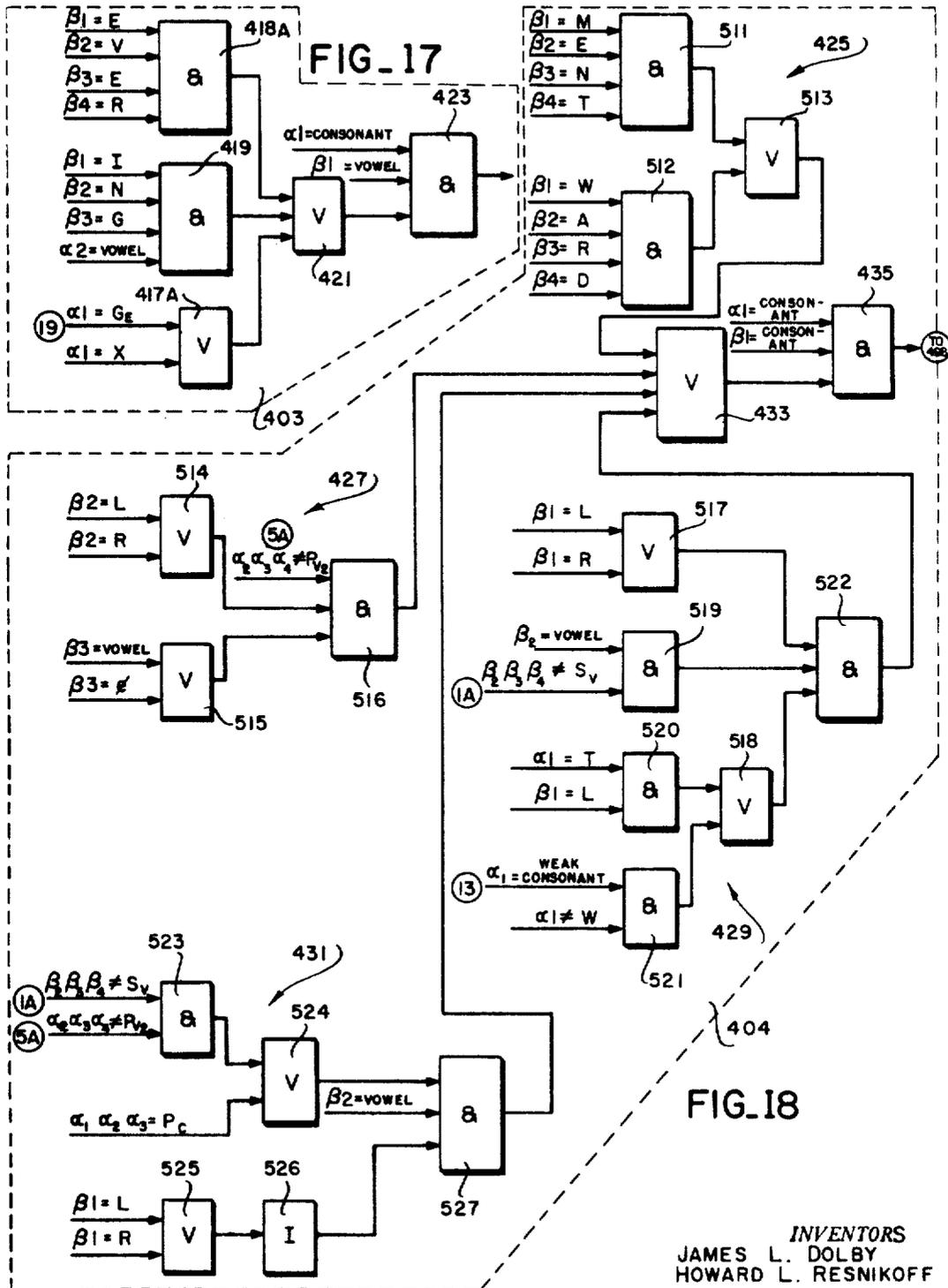
J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 21 of 26



INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 22 of 26

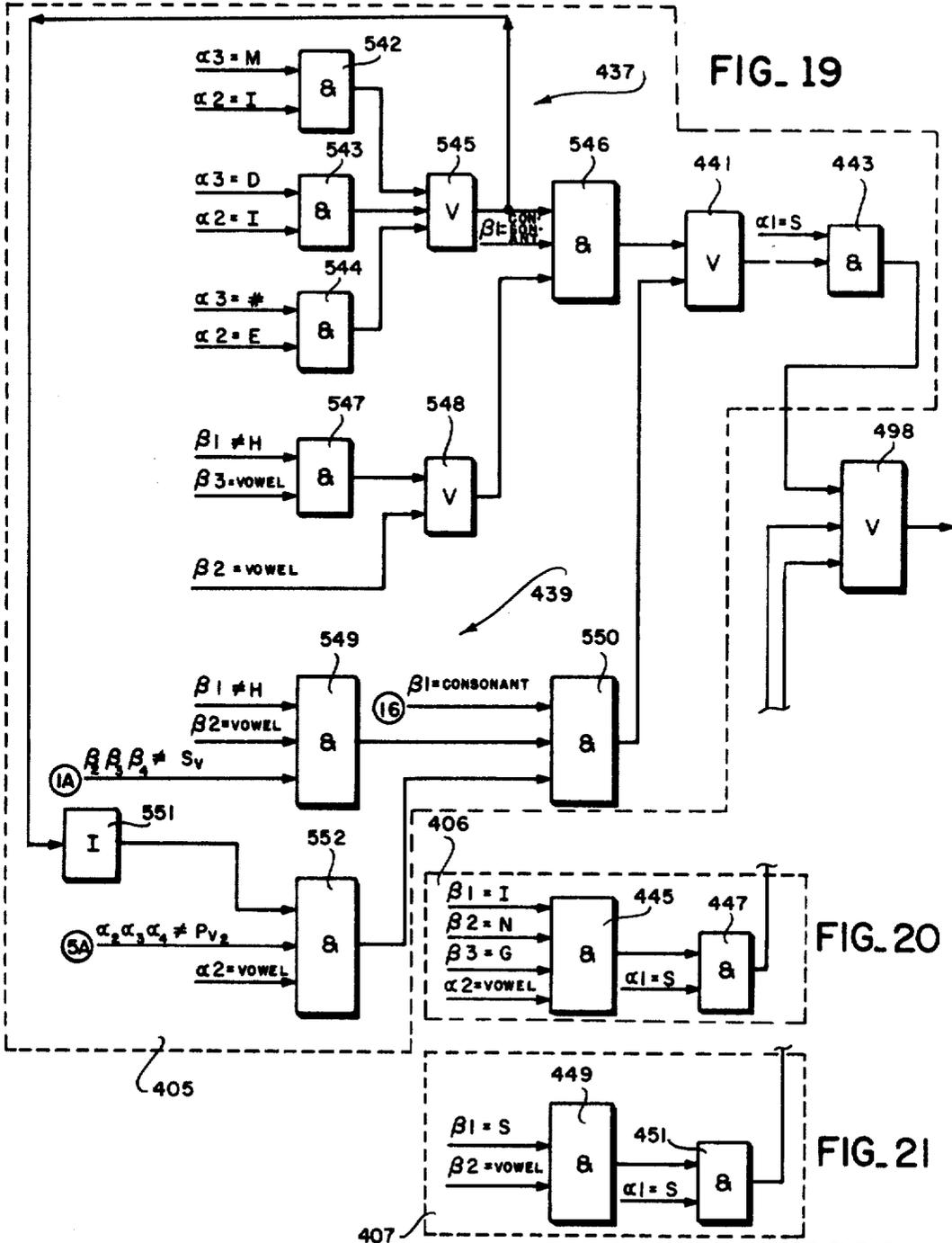


FIG. 19

FIG. 20

FIG. 21

INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

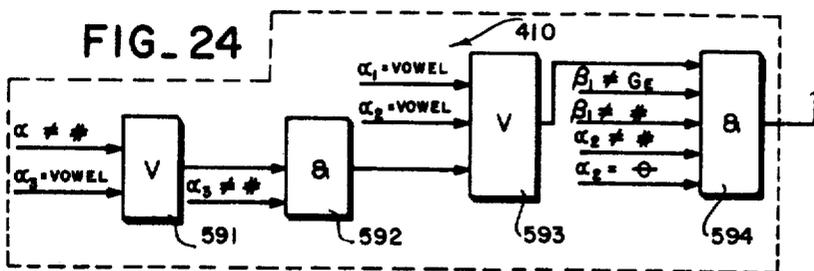
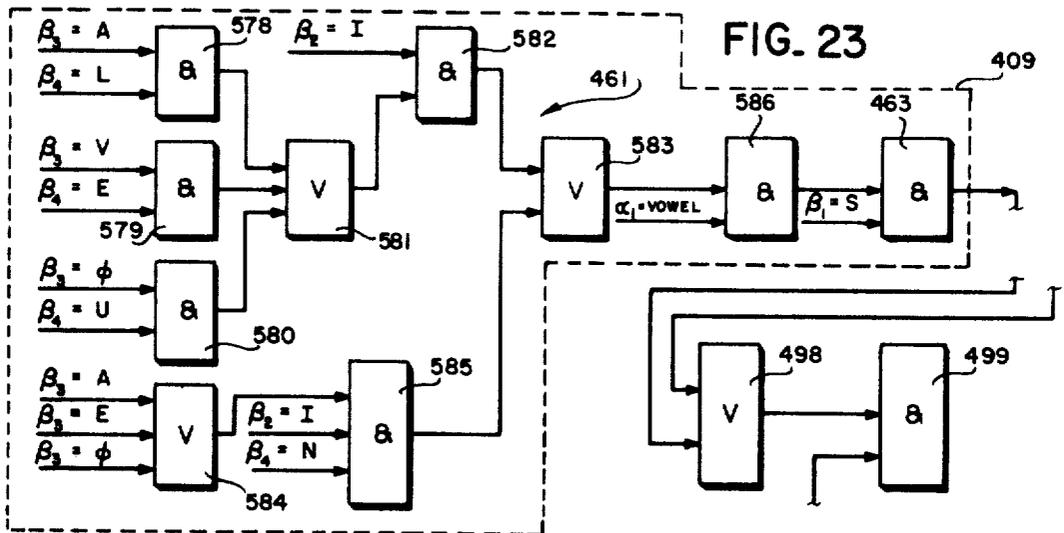
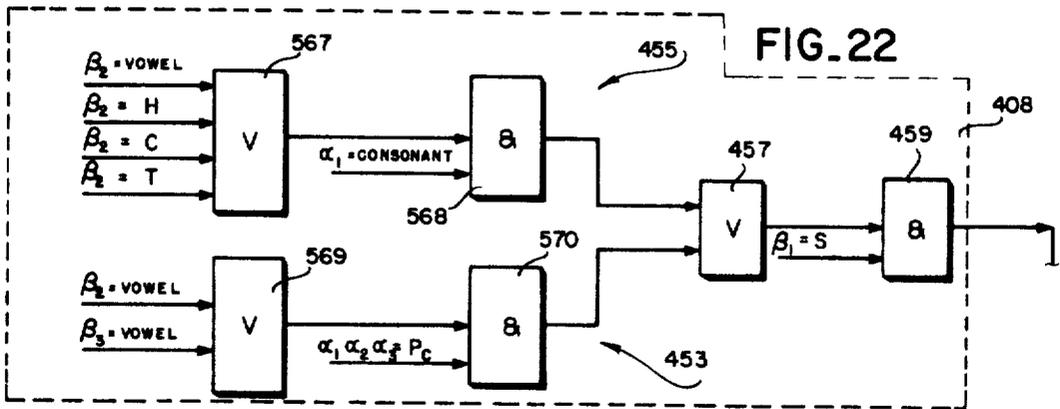
J. L. DOLBY ET AL

3,439,341

HYPHENATION MACHINE

Filed Aug. 9, 1965

Sheet 23 of 26



INVENTORS  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF

By *George Sullivan*  
Agent

April 15, 1969

J. L. DOLBY ET AL  
HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 24 of 26

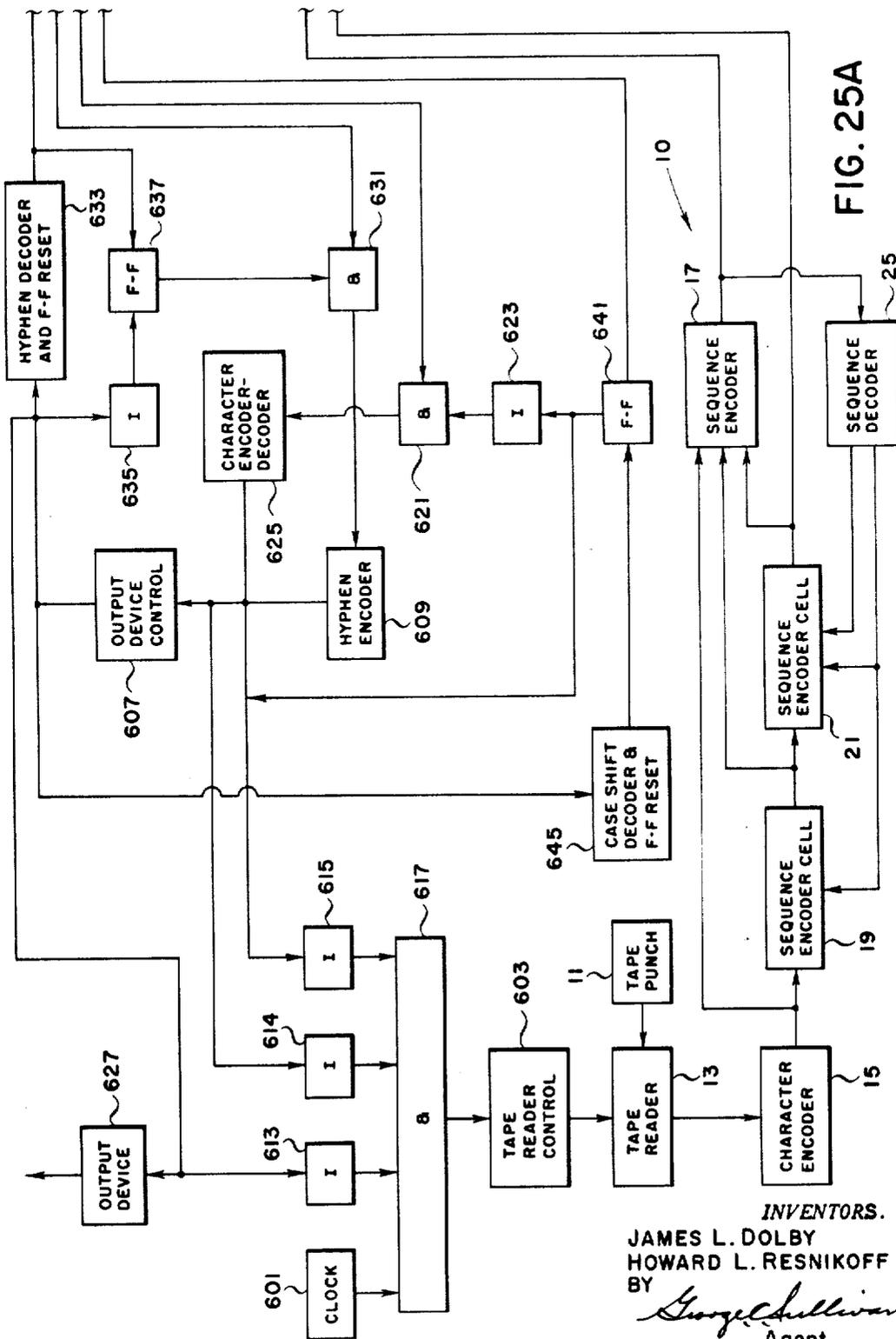


FIG. 25A

INVENTORS.  
JAMES L. DOLBY  
HOWARD L. RESNIKOFF  
BY *George Sullivan*  
Agent



April 15, 1969

J. L. DOLBY ET AL  
 HYPHENATION MACHINE

3,439,341

Filed Aug. 9, 1965

Sheet 26 of 26

TABLE IV

TABLE OF LETTER STRINGS FOR HYPHENATION DEVICE OF FIG. 25									
CHARACTER GROUP	LETTER STRINGS		X <sub>6</sub>	X <sub>5</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>
G <sub>b</sub>	CH		UPPER & LOWER CASE	1	1	0	1	0	1
G <sub>b</sub>	PH			1	1	0	1	1	0
G <sub>b</sub>	TH			1	1	0	1	1	1
G <sub>b</sub>	GH			1	1	1	0	0	0
G <sub>b</sub>	WH			1	1	1	0	0	1
G <sub>b</sub>	DG			1	1	1	0	1	0
G <sub>b</sub>	QU			1	1	1	0	1	1
G <sub>e</sub>	WL		UPPER & LOWER CASE	0	1	1	0	0	0
G <sub>e</sub>	WN			0	1	1	0	0	1
G <sub>e</sub>	CK			0	1	1	0	1	0
G <sub>e</sub>	GHT			0	1	1	0	1	1
G <sub>e</sub>	TCH			0	1	1	1	0	0

FIG. 26

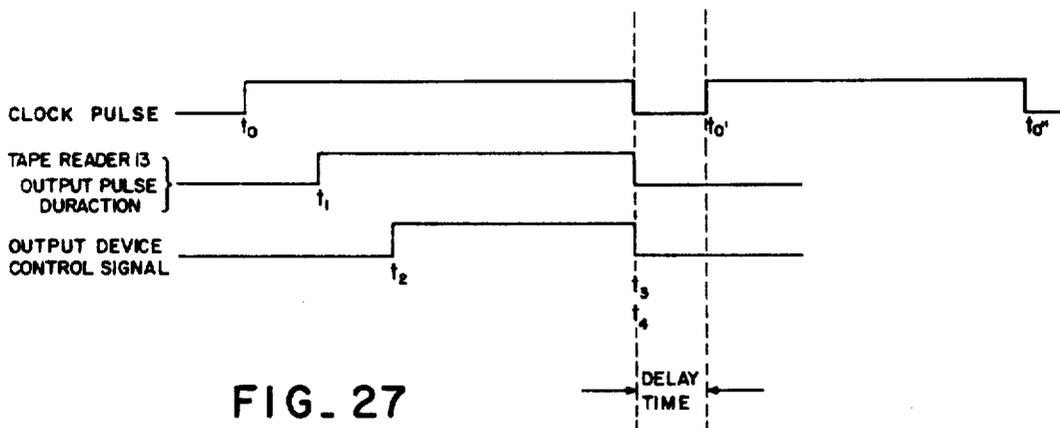


FIG. 27

INVENTORS  
 JAMES L. DOLBY  
 HOWARD L. RESNIKOFF

By *George C. Sullivan*  
 Agent

1

3,439,341

**HYPHENATION MACHINE**

James L. Dolby, Los Altos, Calif., and Howard L. Resnikoff, Princeton, N.J., assignors to Lockheed Aircraft Corporation, Burbank, Calif.

Filed Aug. 9, 1965, Ser. No. 478,127  
Int. Cl. G06f 7/08, 15/20; G06k 9/10

U.S. Cl. 340—172.5

6 Claims

**ABSTRACT OF THE DISCLOSURE**

A digital device for hyphenating and justifying original textual composition into a second form of textual composition having lines of pre-selected number of words per line which is accomplished by the examination of a string of alpha, numeric and other punctuational characters taken from the original textual composition to determine whether the characters being examined define one or more of the conditions necessary for splitting, hyphenating and/or punctuation by the device. The device contains a plurality of circuits which may be termed syllabification logic circuits which define logic equations and conditions which, if satisfied by a string of characters under examination from the textual composition, will be hyphenated, justified and/or otherwise punctuated. The system provides a means for automatically determining whether it is gramatically possible to syllabify a given string of characters without the assistance of manual aids.

The present invention relates to automatic word splitting and justifying equipment used in the composition of text material for printing and more particularly to a unique technique for determining the split points of words in accordance with those established by accepted authorities.

In the prior art several common approaches have been employed to solve the problem of word splitting. The oldest and still most widely used approach is to have the operator of the typesetting equipment indicate the proper position for a word split by actually typing a hyphen at the proper point in the text. To facilitate this process, an indicator is provided to indicate to the operator when a sufficient amount of material has been prepared so that the automatic justification device of the machine can fill the line with spaces (or "lead"). The indicator also informs the operator how much room is left in the line if the line is to fit within the desired length. A more recent variant of this procedure permits the operator to prepare the entire text without regard to the word splitting problem but then requires another operator to intervene whenever the subsequent justification process cannot meet stipulated requirements without splitting a word.

Another approach to the problem has been provided by incorporating a general purpose electronic computer as part of the composing room machinery so that a fairly extensive set of logic rules and/or a fairly extensive dictionary can be made a part of the process. Users of this technique have generally found it necessary to employ a final human proofing operation to eliminate grammatical errors introduced by the word splitting process.

A further approach to the problem has been to build in the simplest possible set of word splitting rules, thus eliminating the need for a general purpose computer, and to train the operator to anticipate the errors that such a

2

device is likely to make so as to eliminate them by a special code before they occur. It is estimated that such a set of rules leads to errors in about 25% of the cases encountered in typical text material. It should be noted that the operator of this third type of machine, although he is relieved from the necessity of making 75% of the splits, must be more extensively trained than the operator of the first type since he must know the vagaries of the machine logic and also the proper word splitting rules to operate the machine efficiently.

Accordingly, a principal object of the present invention is to provide a machine which eliminates or minimizes the shortcomings of prior art to solving the problems attendant with word splitting and justifying machines and approaches thereto.

Another object of the present invention is to provide a machine which automatically performs a number of composing operations heretofore carried out manually and mentally.

Still another object of the present invention is to provide a device which eliminates sources of human error by providing a fully automatic computerized composing machine to substantially reduce the cost of preparing printed text.

Yet another object of the present invention is to provide a device for examining each character of a textual composition to determine where each word of the text should be split.

Still a further object of the present invention is to provide a device which examines a string of characters to determine between what two adjacent characters the string may be split.

Yet a further object of the present invention is to provide a machine which automatically justifies and hyphenates a string of characters of a textual composition.

While another object of the present invention is to provide a device which performs a number of composing operations owing to the presence of logic rules incorporated into the functions of the device.

While a further object of the present invention is to provide a method wherein logical design equations are utilized for word splitting of textual compositions presented as printed text.

In accordance with the present invention, the foregoing objects are achieved through the use of a machine which has a plurality of built in logical equations and rules applicable to word splitting and line justifying to thereby eliminate the need for an operator trained in the mechanics of word splitting. In fact, the only useful function the operator may be required to perform is to hyphenate extremely lengthy compound words. This would tend to reduce further the possible error rate of the system.

The present invention does not require the use of a general purpose computer or even a comparable amount of logic or stored memory capability. Hence it can be manufactured at cost comparable to devices that have little or no word splitting capability but operate at the same speed. In addition, the present invention has a relatively low error rate, on the order of 1 to 2% of the words actually split, so that the amount of proof reading necessary to correct spelling and other operator errors is substantially eliminated. It has been found that in typical applications, less than one in four lines of type end in a split word and therefore the number of corrections necessary is estimated to be less than one per 250 lines of output.

The means of determining whether it is grammatically possible to syllabify a given string of letters at a given point is based upon certain properties of the theory of syllabification derived in accordance with the present invention. It has been discovered that there are certain character sequences that may always be split at a given point regardless of the sequences that surround the given sequence. Thus, by way of example, the sequence "ghtf" may always be properly split as "ght-f" regardless of the surrounding characters, as in the case of the word "straight-forward." On the other hand, sequences exist that may only be split properly if considerable information is known about the grammatical surroundings. Thus, by way of example, the word "produce" may be properly split as "prod-uce" if it is used as a verb. Consequently, an automatic word splitting device, if it is to agree with standard or acceptable grammatical usage as indicated by standard dictionaries and style manuals, must not only be capable of splitting letter strings, but must also be capable of resolving important syntactic ambiguities. Hence not even a complete store of the information in a dictionary is sufficient to resolve all word breaking situations.

To continue, it has been also discovered that it is possible to determine whether a particular character string can be split unambiguously without recourse to larger context. Furthermore, it has been discovered that trained compositors do not in practice operate to minimize the amount of space inserted in text. Experience with the logic used in the present invention shows that both the proportion of lines ending in split words and the amount of space introduced per line agree substantially with what is generally accomplished by expert compositors. It has also been established in accordance with the present invention, that for text of material ranging in line length from that of editorial page composition to book composition, the maximum space introduced by the logic herein described tends to be slightly less than that allowed by good composition practice.

Generally, the logic herein described provides for grammatically proper word splitting. However, it is important to note that authorities differ on certain questions in this area. For instance, some dictionaries indicated that the word "peculiar" should be split "pecu-liar" (as does the present system) while others prefer "pecul-iar." Further, it is possible to confound almost any logical scheme with material that compounds many words without the use of hyphens. Thus, the performance of the device of the present invention will generally be enhanced if the operator is instructed to minimize the use of word compounds without hyphens.

The novel features which are believed to be characteristic of the invention both as to its organization and method of construction and operation, together with further objects and advantages thereof, will be better understood from the following description considered in connection with the accompanying drawings in which illustrative embodiments of the invention are disclosed by way of example. It is expressly understood, however, that the drawings are for the purpose of illustration and description only and do not define limitation of the invention.

In the drawings:

FIGURE 1 is a block diagram of the over-all hyphenating-justifying system of the present invention;

FIGURES 2A, 2B, 2C and 2D illustrate in greater detail the system generally shown in FIGURE 1;

FIGURES 3A and 3B and 6 through 14 are detailed illustrations of logic circuits for the class decoder shown in FIGURES 1 and 2;

FIGURES 4A-4D and 15 through 24 are detailed illustrations of logic circuits for the hyphenation devices shown in FIGURES 1 and 2;

FIGURES 5A, 5B and 5C are tables which illustrate the type of information obtained from sequence encoder shown in FIGURE 2A;

FIGURES 25A and 25B are another embodiment of

the present invention which provides a continuous hyphenated output;

FIGURE 26 is a table of letter strings for the hyphenation device shown in FIGURE 25; and

FIGURE 27 is a timing diagram illustrating the timing signals of the system shown in FIGURE 25.

With reference now to the drawings, there is illustrated in FIGURE 1 a block diagram of the over-all hyphenating-justifying system of the present invention. The input to the system is illustrated as tape punch 11 which in practice would typically consist of an operator punching binary coded paper tape from written text material. This coded tape is then conveyed to tape reader 13 the function of which is to convert the tape information into electrical information which, in accordance with the present invention, is transmitted to the character encoder 15 during the read-in period and to output device 117 during the read-out period. The control of the tape reader is achieved by means of tape reader control 79 which functions in the hereinafter-described manner.

The read-in period is herein defined as that period when a complete line is read by tape reader 13 and transferred to the system.

The beginning of the line occurs when the first character of that line is read by reader 13 and the end occurs when the last character has been read by reader 13. As will hereinafter become apparent, the last character read by reader 13 may or may not be the last character of the hyphenated and justified line printed during readout. Character encoder 15 converts the output of tape reader 13 into binary code signals. In the present invention each character is represented by a six-bit code with a seventh bit denoting whether the character is in the upper or lower case which is important for determining the width of characters. It will be appreciated by those skilled in the art that codes having a different number of bits may be employed and the primary reason for using character encoder 15 is that the output code of commercially available tape readers is frequently in a code format that is not compatible with some of the specific requirements of the present invention.

The function of character counter 105 is to count the number of characters read into sequence decoding-encoding system 10 and will continue counting until tape reader control 79 provides an output signal that inhibits further read-in operation of tape reader 13. Tape reader control 79 will receive a signal from line length counting system 68, the operation of which will hereinafter be described, denoting the end of line condition. During the read-in period, character counter 105 adds the number of characters transmitted to system 10 and during the back-up period, character counter 105 sequentially counts down to zero the number of characters which have been added thereby returning the tape in tape reader 13 to its original position. In this connection it should be noted that the tape reader is characterized as a reversible tape reader.

It should be further noted that the illustrated tape input is only for purposes of illustration and is not to be construed as a limitation of the invention.

The function of sequence decoding-encoding system 10 is to group letter sequences in a manner which will hereinafter be described for purposes of satisfying unique hyphenation logic as well as for simplifying the logic required in the various networks. During typical operation most of the characters generated by character encoder 15 will be transmitted directly through system 10 to shift register 43; however, there will be certain letter sequences where system 10 will combine these sequences into a single code which is then transmitted to shift register 43.

Shift register 43 consists of 8 cells ( $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$ ) each of which will store a single character. Cells  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  each consist of seven flip-flops, where the seventh provides upper and lower case information. The information contained in cell  $\alpha 1$  is transmitted to line length counting system 68

for determining the width of the last character investigated in the line. It should be noted that the characters move into the shift register from left to right (from  $\beta 4$  to  $\alpha 4$ ) and the word "PRIVATE" would appear as illustrated. Cell  $\beta 4$  contains a symbol denoting a space between words. When the next new character is shifted into the shift register, the new character will replace the space (#) in cell  $\beta 4$  and all characters are shifted one cell to the right with the letter "P" being replaced by the letter "R" in cell  $\alpha 4$ . The letter "P" is then "lost" from the logic of the system; however, it is still retained on the tape in tape reader 13 for readout. The point of investigation of a character string is between the  $\alpha 1$  and  $\beta 1$  cells (AV) and to determine when there is a proper split point between these cells, the information in cells  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  is analyzed by hyphenation logic device 65 which is of significant essence to the present invention. Class decoder 63 also examines the information in cells  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  which then transmits this information to hyphenation logic device 65.

The information in each of the cells is decoded by character decoders (47-55) the output of which are illustrated as a single line but may be considered as 8 lines, one for each character decoder.

In practice there may be several hundred lines which connect the eight character decoders to class decoder 63, hyphenation logic device 65 and line ending logic 67. The output of character decoders 47 through 55 is in single bit form rather than in multiple bit form which makes it possible to simplify the logic of the present invention.

Class decoder 63 not only receives information from character decoders 47 through 55 but also from each of the cells of shift register 43. The information received by class decoder 63 from character decoders 47 through 55 is used to determine the following:

- (1) Vowel beginning suffixes ( $S_v$ )
- (2) Consonant beginning suffixes ( $S_c$ )
- (3) Consonant ending prefixes ( $P_c$ )
- (4) Vowel ending prefixes ( $P_{v1}$ ) and
- (5) Vowel ending prefixes ( $P_{v2}$ ).

The information received by class decoder 63 directly from the shift register cells is used to determine the following:

- (1) Vowel logic (V)
- (2) Weak consonant logic (W)
- (3) Strong consonants (S)
- (4) Type "b" letter groups ( $G_b$ ) and
- (5) Type "e" letter groups ( $G_e$ )
- (6) Space or punctuation (#)
- (7) Special "S" rule (S)

The foregoing is set forth in Tables I and II of FIGURES 5A and 5B. This information, which is decoded by class decoder 63, is converted to single bit information which is then used in hyphenation logic device 65 for purposes of hyphenating words in accordance with the present invention.

Hyphenation logic device 65 is a digital device which will provide an output signal when a word may be split between the  $\alpha 1$  and  $\beta 1$  cells of shift register 43. This is achieved by several logic networks the details of which will hereinafter be described and which will be generally referred to as follows:

- (1) Vowel-vowel logic network (VV-1)
- (2) Vowel-consonant logic network (VC-1 to 5)
- (3) Consonant-vowel logic network (CV-1 to 3)
- (4) Consonant-consonant logic network (CC-1 to 4)
- (5) "S" consonant logic network (SC-1 and 2)
- (6) "S" vowel logic network (SV)
- (7) "S-S" logic network (SS)
- (8) Consonant "S" logic network (CS-1 and 2)
- (9) Vowel "S" logic network (US)
- (10) Inhibit logic.

It will be particularly noted that the above-mentioned logic networks depend upon whether or not the  $\alpha 1$  or  $\beta 1$  positions are consonants, vowels or the letter "S." This basic information is used with information contained in cells  $\alpha 2$ ,  $\alpha 3$ ,  $\alpha 4$ ,  $\beta 2$ ,  $\beta 3$  and  $\beta 4$  of shift register 43. The amount of information required from these other cells will depend upon the characteristics of the particular logic network in question the details of which will be hereinafter considered.

Line ending logic device 67 receives its information directly from character decoders 47 through 55 and will provide an output signal under certain circumstances such as when  $\beta 1$  is a space and  $\alpha 1$  is not a space which would denote the end of a word or when  $\alpha 1$  is a code representing the end of paragraph, etc.

The outputs of hyphenation logic device 65 and line ending logic device 67 are transmitted both to line length counting system 68 and space counting system 92. Line length counting system 68, the details of which will be hereinafter described with respect to FIGURES 2A, 2B, 2C, and 2D, functions to provide an output signal to tape reader control 79 denoting when the end of line condition exists. Line length counting system 68 also provides a signal to space insertion logic 118. Space counting system 92 counts the spaces between words in the line counted by line length counting system 68 and conveys this information to space insertion logic 118. It will be particularly noted that a line must end at either the position denoted by line ending logic 67 or at the position denoted by hyphenation logic device 65. When the end of line condition is established by line length counting system 68 then tape reader control 79 provides an output signal to tape reader 13 which stops tape reader 13 and causes it to back up to its original position, as established by the number of characters counted by character counter 105. Tape reader 13 will now read-out and at each space between words space insertion logic 118 will provide information to output device 117 denoting the number of spaces to be added for proper justification. An output signal will be provided from line length counting system 68 denoting when the read-out operation should stop which will occur when the system has a zero count as will hereinafter be explained.

In view of the foregoing, it can be seen that the tape used by tape reader 13 serves as the memory for read-out of all characters processed during the read-in period. In summary, the complete cycle of operation is as follows:

- (1) Read-in—Read-in of a new line is started by step (3) and is stopped by line length counting system 68.
- (2) Back-up—Back-up is started by the "stop" signal from system 68 (step (1)). During read-in, character counter 105 adds the characters and back-up is stopped by character counter 105 when the number of characters counted during back-up equals the number of characters added during read-in. This is the beginning of the line read-in during step (1).

(3) Read-out—Read-out is started by the "stop" signal from counter 105 (step (2)). Read-out is stopped by line length counting system 68.

(4) Repeat steps (1), (2) and (3) above for the next line of printed output.

In FIGURES 2A, 2B, 2C and 2D is illustrated in greater detail the Hyphenator-Justifier Device generally shown in FIGURE 1. In order to initiate operation of the FIGURE 2 device, an operator transforms written information into a digital format by means of tape punch 11. The output of tape punch 11 is then transferred to tape reader 13 which converts this information into electrical signals having a predetermined number of bits which is transferred then to character encoder 15. The output of character encoder 15 is simultaneously applied to the inputs of sequence encoder 17 and sequence encoder cell 19. The output of sequence encoder cell 19 is simultaneously applied to sequence of encoder 17 and to the input of sequence encoder 21 the output of which is simultaneously

applied to sequence encoder 17 and to cell 23. The output of sequence encoder 17 is simultaneously applied to cell 23 and to sequence decoder 25. Sequence decoder 25 has 2 outputs, 27 and 29. Output 27 serves to clear sequence encoder cell 21 of 2 letter sequence and output 29 serves to clear both sequence encoder cell 19 and sequence encoder cell 21 of 3 letter sequences.

The function of the above-described sequence encoder cells and sequence decoder is to combine sequences of letter strings into common codes. In Tables I, II and III are illustrated the sequence groups  $G_b$  and  $G_e$  which have the letter strings CH, PH, TH, GH, WH, DG and QU for the  $G_b$  group and WL, WN, CK, GHT and TCH for the  $G_e$  group. The final "e" signal is generated by the sequence encoder and sequence encoder cell. For purpose of use with the hereinafter described hyphenation logic, each of these sequence groups ( $G_b$  and  $G_e$ ) could each have a single code for all of the letter strings within that group. However, from Tables I, II and III it can be seen that the code is modified to take into account the width of letters, upper and lower case letters, and distinction between various letters for use in conjunction with the justification logic, as distinguished from the hyphenation logic, of the present invention.

Typical operation of sequence encoder system 10, which includes circuits 17, 19, 21, 25, 37, 29 and 41, is as follows: Assuming the letter sequence "CH" is stored such that the letter "C" is stored in sequence encoder cell 21 and the letter "H" is stored in sequence encoder cell 19, in this event coded outputs appear at lines 31 and 33 which would respectively represent these characters and would appear at the input of sequence encoder 17. Sequence encoder 17 would recognize these codes and provide a single output code in line 35 which would represent the letter sequence "CH" (see  $G_b$  group of Tables I and III). This sequence code will therefore appear at the input of cell 23 and sequence decoder 25. Upon occurrence of the next clock pulse this code would be transferred into cell 23 and sequence decoder 25 would provide an output on line 27 which would clear sequence encoder cell 21. In the event sequence encoder cell 21 is empty, a signal will be applied to inverter 37 the output of which is applied to AND gate 39. Therefore, when a clock pulse is applied to AND gate 39 and there is no signal applied from the output of inverter 37 then the clock pulse will not be applied to the cells of shift register 43 and any information stored in sequence encoder cell 19 will be shifted to sequence encoder cell 21. As a result there will be no gap in information which would otherwise be the case if the information in cell 23 had been shifted to the next cell in shift register 43.

The next situation occurs when a three-letter sequence exists as in the  $G_e$  group of Table II. For example, if the letter sequence "GHT" is involved then sequence encoder cell 21 contains the letter "G," sequence encoder cell 19 contains the letter "H" and character encoder 15 contains the letter "T." This coded information representing the letter sequence G, H and T is respectively applied through lines 31, 33 and 45, to the input of sequence encoder 17 which will provide a single output code which represents this letter sequence (see Tables II and III). Sequence decoder 25 will sense this code and provide an output signal on line 29 which will clear sequence encoder cells 19 and 21. The clock output pulses from clock 41 are then prevented by AND gate 39 from being applied to shift register 43 for two clock pulses since sequence encoder cell 21 will be empty for two pulses and therefore provide two output pulses which are inverted by inverter 37. Therefore, no gap of information will exist between cell 23 and sequence encoder cell 21 as the character stored in character encoder 15 will now be in sequence encoder cell 21 and the code for the three letter sequence "GHT" is stored in cell 23.

Another function of sequence encoder 17 pertains to the final E ( $\epsilon$ ) logic. Final E ( $\epsilon$ ) is defined as an E

which is followed by a space or a punctuation or an E which is followed by a D, S or R which is followed by a space or punctuation. Sequence encoder 17 functions to determine whether or not an E is a final E. If it is a final E then sequence encoder 17 will provide a final E code in place of the existing E code. Since this does not involve using a single code for a group of letters, cells 19 and 21 are not cleared as was the case with the previously described encoding. It should also be noted that character decoder 49 will decode the  $\epsilon$  rather than class decoder 63 which decodes the other coded characters from sequence encoder 17. An example of how this operates is as follows: Assume an E is located in sequence encoder 21, a D is located in sequence encoder cell 19 and a space is represented at the output of character encoder 15. Under this set of conditions sequence encoder 17 will provide an  $\epsilon$  output code which will be transferred to cell 23 on the immediately following clock pulse. Upon the occurrence of this immediately following clock pulse, the D in 19 is transferred to sequence encoder cell 21, the space at the output of character encoder 15 is transferred to sequence encoder cell 19 and there was no transfer of the letter E from sequence encoder cell 21 to cell 23 since it was replaced by the  $\epsilon$  code from sequence encoder 17.

As previously explained, the output of the sequence encoder system 10 is in binary form and is applied to the  $\beta_4$  cell of shift register 43. The outputs of this system will be either the same as the output of character encoder 15 or a new code will represent certain letter strings or an  $\epsilon$  which will be obtained from the output of sequence encoder 17. Therefore, the coded inputs to  $\beta_4$  cell 23 will represent either single letters, spaces or punctuations, hyphens, end of paragraphs, numbers, letter groups or a final E.

Shift register 43 consists of eight cells referred to as  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$  and  $\alpha_4$  and  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$  and  $\beta_4$ . In the present embodiment of this invention each cell consists of 7 flip-flops the first 6 of which are used for character identification. In addition, the coding is such that the fourth, fifth and sixth flip-flops are used for vowel identification and the seventh is used to denote upper and lower case. As will be apparent to one skilled in the art that different codes may be devised and a greater or lesser number of flip-flops may be used which would of course depend upon the coding. However, it has been found that the employed coding has unique aspects which provide advantages for performing the hyphenation and justification in the present invention.

The outputs of cells  $\beta_1$  through  $\beta_4$  are respectively applied to the inputs of character decoders 47, 48, 49 and 50. The outputs of cells  $\alpha_1$  through  $\alpha_4$  are respectively applied to the inputs of character decoders 52, 53, 54 and 55.

The outputs of character decoders 47-55 are applied to the inputs of class decoder 63, hyphenation logic device 65 and end of line logic device 67. It will be appreciated that for reasons of simplicity, single-line connections are illustrated whereas in actual practice each of the characters decoded by character decoders 47-55 have separate line outputs some of which are connected to the class decoder, some of which are connected to the hyphenation logic and some of which are connected to the line ending logic in a manner which will hereinafter be described with respect to the particular operation of these elements.

It will also be noted from FIGURES 2A and 2D that the output of each of the  $\alpha_1$  through  $\alpha_4$  and  $\beta_1$  through  $\beta_4$  cells of shift register 43 are directly applied to the input of class decoder 63. It will be appreciated that in actual practice these single line connections will consist of many lines since binary information is being transferred from the first 6 flip-flops of each cell of the shift register to several different elements of class decoder 63. Each of character decoders 47-55 decodes different in-

formation from each of the cells with which it is associated and the information which each decodes is set forth in the block diagram of the decoders shown in FIGURE 24. For example, character decoder 50 will provide a single output pulse when the flip-flops of the  $\beta 4$  cell contain the binary information designating any of S, E, O, U, Y, I, N, R, M, L, C, D, G, H, T, or #. The remaining character decoders operate in like manner with respect to the characters they sense.

The function of the hyphenation logic element 65, the detailed operation of which are hereinafter described, is to identify permissible split locations in words of the text under investigation. If there is a permissible split between the letters occurring in the  $\alpha 1$  and  $\beta 1$  cells, then there will be an output from the hyphenation logic device 65. Line ending logic device 67 supplements the operation of the hyphenation logic device 65 inasmuch as it provides an output when there is a proper end of line condition even though there is no output from hyphenation logic device 65. This would occur, for example, when the  $\alpha 1$  cell is a hyphen and the  $\beta 1$  cell is a space or is not a punctuation which will indicate a proper ending at the hyphen of a hyphenated word. In addition, line ending logic will provide an output when  $\alpha 1$  is not a space and  $\beta 1$  is a space which indicates the ending of a word. Finally, line ending logic 67 will provide an output when  $\alpha 2$  is a number,  $\alpha 1$  is a punctuation and  $\beta 1$  is a number which makes it possible to split numbers after the punctuation.

The following description pertains to the read-in operation and more particularly to the justification techniques which are employed in the present invention. As shown in FIGURE 2B, AND gates 70 and 71 respectively receive signals from character encoder 15 and the  $\alpha 1$  cell and have time signals  $T_1$  and  $T_2$  respectively applied to their inputs. A signal occurs at the  $T_1$  input only during the read-in time and during this time a signal does not occur at the  $T_2$  input. Conversely, a signal occurs at the  $T_2$  input only during the read-out time but no signal occurs during the read-out time at the  $T_1$  input. The following discussions will relate only to the read-in period and therefore no signal will appear at output of AND gate 71.

The width of character in the  $\alpha 1$  cell is necessary information for purpose of justification since either that character in cell ( $\alpha 1$ ) or a hyphen will be at the end of the line. Therefore, the information in cell ( $\alpha 1$ ) is transmitted through AND gate 70, through OR gate 72 to the input of width encoder 73 for purpose of justification. It will be appreciated that the information being transmitted from cell ( $\alpha 1$ ) through AND gate 70 and OR gate 72 will be 7 bit information which will identify the particular character stored in cell  $\alpha 1$  as well as its upper or lower case characteristics which will affect the character's width. Width decoder-encoder 73 decodes the 7-bit code to a 4-bit code which denotes the width of the particular character in question. The techniques for accomplishing this are conventional and will be obvious to one skilled in the art. One approach could be to decode each 7-bit character into a 4-bit character denoting its width. In the alternative, it would be possible to group characters having the same width together and provide 4-bit coded information only as to groups. As previously explained, the 7th bit contains information as to whether the character is upper or lower case which will provide in many instances a different width for the same character and this information is therefore necessary. The 4-bit information represents the actual unit width of each letter by the binary number system and it should be noted that all 4-bits will generally be necessary to represent the width of a group of letters as coded by sequence encoder 17. The output of width encoder 73 is transmitted to width counter 74 which counts down to zero during each character count and subtracts one unit at a time from line counter 75. Line width counter 75 functions to provide the difference between a predetermined line length

setting and the summation of character widths that were contained in the  $\alpha 1$  cell. That is, a setting of 100 units, for example, may be introduced into line length counter 75 by line length setting device 77 by conventional binary logic schemes. Then the character width contained in width counter 74 is subtracted from the 100 units by conventional binary counting techniques. If, for example, the character "A" in the  $\alpha 1$  cell had 3 units of width, then the output from line length counter 75 would have a binary number of 97. It should be noted that the output of line length counter 75 must contain a larger number of bits, for example, 9 bits, since it must be able to handle relatively large numbers as may be encountered in long lines such as in books and magazines.

In summary, line length counter 75 provides the difference between the line length as established by line length setting device 77 and the summation of the character widths transmitted from width encoder 73. As will be hereinafter described in greater detail, line length counter 75 will continue counting downwards until a predetermined negative number is reached at which time this condition will be sensed by end of read-in device 78 which will actuate tape reader control 79 which will then stop tape reader 13.

Concurrent with this width counting process the outputs of hyphenation logic device 65 and line ending logic device 67 are used to determine the end of line location. This is achieved by transmitting the information in line length counter 75 to end of line location register 80 through "AND" gate 81 which will occur if either of the following conditions exist:

(1) The first condition exists if the output of line length counter 75 is equal to or greater than 3 and there is an output signal from hyphenation logic 65.

(2) The second condition exists when the output of line length counter 75 is equal to or greater than zero and there is an output from line ending logic 67.

The first condition is implemented by transmitting the output of line length counter 75 to the input of device 83 the output of which is applied to AND gate 85 along with the output of hyphenation logic device 65. The output of AND gate 85 is then transmitted to OR gate 87 the output of which is applied to AND gate 81. The function of device 83 is to take into account the addition of a hyphen to the end of the line which in this instance will be considered as having a 3-unit length, when hyphenation logic device 65 identifies a split point. This hyphen is never received by the  $\alpha 1$  cell since it is separately added at the end of the line during read-out as will hereinafter be described. Device 83 prohibits the ending of the line if there are less than 3 units available and the word would otherwise be hyphenated.

The details of device 83 are not described here since it consists of a group of conventional binary logic circuits permitting passage of signal only when a predetermined number is exceeded.

The technique for implementing the above-mentioned second condition is by applying the output of line length counter 75 to the input of device 89 the output of which is applied to the input of AND gate 91 along with the output of line ending logic device 67 wherein the output of AND gate 91 is applied to OR gate 87 the output of which is applied to AND gate 81. The function of device 89 is to prohibit the transmission of a signal to AND gate 91 when the output of line counter 75 is less than zero. When this condition exists there will be no output from AND gate 91 even though there is a line ending condition signal from line ending logic device 67.

In summary, during the addition of each new character to the machine the remaining units of length of line, as stored in length counter 75, are transmitted to end of line location register 80 when either of the above-mentioned two conditions exist. This process will continue until the smallest positive line length is remaining in length counter 75 and neither of these two conditions

exist. However, tape reader will continue operation and reading information into the system until line length counter 75 has gone negative or to zero as determined by the setting of end of read-in device 78. This is necessary to be sure that the last possible character has been examined. That is, if the word is to be hyphenated and the units in counter 75 are 2 or 1 (not zero), then no output will be received from AND gate 85 and the two units will not be transferred from length counter 75 to end of line location register 80. This is because the width of a hyphen is 3 units, and if the hyphen were added and the line ended at this point, there would be too many units in the line. However, reader 13 will read the next character so that length counter 75 is zero or less than zero which is sensed by device 78 to stop operation. Zero is the limiting condition since the present invention does not provide a method for compressing the space between words.

While the length of line is being counted, it is necessary to determine the number of spaces between words so that justification of the line may be accomplished. This is achieved by applying the output of AND gate 70 to the input of space counter 93, of space counter system 92, which will decode the space code located in the  $\alpha 1$  cell and will therefore add all of the spaces which have occupied this cell. The space count in space counter 93 is transmitted to the input of AND gate 94 along with the output of OR gate 87. Therefore, space register 95 will have the same count as space register 93 provided there is an output from OR device 87 which denotes a proper end of line condition. It is necessary to have this logic since space counter 93 may recognize another space in cell  $\alpha 1$  while line length counter 75 is going negative and yet this space will not be included in the justified line. Another reason is that there are situations in which it will be improper to split the last word investigated and therefore it is incorrect to count the space between this last word and the immediately preceding word. The reason for this is that the end of line condition will be at the last letter of the above-mentioned immediately preceding word and therefore the space between these last two words cannot be used for purposes of justification.

For reasons which will hereinafter become apparent, end of line condition device 97 is provided for use during the read-out operation. This device includes AND gates 98, 99, 100, each of which has one input connected to the output of OR device 87. AND gate 98 has the other input connected to the end of paragraph output from line ending logic 67. AND gate 99 has the other input connected to all other end of line conditions of line ending logic 67. AND gate 100 has the other input connected to the output of hyphenation logic 65. Each of these AND gates is respectively connected to flip-flops 101, 102 and 103.

The following discussion relates to the read-out phase which may include (1) space insertion between words, (2) hyphenation of a word, (3) the end of paragraph condition, or (4) end of line condition. The first step of the read-out phase is to return the tape of tape reader 13, which reads during the read-in period, to the beginning of the line. This is achieved by means of tape reader control 79 which is responsive to the output of character counter 105. During the read-in operation, character reader 105 counts each character that is transmitted to the input of sequence encoder cell 19. In this manner the total number of characters, as distinguished from the number of widths, is known when end of read-in device 78 provides an output signal which stops tape reader device control as previously explained. As a result, tape reader 13 is caused to reverse its operation for the exact number of characters stored in character counter 105.

It is important to note that AND gates 71 and 107 respectively have input signals applied to them only during the read-out operation which is denoted by the sym-

bol  $T_2$ . Therefore, AND gate 71 will transmit encoded character information during read-out which is used to determine when the end of line condition occurs during read-out, the details of which will be hereinafter explained. The function of gate 107 is to transmit the tape reader output through AND gate 109, AND gate 11, OR gate 113, to output device control 115 and then to output device 117. If the text were such that there were no spaces to be inserted and the last word of the line was not to be hyphenated or there was an end of paragraph condition then the output of tape reader 13 would be directly communicated to the output device 117 without any inhibiting by AND gates 109 and 111 as will become apparent.

When end of line location register 80 has any positive number stored therein, then there will be space insertion provided it is not an end of paragraph condition. In order to determine the number of spaces to be inserted it is necessary to know both the number of spaces between words (as determined by space register 95) as well as the number of units to be inserted (stored in end of line location register 80). Assume it is a line ending condition but it is neither an end of paragraph nor a hyphenation, then there will be an output from flip-flop 102 which is applied to AND gate 119. Also connected to AND gate 119 is the binary number representing the units stored in end of line location register 80 which is then transmitted to divider device 121. If the end of line condition is the hyphen of a word that is hyphenated by hyphenation logic device 65, then a signal is applied from flip-flop 103 to the input of AND gate 123. The other input to AND gate 123 is the binary form of the units stored in end of line location register 80. Since a signal is applied to AND gate 123 from flip-flop 103 the binary form of the units in register 80 are transmitted to a conventional subtractor device 125 which reduces the binary count of register 80 by 3 which is then applied to the input of divider circuit 121. Therefore, the input to divider circuit 121 represents the number of units to be added to the line in question. The other input to divider circuit 121 represents the total number of spaces between words in the line which are to be increased in width by the number of units applied to the other input of the divider. Assuming there are 3 spaces and 11 units applied to the divider, then the quotient ( $q$ ) is 3 and the remainder ( $r$ ) is 2. The quotient is applied to the input of counter 127 and the remainder is applied to the input of counter 129. The detailed operation of counter 127 and 129 will not be described since their operation may be implemented in many different ways by one skilled in the art. However, it should be noted that counters 127 and 129 are to be inhibited, for reasons which will hereinafter become apparent, when there is an output signal from space code decoder 131 which senses the code indicating a space being transmitted from output device control 115 to output device 117. From the example given, counter 127 will contain the binary number representing 3 units. This number will be transmitted to the input of unit space code generator 133 and will also be transmitted to inverter 135 the output of which is applied to AND gate 137. The function of inverter 135 and AND gate 137 is to prevent counter 129 from transmitting a signal to unit space generator 133 when counter 127 is in operation. However, when counter 127 has completed its count, counter 129 will transmit its output signal through AND gate 137, the other input of which is now receiving a signal, to unit space code generator 133.

When unit space code generator 133 receives the binary signal representing 3 units from counter 127 they will be decoded and generator 133 will provide 3 individual unit space signals which will be applied to OR gate 113 and will cause output device control 115 to advance output device 117 by 3 unit spaces. This addition of unit spaces will occur only when space code decoder 131 has sensed that a space had been advanced from output device control 115.

Counter 129 inserts one unit space immediately after counter 127 had inserted its 3 unit spaces. After counter 128 has inserted its one unit space it counts down the remainder by one unit. Therefore, in the example given there would be 4 additional unit spaces inserted between the first and second words, 4 additional unit spaces inserted between the second and third words and only 3 additional unit spaces inserted between the third and fourth words, since the counter 129 had already run down to zero.

Therefore, counters 127 and 129 had been enabled by the output of the space code decoder 131 and immediately following the conventional space between words there will be added the unit spaces as dictated by the output of unit space generator 133. It should be particularly noted that when the counters have been enabled and there is an output from unit space code generator 133 that inverter 139 will prevent AND gate 109 from transmitting the tape reader signal to AND gate 111. Therefore, unit spaces may be added while suppressing any new characters which would otherwise be available from tape reader 13. It will also be noted that the output of inverter 139 is connected to tape reader control 79 which will cause tape reader 13 to stop operation when there is no signal output from inverter 139.

The next phase of operation pertains to identifying the end of line condition. In order to achieve this function, OR gate 113 is provided which (1) gates the characters read by tape reader 13 through AND gate 107, and (2) gates the unit spaces added through unit space CODE generator 133 and (3) gates the hyphens which are to be separately added in the manner hereinafter described. All of this information is transmitted through AND gate 141 during the read-out operation since there is a signal also existing at the  $T_2$  input of AND gate 141. The output of AND gate 141 is transmitted through OR gate 143 and is encoded by character encoder 15 and then transmitted through AND gate 71. Since there is an input signal at the  $T_2$  input of AND gate 71, the encoded signal is applied through OR gate 72 to width encoder 73 which converts the code into width code which is then transmitted to width counter 74 which subtracts from line length counter 75. When line length counter 75 has been decreased to zero units, decoder device 145 senses this condition which stops output device 117 through output device control 115 and also starts tape reader control 79. In addition, the output of decoder 145 starts timer device 147 which provides a signal  $T_1$  the function of which has been previously described.

When a hyphen condition exists a signal will be applied to one input of AND gate 149 from the output of flip-flop 103 of end of line condition device 97. When the output of length counter 75 is 3 then decoder 151 will apply the other input to AND gate 149 the output signal of which is applied in parallel to both inverter 153 and to OR gate 113. This signal is coded as a "hyphen" and output device control 115 will therefore cause output device 117 to print out a hyphen which will consume the last three units of the line. During this hyphen insertion step inverter 153 functions to inhibit read-out of characters from AND gate 109 and also cause tape reader control 79 to stop tape reader 13 (in this instance no signal from inverter 153 indicates a stop condition). From this it can be seen that a hyphen is inserted at the proper place in the hyphenated word as well as at the end of the line. It will be appreciated that the previously described justifying process will, if units are to be added, take place prior to the insertion of this hyphen.

In addition to the foregoing, it is necessary to take into account the end of paragraph condition since there will be no space insertion or hyphenation when the end of paragraph condition exists. To accomplish recognition of an end of paragraph condition the output of output device control 115 is transmitted to end of paragraph decoder 154 the output of which is transmitted to AND

gate 155. The other input to AND gate 155 is connected to flip-flop 101 which indicates the end of paragraph condition. When these conditions exist, then there is an output from AND gate 155 which is applied to output control device 155 which stops output device 117 and causes it to shift to the beginning point of the next line. The output from "AND" gate is also transmitted through line 156 to both tape reader control 79, which starts tape reader 13, and to timer 147, which initiates its  $t_1$  output signal. Therefore, the read-in cycle of the next line is started.

It should be noted that if the immediately preceding line ends at the end of a word then the code of the first character at the beginning of the next line will be a space. This first space in the next line must be ignored by the system since, if not excluded, it would result in indenting the next line by a space. First space decoder device 157 performs the function of decoding the space preceding the first character in a new line in which instance an output signal is provided and inverter device 159 inhibits AND gate 161. Therefore, the space which immediately precedes the first character is not transmitted through AND gate 161 and is excluded from the system and is not operated on by the previously-described system. All other characters in that line, including spaces, will be transmitted through AND gate 161. Other spaces in the line are not sensed because device 157 includes a flip-flop which is set in the one state only with the leading edge of the first character of the line.

#### Method of hyphenation

The previously-described system shown in FIGURES 1 and 2 specifically sets forth the justification techniques and generally described the system arrangement for the hyphenation mechanization of the present invention. The foregoing included a complete discussion of sequence encoder system 10, shift register 43 and character decoders 47-55, and only a general discussion of class decoder 63 and hyphenation logic device 65. The following discussion pertains to the hyphenation principles of the present invention and its implementation will utilize five basic elements which are as follows:

- (1) Sequence encoder system 10 (supra and Tables I, II and III)
- (2) Cells  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  of shift register 43
- (3) Character decoders 47 through 55
- (4) Class decoder 63, and
- (5) Hyphenation logic device 65

Another embodiment of the present invention (FIG. 25) pertains only to a hyphenation device, as distinguished from the justifying-hyphenating device (FIGS. 1 and 2), and will include these same five basic elements. The differences and similarities between these embodiments will become apparent after a detailed consideration of the same. However, it should now be particularly noted that identically the same hyphenating principles and logic are employed in these two embodiments. At the outset each of the above-mentioned basic elements will be considered separately and their interrelationship will be described.

#### Sequence encoder system

A detailed description of the logic of sequence encoder system 10 is not presented since its implementing logic is considered conventional. However, the method of establishing classes, and the heretofore-described arrangement of logic elements (such as sequence encoder 17, sequence encoder cells 19 and 21 and sequence decoder 25) are considered unique to the present invention. There are two basic classes of information obtained from sequence encoder 17 which are shown in Tables I, II and III and are herein referred to as the  $G_b$  and  $G_e$  classes. As previously described, these class codes include information describing the width of characters in each class which is essential for purposes of justifica-

tion. Another important facet of sequence encoder 10 is that its use makes it possible to reduce the number of components and simplify the logic by combining letter groups into a single code. More important, however, is that these classes are an integral part of the hyphenation principles of the present invention.

Referring now to Tables I and III, in particular the  $G_b$  class, it can be seen that the letter sequences CH, PH, TH, GH, WH, DG and QU are defined by single codes and are transmitted directly to class decoder 63 and to hyphenation logic device 65. It has been discovered that it is possible and desirable to treat each of these letter strings as a single consonant for identification purposes, and, in addition, they may be treated identically for purpose of hyphenation as defined by the present invention. An example of how the  $G_b$  sequence is unique to the hyphenation logic is as follows: If the letter string PH had not been combined into a single code ( $G_b$ ) and treated as a single consonant, the word "PHOSPHOR" would be improperly split at the letters "P" and "H" (second occurrence). However, when combined into a single code, then the word "PHOSPHOR" is properly split between the letters "S" and "PH" since the letter string "PH" is considered a single consonant.

The next group of sequence encoding of characters is generally referred to as the  $G_e$  group which includes the letter strings WL, WN, CK, GHT, and TCH. This grouping of letter strings is used in the hereinafter described inhibit logic device 410 and vowel-consonant logic of device 402 and hyphenation logic device 65. The letter strings of the  $G_e$  group are typically syllable ending strings as can be seen in the examples: OWLISH, OWN-ER, BACK-ING, LIGHT-HOUSE and so forth. Further these strings are almost never split internally. Their utility in the present machine occurs because it is basically difficult to split the sequence Consonant-Vowel except when these strings or certain affixing elements are available.

#### Output of shift register 43

As previously described, the output of sequence encoder 17 is initially transmitted to cell 23 ( $\beta_4$ ) of shift register 43 along with the characters encoded by character encoder 15 but not converted by sequence encoder 17. The information in the cells of the shift register is serially shifted from left to right at the occurrence of each consecutive clock pulse from clock 41 which is transmitted through AND gate 39 which is controlled in the heretofore described manner. The information stored in the cells of the shift register are sensed by two different techniques. The first is by character decoders 47-55 which provide a single pulse output for each sensed code. The second is by class decoder 63 which also provides a single pulse output for each sensed code. The output of each of the shift register cells which is sensed by class decoder 63, hyphenation logic device 65 and end of line logic device 67 is in 6-bit binary form. The 7th bit, which is used to indicate upper or lower case, is carried only by the  $\beta_1$ - $\beta_4$  and  $\alpha_1$  cells and is sensed by AND gate 70 from the  $\alpha_1$  cell which information is used by line length counting system 68 for purposes of justification. It is therefore obvious that the  $\alpha_2$ ,  $\alpha_3$  and  $\alpha_4$  cells need contain only 6 bits.

#### Character decoders 47-55

Each of character decoders 47-55 decodes those characters (6 bit format) which are shown in the associated blocks of FIG. 2A. The output of each character decodes in a single pulse and occurs when the information in the associated cell corresponds to the decoding capability of that character decoder. The output of character decoders 47-55 is applied to class decoder 63, hyphenation logic device 65 and line ending logic device 67 in the heretofore and hereinafter described manners.

#### Class decoder 63

Class decoder 63 (shown in FIGURE 3 and FIGURES 6-14) and hyphenation logic device 65 (shown in FIGURES 4A-4D and FIGS. 15-24) are interdependent but for purpose of explanation will be considered separately. Class decoder 63 will be considered first since all of its output information is used by hyphenation logic device 65 which is for purposes of hyphenating words.

The class decoder receives two basic types of information which are: (1) information directly from character decoders 47 through 55 and (2) information directly from cells  $\alpha_1$  through  $\alpha_4$  and  $\beta_1$  through  $\beta_4$ . It will be recalled that the information from the character decoders is in the form of a single pulse which is transmitted from each character decoder by a predetermined wire for each decoded character and that the information from the cells is in 6 bit format information. Class decoder 63 is shown in block diagram form in FIGURE 3 and includes elements 201 through 209. The inputs and outputs of each of these elements are shown on the respective blocks of the block diagram. In addition, the details of each of elements 201 through 209 are shown in FIGURES 6 through 14, respectively, and are cross referenced by these figure numbers in the individual blocks.

Elements 201 through 205 receive their inputs directly from the character decoders whereas elements 206 through 209 receive their inputs directly from the cells of the shift register. The basic reasons for this are simplification of logic and simplification of class coding. The information from elements 201 to 205 relate to prefix and suffix condition. Whereas, the information from elements 206 to 209 relate to vowel and consonant conditions and to the  $G_e$  group. For purpose of the present invention, the  $G_b$  and  $G_e$  group are also considered to be consonants.

The purpose of device 201 the details of which are shown in FIGURE 6A and 6B is to provide information about vowel beginning suffixes. It has been discovered that the following strings of vowel beginning suffixes ( $S_v$ ) have unique advantages for purposes of hyphenation:

al#, ed#, ee#, er#, es#, y#, ly#, abl, age, anc, ant, ari, ary, ato, enc, ern, ery, est, ian, ibl, ice, ily, ine, ing ish, ism, ist, ite, itu, ity, ize, ous, oid, ope, ose, ier#, ies#, ied#.

While the foregoing discussion has been concerned with vowel beginning suffixes ( $S_v$ ) it should be noted that a distinction between the final "e," that is where the letter "e" is followed by # or by any one of d#, s# or r#, and an e without these associated endings is indicated. The particular set of suffixes all being with a vowel and occur frequently after a double consonant string as in the word RENTER. The basic logic presupposes that all consonant pairs can be split except in certain specific situations such as the present one. Thus inclusion of er# in the  $S_v$  logic insures that the incorrect split REN-TER will not occur.

A single pulse is obtained at the output of device 201 (shown as encircled 1) when any one of the above-listed strings do not respectively occur at the  $\beta_2$ ,  $\beta_3$ ,  $\beta_4$  cells. The signal is needed to suppress improper splits, that is CHAR-TIST would be an improper split.

The purpose of device 202, the details of which are shown in FIGURE 7, is to identify consonant beginning suffixes ( $S_c$ ) as defined by the present invention. It has been discovered that the following letter strings, consonant-beginning suffixes have unique advantages for purposes of hyphenation:

cial, cian, cien, cion, cive, ciou,  
gial, gian, gien, gion, give, giou,  
tial, tian, tien, tion, tive, tiou,  
bil, cal, mat, ment, fore, tory, ture,  
ful, less.

As single pulse is obtained at the output of device 202 (shown as encircled 2) when any one of the above-listed

string respectively occur at the  $\beta 1$ ,  $\beta 2$ ,  $\beta 3$ ,  $\beta 4$  cells. It should be particularly noted that the unique characteristic of each of these strings is that the beginning character of each string is a consonant and a split will always occur before these letter strings. When included, the ( $S_c$ ) set of suffixes have the property that they almost always cause a split off in English words. Thus, this list simplifies the difficult problem of obtaining Vowel-Consonant splits in such words as SPE-CIAL, SPA-TIAL, for examples.

The purpose of device 203, the details of which are shown in FIGURE 8, is to provide information pertaining to consonant ending prefixes ( $P_c$ ). It has been discovered that the following group of letter strings have unique characteristics for purposes of hyphenation:

#ac, #ad, #al, #ex, out, sub, sun, #em, #en, #im, #in, #un, com, con.

A single pulse is obtained at the output of device 203 (shown as encircled 3) when any one of the above-listed strings respectively occur at the  $\alpha 1$ ,  $\alpha 2$ ,  $\alpha 3$  cells. The unique common characteristic of this group of character strings is that the prefix ends with a consonant. The  $P_c$  set of prefixes is the complementary set to the  $S_c$  set of suffixes and its inclusion permits the splitting of such words as EX-IT or OUT-ING, for examples.

Devices 204 and 205, the details of which are respectively shown in FIGURES 9 and 10, may be considered together since they examine the same letter strings. One distinction between these devices is that device 204 considers the letter strings in cells  $\alpha 1$ ,  $\alpha 2$  and  $\alpha 3$  whereas device 205 considers the same letter strings in cells  $\alpha 2$ ,  $\alpha 3$  and  $\alpha 4$ . Another distinction is that device 204 provides an output when the letter strings described below exist whereas device 205 provides an output signal only when these letter strings do not exist. The purposes of each of these devices is to identify vowel ending prefixes ( $P_v$  or  $P_{v1}$  for device 204 and  $P_{v2}$  for device 205). It has been discovered that the following group of letter strings have unique characteristics for purposes of hyphenation:

re, be, de, pro

A single pulse is obtained at the output of device 204 (shown as encircled 4) when anyone of the above-listed strings respectively occur at the  $\alpha 1$ ,  $\alpha 2$ ,  $\alpha 3$  cells. Conversely, no pulse is obtained at the output of device 205 (shown as encircled 5) when any one of the above-listed strings respectively occur at the  $\alpha 2$ ,  $\alpha 3$ ,  $\alpha 4$  cells. A pulse will occur at all other times.

It should be particularly noted that it is not required that any of these letter strings be preceded by a space or punctuation. Therefore, "RE" in "FREQUENCY," the "RE" in "RETREAT" (both occurrences) are all properly treated in the same manner. It should be noted that the double use of this list in the logic is to take advantage of the fact that if it is proper to split the word RE-TREAT after the RE then it is certainly improper to split the same word RET-REAT. Thus in one position, the signal from this set provides the possibility of a split (if certain other conditions are met while in the other position the corresponding signal specifically suppresses a split.

The purpose of device 206, the details of which are shown in FIGURE 11, is to determine when the  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  cells contain a vowel (V). It will be noted, however, that in certain instances it is necessary to know when  $\beta 4$  is not a vowel and therefore device 206 also provides this information.

It should be particularly noted that the hyphenation logic is based upon the concept that a final E is not a vowel. For purpose of the present invention a final E is defined as letter E followed by a space or punctuation or by any one of d\_#, S\_# or R\_#.

As previously described, sequence encoder 17 also functions to identify and separately code each final E

according to the definition given above. Since the final E is separately coded, device 206 distinguishes between the conventional and final E and therefore will not include a final E as defining a vowel.

This discovery is extremely important in bringing about proper word splitting. The final E is frequently used in English to signal the modification of pronunciation of a preceding vowel rather than the existence of another syllable. Thus the final E in RATE serves to differentiate the word phonetically and graphemically from the word RAT and does *not* indicate the existence of a second syllable of the form RA-TE or RAT-E as might be the case in other languages.

The purpose of device 207, the details of which are shown in FIGURE 12, is to determine when a weak consonant (W) does or does not occur. It has been discovered that the following consonants, separately identified as weak consonants (W), have unique characteristics for purposes of hyphenation: l, m, n, r, w. It is called the set of weak consonants. These consonants are considered significant when they appear at the  $\alpha 1$ ,  $\beta 1$  and  $\beta 4$  positions because these consonants play a somewhat different role than the other consonants when they occur in strings of consonants, primarily because they do *not* signal a split condition. Thus, as has been previously noted, one can frequently split two successive consonants but this rule does not hold when the second consonant is weak. Accordingly, a single pulse is obtained at the output of device 207 when any of the  $\alpha 1$ ,  $\beta 1$  and  $\beta 4$  cells contain a weak consonant (shown as outputs encircled 13, 14 and 15 respectively, in FIGURE 3). Also, no pulse is obtained at the outputs (shown as encircled 15A and 15B) of device 207 when  $\beta 4$  and  $\beta 1$  respectively contain weak consonants (W).

The purpose of device 208, the details of which are shown in FIGURE 13, is to determine when  $\beta 1$  contains a consonant (C) and also to determine when  $\beta 1$  is not a  $G_e$  letter group. It should be noted that for purposes of the present invention consonants are defined as follows:

b, c, d, f, g, h, p, t, x,  $\phi$ ,  $G_b$ ,  $G_e$ , l, m, n, r, w, q, v

This definition of consonants is significant because, in essence, all characters other than the vowels or marks of punctuation are considered consonants as a matter of selection in accordance with the present inventive concept. Accordingly, a single pulse is obtained at the encircled 16 output of device 208 when the  $\beta 1$  cell contains a consonant. Also a single pulse is obtained at the encircled 17 output of device 208 when the  $\beta 1$  cell does not include a character of the  $G_e$  group type.

The purpose of device 209, the details of which are shown in FIGURE 14, is to determine when the  $\alpha 1$  cell contains a consonant (encircled 18 output) and also to determine when the  $\alpha 1$  cell contains a consonant (encircled 18 output) and also to determine when the  $\alpha 1$  cell contains a  $G_e$  letter group (encircled 19 output). The definition of a consonant is the same as given above in the description of device 208.

The importance of devices 206, 207, 208 and 209 is that these devices provide information as to whether or not the  $\alpha 1$  and  $\beta 1$  cells contain consonants or vowels. This information is of particular importance inasmuch as one of the major discoveries of the present invention is considered to be the use of vowel and consonant information on each side of the point of investigation (i.e., the  $\alpha 1$  and  $\beta 1$  cells).

#### Hyphenation logic

FIGURES 4A-4D are block diagrams showing the inputs and outputs of the various logic networks of hyphenation logic device 65 of the present invention. The logic networks are identified by reference numerals 401 through 410 the details of which are respectively shown in FIGURES 15 through 24. The primary function of hyphen-

ation logic device 65 is to provide a signal at the output of AND gate 499 when the word being examined should be split between the characters contained in the  $\alpha 1$  and  $\beta 1$  cells of shift register 43. As previously explained, the information processed by the hyphenation logic is derived from two primary sources which are (1) character decoders 47 through 55 (shown in FIGURES 1 and 2A and 2D) and (2) class decoder 63 (shown in FIGURES 1, 2D, 3 and 6 through 14). There are three basic logic network systems that make up hyphenation logic device 65 which are (1) consonants-vowels logic networks 401-404, (2) S logic networks 405-509, and (3) inhibit logic network 510.

As previously indicated, the consonants and vowels at the  $\alpha 1$  and  $\beta 1$  positions play a very important part in the hyphenation logic of the present invention. To implement this aspect of the present invention, logic networks 401, 402, 403 and 404 respectively depend upon the  $\alpha 1$  and  $\beta 1$  cells being occupied by vowel-vowel, vowel-consonant, consonant-vowel and consonant-consonant characters. Contrary to traditional grammar, it has also been discovered that the letter "S" may be considered either as a vowel or a consonant. Therefore, separate logic networks 405, 406, 407, 408 and 409 have been devised to implement this concept. This has been implemented by having logic networks 405, 406, 407, 408 and 409 respectively depend upon the  $\alpha 1$  and  $\beta 1$  cells being occupied by "S" consonant, S-vowel, S-S, consonant-S and vowel-S characters. Inhibit logic network is based upon a special set of restraints and functions to prevent word splitting which would be normally performed by the other logic networks.

In practice if inhibit logic network 410 has a zero output pulse denoting a restraint condition then there will be no output from AND gate 499 even though there is an output from OR gate 498 which denotes that one of the other logic networks recognizes a split condition. In order for there to be a split, as denoted by the output from AND gate 499, it is necessary that there be an output pulse from inhibit device 410 as well as an output pulse from OR gate 498. It should also be noted that the hyphenation logic operates in parallel which is highly advantageous from the standpoint of speed of operation. That is, each of logic elements 401 through 410 will simultaneously consider the information contained in the  $\beta 1$  through  $\beta 4$  and  $\alpha 1$  through  $\alpha 4$  cells and if a proper split condition prevails then there will be an output from one of logic networks 401 through 409 as well as an output from inhibit logic network 410.

Systematic statistical analysis investigation and observation of the English language has resulted in the unique set of conditions which provide the criteria for designing the present hyphenation system.

The purpose of vowel-vowel logic network 401 including AND gate 418, the details of which are shown in FIGURE 15, is to split a word if  $\alpha 1$  is a vowel,  $\beta 1$ ,  $\beta 2$  and  $\beta 3$  are either ING or ONE. Initially this may appear to be a straightforward condition since ING, for example, is a typical ending of words. However, it will be noted that this rule will not split the word BR/ING since  $\alpha 1$  is an R which is not a vowel; whereas it will split the word BE/ING since E is a vowel. As will hereinafter be considered, other conditions are employed to also examine the ING letter string.

Vowel-consonant logic system 402 consists of a plurality of logic networks 412, 413, 414, 415 and 416, each of which has the output connected to OR gate 417 the output of which is connected to AND gate 418 along with the encircled 6 and encircled 16 inputs. The encircled 6 input is from vowel logic network 206 and provides a signal to AND gate 418 when  $\alpha 1$  is equal to a vowel and the encircled 16 input is from logic network 208 and provides a signal when  $\beta 1$  is equal to a consonant. As previously noted, the splitting of the sequence vowel-consonant provides certain problems. The logic given here is com-

pounded of several specific results where such a sequence can be split without complete analysis of the word in question. The logic rules are constructed solely to attain as much splitting power as possible with limited access to the context of the sequence in question and do not, in general, correspond to any well-known rules of linguistics.

The purpose of vowel-consonant logic network 412 including AND gate 418, the details of which are shown in FIGURE 16, is to indicate a split condition if  $\alpha 1$  is a vowel (V),  $\beta 1$  is a consonant (C),  $\alpha 1$  is equal to U,  $\beta 2$  is a vowel and  $\beta 2$ , 3 and 4 is not a vowel beginning suffix ( $S_v$ ). This rule may be conveniently referred to as the U rule and was discovered as a result of an attempt to show that no simple rule of this form could exist. The study instead showed that such a rule did exist for the letter U but not for any other vowel. For example, it will split BU-TANE.

Vowel-consonant logic network 413 including AND gate 418, which are shown in detail in FIGURE 16, indicates a split condition provided  $\alpha 1$  is a vowel,  $\beta 1$  is a consonant,  $\alpha 1$ ,  $\alpha 2$  and  $\alpha 3$  is a vowel ending prefix which is denoted by the encircled 4 input and  $\beta 1$  is not a C or a V,  $\beta 2$  is a vowel (V) other than a U,  $\beta 3$  is not an S and  $\beta 4$  is not an S, a weak consonant (W), a vowel (V) or a space or punctuation (#). This set of conditions is useful in splitting the following words: DE-BATE and RE-TAKE, for examples.

Vowel-consonant logic network 414 including AND gate 418, the details of which are shown in FIGURE 16, indicates a split condition provided  $\alpha 1$  is a vowel (V),  $\beta 1$  is a consonant (C),  $\beta 2$  is a vowel (V) and  $\alpha 1$ ,  $\alpha 2$ ,  $\alpha 3$  and  $\alpha 4$  respectively equal E, M, O, and S or  $\alpha 1$ ,  $\alpha 2$  and  $\alpha 3$  respectively equal Y, N, A. To provide a proper split condition,  $\beta 2$  must be a vowel otherwise the logic would incorrectly split SOMERSET or similar forms. The correct split would be SOME-RSET.

Vowel-consonant logic network 415 including AND gate 418, the details of which are shown in FIGURE 16, splits a word if  $\alpha 1$  is a vowel (V),  $\beta 1$  is a consonant (C),  $\beta 2$  and  $\beta 3$  respectively equal L and final E and  $\beta 1$  is not a weak consonant (W). This set of conditions distinguishes between the situations such as TABLE, which should be split TA-BLE and EARLE which should not be split EA-RLE.

Vowel-consonant logic network 416 receives its signal directly from logic network 202 and a word split is proper if  $\beta 1$ ,  $\beta 2$ ,  $\beta 3$ ,  $\beta 4$  is a constant beginning suffix ( $S_c$ ) provided  $\alpha 1$  is a vowel and  $\beta 1$  is a consonant are applied to the inputs of AND gate 418. The details of logic network 202 have been discussed hereinabove and no discussion will be given here.

Consonant-vowel logic system 403 includes consonant-vowel logic networks 417, 418, 419, the outputs of which are transmitted to OR gate 421 the output of which is transmitted to AND gate 423 along with the inputs denoted as encircled 9 and encircled 18 which are respectively  $\beta 1$  equals a vowel and  $\alpha 1$  equals a consonant.

Consonant-vowel logic network 417 including AND gate 423, the details of which are shown in FIGURE 4B, will split a word if  $\alpha 1$  is an X or if  $\alpha 1$  equals the  $G_e$  letter string. The importance of this aspect of the circuit has been considered in connection with the discussion of the letter strings of the  $G_e$  group set forth hereinabove.

Consonant-vowel logic network 418A including AND gate 423, the details of which are shown in FIGURE 4B, will indicate a split condition provided  $\alpha 1$  is a consonant,  $\beta 1$  is a vowel and  $\beta 1$ ,  $\beta 2$ ,  $\beta 3$  and  $\beta 4$  are respectively E, V, E, R. The logic rules are organized by the various combinations of consonant, vowel and S at the critical  $\beta 1$ ,  $\alpha 1$  positions. Thus each rule presupposes a particular combination and is constructed with that combination in mind.

Consonant-vowel logic network 419 including AND gate 423, the details of which are shown in FIGURE 17, will split a word if  $\alpha 1$  is a consonant,  $\beta 1$  is a vowel, and  $\beta 1$ ,  $\beta 2$  and  $\beta 3$  are respectively ING. This rule should be

compared with the rule implemented by the vowel-vowel logic network 401 which also splits ING. The sequence ING is, of course, a frequently occurring sequence because of the many formations of present participial forms in English. It is necessary to add the restriction that  $\alpha 2$  is a vowel to take care of the many cases where the final consonant of the verb is doubled in the participial form and the correct split is between the doubled consonants rather than after the second consonant, for example the proper split would be WED-DING and not WEDD-ING.

Another major logic system of hyphenation logic device 65 is shown in FIGURE 4B and is identified as consonant-consonant logic system 404 which includes consonant-consonant logic networks 425, 427, 429 and 431, the outputs of which are connected to the input of OR gate 433 the output of which is applied to AND gate 435 along with the signals denoted as circle 16 and circle 18 which are respectively  $\beta 1$  equals a consonant and  $\alpha 1$  equals a consonant.

Consonant-consonant logic network 425 including AND gate 435, the details of which are shown in FIGURE 4B, splits a word if  $\alpha 1$  is a consonant,  $\beta 1$  is a consonant and  $\beta 1, \beta 2, \beta 3, \beta 4$ , respectively equal MENT or respectively equal WARD.

Consonant-consonant logic network 427, including AND gate 435, the details of which are shown in FIGURE 4B, will split a word if  $\alpha 1$  is a consonant,  $\beta 1$  is a consonant,  $\beta 2$  equals L or R,  $\beta 3$  is a vowel or a final E provided  $\alpha 2, \alpha 3$  and  $\alpha 4$  are not a vowel ending prefix ( $P_{v2}$ ). The vowel ending prefix input is denoted as encircled 5 which is obtained from previously described logic network 205.

Consonant-consonant logic network 429 including AND gate 435, the details of which are shown in FIGURE 18, splits a word if  $\alpha 1$  is a consonant,  $\beta 1$  is a consonant,  $\beta 1$  equals L or R,  $\beta 2$  is a vowel and either  $\alpha 1, \beta 1$  respectively equal TL or  $\alpha 1$  equals M, N, L or R provided  $\beta 2, \beta 3$  and  $\beta 4$  is not a vowel beginning suffix ( $S_v$ ). It should be noted that by definition weak consonants are defined M, N, L, R and W; however, the W was excluded as will be apparent from the detailed description of 429 shown in FIGURE 18C. As noted above, one of the counter-examples to the notion of splitting successive consonants occurs when the second consonant is weak. This rule provides the counter-counter-logic for the particular case when both of the consonants are weak.

Consonant-consonant logic network 431 including AND gate 435 the details of which are shown in FIGURE 18, split a word if  $\alpha 1$  is a consonant,  $\beta 1$  is a consonant,  $\beta 1$  is not an L or R,  $\beta 2$  is a vowel and either  $\alpha 1, \alpha 2, \alpha 3$  is a consonant ending prefix ( $P_c$ ) or neither  $\beta 2, \beta 3, \beta 4$  is a vowel beginning suffix ( $S_v$ ) nor  $\alpha 2, \alpha 3, \alpha 4$  is a vowel ending prefix ( $P_v$ ). This rule provides the necessary structure for the system to take into account the basic preferential relations that occur between and among the various affixing elements insofar as these elements affect the rules of hyphenation.

#### "S" Logic systems

On FIGURES 4C and 4D are illustrated by block diagram the "S" logic systems which make up an important part of Hyphenation Logic device 65. This aspect of the present invention consists of S-consonant logic system 405, S-vowel logic system 406, S-S logic system 407, consonant-S logic system 408 and vowel-S logic system 409.

The "S" consonant logic system 405 includes S-consonant logic network 437 and S-consonant logic network 439, the outputs of which are connected to OR gate 441, the output of which is connected to AND gate 443 along with  $\alpha 1$  equals S.

S-consonant logic network 437 including AND gate 443, the details of which are shown in FIGURE 19, split a word if  $\alpha 1$  is an S,  $\beta 1$  is a consonant,  $\alpha 1, \alpha 2, \alpha 3$  equals any of MIS, DIS, ---#ES and either or both

$\beta 2$  and  $\beta 3$  are vowels unless  $\beta 1$  is an H in which instance  $\beta 2$  must be a vowel. This rule is comparable to the rule involving consonantal ending prefixes in the consonant logic above. It provides the necessary logic to recognize the S-ending prefixes and make use of them for hyphenation points.

S-consonant logic network 439 including AND gate 443, the details of which are shown in FIGURE 19, splits a word if  $\alpha 1$  is an S,  $\beta 1$  is a consonant,  $\alpha 2$  is a vowel,  $\beta 1$  is not an H and  $\beta 2$  is a vowel, and  $\beta 2, \beta 3, \beta 4$  is not a vowel beginning suffix ( $S_v$ ) and  $\alpha 2, \alpha 3, \alpha 4$  is not a vowel ending prefix ( $P_{v2}$ ) and  $\alpha 3, \alpha 2$  is not respectively equal to MI, DI and #E. This rule provides a means of hyphenating between an S and a following consonant when the S is not the last letter of an S-ending prefix.

S-vowel logic system 406 includes S-vowel logic network 445 and AND gate 447 which receives the output of S-vowel logic network 445 along with a signal when  $\alpha 1$  equals S. S-vowel logic element 445 including AND gate 451, the details of which are shown in FIGURE 20, split a word if  $\alpha 1$  is an S,  $\beta 1$  is a vowel,  $\alpha 2$  is a vowel and  $\beta 1, \beta 2$  and  $\beta 3$  equal ING respectively. This rule is exactly the same as the vowel-consonant ING rule hereinabove discussed except that the consonant has been replaced by an S. The need for the preceding vowel is the same as before.

SS logic system 407 includes SS logic network 449 the output of which is connected to AND gate 451 along with the  $\alpha 1$  equals S condition. S-S logic network 449, including AND gate 451, the details of which are shown in FIGURE 21, will split a word if  $\alpha 1$  is an S,  $\beta 1$  is an S and  $\beta 2$  is a vowel.

Consonant-S logic system 408 includes consonant-S logic network 453 and consonant-S logic network 455 the outputs of which are applied to OR gate 457 the output of which is applied to AND gate 459 along with the input  $\beta 1$  and equals S.

Consonant-S logic network 453 including AND gate 459, the details of which are shown in FIGURE 22, will split a word if  $\alpha 1$  is a consonant,  $\beta 1$  is an S,  $\alpha 1, \alpha 2$  and  $\alpha 3$  is a consonant-ending prefix ( $P_c$ ) and either or both of  $\beta 2$  or  $\beta 3$  are vowels. This rule provides a means of recognizing consonant-ending prefixes when they occur before an S and allowing a hyphenation point after the prefix.

Consonant-S logic network 455 including AND gate 459, the details of which are shown in FIGURE 22, will split a word if  $\alpha 1$  is a consonant,  $\beta 1$  is an S, and  $\beta 2$  is either a vowel, H, C or T. It should be noted that in element 455 it is necessary to bring in a separate  $\alpha 1$  equals a consonant signal whereas in device 453 it is not necessary since the consonant ending prefix ( $P_c$ ) includes the condition where  $\alpha 1$  must equal a consonant.

Vowel-S logic system 409 includes vowel-S logic network 461 the output of which is applied to AND gate 463 along with the condition  $\beta 1$  equals S.

Vowel-S logic network 461 including AND gate 463, the details of which are shown in FIGURE 23, will split a word if  $\alpha 1$  is a vowel,  $\beta 1$  is an S and  $\beta 1, \beta 2, \beta 3, \beta 4$  is an S suffix ( $S_s$ ) which is defined as follows: ( $S_s$  = sial, sian, sien, sion, siou or sive). This rule is comparable to the corresponding rule in the vowel consonant logic and takes advantage of the fact that certain S-beginning sequences are almost invariably split off.

Inhibit logic device 410 functions to prevent a signal from appearing at the output of AND gate 499 even though one of the above-mentioned hyphenation logic systems applies a signal to OR gate 498 thereby indicating a proper split condition. No output signal from inhibit logic device 410, the details of which are shown in FIGURE 24, will occur under the following circumstances:  $\beta 1$  equals  $G_o$  or  $\alpha 1$  equals # or  $\alpha 2$  equals # or when  $\alpha 3$  equals # and  $\alpha 2$  is not a vowel and  $\alpha 1$  is not a vowel or when  $\alpha 4$  equals # and  $\alpha 3$  is not a vowel and  $\alpha 2$  is not a vowel and  $\alpha 1$  is not a vowel. These conditions insure that (1) one never permits a syllable to begin with one of the

typically "ending" sequences; (2) that one always has a preceding vowel so that the word segment prior to the hyphenation point is actually a possible syllable; and (3) that one never splits off a single letter from the following sequence, a generally accepted dictum of the printing trade.

#### Detailed logic networks of class decoder 63

The following description pertains to the details of the various logic networks of class decoder 63. As previously indicated, class decoder 63 has two basic parts, the first of which relates to the logic networks for implementation of the  $S_v$ ,  $S_c$ ,  $P_c$  and  $P_v$  conditions and the second consists of networks which provide  $G_e$  group and vowel-consonant information about the different cells of the shift register. The information provided at the input of the  $S_v$ ,  $S_c$ ,  $P_c$ ,  $P_v$  logic networks (201-205) is obtained directly from character decoders 47 through 55 and the information provided at the inputs of the  $G_e$  group and vowel-consonant logic networks (206-209) is obtained directly from the flip-flops of each cell of shift register 43.

#### Vowel beginning suffixes ( $S_v$ ) logic network

In FIGURES 6A and 6B is illustrated the specific logic networks of vowel beginning suffix ( $S_v$ ) logic network 201 which provide an output signal when the following character strings exist at the  $\beta_2$ ,  $\beta_3$  and  $\beta_4$  positions of shift register 43:

al#, ed#, ee#, er#, es#, y#, ly#, abl, age, anc,  
ant, ari, ary, ato, enc, ern, ery, est, ian, ibl,  
ice, ily, ine, ing, ish, ism, ist, ite, itu, ity,  
ize, ous, oid, ope, ose, ier, ies, ied

Network 201 includes AND gates 211 through 221 the inputs of which are illustrated in FIGURE 6A. OR gates 223 through 227 have inputs as illustrated and outputs which are respectively connected to AND gates 213, 214, 216, 218 and 221. AND gates 211, 212 and 213 are connected to the input of OR gate 228 the output of which is connected to AND gate 229 the other input of which  $\beta_2$  equals O. The outputs of AND gates 214 through 221 are connected to the input of OR gate 231 the output of which is connected to AND gate 232 along with the  $\beta_2$  equals I input. AND gate 233 has the inputs illustrated and its output along with those of AND gates 229 and 232 are respectively illustrated by X—X, Y—Y, and Z—Z which are connected to the input S of OR gate 235 of FIGURE 6B.

Referring now to the logic networks of FIGURE 6B, it can be seen that AND gates 236 through 241 have inputs as illustrated including these inputs from OR gates 243 and 244 which are respectively applied to AND gates 239 and 241. Each of AND gates 236 through 241 have their outputs connected to OR gate 246 the output of which is connected to AND gate 247 along with the input signal  $\beta_2$  equals A. AND gates 248 and 250 through 253 have the inputs as illustrated and AND gates 251 through 253 also receive input signals from OR gates 254, 255, 256, respectively, which have input signals as illustrated. The outputs of each of AND gates 250 through 253 are connected to OR gate 258 the output of which is connected to AND gate 259 along with the input signal  $\beta_2$  equals E. The output of each of AND gates 247, 248 and 259 is connected to the input of OR gate 235.

It can therefore be seen that a signal will occur at the output of OR gate 235 when  $\beta_2$ ,  $\beta_3$ ,  $\beta_4$  are equal to the above-mentioned  $S_v$  character groups. However, this information is not, per se, useful by the hereinbefore-described rules and is therefore transmitted to inverter 261 which will provide an output when  $\beta_2$ ,  $\beta_3$  and  $\beta_4$  are not a vowel-beginning suffix  $S_v$ .

A typical example of the operation of vowel-beginning suffix logic shown in FIGURES 6A and 6B is as follows: Assuming that  $\beta_2$ ,  $\beta_3$ , and  $\beta_4$ , respectively, equal ING. When this condition prevails a  $\beta_3$  equals N signal will be

applied to the input of OR gate 231,  $\beta_4$  equals G signal will be applied to the input of AND gate 217 and  $\beta_2$  equals I signal will be applied to the input of AND gate 218 and  $\beta_2$  equals I signal will be applied to the input of AND gate 232. Accordingly, a signal will appear at the output of AND gate 232 which will be applied to the input of OR gate 235 and inverter 261 will invert this signal to a zero output condition. It can be seen that this information is then used by vowel-consonant logic network 412 which will indicate a split condition if  $\alpha_1$  equals U,  $\beta_2$  is a vowel and  $\beta_2$ ,  $\beta_3$  and  $\beta_4$  is not a vowel-beginning suffix ( $S_v$ ). Assume that the word CHARTIST is being examined. In this instance  $\beta_2$ ,  $\beta_3$ ,  $\beta_4$  is a vowel-beginning suffix ( $S_v$ ) and therefore the word is not split between the  $\alpha_1$  and  $\alpha_2$  letters.

#### Consonant-beginning suffixes ( $S_c$ )

The consonant-beginning suffix logic network 202 is illustrated in FIGURE 7 and has application only with vowel-consonant logic system 402. That is, if  $\alpha_1$  is a vowel and  $\beta_1$  is a consonant then the word is split if  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$ ,  $\beta_4$  is a consonant-beginning suffix ( $S_c$ ). The following letter strings are defined as consonant-beginning suffixes ( $S_c$ ):

cial, cian, cien, cion, cive, ciou,  
gial, gian, gien, gion, give, giou,  
tial, tian, tien, tion, tive, tiou,  
bil, cal, mat, ment, fore, tory, ture,  
ful, less

Consonant-beginning suffix logic network 202 includes AND gates 272 through 275 having the illustrated inputs and the outputs being connected to OR gate 276. AND gate 272 also receives a signal from OR gate 277 which has the illustrated inputs. OR gate 278 has the illustrated inputs and its output is connected to AND gate 279 which has inputs from OR gate 276 and the  $\beta_2$  equals I condition. AND gates 280, 281 and 282 have inputs as illustrated and have their outputs connected to OR gate 283 the output of which is connected to AND gate 284 along with the signal  $\beta_3$  equals L. AND gates 285 and 286 have inputs as illustrated and their outputs are connected to OR gate 287 the output of which is connected to AND gate 288 along with  $\beta_2$  equals E. AND gates 290, 291 and 292 have inputs as illustrated and have outputs connected to OR gate 293 the output of which is connected to AND gate 294 along with  $\beta_3$  equals R. AND gate 295 has the input  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$  respectively equal to MAT. AND gates 284, 279, 288, 294 and 295 have their inputs connected to OR gate 296 the output of which is directly connected to OR gate 417 of FIGURE 4A and FIGURE 16.

#### Consonant-ending prefixes ( $P_c$ )

The consonant ending prefixes  $P_c$  logic network is illustrated in FIGURE 8 and has application in consonant-consonant logic network 431 and in consonant-S logic network 453. For example, in consonant-S logic network 453 if  $\alpha_1$  is a consonant and  $\beta_1$  is the letter S, then the word will be split provided  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  is a consonant-ending prefix ( $P_c$ ) and either or both  $\beta_2$  and  $\beta_3$  are vowels. The following strings are the consonant ending prefixes ( $P_c$ ):

#ac, #ad, #al, #ex, out, sub, sun,  
#em, #en, #im, #in, #un, com, con

Consonant ending prefixes logic network 203 includes AND gates 301 through 304 the inputs of which are as illustrated and the outputs of which are connected to OR gate 305. OR gate 306 has inputs as illustrated and the output is connected to the input of AND gate 303 along with the signal  $\alpha_2$  equals A. OR gate 307 performs the function of transmitting its output signal to OR gate 308 and to AND gates 309 and 301 which have the illustrated inputs. OR gate 311, having inputs  $\alpha_1$ =N or B has its output connected to AND gate 312 along with the in-

puts illustrated. AND gates 313 and 314 have the illustrated inputs. AND gates 309, 312, 313 and 314 have their outputs connected to the input of OR gate 315.

#### Vowel-ending prefixes ( $P_v$ )

The vowel-ending prefix logic networks 204 and 205 are illustrated in FIGURES 9 and 10 and are identical in logic except that the inputs to the FIGURE 9 network utilize signals from  $\alpha 1$ ,  $\alpha 2$  and  $\alpha 3$  whereas the signals to the FIGURE 10 network are from  $\alpha 2$ ,  $\alpha 3$  and  $\alpha 4$ . The same letter combinations are used in both instances. From FIGURES 3 and 4 it can be seen that the vowel-ending prefixes  $P_{v1}$  and  $\neq P_{v2}$  are used in conjunction with the logic networks 413, 427, 431 and 439. The following letter strings are considered vowel-ending prefixes ( $P_v$ ): re, be, de, pro.

Vowel-ending prefix ( $P_{v1}$ ) logic network 204 is shown in FIGURE 9 and includes OR gate 321 having the inputs  $\alpha 2$  equals R, B, and D the output of which is connected to AND gate 322 along with  $\alpha 1$  equals E the output of which is then connected to OR gate 323. In addition, AND gate 324 has inputs  $\alpha 1$  equals O,  $\alpha 2$  equals R and  $\alpha 3$  equals P the output of which is connected to OR gate 323.

Vowel-ending prefix ( $P_{v2}$ ) logic network 205 is shown in FIGURE 10 and includes OR gate 327 having the inputs  $\alpha 3$  equals R, B and D the output of which is connected to AND gate 328 along with  $\alpha 1$  equals E the output of which is then connected to OR gate 329. In addition, AND gate 330 has inputs  $\alpha 2$  equals O,  $\alpha 3$  equals R and  $\alpha 4$  equals P the output of which is connected to OR gate 329. The output of OR gate 329 is then inverted by inverter 331 to provide an output signal when  $P_{v2}$  does not equal a vowel-ending prefix.

From the foregoing it can be seen that the word FRE/QUENCY and the word RE/TREAT would be broken in the manner illustrated since this rule does not require that the letter strings be preceded by a space or punctuation.

#### Vowel logic (V)

The details of vowel logic network 206 are illustrated in FIGURE 11 and, as previously indicated, this network receives its input signals directly from the flip-flops of each cell of shift register 43. This network includes AND gates 338 through 344 which have the inputs as illustrated. The coding for these inputs is such that the bar above either the  $\alpha$  or  $\beta$  symbols indicate a zero output condition of the sensed flip-flop whereas if no bar appears, then the output of the flip-flop is a "one." The number subscripts coupled with the  $\alpha$  and  $\beta$  symbols indicate the following: the first number identifies the cell, the second number is only for separation and is not significant and the third number identifies the particular flip-flop in the cell as identified by the first number. As previously indicated, the letters A, E, I, O, U and Y are vowels. However, it is important to note that a final E is not a vowel but is considered as a consonant. The particular coding of the vowels is illustrated in Table II, FIGURE 5B. The vowels in the vowel class (V) have common code  $X_3=0$ ,  $X_4=0$  and  $X_5=0$ . The code represented by the bits  $X_0$ ,  $X_1$  and  $X_2$  denote the particular vowel and are not important in the consideration of the vowel logic network of FIGURE 11. Typical operation of the FIGURE 11 network is as follows: Referring to AND gate 339 assume that the third, ( $X_3$ ), fourth, ( $X_4$ ), and fifth ( $X_5$ ) flip-flops of the  $\beta 4$  cell are respectively in the 0, 0 and 0 states, then there will be an output signal from AND gate 339 which indicates that  $\beta 4$  is equal to a vowel in accordance with the code shown in Table II. For reasons which will hereinafter become apparent, it is important to know when  $\beta 4$  is not a vowel and therefore inverter 345 is employed to provide an output signal when  $\beta 4$  is not a vowel. That is, when any of flip-flops  $X_3$ ,  $X_4$  or  $X_5$  of the  $\beta 4$  cells are in a "one" state,

then  $\beta 4$  is not a vowel and inverter 345 will provide an output signal. It should be noted that AND gate 239 is actually operated by actually applying a positive signal to its inputs when it is desired to obtain a positive signal from its output. Although the inputs indicate zero conditions of the  $X_3$ ,  $X_4$  and  $X_5$  flip-flops of the  $\beta 4$  cell, the actual signals obtained from these flip-flops will be "one" signals and this is possible since the other side of the zero state of the flip-flop is a "one" state and it is only a matter of wiring to achieve this desired result. The vowel information obtained from vowel logic device 206 is used in nearly all of the logic networks of the hyphenation logic device shown in FIGURES 4A through 4D and FIGURES 15 through 24.

#### Weak consonant logic (W)

The details of weak consonant logic network 207 are illustrated in FIGURE 12. This network provides information about weak consonants which consist of the consonants N, R, W, M and L. Weak consonant information is used in connection with the logic networks 413, 415 and 429. As in the case of vowel logic network 206, the input conditions of the weak consonant network are obtained directly from the flip-flops of each cell of shift register 43 and the input coding shown in FIGURE 12 is interpreted in the manner above described with relation to the FIGURE 11 logic network.

Logic network 207 includes AND gates 351, 352 and 353 having inputs as illustrated. OR gates 354, 355 and 356 receive signals from AND gates 357, 358 and 359 and each of these gates have input conditions as illustrated. AND gates 360, 361 and 362 receive input signals respectively from OR gates 354, 355, 356 and from AND gates 351, 352 and 353. Output signals from AND gates 360, 361 and 362 respectively indicate that  $\alpha 1$ ,  $\beta 1$  and  $\beta 4$  equal weak consonants. However, inverters 363 and 364 are provided to indicate when  $\beta 1$  and  $\beta 4$  are respectively not weak consonants. It will be appreciated from observing the inputs to the logic network of FIGURE 12 that signals will be received when flip-flops  $X_3$ ,  $X_4$  and  $X_5$  are respectively in the 1, 0, 0 states of the  $\alpha 1$ ,  $\beta 1$  and  $\beta 4$  cells.

In order to minimize the number of logic elements within the system, it is desirable to have the shortest code possible and to achieve this result it is often necessary to overlap the bit states of the particular code in question. Therefore, the logic network 207 is provided with AND gates 357 and OR gate 354 which will prevent an output signal from occurring from AND gate 360 even though flip-flops  $X_3$ ,  $X_4$  and  $X_5$  of the  $\alpha 1$  cell are equal respectively to 001. From Table I it can be seen that if the  $X_2$  flip-flop of the  $\alpha 1$  cell is equal to a zero, then outputs will be obtained for the weak consonants N, R, W, M and L since that signal is applied directly to OR gate 354. A signal will also be received from the output of OR gate 354 when flip-flops  $X_0$  and  $X_1$  of the  $\alpha 1$  cell are in the "0" states since they are applied directly to AND gate 357. Accordingly, output signals will be received from AND gate 360 only when weak consonant conditions exist even though other codes may have flip-flops  $X_3$ ,  $X_4$  and  $X_5$  respectively in the 0, 0 and 1 states as is the case with strong consonants, final E, B, and C. A similar analysis is applicable to AND gate 358 and OR gate 355 for the  $\beta 1$  condition and for AND gate 359 and OR gate 356 for the  $\beta 4$  condition.

#### Consonant logic for $\beta 1$ cell

Logic network 208 which is shown in FIGURE 13 provides information as to whether or not the  $\beta 1$  cell contains a consonant or is not a member of the  $G_e$  group. By definition a consonant (C) is equal to any of the strong consonants (S), or weak consonants (W) or the letter strings in the  $G_b$  or  $G_e$  groups. (See Tables I, II and III.) AND gate 372 is connected to  $X_3$ ,  $X_4$  and  $X_5$  flip-flops of the  $\beta 1$  cell and will provide an output signal when the

$X_3$ ,  $X_4$  and  $X_5$  flip-flops are respectively in the 1, 1 and 0 states which is the  $G_e$  code set forth in Table II of FIGURE 5B. The output of AND gate 372 is transmitted to inverter 373 and to OR gate 374. Inverter 373 will provide an output signal when  $\beta 1$  is not  $G_e$  which information is necessary in the inhibit logic shown in FIGURE 24. Also connected to OR gate 374 is  $\beta 105$  which will result in an output signal from OR gate 374 when the  $X_5$  flip-flop of the  $\beta 1$  cell is in the 1 position which code identifies string consonants (S), weak consonants (W) and the  $G_b$  letter group as defined in Table I. Therefore, OR gate 374 will indicate a  $\beta 1$  consonant condition when there exists either a strong consonant (S), a weak consonant (W), a  $G_b$  group or a  $G_e$  group.

#### Consonant logic for $\alpha 1$ cell

Logic network 209 which is shown in FIGURE 14 provides information as to whether or not the  $\alpha 1$  cell contains a consonant or is a member of the  $G_e$  group. This network includes AND gate 381 and OR gate 382. AND gate 381 includes the inputs  $\alpha 103$ ,  $\alpha 104$  and  $\alpha 105$  and will therefore provide an output when the  $G_e$  letter groups occur in the  $\alpha 1$  cell. The output of AND gate 381 is made directly available to the consonant-vowel logic network 417. In addition, this signal is applied to the input of OR gate 382 along with the  $\alpha 105$  signal and therefore OR gate 382 will indicate a consonant condition at cell  $\alpha 1$  when there exists either a strong consonant (S), a weak consonant (W), a  $G_b$  group or a  $G_e$  group.

#### Hyphenation system

The basic function of the logic networks of the hyphenation system is to break or hyphenate words with a high degree of accuracy in accordance with the teachings of the present invention. In view of the foregoing discussion pertaining to class decoder 63, it will be appreciated that the logic network of the hyphenating system is dependent upon and must be considered with the information provided by the class decoder. All of the information provided to the input networks of the hyphenation system is in the form of single pulses as distinguished from the 6 bit information used in logic networks 206, 207, 208 and 209 of the class decoder. It should be particularly noted that the hyphenation system is a parallel network as distinguished from serial network. That is, each of the logic networks within the system simultaneously receives information about the information contained in the  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  cells if such information is part of that required by the logic network in question. This same parallel system is also employed in the class decoder since it receives its information in parallel from the cells of shift register 43 and from character decoders 47 through 55.

Referring now to the hyphenation system as generally illustrated in FIGURES 4A through 4D and specifically illustrated in FIGURES 15 through 24, it will be particularly appreciated that there are three primary forms of logic networks. These are (1) where the  $\alpha 1$  and  $\beta 1$  cells must contain vowel and consonant information in any of the combinations vowel-vowel, vowel-consonant, consonant-vowel or consonant-consonant and (2) the special "S" rules wherein the  $\alpha 1$  and  $\beta 1$  cells take the form of "S"-C consonant, "S"-V vowel, "S", "S", consonant-"S", vowel-"S" and (3) the inhibit logic which takes into account special conditions.

The first grouping of logic networks evolve about consonant and vowel information. It has been discovered that it is possible to properly split words by examining each side of the split location for consonant and vowel conditions and once these conditions have been sensed, it is then possible to apply a special set of constraints by examining the remaining characters in the letter string. This is achieved by logic systems 401, 402, 403 and 404. As has been disclosed hereinabove, it has been found appropriate to organize the system according to the vowel-consonant-S situa-

tion in the  $\alpha 1$ ,  $\beta 1$  positions. Thus many of the rules, if separately stated, would require a specific determination of which class the  $\alpha 1$  and  $\beta 1$  characters fell in. By organizing the rules in terms of the various possibilities that can occur, the system is simplified and the internal bookkeeping in our computer simulation provides a simpler structure for the machine itself. Nearly the same result would have occurred if we had determined every rule separately and then searched for the simplest logical structure for accomplishing all the rules simultaneously.

The second grouping of logic networks centers around the letter "S" which, according to the present invention, is not considered as either a vowel or a consonant. The letter "S," as distinguished from conventional grammatical treatments, is considered a special case and by using this special case in the present invention, it is possible to realize a greater number of and more accurate split conditions.

The third primary logic network pertains to inhibit logic which functions to prevent the splitting of words when certain conditions exist. This inhibiting occurs even though one of the other logic networks indicates that there is a proper split condition. Stated in a different manner, the over-all logic consists of a set of "split-now" rules and "don't-split-now" rules and a set of precedence relations to determine which overrides in combinations.

The vowel-vowel logic network 401 is generally shown in FIGURE 4A and specifically shown in FIGURE 15. The vowel-consonant logic system 402 is generally illustrated in FIGURE 4A and is specifically illustrated in FIGURE 16. The consonant-vowel logic system 403 is generally shown in FIGURE 4B and is specifically shown in FIGURE 17 and the consonant-consonant logic system 404 is generally shown in FIGURE 4B and is specifically shown in FIGURE 18. The "S" logic networks consist of the "S" consonant logic network 405 (FIGURES 4C and 19), the "S" vowel logic network 406 (FIGURES 4C and 2B), the "S" "S" logic network 407 (FIGURES 4C and 21), the consonant-S logic network 408 (FIGURES 4D and 22) and the vowel-S logic network 409 (FIGURES 4D and 23).

#### Vowel-vowel (VV) logic network

The details of the vowel-vowel logic network 401 are illustrated in FIGURE 15 and include AND gates 383 and 384 having the inputs illustrated and having the outputs connected to OR gate 385. The output of OR gate 385 is connected to AND gate 386 along with the inputs  $\alpha 1$  equals a vowel and  $\beta 2$  equals N. It will be noted that the output of AND gate 386 is applied to OR gate 498 which will then indicate whether or not there is a proper split condition. The logic network set forth in FIGURE 15 is the implementation of the rule that if  $\alpha 1$  is a vowel and  $\beta 1$  is a vowel and  $\beta 1$ ,  $\beta 2$  and  $\beta 3$  equals ING or ONE, then the word should be split. From FIGURE 15 it can be seen that  $\beta 1$  equals a vowel is not used in the logic since  $\beta 1$  equals the vowel "I" which is applied to AND gate 383. The same is applicable to  $\beta 1$  equals O. Therefore, there will be no output signal from AND gate 386 unless  $\beta 1$  was either the vowel "I" or the vowel "O."

#### Vowel-consonant (VC) logic networks

Vowel-consonant logic network 412 is illustrated in FIGURE 16 and consists of an AND gate to which the signals  $\alpha 1=U$ ,  $\beta 2=\text{vowel}$  and  $\beta 2$ ,  $\beta 3$ ,  $\beta 4 \neq S_v$ . This is referred to as the U rule and results in splitting a word when  $\alpha 1=U$ ,  $\beta 1=\text{a consonant}$ ,  $\beta 2=\text{a vowel}$  and  $\beta 2$ ,  $\beta 3$  and  $\beta 4$  is not a vowel-beginning suffix ( $S_v$ ) which was previously described. It will be appreciated that the output of AND gate 412 applied to OR gate 417 the output of which is applied to AND gate 418 along with  $\alpha 1=\text{a vowel}$  and  $\beta 1=\text{a consonant}$ . From this it can be seen that the vowel-consonant logic for the  $\alpha 1$  and  $\beta 1$  cells are applied directly to AND gate 418 and therefore reduces the logic necessary for the vowel-consonant logic networks 412, 413, 414 and 415.

Logic network 413, shown in FIGURE 16, includes AND gates 391, 392 and 393. The inputs to the AND gates are as illustrated and result in splitting a word if  $\alpha 1$ =a vowel,  $\beta 1$ =a consonant,  $\alpha 1, \alpha 2, \alpha 3$  is a vowel ending prefix ( $P_v$ ),  $\beta 1$  is not the letter C or the letter V,  $\beta 2$  is a vowel ( $\neq U$ ),  $\beta 3$  is  $\neq S$  and  $\beta 4$  is  $\neq S$ , weak consonant (W), vowel (V) or #.

Logic network 414 provides the necessary logic for splitting a word where the letter strings  $\alpha 3, \alpha 2, \alpha 1$  respectively equal ANY or  $\alpha 4, \alpha 3, \alpha 2$ , and  $\alpha 1$  respectively equal SOME where  $\beta$ =a vowel,  $\alpha 1$ =a vowel and  $\beta 1$ =a consonant. This is implemented by using AND gates 394 and 395 having the inputs illustrated. The outputs of these AND gates are connected to OR gate 396 the output of which is connected to AND gate 397 along with the input  $\beta 2$ =a vowel. The output of AND gate 397 is then connected to OR gate 417 the output of which is connected to AND gate 418 which is then connected to OR gate 498.

Logic network 415 consists of AND gate 398 the output of which is connected to OR gate 417 which is connected to AND gate 418 along with  $\alpha 1$ =a vowel and  $\beta 1$ =a consonant. The inputs into AND gate 398 are as illustrated and will, along with the inputs to AND gate 418, split a word provided  $\alpha 1$ =a vowel,  $\beta 1$ =a consonant,  $\beta 2$  and  $\beta 3=L\phi$  and  $\beta 1$  is not a weak consonant.

There is a fifth situation in which the word is split provided  $\alpha 1$ =a vowel,  $\beta 1$ =a consonant and  $\beta 1, \beta 2, \beta 3$  and  $\beta 4$  is a consonant-beginning suffix ( $S_c$ ). This is illustrated in FIGURE 16 as being a signal from logic element 202 of FIGURES 3 and 7 which is applied to OR gate 417 the output of which is applied to AND gate 418 along with  $\alpha 1$ =a vowel and  $\beta 1$ =a consonant.

#### Consonant-vowel (CV) logic networks

Consonant-vowel logic system 403 consists of the logical implementation of three basic rules the details of which are set forth in FIGURE 17.

The first logic network 417A consists of OR gate 417A having the inputs  $\alpha 1=G_e$  and  $\alpha 1=X$ . The output of OR gate 417A is applied to OR gate 421, the output of which is applied to AND gate 423 along with  $\alpha 1$ =a consonant and  $\beta 1$ =a vowel. Accordingly, a word will be split provided  $\alpha 1$ =a consonant,  $\beta 1$ =a vowel and either  $\alpha 1=X$  or  $\alpha 1=G_e$ .

In FIGURE 17 is also illustrated the EVER rule which is implemented by AND gate 418A wherein  $\beta 1, \beta 2, \beta 3, \beta 4$  equals EVER.

The next rule of the consonant-vowel logic network is implemented by the use of AND gate 419 which has the inputs  $\beta 1=O, \beta 2=N, \beta 3=G$  and  $\alpha 2$ =a vowel. The output of AND gate 419 is applied to OR gate 421 the output of which is applied to OR gate 423.

#### Consonant-consonant (CC) logic network S

Consonant-consonant logic system 404 consists of four different logic networks 425, 427, 429, 431. Consonant-consonant logic network 425 consists of AND gates 511 and 512 having the illustrated inputs and outputs of which are connected to OR gate 513 the output of which is connected to OR gate 433 the output of which is connected to AND gate 435 along with the inputs  $\alpha 1$ =a consonant and  $\beta 1$ =a consonant. Network 425 results in splitting a word when  $\alpha 1$ =a consonant,  $\beta 1$ =a consonant and  $\beta 1, \beta 2, \beta 3$  and  $\beta 4$ =MENT or WARD.

Logic network 427 includes OR gates 514 and 515 the outputs of which are connected to AND gate 516 along with the input  $\alpha 2, \alpha 3, \alpha 4 \neq P_{v2}$ , (vowel ending prefix). Therefore, a word will be split provided  $\alpha 1$ =a consonant,  $\beta 1$ =a consonant,  $\beta 2=L$  or R,  $\beta 3$ =a vowel or a final E provided  $\alpha 2, \alpha 3, \alpha 4$  is not a vowel prefix.

Logic network 431 includes AND gate 523 having the illustrated inputs the output of which is connected to OR gate 524 having the input  $\alpha 1, \alpha 2, \alpha 3$ =a consonant ending prefix ( $P_c$ ), OR gate 525 has the inputs  $B1=L$

and  $B1=R$ , the output of which is applied to inverter 526. The outputs of OR gate 524, inverter 526 and the condition  $\beta 2$ =a vowel are applied to AND gate 527 the output of which is connected to OR gate 433. Therefore, a word will be split provided  $\alpha 1$ =a consonant,  $\beta 1$ =a consonant,  $\beta 1$  is not equal to L or R,  $\beta 2$  is a vowel and either  $\alpha 1, \alpha 2, \alpha 3$  is a consonant ending prefix ( $P_c$ ) or neither  $\beta 2, \beta 3, \beta 4$  is a vowel beginning suffix ( $S_v$ ) nor  $\alpha 2, \alpha 3, \alpha 4$  is a vowel ending prefix ( $P_v$ ).

#### The letter "S" logic networks

It has been discovered that the letter "S" plays a unique position in linguistics and has several interrelationships with other letter groupings, vowel and consonant conditions which bear upon correct hyphenation. Because of its unique position, it has been classified neither as a consonant or a vowel according to the present invention.

In FIGURE 19 is illustrated the "S" consonant logic system 405 which includes logic networks 437 and 439. Logic network 437 includes AND gates 542 through 544 having the illustrated inputs wherein the output of each is connected to OR gate 545 the output of which is connected to AND gate 546. AND gate 547, having the inputs illustrated, is connected to OR gate 548 along with  $\beta 2$ =a vowel the output of which is connected to AND gate 546. AND gate 546 has the input  $\beta 1$ =a consonant and has its output connected to OR gate 441 the output of which is connected to AND gate 443 along with the input  $\alpha 1=S$ . The output of AND gate 443 is then connected to the input of OR gate 498. It can therefore be seen that logic network 437 will split a word if  $\alpha 1=S, \beta 1$ =a consonant,  $\alpha 1, \alpha 2, \alpha 3$ =mis, dis, # es and either or both  $\beta 2, \beta 3$  are vowels, unless  $\beta=h$  in which case  $\beta 2$  must be a vowel.

Logic network 439 includes AND gate 549 having the inputs illustrated the output of which is connected to AND gate 550. Network 439 also obtains information from OR gate 545 of network 437 which is applied to inverter 551 the output of which is applied to AND gate 552 along with the inputs illustrated. The output of AND gate 552 is connected to AND gate 550 along with  $\beta 1$ =a consonant the output of which is connected to OR gate 441 which is then connected to AND gate 443. From this it can be seen that a word will be split provided  $\alpha 1=S, \beta 1$ =a consonant,  $\alpha 2$ =a vowel,  $\beta 1$  is not an H,  $\beta 2$ =a vowel. Unless  $\beta 2, \beta 3, \beta 4$  is a vowel beginning suffix or  $\alpha 2, \alpha 3, \alpha 4$  is a vowel ending prefix provided, however, that  $\alpha 1, \alpha 2, \alpha 3$  does not equal MIS, DIS, # ES.

In FIGURE 20 is illustrated the "S" vowel logic network which is a special situation involving the letter string ING. The inputs to AND gate 445 are as illustrated and its output is connected to AND gate 447 along with  $\alpha 1=S$ . Therefore, a word will be split provided  $\alpha 1=S, \beta 1$ =a vowel,  $\alpha 2$ =a vowel and  $\beta 1, \beta 2, \beta 3=ING$ . It will be noted that it is not necessary to insert  $\beta 1$ =a vowel into this logic network since it is  $\beta 1=I$  which is a more limited case but falls within the definition of a vowel.

In FIGURE 21 is illustrated logic network 407. This logic network is implemented by applying  $\beta 1=S$  and  $\beta 2$ =a vowel to AND gate 449 the output of which is connected to AND gate 451 along with  $\alpha 1=S$ . It can therefore be seen that a word will be split if  $\alpha 1=S, \beta 1=S$  and  $\beta 2$ =a vowel.

Consonant-S logic network 408 is shown in FIGURE 22 and includes consonant-S logic network 453 and consonant-S logic network 455. Network 455 includes OR gate 567 having the illustrated inputs and its output connected to AND gate 568 along with  $\alpha 1$ =a consonant. The output of AND gate 468 is applied to OR gate 457 the output of which is connected to AND gate 459 along with  $\beta 1=S$ . From this it can be seen that a word will be split if  $\alpha 1$ =a consonant,  $\beta 1=S$ , and  $\beta 2$  is either a vowel, H, C, or T.

Logic network 453 includes OR gate 569 having the inputs  $\beta 2$ =a vowel and  $\beta 3$ =a vowel the output of which is connected to AND gate 570 along with the input condition  $\alpha 1, \alpha 2, \alpha 3=P_c$ . AND gate 570 is connected to OR gate 457 the output of which is connected to AND gate 459 which is connected to OR gate 498 the output of which is connected to AND gate 499. From this it can be seen that a word will be split provided  $\alpha 1$ =a consonant,  $\beta 1$ =an S,  $\alpha 1, \alpha 2, \alpha 3$  is a consonant ending prefix ( $P_c$ ) and either or both of  $\beta 2$  and  $\beta 3$  are vowels.

The last "S" rule logic network (409) is illustrated in FIGURE 23 and includes AND gates 578, 579 and 580 having the illustrated inputs and their outputs connected to OR gate 581 the output of which is connected to AND gate 582 along with  $\beta 2=I$  the output of which is connected to OR gate 583. OR gate 584 has the inputs illustrated and its output connected to AND gate 585 along with the inputs  $\beta 2=I$  and  $\beta 4=N$ . AND gate 585 is connected to OR gate 583 the output of which is connected to AND gate 586 along with  $\alpha 1$ =a vowel, the output of which is connected to AND gate 463 along with  $\beta 1=S$ . AND gate 463 is connected through OR gate 498 to AND gate 499 which will provide an output signal if a hyphenation condition exists. It can therefore be seen that the function of logic network 409 is to split words when  $\alpha 1$ =a vowel,  $\beta 1$ =an S, and  $\beta 1, \beta 2, \beta 3, \beta 4$  is an S suffix of the following types: sial, sian, sien, sion, siou.

In FIGURE 24 is illustrated inhibit logic network 410 which includes OR gate 591, having the inputs illustrated, which is connected to AND gate 592 along with  $\alpha 3 \neq \#$  the output of which is connected to OR gate 593 along with  $\alpha 1$ =a vowel and  $\alpha 2$ =a vowel. OR gate 593 is connected to AND gate 594 along with the inputs  $\beta 1 \neq G_e, \beta 1 \neq \#, \alpha 2 \neq \#,$  and  $\alpha 2=0$ . The output of AND gate 594 is connected to the input of AND gate 499 and will inhibit a split condition in the instances where inhibit logic 410 so dictates.

#### Hyphenation machine

In FIGURE 25 is illustrated another embodiment of the present invention which provides a continuous hyphenated output. This embodiment is a hyphenating machine which provides a continuous output wherein each word is hyphenated at all possible points in accordance with the logic networks of the present invention. That is, there will be no justifying procedures but rather, there will only be a continuous hyphenated output. This machine has particular use where an individual wishes to consider all hyphenation points without having to memorize or consult dictionaries.

The embodiment shown in FIGURE 25 has many elements that are identical to elements shown in the previously-described system. The like elements in these embodiments are represented by the same reference numerals and will not be discussed in detail. Basically, the two systems are identical in the following areas: (1) character encoder 15, (2) sequence decoding-encoding system 10 (some modification in coding to identify particular letter strings), (3) shift register 43, (4) character decoders 47-55, (5) class decoder 63, and (6) hyphenation logic. The primary area of difference is that the FIGURE 25 embodiment does not have justification logic, but rather, it includes hyphenation output logic.

#### Operation

The general operation of the FIGURE 25 hyphenation machine is as follows: Clock device 601 will activate tape reader 13 through AND gate 617 and tape reader control 603. The output of tape reader 13 is encoded by character encoder 15 the output of which is then transmitted in parallel to sequence encoder 17 and sequence encoder cell 19 of sequence encoder system 10. The detailed considerations of sequence encoder system 10 has been previously considered and will therefore not be discussed at this time. As will hereinafter become apparent, it is necessary to identify the particular letters in each

letter string which was not required in the previously-described embodiment. In Table IV, FIGURE 26, is illustrated the code which is used in the present embodiment to identify each letter string as well as grouping. It should be noted that in Table IV there is a separate code for each letter string of the character groups  $G_b$  and  $G_e$ . This was not the case in the previously-described system since the  $G_b$  and  $G_e$  codes related to character widths and did not designate the letter strings. Letter string designation was not important in that embodiment since each letter of the letter string was stored on the tape which was later read back during the read-out operation. However, in the present embodiment it is necessary to identify the particular letters in the letter strings since each character is separately read-out. This is achieved by conventional logic coding techniques in sequence decoder 605 which provides an encoded output for each letter of the letter string as coded in Table IV of FIGURE 26. It should also be noted that it is possible to use essentially the same codes as used in the  $G_b$  and  $G_e$  letter groups of Tables II and III. However, it was necessary to add additional codes to the  $G_b$  group to take into account the DG and QU letter strings and it was unnecessary to use the last two codes of the  $G_e$  letter string of Table II since there were only five  $G_e$  letter strings. Width information is not necessary in the present embodiment.

In FIGURE 27 is illustrated a timing diagram which generally represents timing signals of the system shown in FIGURE 25. Assuming there is no output from any of output device control 607, hyphen encoder 609 or upper case decoder-encoder 611, then each of inverters 613, 614 and 615, which are respectively connected to these outputs, will provide positive signals to the input of AND gate 617. When this condition prevails and time  $T_0$  occurs (see the timing diagram of FIGURE 27), then clock 601 will provide an output signal to AND gate 617 which will result in actuating tape reader control 603. When this occurs a new character, at time  $T_1$ , will be shifted into character encoder 15 and sequence encoder system 10. For all practical purposes all electrical signals of the hyphenation system will simultaneously start their occurrence at time  $T_1$ . That is, the information in the cells will be shifted at time  $T_1$  and at this same time ( $T_1$ ) hyphenation logic device 65 will provide an output signal indicating that a split should occur between the new information contained in the  $\alpha 1$  and  $\beta 1$  cells if so dictated by the hyphenation logic. The reason for the delay between time  $T_0$  and  $T_1$  is because tape reader control 603 and tape reader 13 include mechanical linkage systems which have an inherent time delay usually on the order of several milliseconds. The time  $T_2$  represents the time at which the signal will occur at the output of output device control 607. The reason for the delay between time  $T_1$  and time  $T_2$  is that output device control 607 also includes mechanical linkage which results in a time delay of several milliseconds. All other time delays within the system are of the order of microseconds. All of the pulses shown in FIGURE 27 terminate at time  $T_3$  and the delay between time  $T_3$  and the delay between time  $T_3$  and time  $T_0$ , is to prevent systems oscillation. It should be noted that clock 601 may provide internal clock pulses at a very rapid rate of, for example, several microseconds duration. However, counters are used to provide the necessary signals for control of the various logic elements within the system. For example, a delay is provided between time  $T_3$  and time  $T_0$ , which is achieved by using a counter in clock 601 having the desired time delay.

Assume, for purposes of the present discussion, that the word "PRIVATE" has been shifted from cells  $\alpha 1$  through  $\alpha 3$  and  $\beta 1$  through  $\beta 4$  to cells  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 3$ . This will occur at time  $T_1$  and sequence encoder 605 will not decode the letter V, since it is not a letter string code, and the binary code for the letter V will be transmitted directly to AND gate 621 through sequence decoder 605. Assuming the letter V is a lower

case letter, then there will be no output from upper and lower case decoder-encoder 611 the details of which will hereinafter be described. Therefore, inverter 623 will provide an output signal which is applied to AND gate 621 along with the letter V output from the  $\alpha 1$  cell. As a result the binary code for the letter V will be transmitted to character decoder-encoder 625 which will then provide a binary output to the input of output device control 607 which is compatible with the code of this device. It will be appreciated that character decoder-encoder may in certain instances be eliminated if the output device control is responsive to binary code which is generated by character encoder 15. However, it will generally be found that it is desirable to use an encoder-decoder so that the internal codes need not be identical to the codes of commercially available output devices. Output device control 607 transmits the code, along with other control signals, to output device 627. Simultaneous with the transfer of the letter V from the  $\beta 1$  cell to the  $\alpha 1$  cell sequence class decoder 63 and hyphenation logic 65 will examine all of the characters being shifted in cells  $\alpha 1$  through  $\alpha 4$  and  $\beta 1$  through  $\beta 4$  as hereinbefore described with respect to the previous embodiment. Accordingly, the hyphenation logic will provide an output signal, since the word "PRIVATE" is properly split between V and A, which is transmitted to flip-flop 629 which is then set in the "one" position and therefore will continue to apply a signal to AND gate 631 until it is reset by the output signal from hyphen decoder and flip-flop reset 633. During that period of time ( $T_1$  through  $T_3$ ) when output device control 607 is transmitting the letter "V" output code to output device 627, inverter 635 will cause no signal to be applied to AND gate 631 since flip-flop 637 is not set in the "one" state. However, at time  $T_3$ , the signal from output device control 607 becomes zero which causes an output signal from inverter 635 which sets flip-flop 637 in the "one" state. Therefore, an output signal will be applied to AND gate 631 from flip-flop 637 until such time that flip-flop 637 is reset by the output signal from device 633. As a result, at time  $T_0'$  AND gate 631 will provide an output signal to hyphen encoder 609 which will provide a binary output signal to the input of output device control 607 which will result in print-out of a "hyphen" by output device 627. Simultaneous with the transfer of the output signal from hyphen encoder 609 the same signal is applied to inverter 614 which causes a zero signal to be applied to AND gate 617 which therefore prevents the tape reader control 603 from causing the READ-IN process to start at time  $T_0'$ . This is the case even though clock 601 provides an output signal at time  $T_0'$  which would otherwise cause the machine to shift to the next character. The function of hyphen decoder and flip-flop reset 633 is to sense the hyphen signal being transmitted from output device control 607 to output device 627 and to cause an output signal to occur at the termination of this hyphen signal. This is achieved by conventional techniques which sense the trailing edge as distinguished from the leading edge of the transferred signal. When the hyphen signal has terminated, device 633 provides an output signal which causes both flip-flop 629 and flip-flop 637 to reset to their original zero states. Therefore, AND gate 631 will no longer provide an output signal and the machine is then able to proceed to the next character which will occur at time  $T_0'$ . It can therefore be seen that "hyphens" are inserted into the output while the input is stopped for one cycle of operation.

The function of upper case decoder-encoder 611 is to provide information as to (1) whether or not the character shifted into  $\alpha 1$  cell was an upper or lower case character or (2) to determine whether or not either or both of the  $\alpha 1$  and  $\beta 1$  characters are upper or lower case. Assuming the letter "V" was an upper case character, this would have been decoded by device 611 which would provide an output which would set flip-flop 641 in a "one" state which results in a zero output from inverter 623.

Therefore, no code would have been transmitted from sequence decoder 605 through AND gate 621 to character decoder-encoder 625. As a result, a signal will be transmitted to output device control 607, during a full-cycled operation, which will indicate that the letter V was an upper case letter. Therefore, output device 627 will be set in the upper case state and will be ready to receive the immediately-following characters which will then have the proper case assigned. After the transmission of the upper case designation by device 607, case shift decoder and flip-flop reset 645 will provide an output signal (by sensing the trailing edge) which will reset flip-flop 641 to the "zero" state. This will permit transmission of the code for the letter "V" through device 605 during the next cycle of operation. Therefore, character decoder-encoder 625 will encode the letter "V" to make it compatible with the coding requirements of output device control 607. This signal will persist during the full clock pulse and at its termination the inputs to AND gate 617 will all be positive such that clock 601 may cause the system to shift to the next character. It should be noted that when an upper case condition exists that a zero output signal will be obtained from inverter 615 which will block AND gate 617 during the transmission of the upper case signal.

Assuming that a two or three letter sequence is received and stored in the  $\alpha 1$  cell, the code representing this sequence will be transmitted to sequence decoder 605 and to upper case decoder-encoder 611. It is not possible to know by the coding illustrated in Table IV whether or not the second letter of the letter string is an upper case letter. Therefore, the  $\beta 1$  cell is sensed to determine whether or not the character in that cell is an upper or lower case character. If the  $\alpha 1$  and  $\beta 1$  cells both indicate an upper case character condition then all of the sequence stored in the  $\alpha 1$  cell is considered to be upper case. However, if only the  $\alpha 1$  cell indicates an upper case condition and the  $\beta 1$  cell indicates a lower case condition, then only the first letter of the letter string stored in the  $\alpha 1$  cell is in the upper case. The detailed mechanization for providing the upper and lower case outputs with respect to the letter string stored in sequence decoder 605 is not shown since its implementation would be obvious to one skilled in the art. However, it should be particularly noted that the coding set forth in Table IV specifically identifies the exact letter string under consideration as well as the number of letters in the specific string. Read-out from sequence decoder 605 will be one character at a time from the sensed letter string and the internal logic of devices 605, 611 and 607 are designed in either of the following manners: The first mode would be to have device 611 indicate an upper case condition, then sequence decoder 605 would print out the first character, then device 611 would again indicate an upper case condition, then sequence decoder 605 would print out the next character and if a third character were involved, device 611 would again indicate an upper case condition and sequence decoder 605 would transmit the third and final character. The second mode of operation would be to have device 611 indicate whether 1, 2 or 3 upper case characters were to follow and device 607 would provide a memory for the number of characters to have the upper case. Then sequence decoder 605 would sequentially transmit each letter of the letter string to character decoder-encoder 625 which would then transmit the information to device 607 and finally to output device 627 with the upper case designation.

In view of the foregoing, it can be seen that a digital hyphenation machine is provided which will continuously hyphenate words which are transmitted to its input. This makes it possible for typographers or other individuals who are interested in breaking words to do so by only examining the output of the machine. Therefore, these individuals will not find it necessary to consult dictionaries or to memorize the proper hyphenation points of

words in question. Finally, the tape employed may be fed to an existing justifying machine.

It is to be understood in connection with this invention that the embodiments shown are only exemplary, and that various modifications can be made in construction and arrangement within the scope of the invention as defined in the appended claims. For example, it is possible to modify the embodiment shown in FIGURE 1 by replacing tape punch 11, tape reader 13 and tape reader control 79 with respectively a magnetic tape write, memory storage device and a memory control device. The end result would be substantially the same as that in the illustrated embodiment, but the structure and operation would be different.

What is claimed is:

1. An electronic hyphenation system for automatically hyphenating a string of characters of textual material comprising an input device providing electrical information about each of the characters in said string of characters, control means connecting with said input device for repeatedly selecting a point of investigation in said string of characters, storage means connecting with said input device for storing said electrical information about said string of characters, digital logic circuit means, including a plurality of interconnected preset character logic circuits, connecting with said storage means and said control means for combined encoding as character groups of selected characters in said string of characters and for applying without regard to word completeness selected character groups and characters at selected points on both sides of said point of investigation to said plurality of preset character logic circuits to automatically provide an output hyphenation signal when hyphenation is proper at said point of investigation in said string of characters, said character logic circuits being preset and interconnected in accordance with a plurality of predetermined hyphenation logic equations operative solely from the positions and identities of a limited sample of characters in said string of characters, said storage means and said digital logic circuit means operating independently of whole word comparisons, and output means controlled by said output hyphenation signal from said digital logic circuit means for transmitting said string of characters with proper hyphenation inserted therein.

2. The hyphenation system of claim 1 further including automatic line justifying means integral said control means, said storage means and said digital logic circuit means, and wherein said control means includes character counter means.

3. A hyphenation system for automatically hyphenating a string of characters comprising an output device, an input device providing electrical information about each of the characters in said string of characters, the output of said input device being connected to a sequence encoder system which provides electrical information about sequence of characters, the output of said sequence encoder system being connected to a storage means for storing a string of characters and character sequences, first means for examining character sequences generated by said sequence encoder system and stored in said storage means, second means for examining other characters stored in said storage means, said first and second means being operatively connected to one another and said storage means to provide output information signals which determine where said characters are to be split, a control device for controlling said input device and said output device, said output information signals being operatively connected to said control device for indicating where characters are to be split, said storage means being operatively connected to said control device for transmitting electrical information which defines the characters stored therein, said control device being operatively connected to said input device for applying a control signal thereto to prevent its operation when a word is to be split as indicated by said output information signals from said first and second

means; said control device being operatively connected to said output device for applying a control signal thereto for transmitting said string of characters with proper hyphenation inserted therein.

4. A hyphenation system for hyphenating a string of characters as defined in claim 3 wherein said control device is further defined as including a line length counting device, a tape reader control device, a space counting system and a space insertion logic device, said line length counting device being connected to said tape reader control device and said space insertion logic device, and said space counting system being connected to said space insertion logic device, said tape reader control device being connected to said input device, and said space insertion logic device being connected to said output device.

5. A hyphenating-justifying system for hyphenating and justifying a string of characters of textual material comprising an input device providing electrical information about each of the characters in said string of characters, said input device having a memory containing the characters transmitted from its output, the output of said input device being connected to a sequence encoder system which provides electrical information about sequences of characters, the output of said sequence encoder system being connected to a storage means for storing a string of characters and character sequences, first means for examining character sequences generated by said sequence encoder system and stored in said storage means, second means for examining other characters stored in said storage means, said first and second means being operatively connected to one another and said storage means to provide an output information signal which determines where said characters are to be split, said output information signal being operatively connected to a justifying means for indicating where characters are to be split, said storage means being operatively connected to said justifying means for transmitting electrical information which defines the width of characters transmitted to said storage means, said justifying means operatively connected to said input device for applying a control signal thereto to prevent its operation when an end of line condition exists, third means responsive to the output of said input device for causing said input device to return to a predetermined position for a readout, said justifying means providing signals to said input device during read-out which causes line justification and hyphen insertion when required.

6. A hyphenation-justifying system as defined in claim 5 wherein said justifying means is further defined as including a line length counting system, a tape reader control device, a space insertion logic device, and a space counting system, said line length counting system being connected to said tape reader control device and said space insertion logic device, said space counting system being connected to said space insertion logic device, and said tape reader control device being connected to said input device and said third means.

#### References Cited

##### UNITED STATES PATENTS

3,046,527	7/1962	Rowley et al. ....	340—146.3
2,961,644	11/1960	Garliner .....	340—172.5
3,103,580	9/1963	Foreman .....	235—159
3,289,176	11/1966	Garth et al. ....	340—172.5
3,299,408	1/1967	An Wang et al. ....	340—172.5
3,307,154	2/1967	Garth et al. ....	340—172.5
3,312,953	4/1967	An Wang et al. ....	340—172.5
3,328,764	6/1967	Sorensen et al. ....	340—172.5
2,717,686	9/1955	Seeber .....	197—19
3,297,124	1/1967	Sims .....	197—19

##### OTHER REFERENCES

Lawson, A.: Computers Will Be a Major Factor in Tomorrow's Typesetting. In *Inland Printer/American Lithographer*. 151(5): pp. 66—67. August 1963.

(Other references on following page)

## OTHER REFERENCES

Lawson, A.: Installations Prove Advantages of Computerized Typesetting. In *Inland Printer/American Lithographer*. 151(6): pp. 100-101. September 1963.

Automatic Type Composition by Computer With IBM 1620 Data Processing System. In *Inland Printer/American Lithographer*, 152(1) p. 96. October 1963.

Lawson, A.: First Fully Automatic Computer-Directed Typesetting System. In *Inland Printer/American Lithographer*, 152(5): p. 63. February 1964.

Electronics Review—In the newspapers. In *Electronics*. 37(13): pp. 27-28. Apr. 6, 1964.

Pannullo, J. N.: New Computers Are Changing Printing Industry Production Techniques. In *Inland Printer/American Lithographer*. 153(3): pp. 60-66. June 1964.

5 ROBERT C. BAILEY, *Primary Examiner*.

JOHN P. VANDENBURG, *Assistant Examiner*

U.S. Cl. X.R.

10 197—11, 19; 235—151.22; 340—146.3