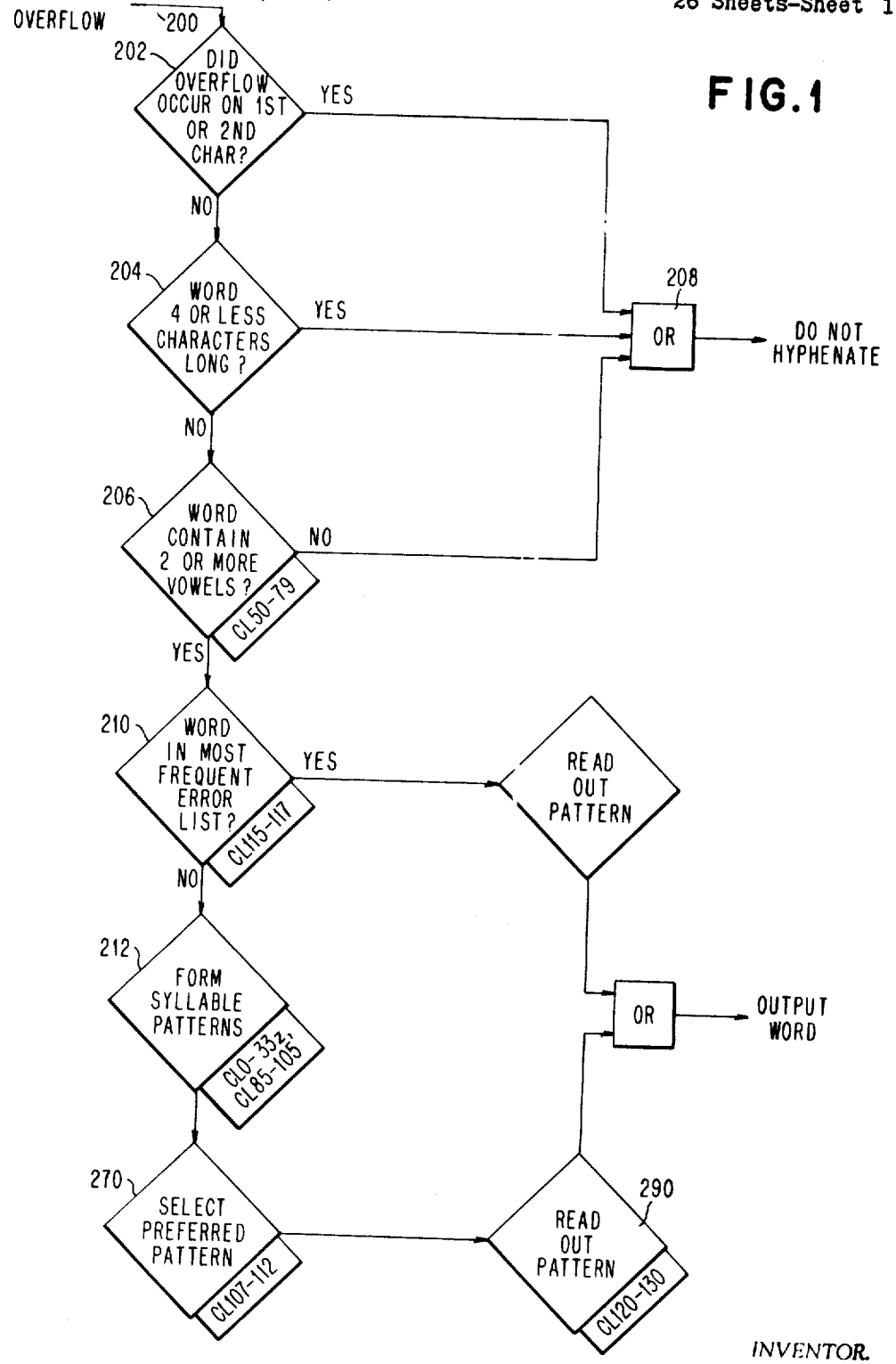
AUTOMATIC HYPHENATION SCHEME

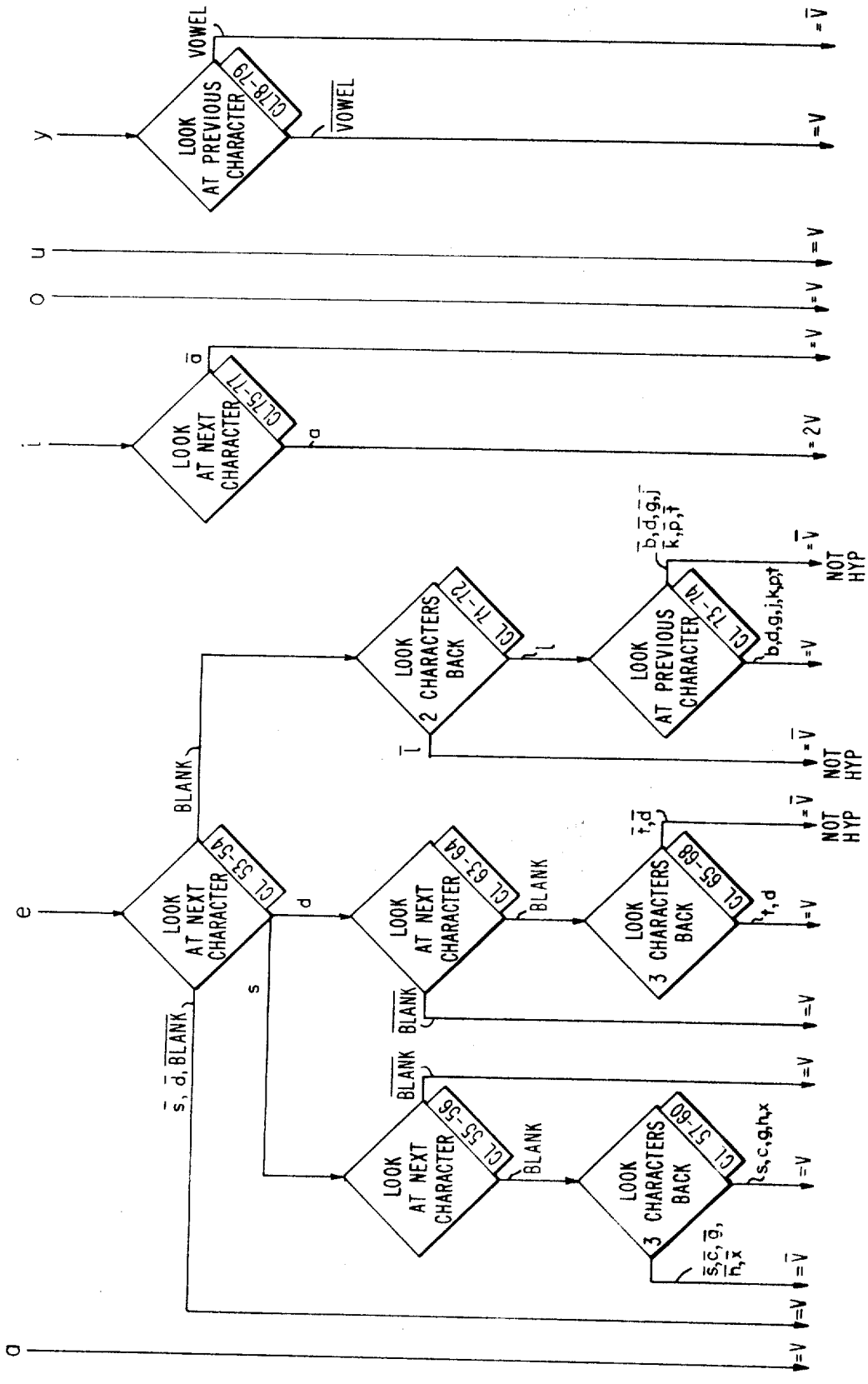
Original Filed June 17, 1964

FIG. 1



INVENTOR.
FREDERICK J. DAMERAU
BY Donald J. Knowlton
ATTORNEY

FIG. 2
FLOW CHART – VOWEL GROUP COUNTING



FORM SYLLABLE PATTERNS

FIG.3A

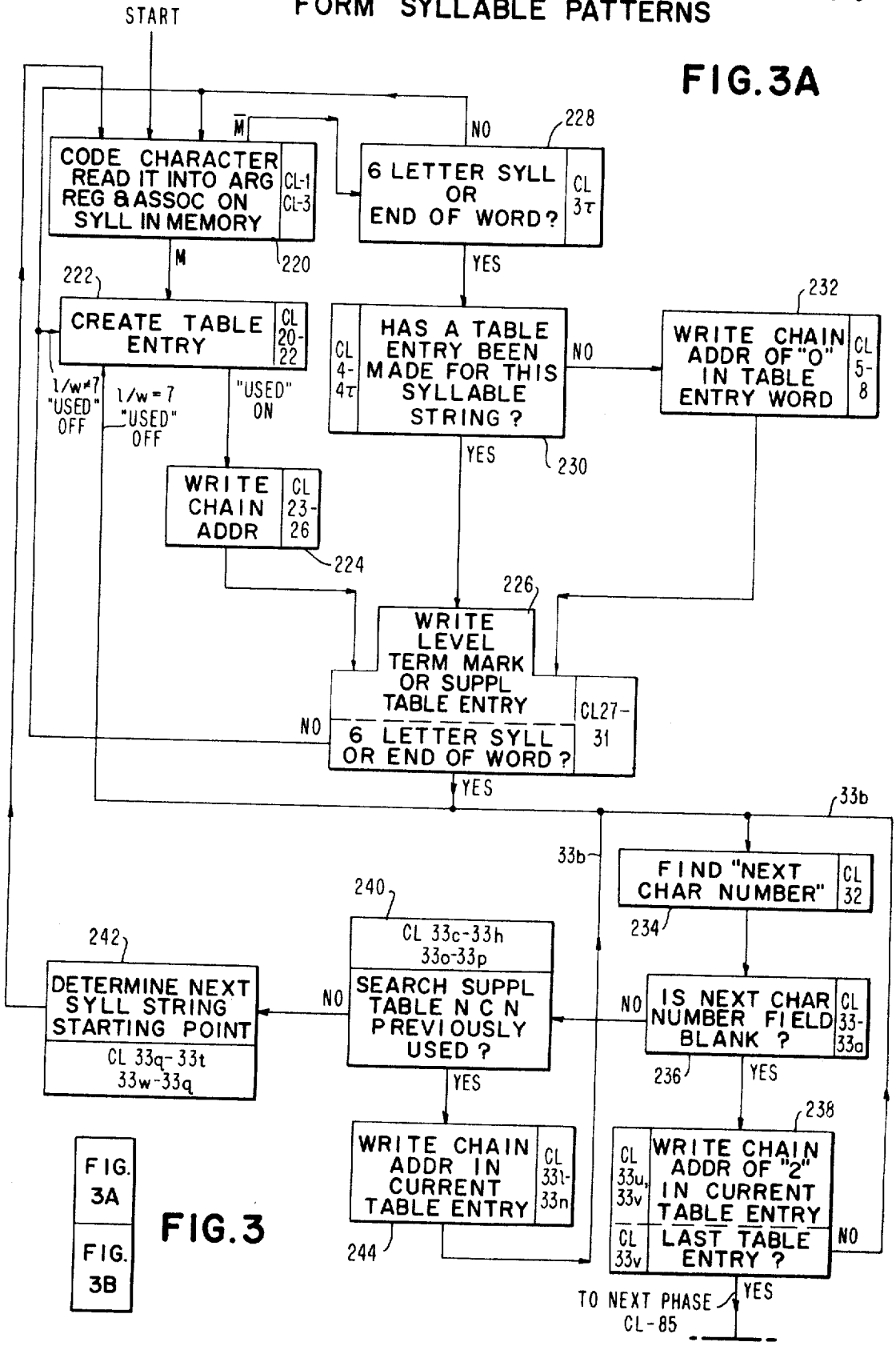


FIG. 3A
FIG. 3B

FIG.3

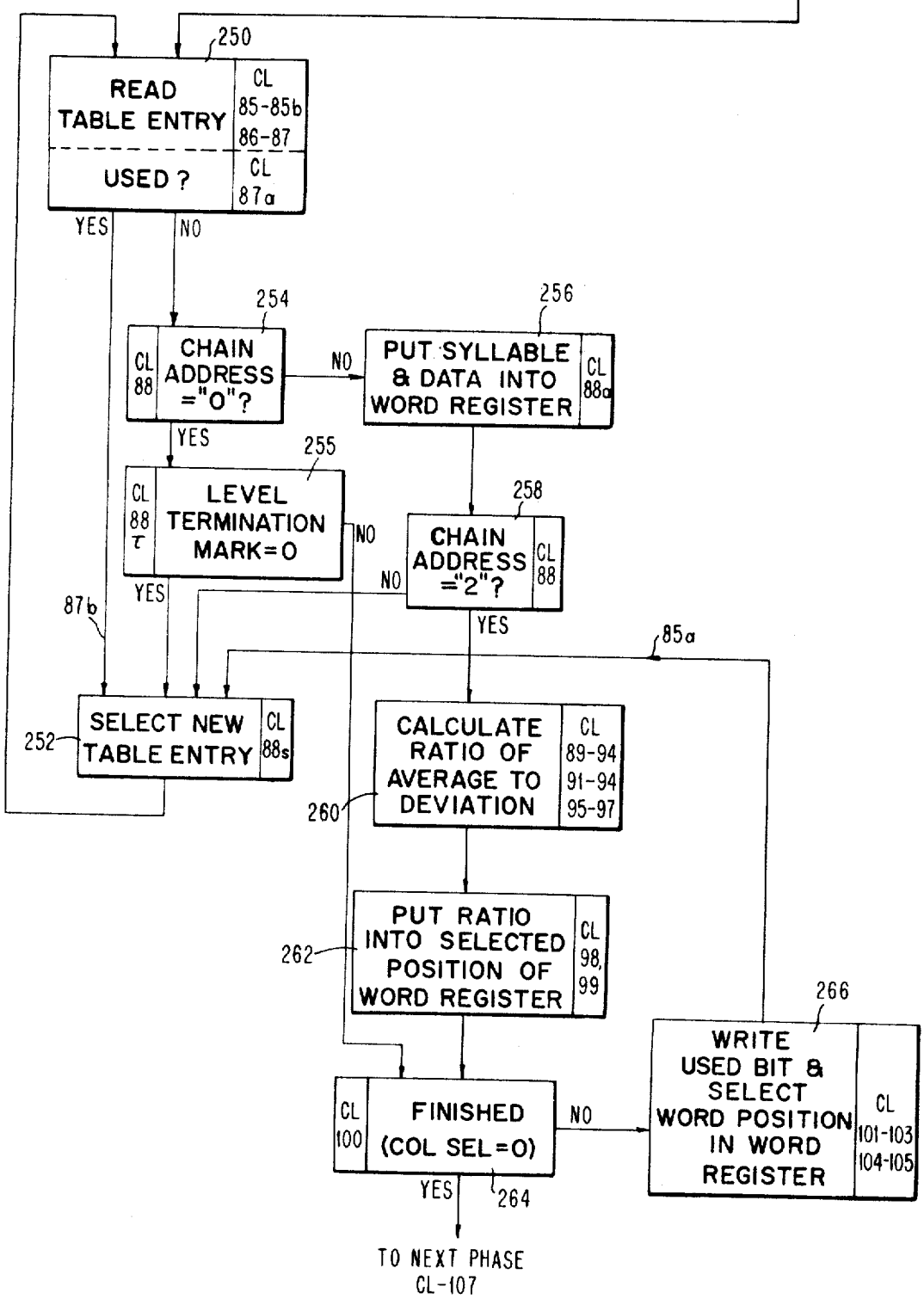
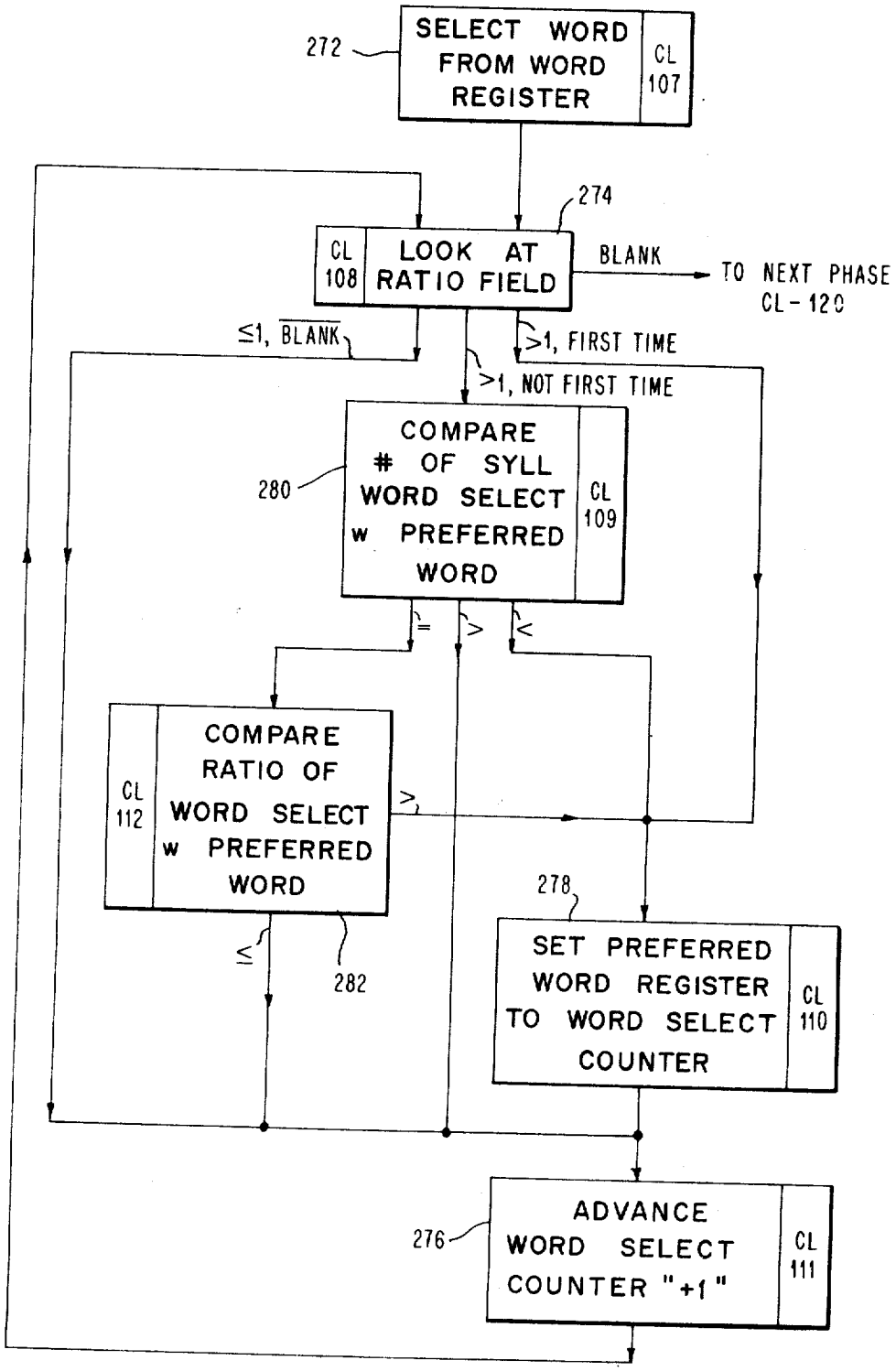

250
READ TABLE ENTRY
CL 85-85b 86-87
USED ?
CL 87a
YES
NO
254
CL 88
CHAIN ADDRESS ="0"?
NO
256
PUT SYLLABLE & DATA INTO WORD REGISTER
CL 88a
YES
255
CL 88 τ
LEVEL TERMINATION MARK=0
NO
258
CHAIN ADDRESS ="2"?
CL 88
NO
YES
87b
YES
85a
252
SELECT NEW TABLE ENTRY
CL 88s
260
CALCULATE RATIO OF AVERAGE TO DEVIATION
CL 89-94 91-94 95-97
262
PUT RATIO INTO SELECTED POSITION OF WORD REGISTER
CL 98, 99
266
WRITE USED BIT & SELECT WORD POSITION IN WORD REGISTER
CL 101-103 104-105
CL 100
FINISHED (COL SEL=0)
NO
YES
264
TO NEXT PHASE CL-107

FIG. 3B

FIG. 4
SELECT PREFERRED PATTERN



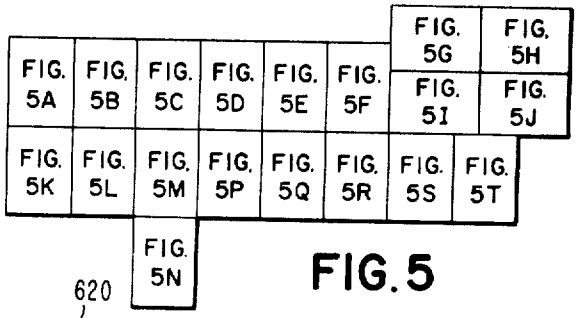


FIG. 5A
FIG. 5B
FIG. 5C
FIG. 5D
FIG. 5E
FIG. 5F
FIG. 5G
FIG. 5H
FIG. 5I
FIG. 5J
FIG. 5K
FIG. 5L
FIG. 5M
FIG. 5P
FIG. 5Q
FIG. 5R
FIG. 5S
FIG. 5T
FIG. 5N

FIG. 5

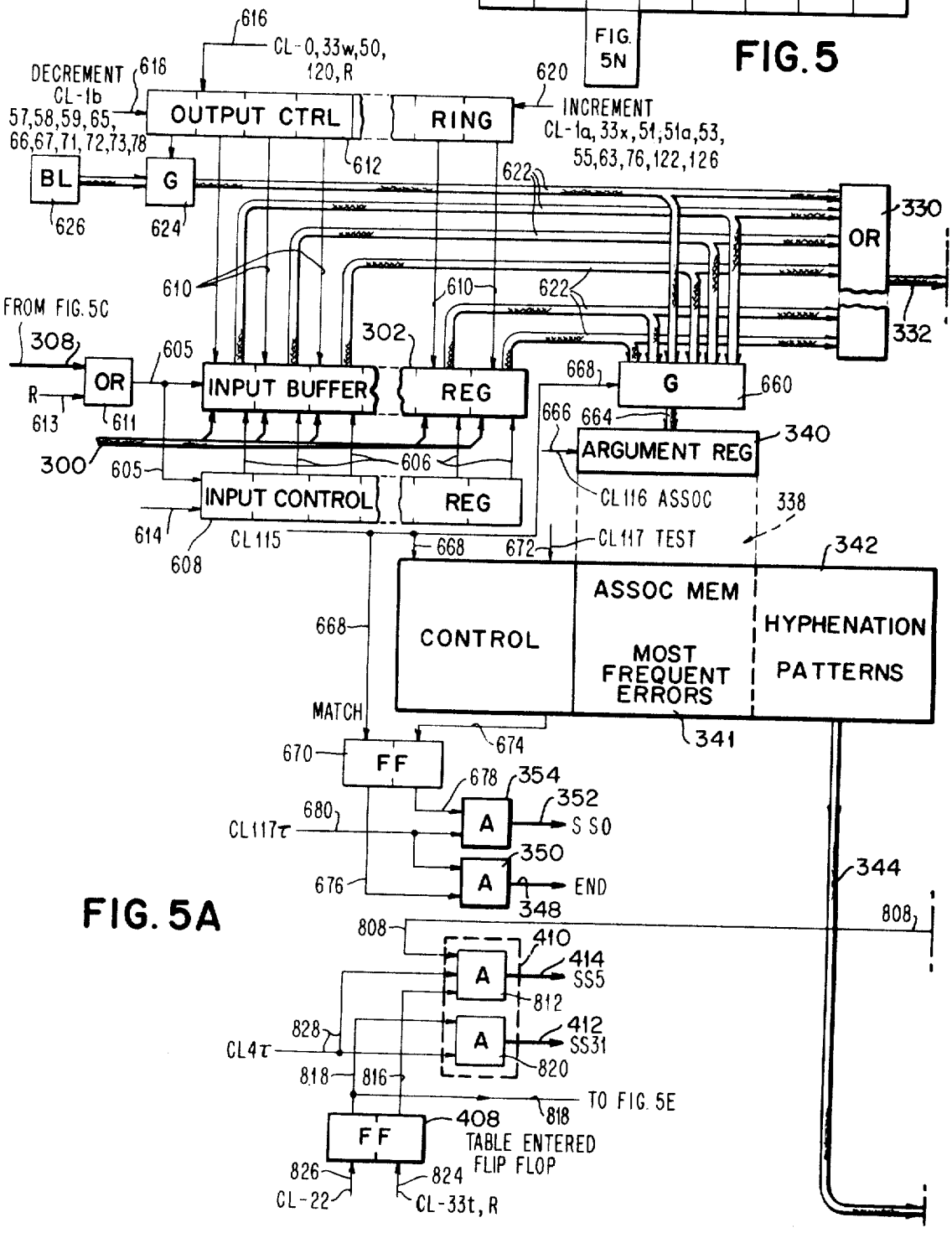

616
CL-0,33w,50,
120,R
DECREMENT
618
CL-1b
57,58,59,65,
66,67,71,72,73,78
620
INCREMENT
CL-1a,33x,51,51a,53,
55,63,76,122,126
OUTPUT CTRL
RING
612
BL
G
626
624
622
330
OR
332
610
610
622
FROM FIG. 5C
308
302
605
OR
INPUT BUFFER
REG
668
G
660
R
613
611
666
664
ARGUMENT REG
340
300
605
606
INPUT CONTROL
REG
CL116 ASSOC
338
614
608
CL115
668
672
CL117 TEST
342
668
CONTROL
ASSOC MEM
MOST
FREQUENT
ERRORS
HYPHENATION
PATTERNS
MATCH
341
670
FF
674
678
354
352
680
CL117τ
A
S SO
350
676
A
END
348
344
808
808
410
414
SS5
812
828
A
412
SS31
CL4τ
818
816
820
TO FIG. 5E
818
408
FF
TABLE ENTERED
FLIP FLOP
826
824
CL-22
CL-33t,R

FIG. 5A

FIG.5B

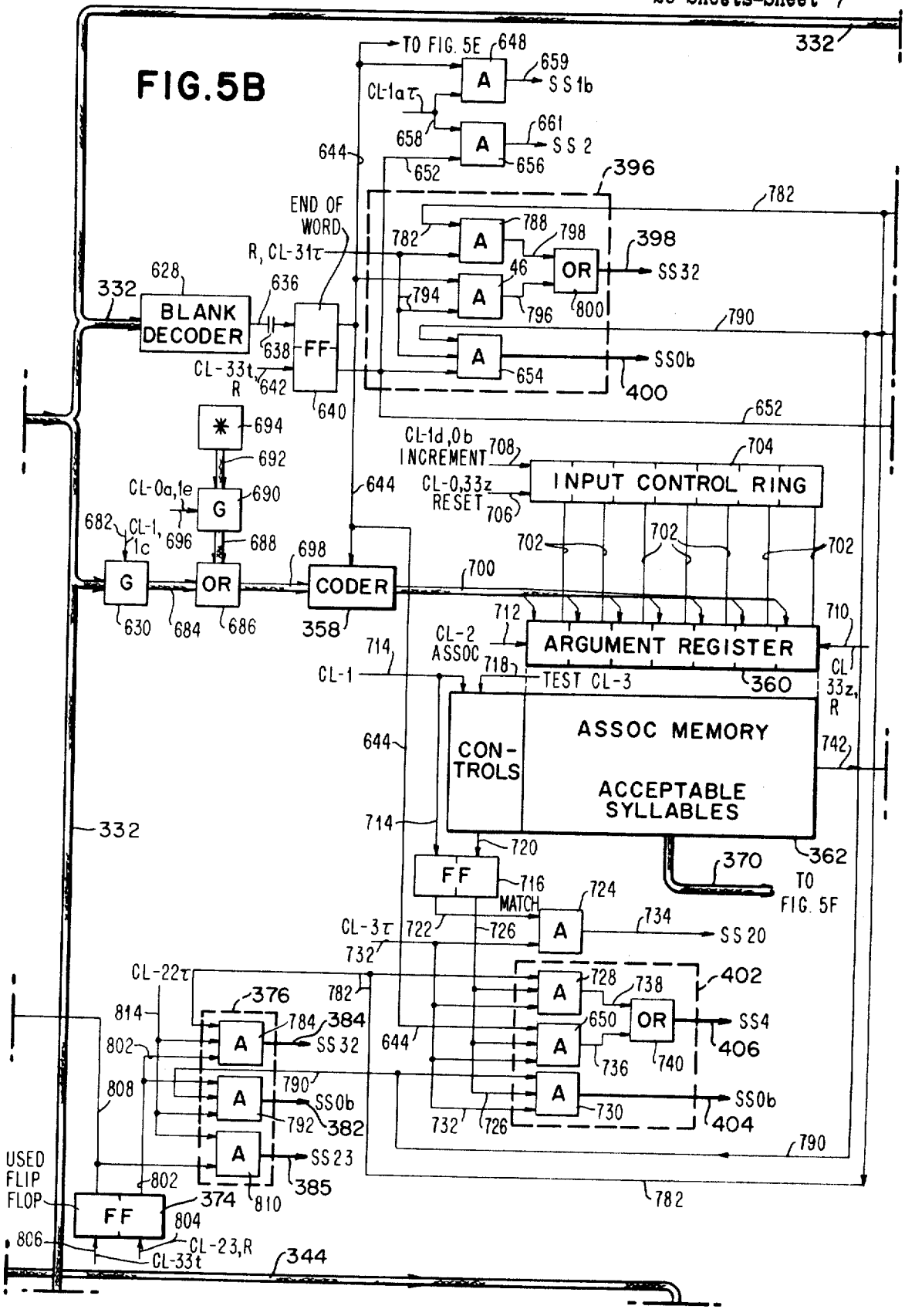

TO FIG. 5E
648
659
SS1b
CL-1aτ
658
661
SS 2
656
652
644
396
782
END OF WORD
R, CL-31τ
788
798
398
SS32
46
OR
794
796
800
790
SS0b
654
400
652
628
BLANK DECODER
636
638
FF
CL-33t, R
642
640
694
692
CL-0a,1e
690
G
682
CL-1, 1c
696
688
698
CODER
700
630
684
686
358
CL-1d,0b INCREMENT
708
704
INPUT CONTROL RING
CL-0,33z RESET
706
702
712
CL-2 ASSOC
ARGUMENT REGISTER
710
CL 33z, R
714
CL-1
718
TEST CL-3
360
CON-TROLS
ASSOC MEMORY
ACCEPTABLE SYLLABLES
742
720
FF
716
MATCH
724
370
362
TO FIG. 5F
CL-3τ
722
726
734
SS 20
732
CL-22τ
376
728
738
402
814
784
384
SS 32
650
SS4
406
802
736
740
808
790
SS0b
792
382
730
404
SS23
385
USED FLIP FLOP
374
810
804
806
CL-23,R
CL-33t
344
332

FIG. 5C

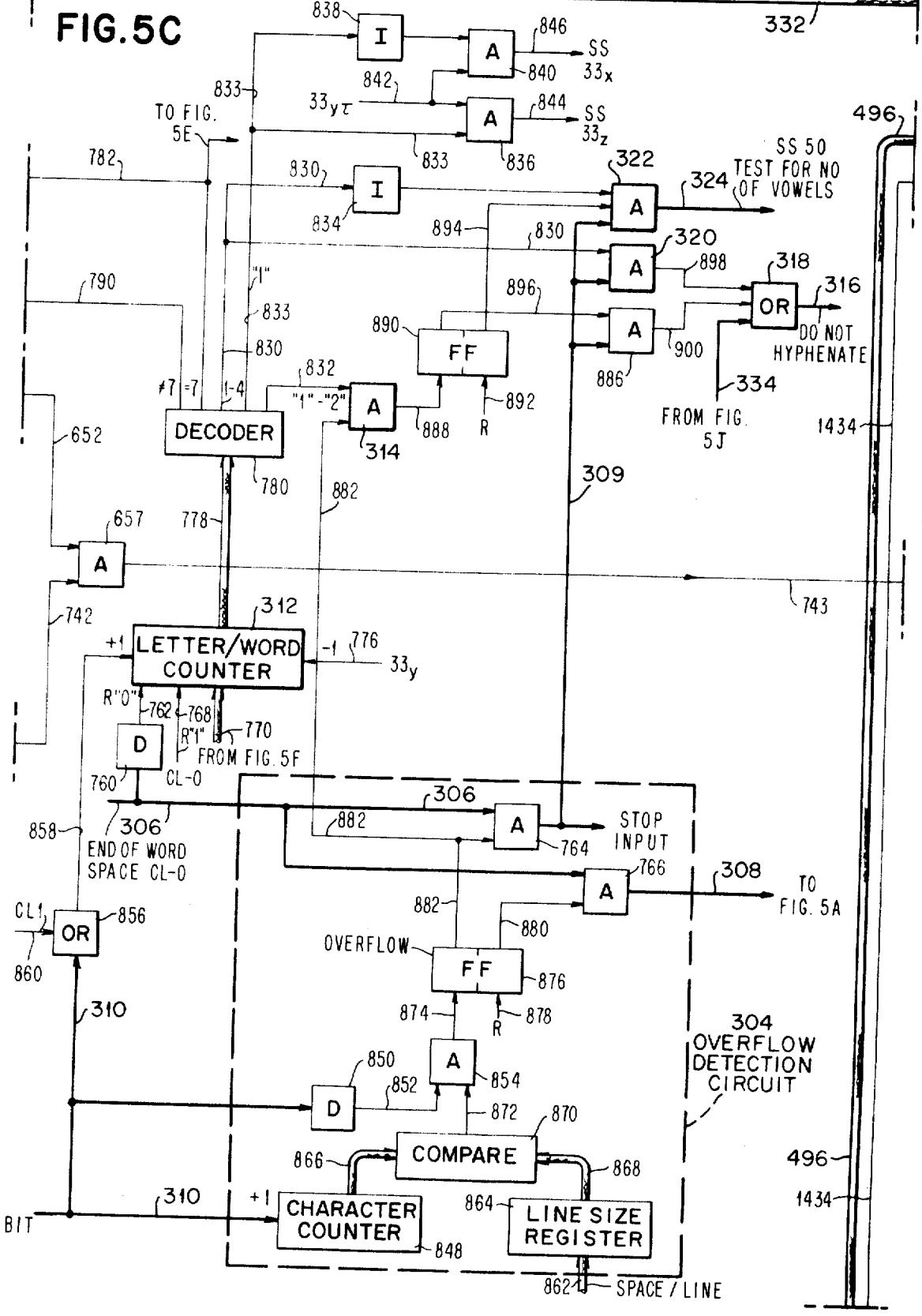
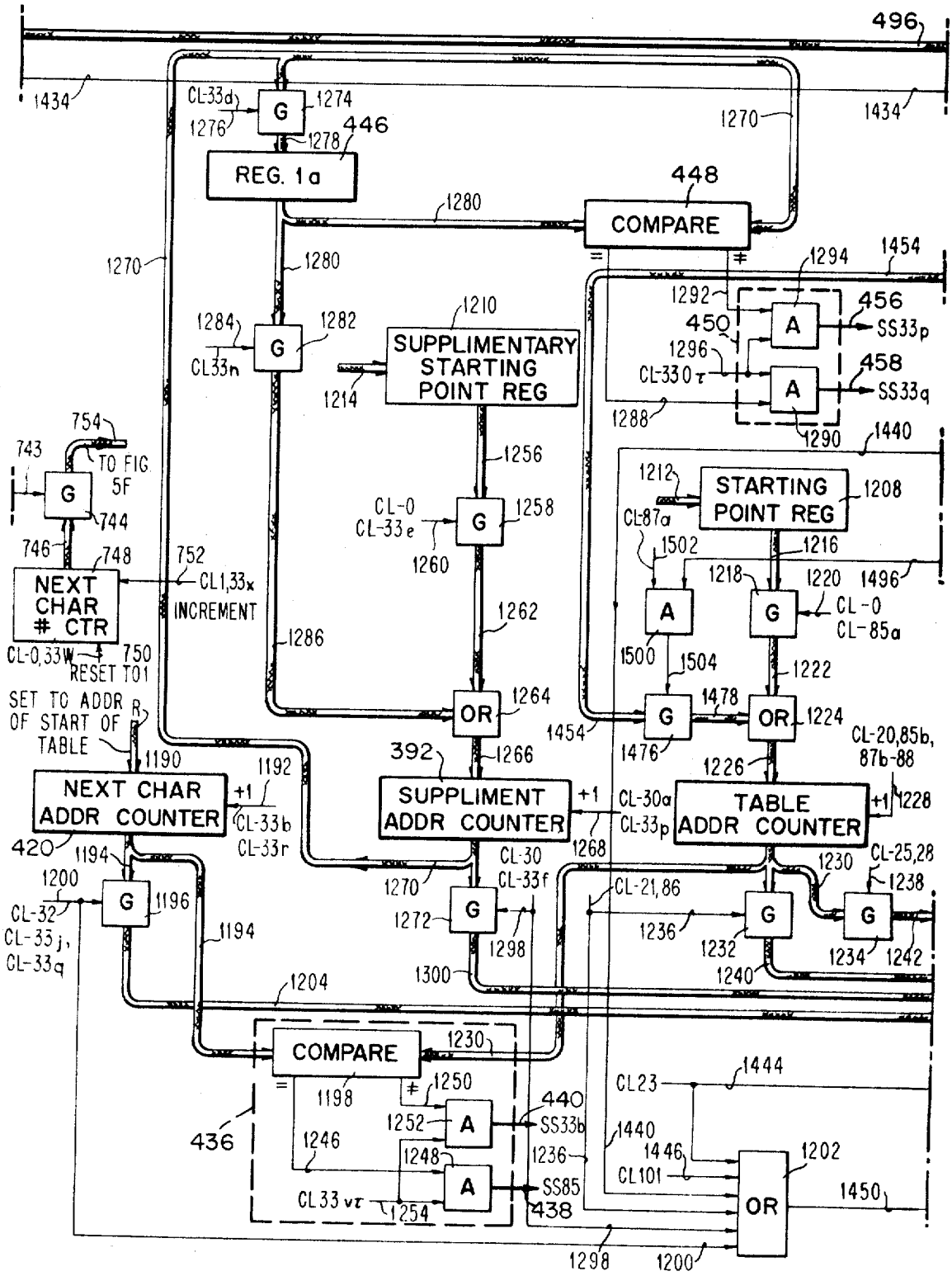

TO FIG. 5E
SS 33x
SS 33z
SS 50 TEST FOR NO OF VOWELS
DO NOT HYPHENATE
FROM FIG. 5J
DECODER
LETTER/WORD COUNTER
FROM FIG. 5F
END OF WORD SPACE CL-0
STOP INPUT
TO FIG. 5A
OVERFLOW
OVERFLOW DETECTION CIRCUIT
COMPARE
CHARACTER COUNTER
LINE SIZE REGISTER
SPACE / LINE
BIT

FIG. 5D


332
496
1434
CL-33d
1276
G
1274
1278
446
REG. 1a
1270
1434
448
COMPARE
1280
1270
1280
1294
1454
1292
456
450
A
SS33p
1296
CL-330τ
458
A
SS33q
1288
1290
1210
1284
1282
G
CL33n
SUPPLIMENTARY STARTING POINT REG
1214
1440
754
743
TO FIG. 5F
G
744
1256
1212
STARTING POINT REG
1208
CL-0
CL-33e
G
1258
CL-87a
1502
1216
746
748
752
1260
1218
1220
1496
NEXT CHAR # CTR
CL1,33x
INCREMENT
A
G
CL-0
CL-85a
1262
1286
CL-0,33W
750
RESET TO1
1500
1504
1222
SET TO ADDR OF START OF TABLE
R
OR
1264
1478
G
OR
1224
1454
1476
CL-20,85b, 87b-88
1190
1192
392
1266
1226
NEXT CHAR ADDR COUNTER
+1
CL-33b
CL-33r
SUPPLIMENT ADDR COUNTER
+1
CL-30a
CL-33p
TABLE ADDR COUNTER
+1
1228
420
1194
1268
CL-30
CL-33f
1230
CL-25,28
1200
1270
CL-21,86
1238
CL-32
CL-33j,
CL-33q
G
1196
1272
G
1236
G
G
1194
1298
1232
1234
1242
1300
1240
1204
COMPARE
1230
1198
1250
1444
CL23
440
1252
A
SS33b
436
1246
1440
1248
1236
1446
1202
A
SS85
CL101
CL33vτ
1254
438
OR
1450
1298
1200

AUTOMATIC HYPHENATION SCHEME

Original Filed June 17, 1964 26 Sheets-Sheet 10

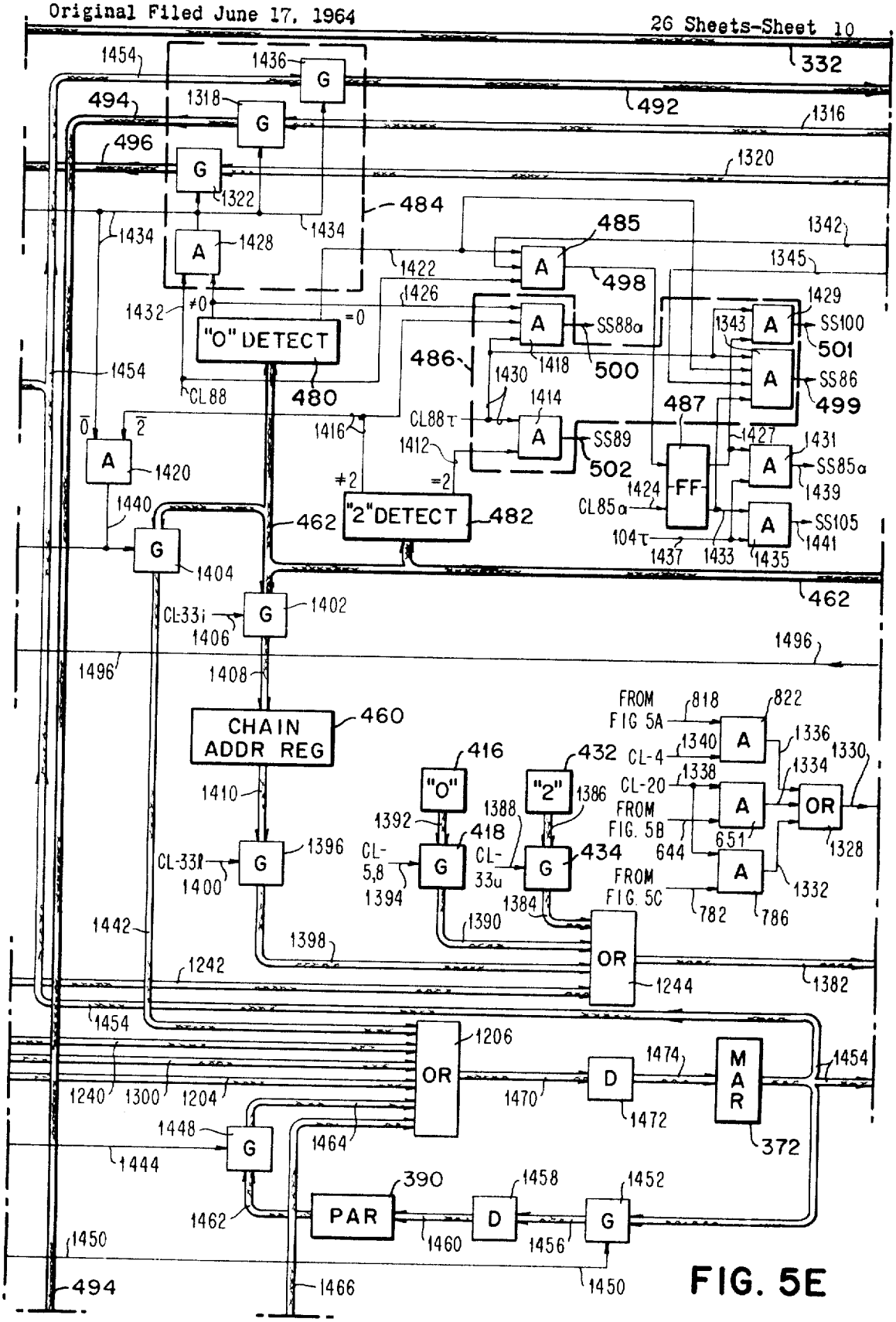


FIG. 5E

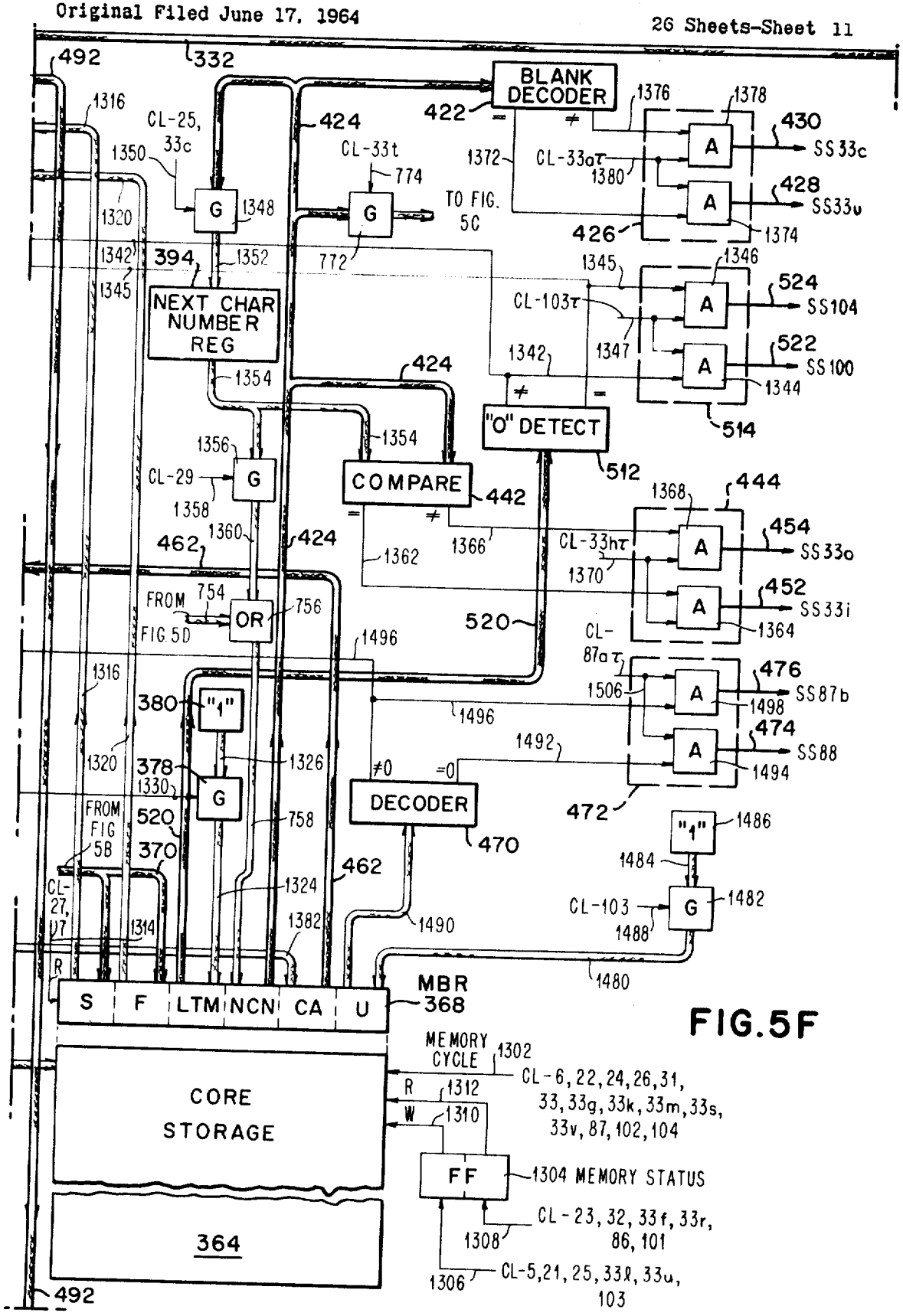
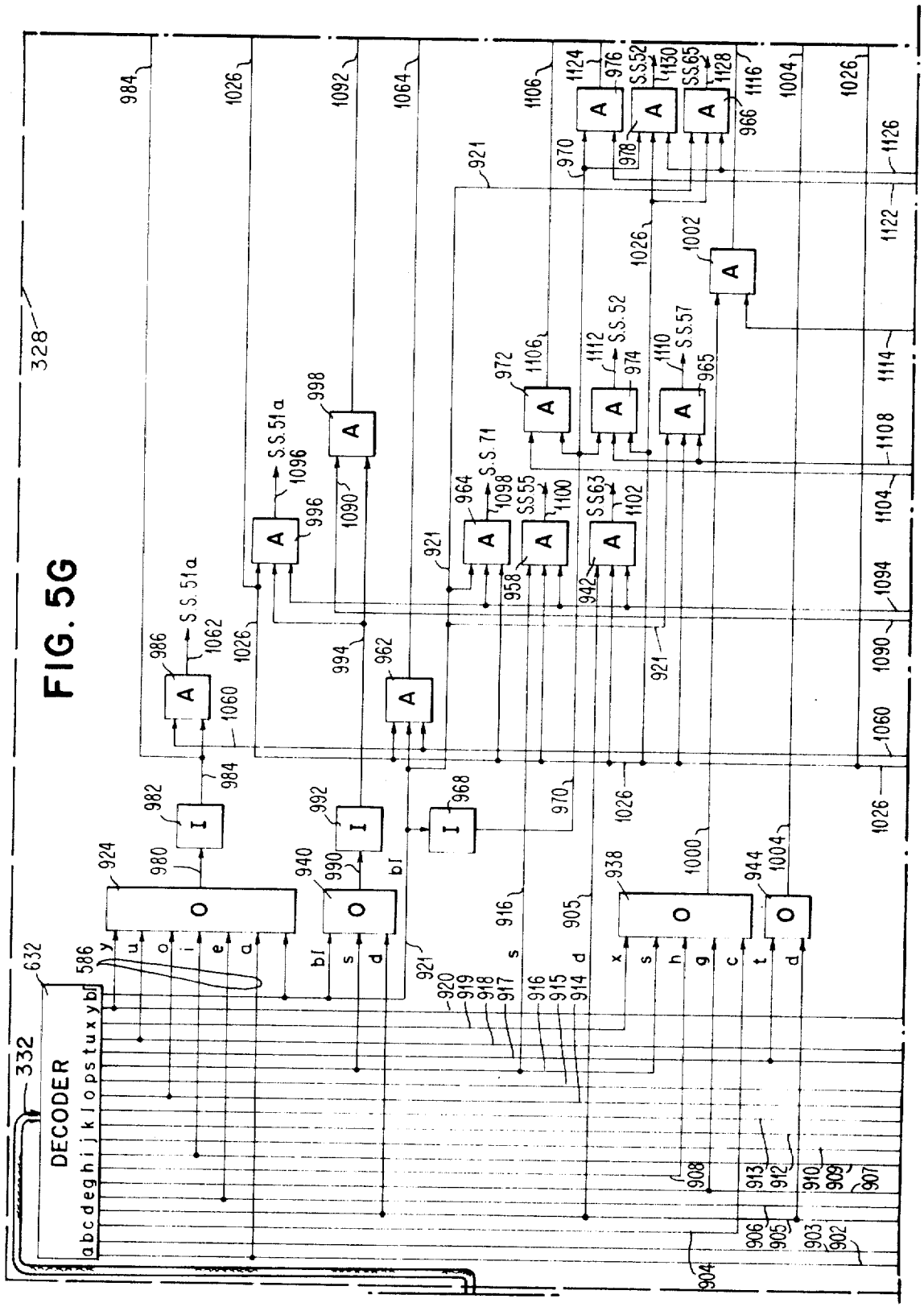

BLANK DECODER
NEXT CHAR NUMBER REG
"O" DETECT
COMPARE
DECODER
OR
"1"
G
A
CL-25, 33c
CL-33t
TO FIG. 5C
CL-33aτ
CL-103τ
CL-29
CL-33hτ
CL-87aτ
CL-103
FROM FIG. 5D
FROM FIG 5B
CL-27, U7
SS33c
SS33u
SS104
SS100
SS33o
SS33i
SS87b
SS88
MBR 368
S F LTM NCN CA U
CORE STORAGE
364
MEMORY CYCLE
CL-6, 22, 24, 26, 31, 33, 33g, 33k, 33m, 33s, 33v, 87, 102, 104
FF
1304 MEMORY STATUS
CL-23, 32, 33f, 33r, 86, 101
CL-5, 21, 25, 33ℓ, 33u, 103
492
332
424
422
426
430
428
394
524
522
514
512
442
444
454
452
476
474
472
470
462
380
378
370
520
1316
1320
1350
1348
1352
1342
1345
1354
1356
1358
1360
1362
1366
1368
1370
1364
1496
1506
1498
1492
1494
1486
1484
1482
1488
1480
1490
1382
1324
1326
1330
758
754
756
1372
1374
1376
1378
1380
1346
1347
1344
774
772
1312
1310
1302
1308
1306

FIG.5F

FIG. 5G


DECODER
a b c d e g h i j k l o p s t u x y b'
A
I
O
S.S.51a
S.S.71
S.S.52
S.S.57
SS55
S.S63
SS52
SS65

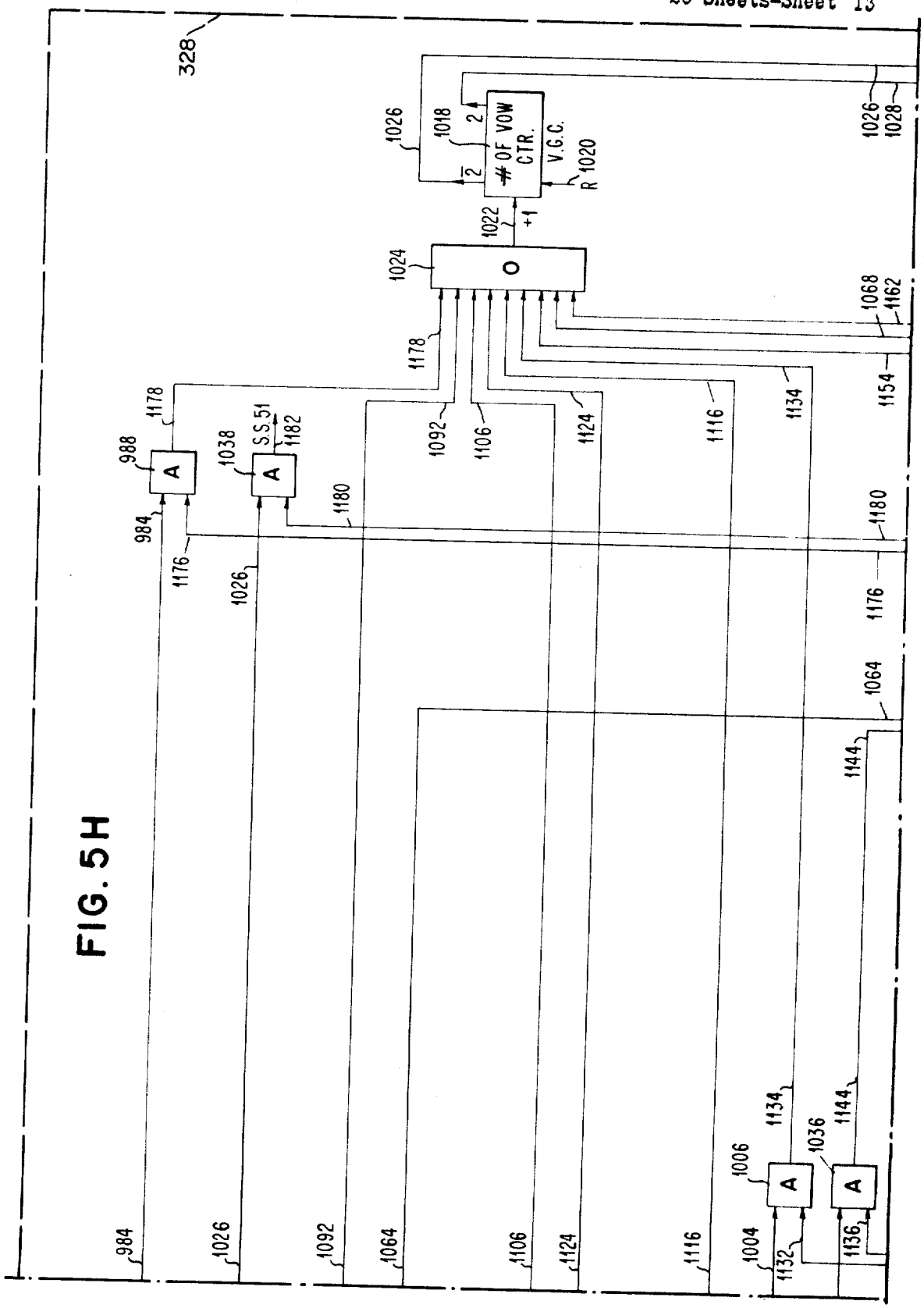

328
1026
1018
2
2
OF VOW CTR.
V.C.C.
R
1020
1022
+1
1024
O
1026
1028
1068
1162
1178
1154
1134
1116
1124
1106
1092
1178
988
A
984
1176
1038
A
S.S.51
1182
1180
1026
1180
1176
1064
1144
1134
1144
1036
1006
A
A
1004
1132
1136
984
1026
1092
1064
1106
1124
1116

FIG. 5H

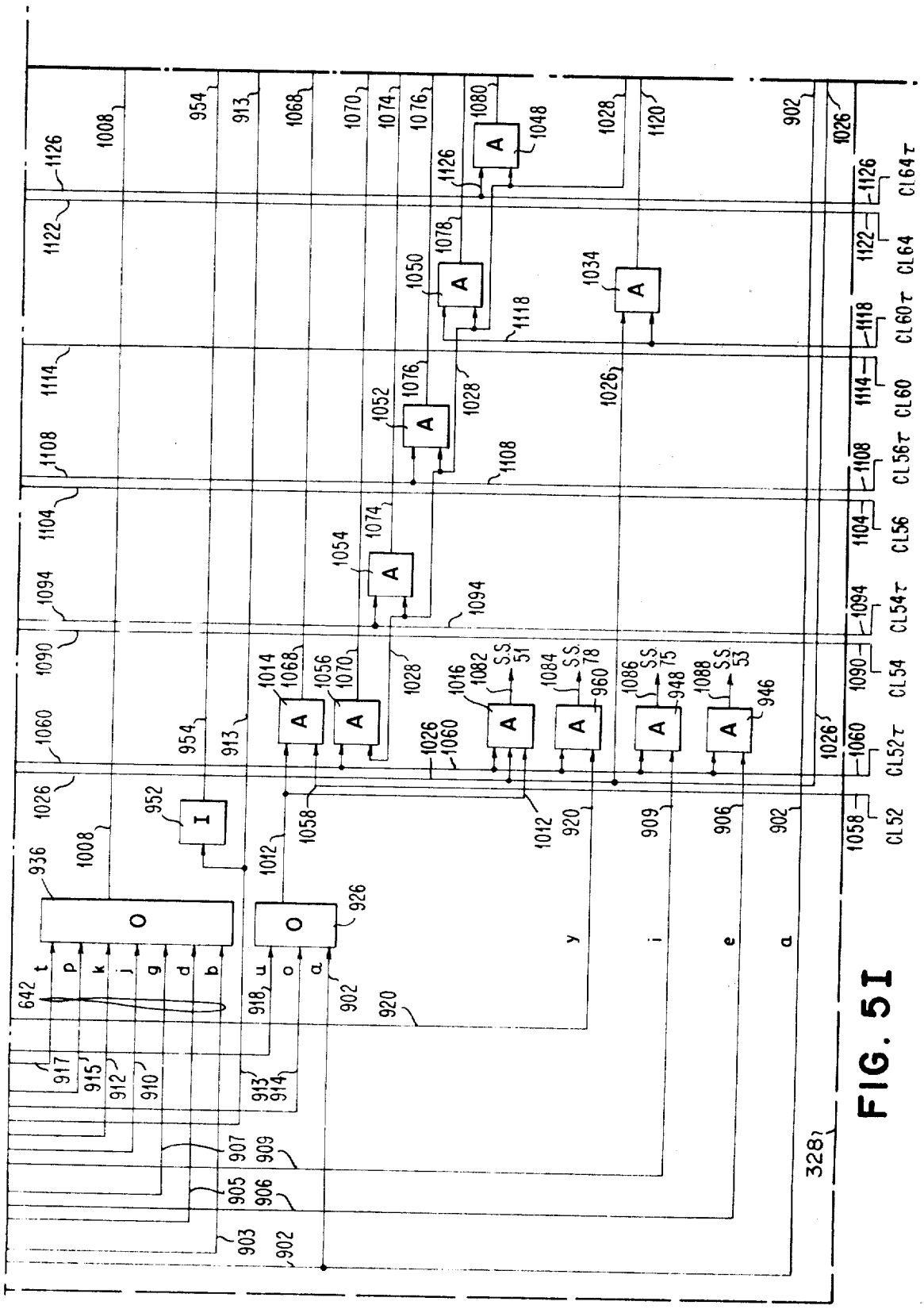

CL52
CL52τ
CL54
CL54τ
CL56
CL56τ
CL60
CL60τ
CL64
CL64τ
S.S. 51
S.S. 78
S.S. 75
S.S. 53
t p k j g d b
u o a
y i e a
O
I
A
1008
1026
1060
1090
1094
1104
1108
1114
1122
1126
954
913
1068
1070
1074
1076
1080
1048
1028
1120
902
1078
1050
1034
1118
1052
1054
1014
1056
1016
1082
1084
960
1086
948
1088
946
1012
1058
920
909
906
936
926
918
642
917
915
912
910
914
907
905
903
328

FIG. 5I

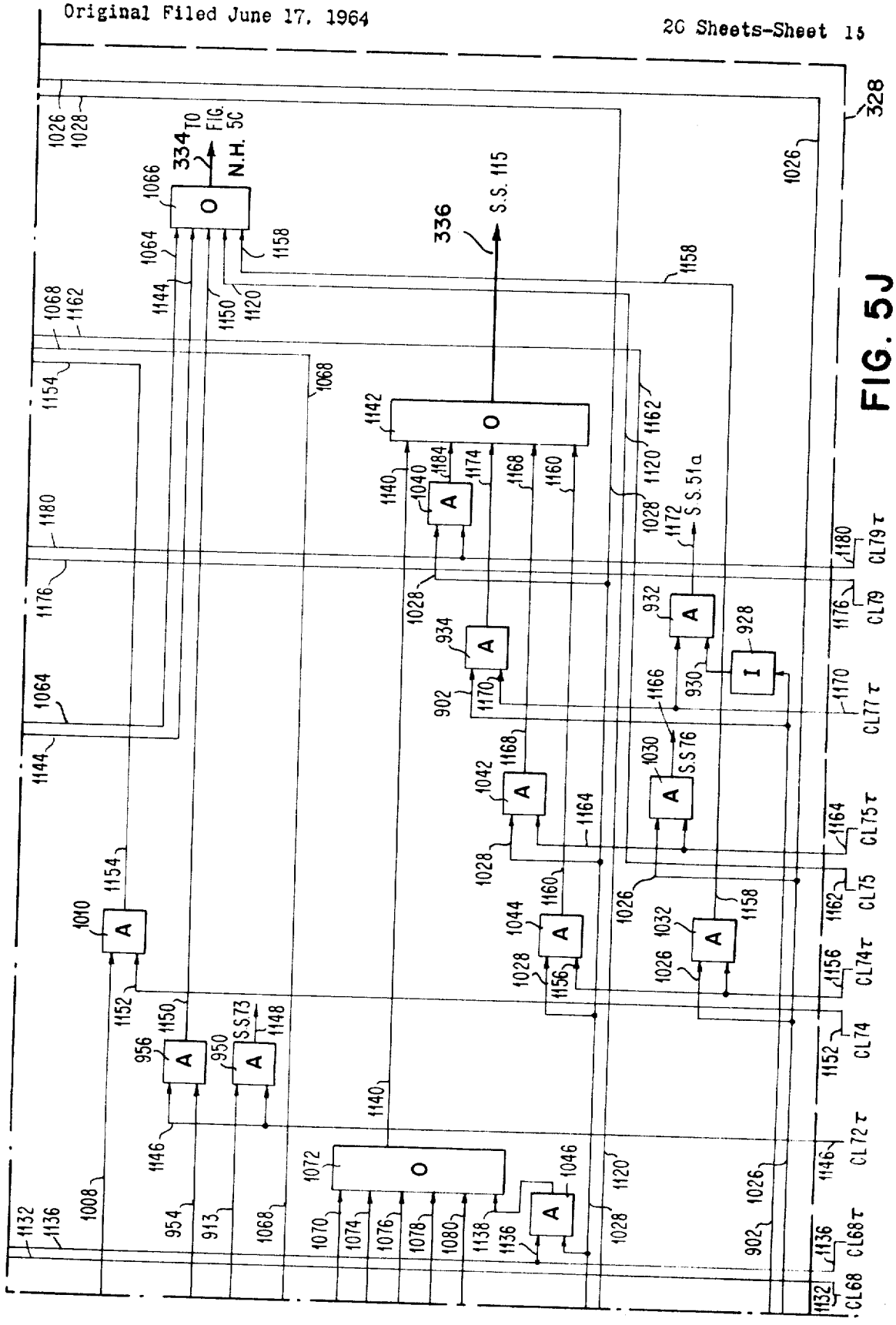

334 TO FIG. 5C N.H.
336 S.S. 115
S.S.73
S.S.76
S.S. 51a
CL79τ
CL79
CL77τ
CL75τ
CL75
CL74τ
CL74
CL72τ
CL68τ
CL68

FIG. 5J

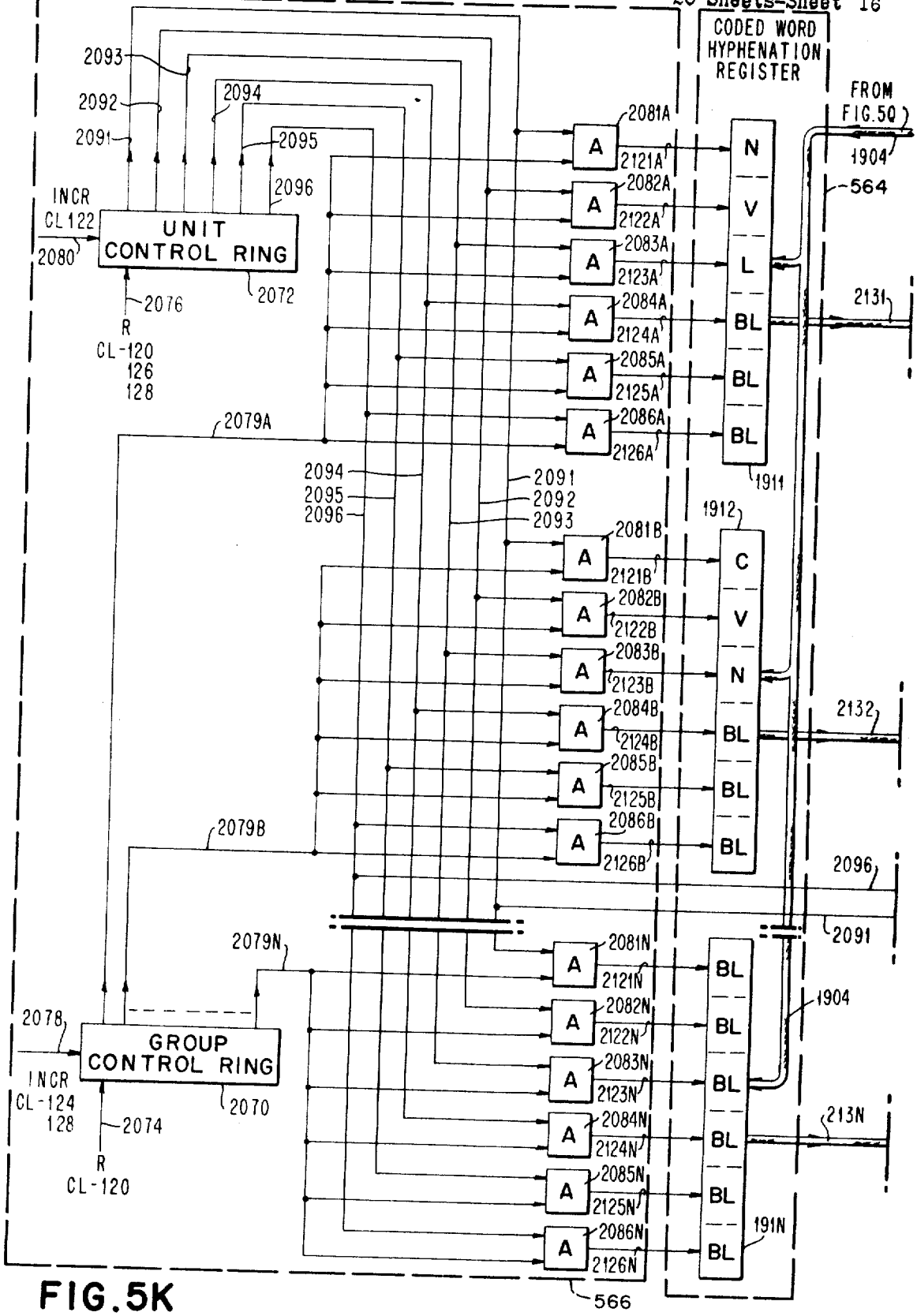

CODED WORD HYPHENATION REGISTER
FROM FIG.5Q
1904
564
2131
2132
2096
2091
1904
213N
2093
2092
2091
2094
2095
2096
INCR CL122
2080
UNIT CONTROL RING
2072
2076
R CL-120 126 128
2079A
2079B
2079N
2078
GROUP CONTROL RING
2070
INCR CL-124 128
2074
R CL-120
2081A
2121A
2082A
2122A
2083A
2123A
2084A
2124A
2085A
2125A
2086A
2126A
2081B
2121B
2082B
2122B
2083B
2123B
2084B
2124B
2085B
2125B
2086B
2126B
2081N
2121N
2082N
2122N
2083N
2123N
2084N
2124N
2085N
2125N
2086N
2126N
A
N
V
L
BL
C
1911
1912
191N
566

FIG.5K

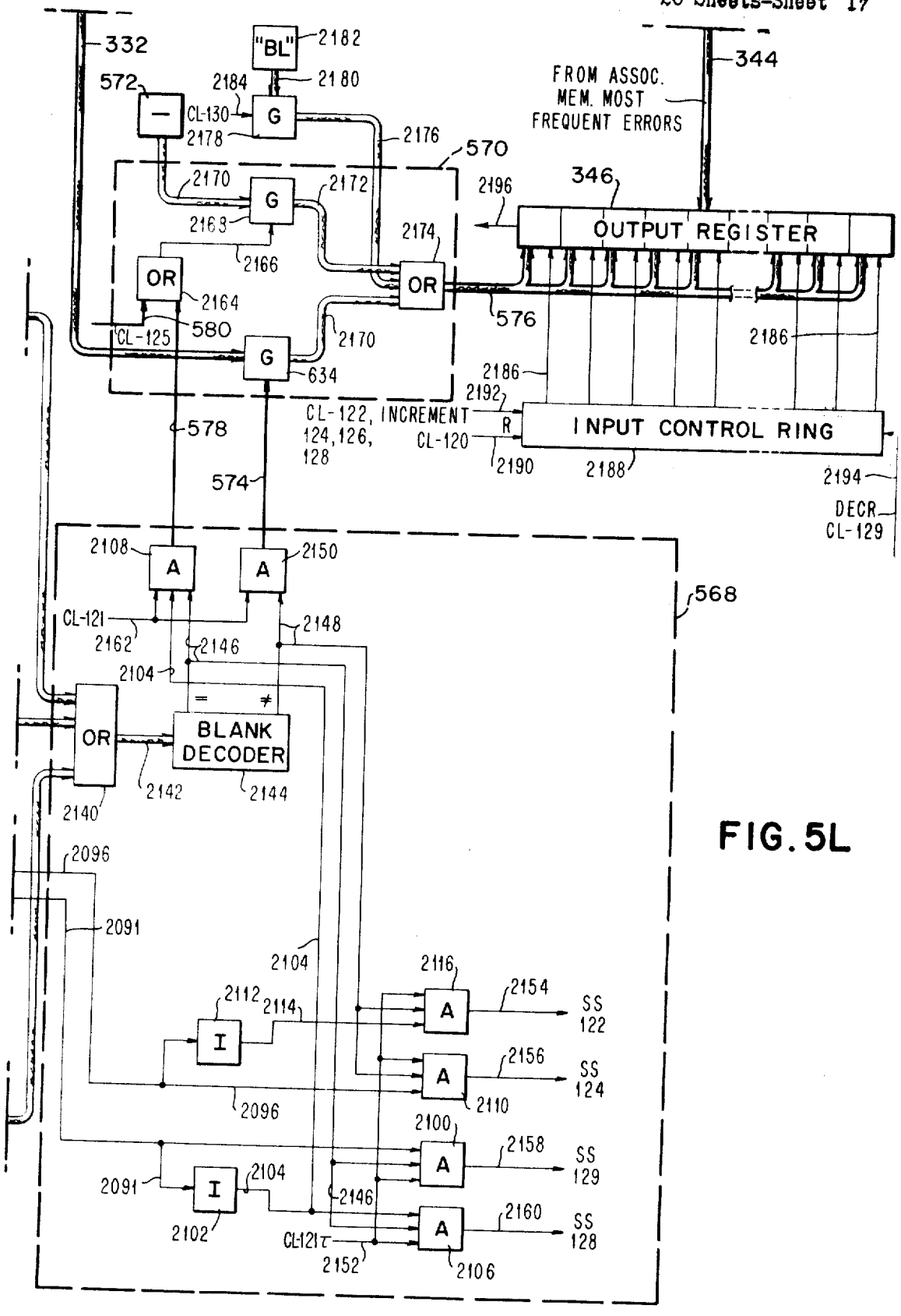
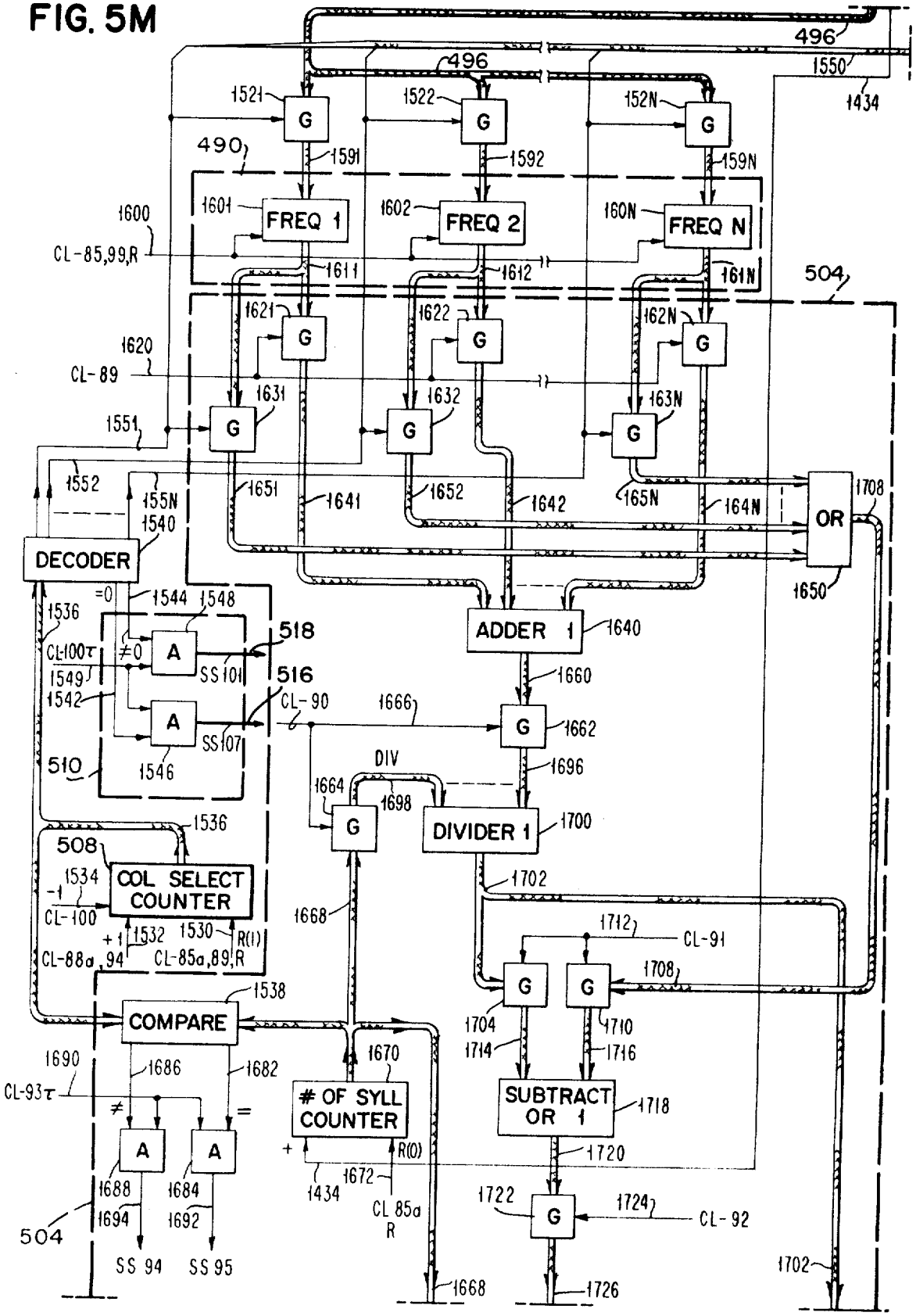

332
"BL"
2182
572
2184
2180
CL-130
2178
G
2176
570
FROM ASSOC. MEM. MOST FREQUENT ERRORS
344
346
2170
2172
2196
OUTPUT REGISTER
2168
2174
2166
OR
2164
576
CL-125
580
2170
2186
634
2186
2192
CL-122, INCREMENT 124,126, 128
R
INPUT CONTROL RING
CL-120
2190
2188
2194
DECR CL-129
578
574
568
2108
A
2150
CL-121
2162
2148
2146
2104
=
≠
BLANK DECODER
2142
2144
2140
2096
2091
2116
2154
SS 122
2112
2114
I
2156
SS 124
2110
2100
2158
SS 129
2091
2104
2146
2102
2160
SS 128
CL-121
2152
2106

FIG. 5L

FIG. 5M


496
1550
1434
1521
1522
152N
G
490
1591
1592
159N
1601
FREQ 1
1602
FREQ 2
160N
FREQ N
1600
CL-85,99,R
1611
1612
161N
504
1621
1622
162N
1620
CL-89
1631
1632
163N
1551
1552
155N
1540
1651
1641
1652
1642
165N
164N
OR
1708
1650
DECODER
=0
1544
1548
1536
518
ADDER 1
1640
CL-100τ
≠0
A
SS 101
1549
516
1542
CL-90
1666
1660
SS 107
1546
510
DIV
1664
1698
1696
DIVIDER 1
1700
1536
508
1534
-1
COL SELECT COUNTER
CL-100
+1
1532
1530
R(1)
CL-88a,94
CL-85a,89,R
1668
1702
1712
CL-91
1708
1538
COMPARE
1704
1710
1714
1716
1690
1686
1682
1670
CL-93τ
≠
=
OF SYLL COUNTER
SUBTRACTOR 1
1718
1720
R(0)
1672
1434
CL 85a R
1722
1724
CL-92
504
1688
1684
1694
1692
SS 94
SS 95
1726

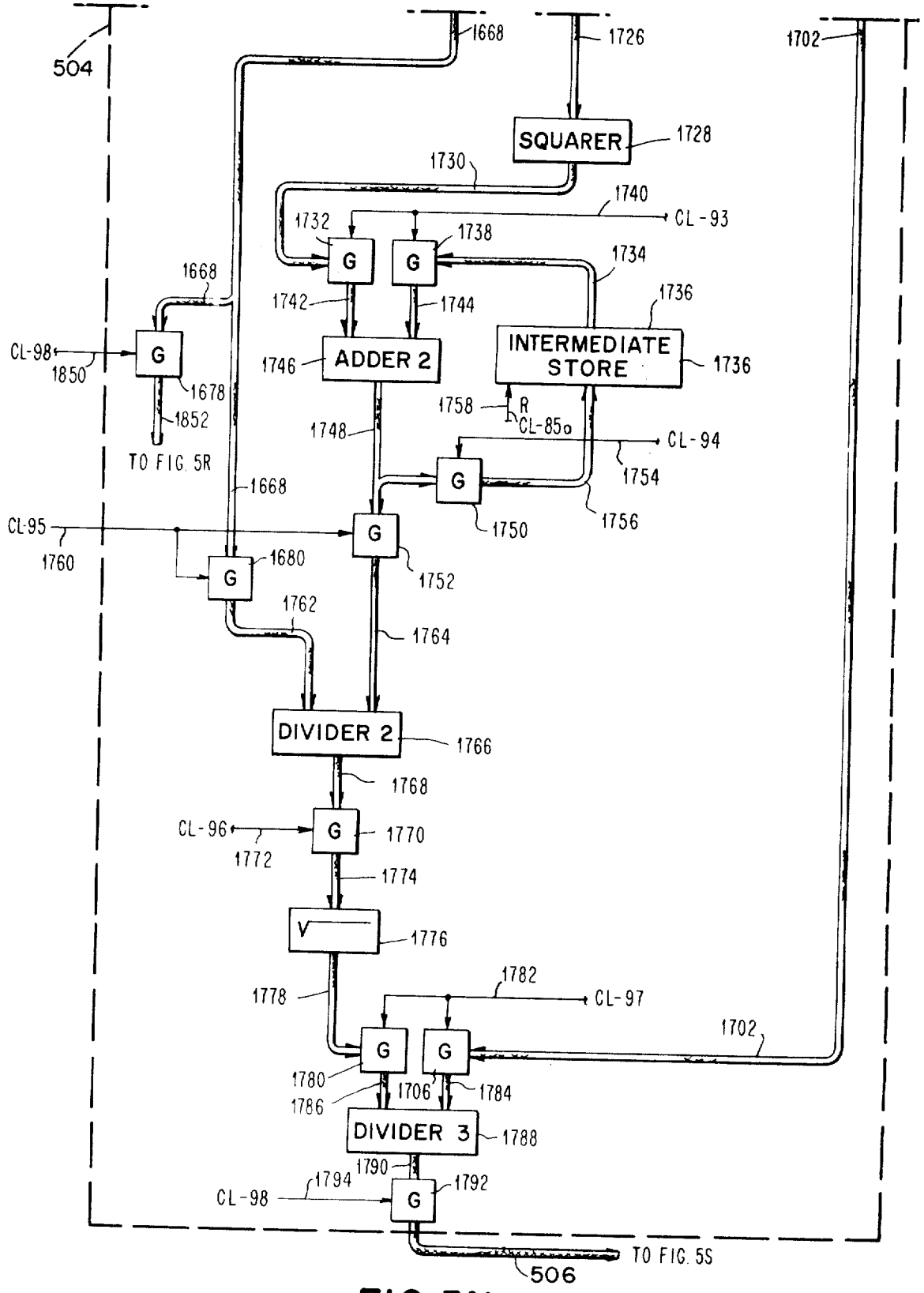

504
1668
1726
1702
SQUARER
1728
1730
1740
CL-93
1732
1738
G
1734
1736
1742
1744
CL-98
1850
1678
1852
TO FIG. 5R
ADDER 2
1746
INTERMEDIATE STORE
1736
1758
R
CL-85a
CL-94
1748
1754
1750
1756
CL-95
1760
1680
1752
1762
1764
DIVIDER 2
1766
1768
CL-96
1772
1770
1774
V
1776
1778
1782
CL-97
1780
1786
1706
1784
DIVIDER 3
1788
1790
1794
CL-98
1792
TO FIG. 5S
506

FIG.5N

FIG. 5P

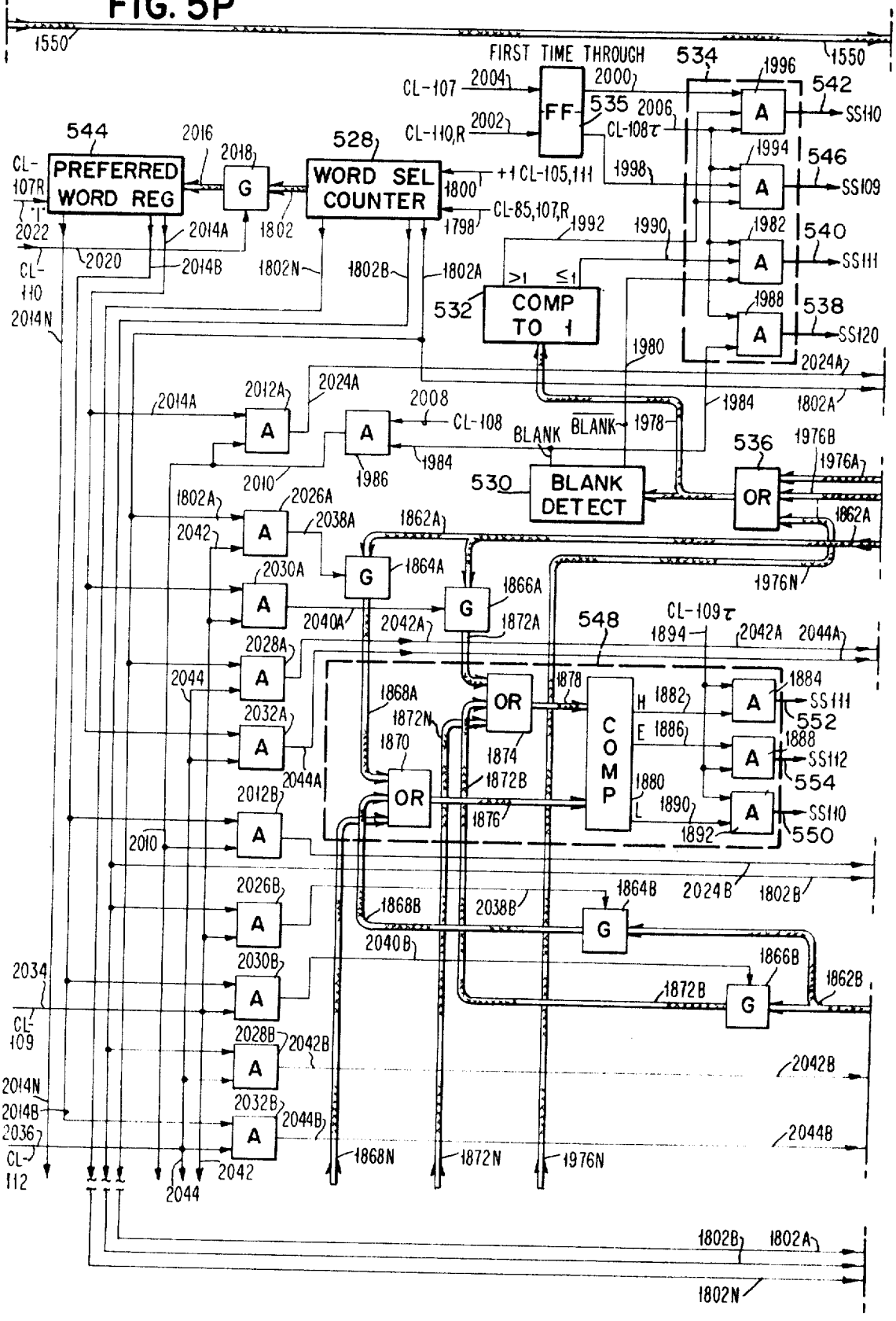

PREFERRED WORD REG
WORD SEL COUNTER
FIRST TIME THROUGH
FF
COMP TO 1
BLANK DETECT
OR
COMP
CL-107
CL-110,R
CL-108τ
+1 CL-105,111
CL-85,107,R
CL-107R
CL-110
CL-109
CL-112
CL-109τ
SS110
SS109
SS111
SS120
SS112
BLANK
BLANK
1550
1800
1798
1802
1802A
1802B
1802N
1868A
1868B
1868N
1872A
1872B
1872N
1976A
1976B
1976N
2000
2002
2004
2006
2008
2010
2012A
2012B
2014A
2014B
2014N
2016
2018
2020
2022
2024A
2024B
2026A
2026B
2028A
2028B
2030A
2030B
2032A
2032B
2034
2036
2038A
2038B
2040A
2040B
2042
2042A
2042B
2044
2044A
2044B
1870
1874
1876
1878
1880
1882
1884
1886
1888
1890
1892
1894
1980
1984
1986
1988
1990
1992
1994
1996
1998
1978
H
E
L
>1
≤1
A
G
OR
COMP
534
535
536
538
540
542
546
548
550
552
554
528
530
532
544

FIG. 5Q

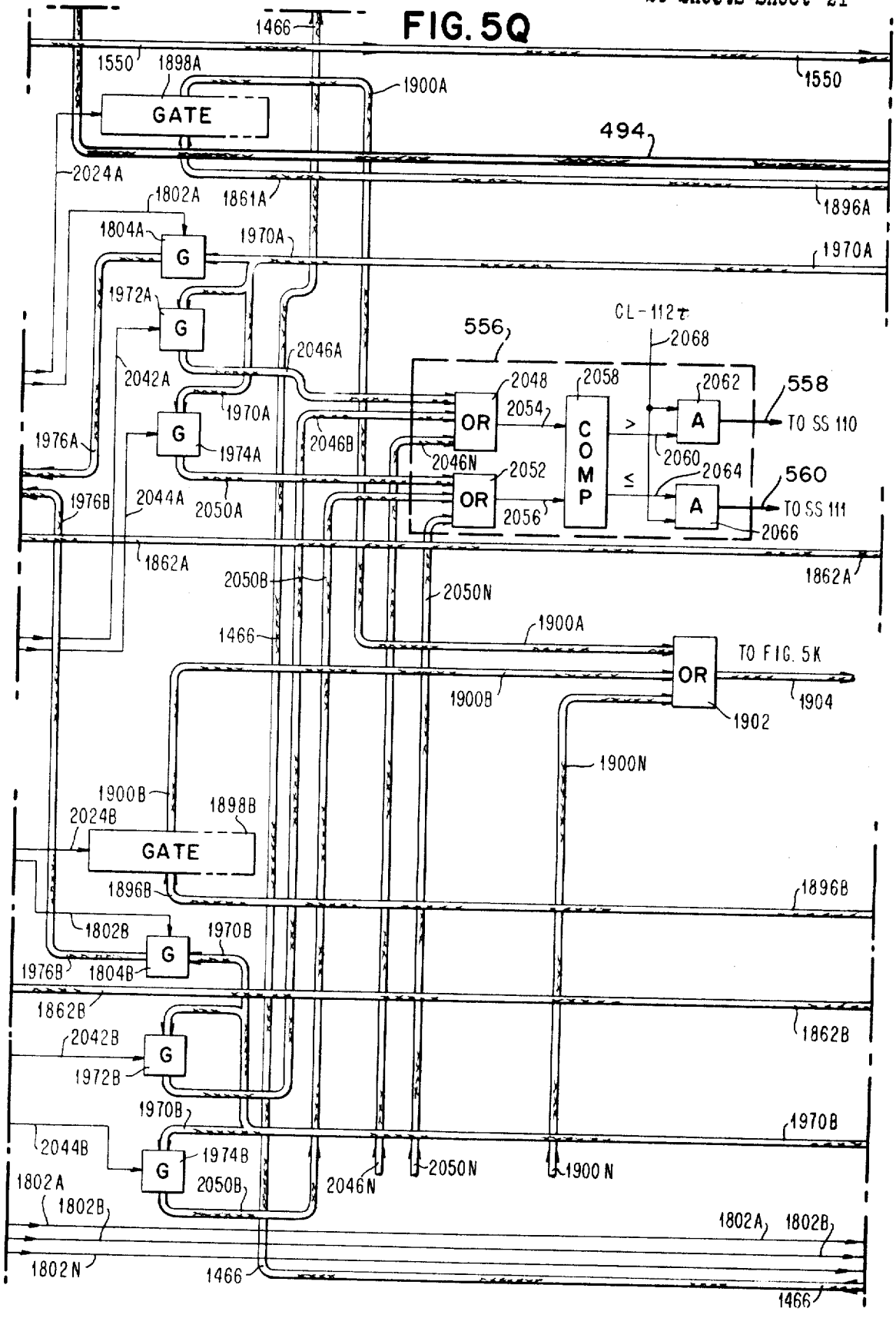

1466
1550
1898A
1900A
GATE
494
1550
2024A
1802A
1861A
1896A
1804A
G
1970A
1970A
1972A
G
CL-112
556
2068
2046A
2042A
2048
2058
2062
558
OR
2054
COMP
>
A
TO SS 110
1970A
1976A
G
1974A
2046B
2046N
2060
2064
560
OR
2052
≤
A
TO SS 111
1976B
2044A
2050A
2056
2066
1862A
1862A
2050B
2050N
1466
1900A
TO FIG. 5K
OR
1900B
1904
1902
1900N
1900B
1898B
2024B
GATE
1896B
1896B
1802B
1970B
G
1976B
1804B
1862B
1862B
2042B
G
1972B
1970B
1970B
2044B
G
1974B
1802A
2050B
2046N
2050N
1900N
1802B
1802A
1802B
1802N
1466
1466

FIG. 5R

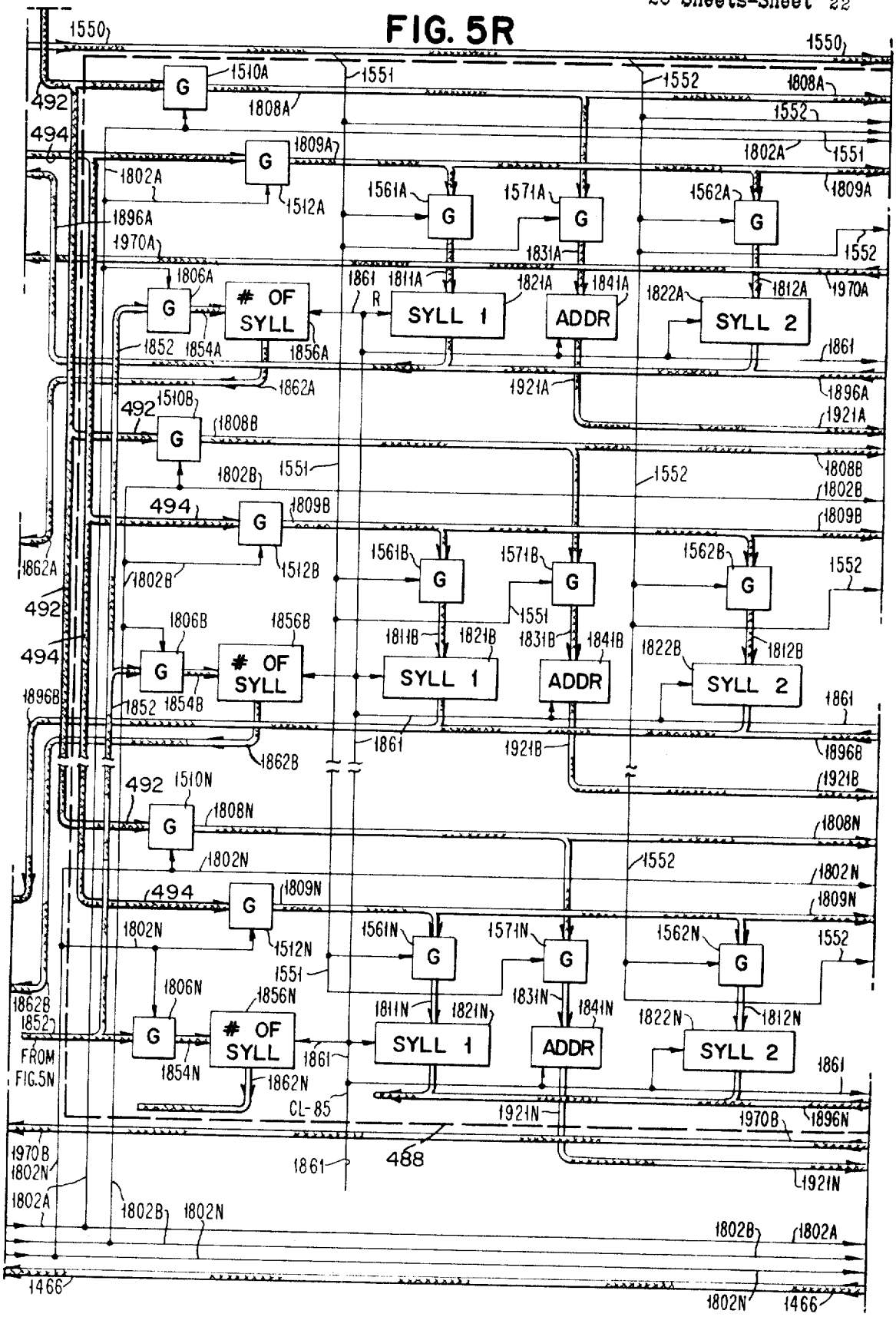
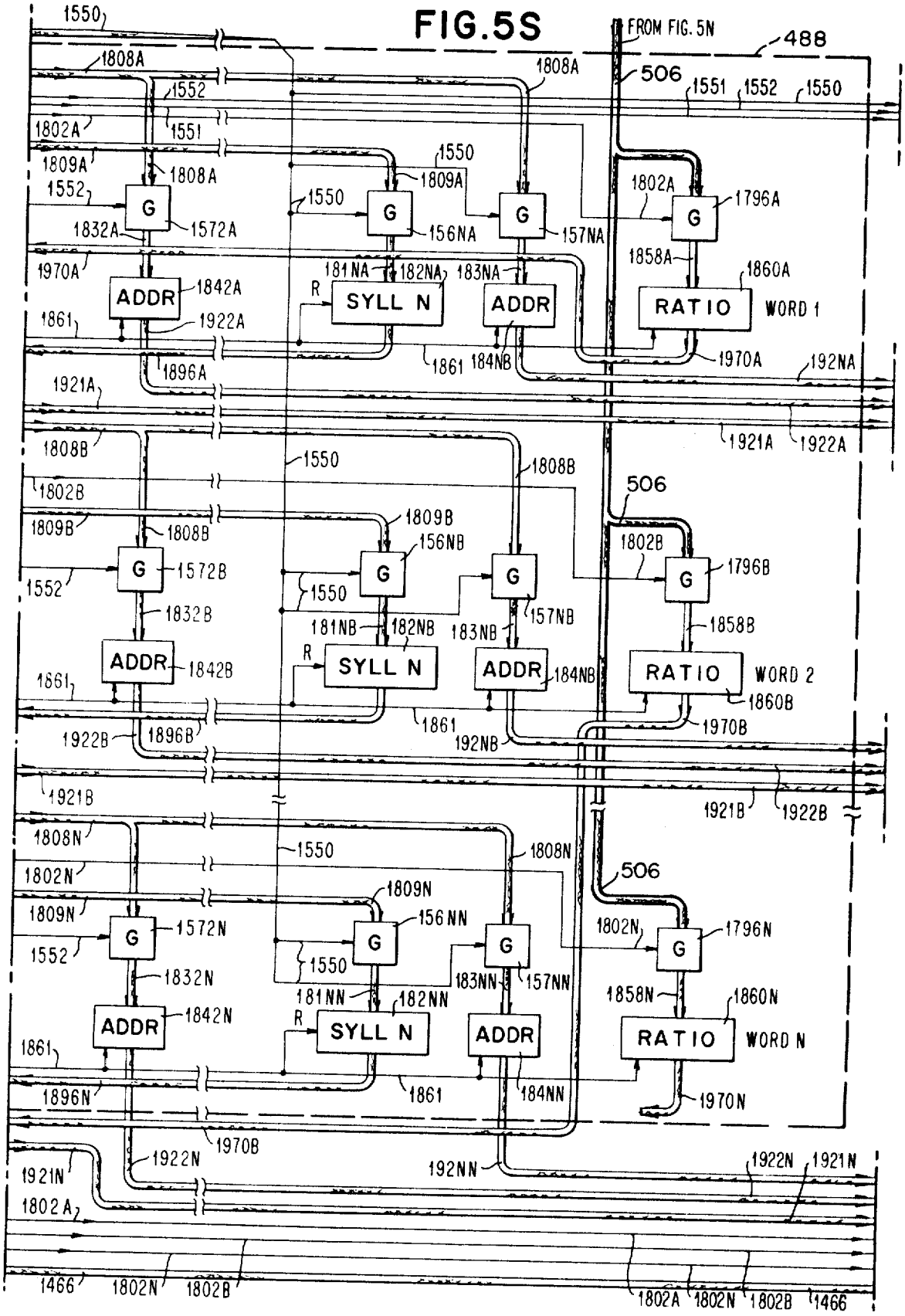

1550
1551
1552
1808A
1510A
492
494
1809A
1802A
1512A
1561A
1571A
1562A
1896A
1970A
1831A
1806A
OF SYLL
1852
1854A
1856A
1861
1811A
1821A
1841A
SYLL 1
ADDR
1822A
1812A
SYLL 2
1862A
1921A
1510B
1808B
1802B
1809B
1561B
1571B
1562B
1806B
1856B
1811B
1821B
1831B
1841B
1822B
1812B
1862B
1510N
1808N
1802N
1809N
1561N
1571N
1562N
1806N
1856N
1811N
1821N
1831N
1841N
1822N
1812N
1854N
FROM FIG.5N
CL-85
488
1921N
1970B
1896N
1466

FIG. 5S


1550
1808A
1552
1551
1802A
1809A
1808A
1552
G
1572A
1832A
1970A
ADDR
1842A
1922A
1861
1896A
1921A
1550
1809A
1550
G
156NA
G
157NA
181NA
182NA
183NA
R
SYLL N
ADDR
1861
184NB
FROM FIG. 5N
488
506
1551
1552
1550
1802A
G
1796A
1858A
1860A
RATIO
WORD 1
1970A
192NA
1921A
1922A
1808B
1802B
1809B
1808B
1550
1809B
156NB
G
1572B
1552
1832B
ADDR
1842B
1861
1922B
1896B
1921B
1550
G
157NB
181NB
182NB
183NB
R
SYLL N
ADDR
184NB
1861
192NB
1808B
506
1802B
G
1796B
1858B
RATIO
WORD 2
1860B
1970B
1921B
1922B
1808N
1802N
1809N
1808N
1550
1809N
156NN
G
1572N
1552
1832N
ADDR
1842N
1861
1896N
1970B
1922N
1921N
1802A
1466
1802N
1802B
G
157NN
181NN
182NN
183NN
R
SYLL N
ADDR
1861
184NN
192NN
506
1802N
G
1796N
1858N
1860N
RATIO
WORD N
1970N
1922N
1921N
1802A
1802N
1802B
1466

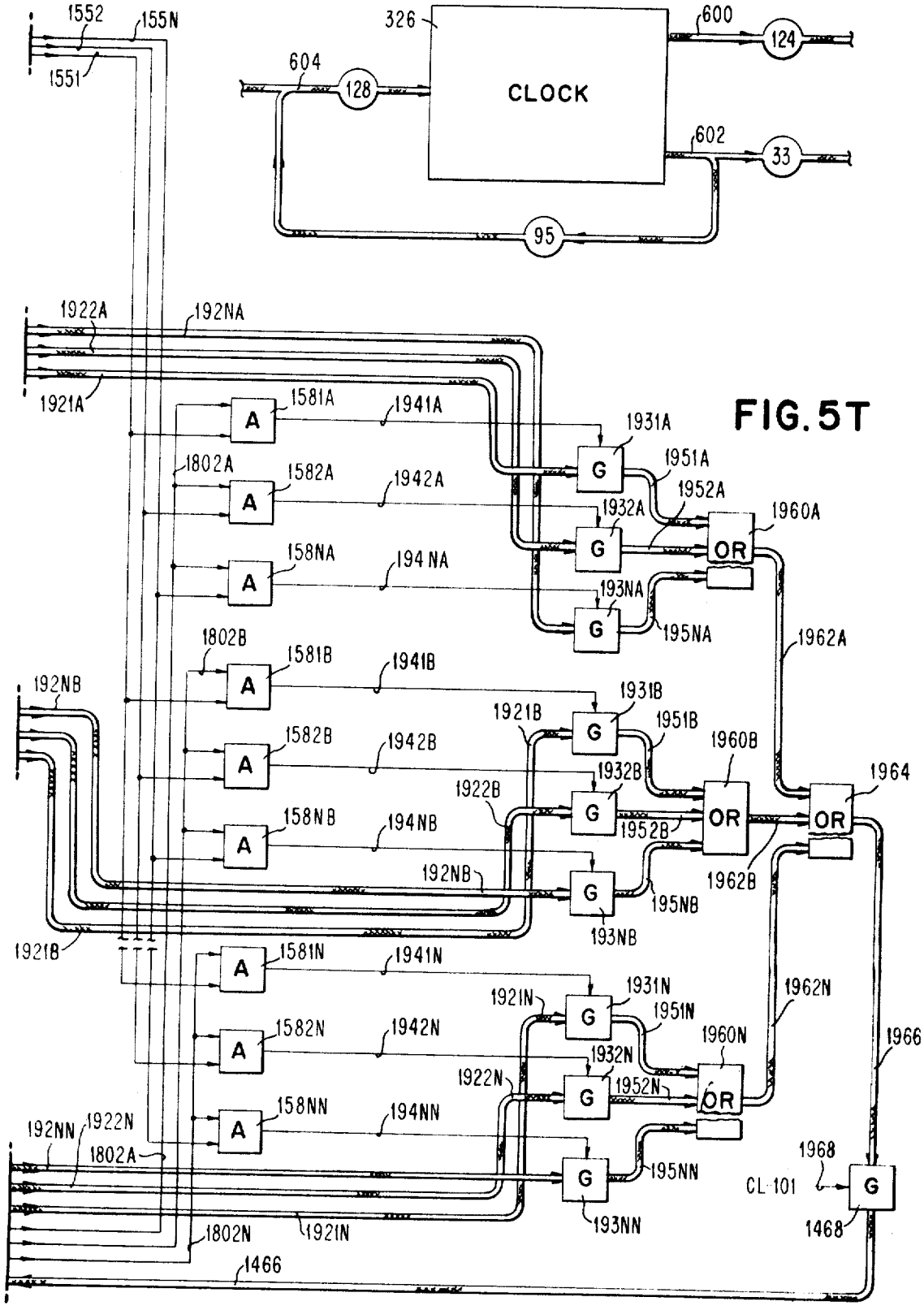

1552
155N
1551
326
CLOCK
600
124
604
128
602
33
95
1922A
192NA
1921A
1581A
1941A
1931A
1951A
1802A
1582A
1942A
1932A
1952A
1960A
158NA
194 NA
193NA
195NA
1962A
1802B
1581B
1941B
192NB
1931B
1921B
1951B
1582B
1942B
1960B
1964
1932B
1922B
158NB
194NB
1952B
1962B
195NB
193NB
1921B
1581N
1941N
1931N
1921N
1951N
1962N
1582N
1942N
1960N
1966
1932N
1922N
1952N
158NN
194NN
192NN
1922N
1802A
195NN
1968
CL 101
193NN
1468
1802N
1921N
1466
A
G
OR

FIG. 5T

TABLE OF POSSIBLE SYLLABLE PATTERNS

ADDRESS	SYLLABLE	RELATIVE FREQUENCY	LEVEL TERMINATION MARK	NEXT CHARACTER NUMBER	CHAIN ADDRESS	USED BIT
100	NV	1723		3	0	0
101	NVL	787	1	4	102	0
102	CV	961		6	0	0
103	CVN	792		7	105	0
104	CVNT	70	1	8	108	0
105	TV	4260		9	111	0
106	TVL	3027		10	0	0
107	TVLE	158	1		2	0
108	V	7233		9	111	0
109	VL	3422		10	0	0
110	VLE	51	1		2	0
111	LE	40	1		2	0

FIG. 6

SUPPLEMENTARY TABLE

NEXT CHARACTER NO.	CHAIN ADDRESS
3	0
4	102
6	0
7	105
8	108
9	111
10	0

FIG. 7

ENTRIES IN WORD REGISTER 488
FOR WORD "MERCANTILE"

WORD #	SYL #	SYL #1	ADDR #1	SYL #2	ADDR #2	SYL #3	ADDR #3	SYL #4	ADDR #4	SYL #5	ADDR #5	RATIO $\left(\frac{\text{AVER F}}{\sqrt{\Sigma \text{ DEVI}^2}}\right)$
1	4	NVL	101	CVN	103	TV	105	LE	111			0.90
2	3	NVL	101	CVN	103	TVLE	107					1.94
3	4	NVL	101	CVNT	104	V	108	LE	111			0.67
4	3	NVL	101	CVNT	104	VLE	110					0.88
5												

FIG. 9

REDUCED CHARACTER SET
CONVERSION TABLE

REDUCED CHARACTERS SET CHARACTERS	NORMAL CHARACTERS
V	a,e,i,o,u
T	b,d,g,j,k,p,t
L	l,r
N	m,n
S	f,s
Z	v,z
C	c
H	h
Q	q
X	x
Y	y
W	w
E	final e
G	final g
*	blank at beginning and end of word

FIG. 8

United States Patent Office 3,537,076 Patented Oct. 27, 1970

3,537,076

AUTOMATIC HYPHENATION SCHEME

Frederick J. Damerau, Mahopac, N.Y., assignor to International Business Machines Corporation, New York, N.Y., a corporation of New York

Continuation of application Ser. No. 375,714, June 17, 1964. This application Nov. 28, 1967, Ser. No. 686,347
Int. Cl. G06d 7/00; G11b 27/00
U.S. Cl. 340—172.5 4 Claims

ABSTRACT OF THE DISCLOSURE

The point or points at which a word of text may properly be hyphenated are determined automatically by a process of first selecting a variety of possibilities and, second, choosing a particular weighted probability from among them. Specifically, the sequence of letters (or characters representing these letters) in the text word under consideration is analyzed in subsets to ascertain all of the possible arrangements of characters within pairs of subsets (each pair embracing a possible hyphen point) that hypothetically could produce acceptable hyphenation of the word. The particular arrangement of these subsets which most probably would produce correct hyphenation then is selected according to established statistical rules. In the disclosed embodiment, the letters of the word to be analyzed are first converted into a reduced character set, which then is broken up in various ways into all of its possible syllable combinations, and out of these possible syllabifications the most probable syllable pattern is selected.

This is a continuation of application Ser. No. 375,714, filed June 17, 1964, now abandoned.

This invention relates to a scheme for hyphenating words in textual material, and more particularly to a statistical scheme for automatically hyphenating words.

The trend toward automation has recently been extended to the newspaper and other publishing industries. This trend has manifested itself in automatic type composing machines and in numerous schemes for automatically justifying lines of type in these machines. However, one problem which has consistently plagued these efforts at automatic line justification is the inability of the machines to properly hyphenate words at the end of a line. Existing and proposed solutions to this hyphenation problem have involved: (a) the use of an enormous dictionary in which all posible words which may appear in the text being printed are stored, and (b) the performance of a table look-up type operation to find the word and its proper hyphenation pattern each time a word is to be hyphenated.

There are two serious disadvantages to the table look-up scheme for automatic hyphenation. First, with almost any sample of text, the number of words involved in much larger than it is economically feasible to store in a high-speed memory such as a magnetic core matrix array. Therefore, the table has to be stored in a low-speed memory. As a result, each table look-up is relatively time consuming, causing the overall speed of the automatic printing operation to be substantially reduced.

Another, more serious, shortcoming of table look-up automatic hyphenation schemes is that no dictionary can ever be complete. This is particularly true in newspaper work where proper names consistently appear in the text to be operated upon. This means that even with a very large dictionary, the machine will still not be able to hyphenate a substantial percentage of the words applied to it, and there will be a need to perpetually update the dictionary in order to maintain any desired degree of hyphenation capability and accuracy.

It is therefore the primary object of this invention to provide an improved automatic hyphenation scheme.

A more specific object of this invention is to provide an automatic hyphenation scheme which does not require the use of a large table of stored entries.

Another object of this invention is to provide a high-speed scheme for automatic word hyphenation.

A further object of this invention is to provide a scheme for automatically hyphenating words, the storage requirements of which are such that all tables may be economically stored in high-speed memory arrays.

A still further object of this invention is to provide an automatic hyphenation scheme which is capable of hyphenating proper names and other unusual words in a sample of text.

Another object of this invention is to provide an automatic hyphenation scheme which is capable of accurately hyphenating a very high percentage of the words applied to it from a given sample of text.

A feature of this invention is the conversion of the word to be hyphenated to a reduced character set and the operation on the characters of the word in the reduced character set in a statistical manner to determine the most probable hyphenation pattern for the word.

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of a preferred embodiment of the invention, as illustrated in the accompanying drawings.

In the drawings:

FIG. 1 is a general flow diagram of the method of this invention.

FIG. 2 is a detailed flow diagram of the test-for-two-or-more-vowels step in the flow diagram of FIG. 1.

FIG. 3 is a diagram illustrating how FIGS. 3A–3B may be combined to form a flow diagram of the form-syllable-patterns step in the flow diagram of FIG. 1.

FIGS. 3A–3B when taken together form a flow diagram of the form-syllable-patterns step in the flow diagram of FIG. 1.

FIG. 4 is a flow diagram of the select-preferred-pattern step in the flow diagram of FIG. 1.

FIG. 5 is a diagram illustrating how FIGS. 5A–5T are combined to form a detailed block diagram of a circuit which may be used to practice the automatic hyphenation scheme of this invention.

FIGS. 5A–5T when taken together form a detailed block diagram of a circuit which may be used to practice the scheme of this invention.

FIG. 6 is a table with illustrative entries used in performing the scheme of this invention.

FIG. 7 is a supplementary table with illustrative entries used in performing the scheme of this invention.

FIG. 8 is a table for converting characters to the reduced character set.

FIG. 9 is an illustrative series of entries in a word register of the embodiment shown in FIGS. 5A–5T.

In addition to the figures described above, pages A1 to A37 at the end of the specification contain an appendix showing in table form the turn-on condition for each of tne single-shot multivibrators in the clock of FIG. 5T, and the functions performed during the period that each of the multivibrators is in its ON state, the functions performed when it switches to its OFF state, and the single-shot (SS) which is in its ON state during the following time period.

INTRODUCTION

This invention provides a statistical scheme for performing automatic word hyphenation. The method is based on two observed facts about word hyphenation. First, in a word, there are letter combinations between

which, as a rule, a hyphen may never occur, and there are letter combinations between which hyphens must fall. Therefore, there are only a limited number of ways in which any given word may be hyphenated. However, since there are generally several ways in which a word might possibly be hyphenated, a weight factor must be introduced in order to enable a proper choice to be automatically made between conflicting patterns. But, since the number of possible syllable types is very large, this scheme would not, at first glance, appear to give any advantage over a straight dictionary look-up scheme. However, a second observable fact is that many letters are equivalent for the purpose of syllabification. For example, in phonetic terms, one stop is the same as another stop; one vowel is the same as another vowel; and one lateral is the same as another lateral for the purpose of syllabification. With the reduced character set to be presented later, all possible hyphenation patterns may be stored in a relatively small dictionary.

The scheme for automatic hyphenation of this invention involves, as a preliminary step, the analyzing of a reasonable sample of the type of text to be automatically printed to determine the syllable patterns which appear in it and the relative frequency of their occurrence. These syllable patterns and their determined relative frequencies are then stored in a high-speed memory array.

Then, when it is determined that a word is to be hyphenated, it is stored in an input register and applied, a character at a time, to a coder which converts it to the reduced character set. The output from the coder is applied to an argument register. The information in this argument register is compared with the possible syllables stored in the high-speed memory, and, when a match is found, the syllable and its relative frequency are transferred to another memory. This practice continues until the maximum number of letters which it has been determined that a syllable can contain have been transferred into the argument register, or until the end of the word is reached. An attempt is then made to find syllables starting with subsequent letters in the word, and, when a successful match is made, the syllable and its associated frequency are also stored in the second memory array. A chain address to the possible syllable preceding that stored is also stored with this syllable. This practice is continued until all possible syllables for the word have been stored.

The relative frequencies stored with the syllables are then used to determine the most likely syllable pattern for the word being operated upon. A suitable criteria for this determination will be presented later. When the most probable hyphenation pattern is determined, the word is hyphenated in accordance with this pattern.

Since many words are not hyphenable, a considerable saving in operating time can be achieved by determining this fact before the above operations are performed on the word. It has been determined that if one of three criteria exists, the word is not hyphenable. First, if the end of line occurs on the first or second letter of the word, the word generally cannot and, as a rule, should not be hyphenated. The circuit of this invention therefore tests to determine whether the end of line occurs on either the first or second letter of a word, and, if this condition is detected, a not-hyphenable indication is given. Secondly, if a word has four or less letters, present practice is not to hyphenate it. A second test is therefore made to determine if the word has four or less letters, and, if this condition exists, the not-hyphenable indication is given. Finally, it has been determined that if the word has less than two vowel groups, it is not hyphenable. The criteria for determining vowel groups will be given later. A final test is therefore made to determine if the word has two or more vowel groups, and, if this test fails, a not-hyphenable indication is given.

While the above scheme accurately hyphenates a sufficient percentages of the words applied to it for most practical applications, it has been found that in any given text sample there is a limited number of words which are hyphenated in some odd manner and would therefore lead to consistent errors with a statistical hyphenation scheme such as that proposed above. The accuracy of the above scheme can therefore be substantially increased by providing a small dictionary of most frequent errors which is quickly checked before the statistical procedure is initiated.

GENERAL DESCRIPTION OF OPERATION

Assume that a determination has been made as to the desired number of index positions in a line of text to be printed and that successive characters of text are fed into and stored on the printing device until an indication is received generally in the middle of a word, that the predetermined number of index positions has been received. An overflow signal is then generated.

Referring now to FIG. 1, it is seen that when the overflow condition occurs on line **200**, the first thing which is done is to determine whether the word is hyphenable or not. This is a three-step operation which involves:

(1) Determining whether the overflow occurred on the first or second character, this being indicated by box **202** in the flow diagram;

(2) Determining whether the word is four or less characters long, this step being illustrated by box **204** in the flow diagram; and

(3) Determining if the word has two or more vowel groups, this being illustrated by box **206** in the flow diagram. If the answer in steps 1 or 2 is "yes," or if the answer in step 3 is "no," then OR box **208** gives an indication that the word in question is not hyphenable. It should be noted that, as in the above sentence, the box numbers in the flow diagrams are used throughout the discussion to designate the various steps in the operation.

The steps performed by boxes **202** and **204** are fairly straight-forward and do not require further explanation, except to illustrate, as will be done later, how they might be accomplished in the embodiment of the invention shown in FIGS. 5A–5T. However, further explanation is required for the vowel-group-counting operation illustrated by box **206**. In this operation, it is vowel groups rather than vowels which are being counted. It is assumed that each syllable contains a vowel group and therefore that the number of syllables will be equal to the number of vowel groups. While the vagaries of English spelling prevent this assumption from being 100% accurate, it is sufficiently accurate for the purpose for which it is being employed. For the purposes of this determination, most vowels and most vowel sequences are counted as one vowel group. The exception to this general rule are as follows:

"i–a" is counted as two vowels; "y" is counted as a vowel only if it follows a consonant; a final "e" is not counted as a vowel unless it is preceded by a stop letter (b, d, g, j, k, p, or t) followed by an "l"; an "e" before a final "d" is counted only if the letter preceding the "e" is a "t" or a "d"; an "e" before a final "s" is counted only if the preceding letter is an "s," a "c," a "g," an "h," or an "x."

For example, in the word "diarchy," the first and second exceptions above are used, leading to a determination that the word has three syllables which, in fact, it does. FIG. 2 is a flow diagram graphically illustrating the above described procedure for counting vowel groups and also illustrating the clock cycles during which the various operation shown therein are performed. The significance of the various clock times appearing in the flow diagrams will be apparent later. This diagram is self-explanatory and will therefore not be described at this point; however, it will be referred to again later to assist in understanding how the circuit of FIGS. 5G–5J operates to perform these functions.

Referring again to FIG. 1, it is seen that once a de-

termination has been made that word is hyphenable, a check is made to determine whether the word is contained in a short list of most frequent errors. This step is indicated by box **210** in the flow diagram. The words in this list are ones which, due to the vagaries of the English spelling, are not hyphenated in accordance with the most probable hyphenation pattern which is determined in a manner to be described presently. Since this list is short, it may economically be stored in a high-speed memory array, and its inclusion does not substantially increase the time required to perform the hypenation routine. While the step required to perform the hyphenation routine. While this step has been included in the preferred embodiment of the invention, it should be understood that it is optional, and satisfactory hyphenation can be achieved without it. However, it has been found that the inclusion of this step can lead to a 15 to 20% increase in the percentage of words which are accurately hyphenated.

As may be seen from the flow diagram of FIG. 1, if a word is found in the most-frequent-error list, the proper hyphenation pattern is read into an output-word register. If the word is not found in the most-frequent-error list, the process proceeds to box **212**, which is the next step in the operation.

As shown by box **212**, the next step in the operation is to form the syllable patterns. This rather involved step is illustrated in detail by the flow diagram of FIGS. 3A–3B. Referring to FIG. 3A, it is seen that the first step in the form-syllable-patterns operation is to code the first character of the word, read it into an argument register, and to associate this syllable on a table of acceptable syllable patterns. This step is represented by box **220** in the flow diagram of FIG. 3A. Included in this box and in the remaining boxes of this flow chart are the clock times, designated CL, during which these operations are performed. While these clock times have no significance for the present discussion, their significance will become apparent later.

Assume first that a match is had on the coded character or group of characters stored in the argument register and being associated on during step **220**. In this event, the circuit proceeds to create a table entry as indicated by step **222**. FIG. 6 shows the table entry format and a group of table entries for an illustrative word. The significance of the particular entries shown in FIG. 6 will be described later. However, at this point the format of the table will be described briefly. The table starts at a particular addresses position, in FIG. 6 the address **100**, and succeeding entries are stored in succeeding address positions in the memory. Each entry contains a possible syllable and a figure representing the relative frequency for that possible syllable. As indicated previously, the relative frequency figure is derived by analyzing a representative sample of text of the type to be hyphenated. In addition, each table entry has four additional fields. A level termination mark is stored in the first of these fields if the syllable of that entry is the last possible syllable which starts with the the letter that that syllable starts with. The second of the fields is the next-character-number field. This field contains the character number of the character following the last character in the syllable of that entry. The third additional field is a chain-address field. This field contains a zero if there is no possible syllable which starts with the letter following the last letter of the syllable of the entry; a two if the syllable of the table entry is the last syllable of the word (i.e. contains the last letter of the word); and the table address of the first table entry starting with the letter following the last letter of its own syllable if such an entry exists. The last additional field is a used-bit field. This field initially contains all zeros. The significance of a one bit in this field will be described later.

During step **222** of FIG. 3A, only the syllable and relative frequency fields of the table entry are filled. In an appropriate situation, the level-termination-mark (LTM) field of the entry may also be filled during this step.

Once a table entry has been created, one of three steps may follow. If this is the first table entry starting with a particular letter other than the first letter of the word, then a flip-flop in the circuit referred as the used flip-flop is in its ON state. The significance of this flip-flop being in its ON state is to indicate to the circuit that the address in the table at which the present entry is being stored should be read into the chain-address field of a previous entry, the syllable of which ends with the letter preceding the letter which the syllable of this entry starts with. Referring to the flow diagram of FIG. 3A, it is seen that, after a table entry has been made, if the used flip-flop is in its ON state, the process proceeds to step **224**, during which the chain address is written into the proper position in the table. From step **224**, the circuit proceeds to step **226**, during which step an entry is made in what will be referred to as the supplementary table.

FIG. 7 shows the format for the supplementary table. The entries shown in FIG. 7 relate to the same illustrative example as that which gave rise to the entries shown in previously mentioned FIG. 6. Comparing FIGS. 6 and 7, it is seen that the next-character-number field of the supplementary table, FIG. 7, contains all character numbers which appear in the next-character-number field of the table, FIG. 6, and the chain-address field of the supplementary table contains the chain addresses from the table of FIG. 6 corresponding to the next character numbers. The purpose of the supplementary table is to avoid repeated look-ups when two different syllable strings are identical from a certain point. Before the chain address for a particular entry is determined and entered, a search is made in the supplementary table to see if a chain address for this next character number has been previously determined. If it has been previously determined, then this chain address is entered into the chain-address field of the table entry, and no further determination is required.

It is noted that in addition to a table entry being created during step **222**, if a syllable containing six letters is in the argument register, a level termination mark is, as was mentioned previously, written into the level-termination-mark field of the table entry being created. The reason for this is that, for the embodiment of the invention being described, syllables over six letters long are not considesed. A counter called the letter-per-word counter (abbreviated l/w) keeps track of the number of characters read into the argument register and has a count of seven in it when there are six characters in the argument register. The notation $l/w=7$ on the flow chart of FIG. 3A therefore means that a six-letter syllable is in the argument register. After the table entry, including the level termination mark, has been created in step **222**, if a syllable containing six letters is in the argument register, the circuit proceeds to find the next-character number in a manner to be described later. Finally, if, after a table entry is created, it is found that the used flip-flop is in its OFF state and that a syllable containing less than six letters is in the argument register, the process returns to step **220**.

Assume now that a match is not had on the character or group of characters being associated on during step **220**. The next operation is then that indicated by box **228**, namely a determination as to whether the argument register contains a six-letter syllable or whether the end of the word has been reached. If neither of these conditions exists, step **220** is returned to, causing an additional character to be coded and read into the argument register and a new association to be made.

As was indicated previously, it has been assumed, for purposes of the lilustrative embodiment, that syllables in excess of six letters will not be encountered. Therefore, when six characters have been read into the argument register, it is assumed that all possible syllables beginning with the first letter in the argument register have been

found and the circuit is ready to seek additional syllables starting with later letters in the word. Of course, when the end of word is reached, there can be no additional syllables beginning with the first letter in the argument register, and, under this condition, the circuit is also ready to seek syllables beginning with subsequent letters in the word. Therefore, if either of the tests indicated by box **228** lead ot an affirmative answer, the operation proceeds to step **230** during which a determination is made as to whether a table entry has been made for any of the letters starting with the first letter in the argument register.

Assume that the determination made in box **230** indicates that a table entry has not been made for this syllable string. This means that there are no possible table entries which can start with the first letter now in the argument register and therefore any possible syllable which ended with the letter preceding this first letter in the argument register is not a valid syllable for the word being tested. This is indicated by writing a zero into the chain-address field of this previously recorded table entry, thereby indicating that no subsequent syllable is chained to this syllable and threefore it is an invalid one. In the flow chart of FIG. 3A, this step is indicated by the box **232**. If the operation of box **230** indicates a table entry has been made for the particular syllable string being investigated, the circuit proceeds to previously described step **226**. The circuit also proceeds to step **226** after a chain address of zero has been entered into the proper table entry during step **232**. If step **226** is arrived at from step **230**, during step **226** a level termination mark is written into the LTM field of the last entry starting with the letter which is then the first letter in he argument register.

If the argument register contains a six-letter syllable or the end of the word has been reached, the process proceeds from step **224** to step **234**, regardless of whether a table entry has been created or not. During step **234**, the next-character-number (NCN) field of the first table entry whose NCN field has not been previously looked at is found. The process then proceeds to step **236** during which this field is looked at to determine if it is blank. If this field is blank, it means that the syllable just looked at is a possible last syllable for the word, and a two is then written into the chain address of this table entry during step **238**. A determination is then made as to whether this is the last table entry. If it is not the last table entry, the circuit proceeds back to step **234** to find a new next-character number. If it is the last table entry, the circuit proceeds on to the next step in the forming of the syllable patterns, which step will be described later.

If during step **236** it is found that the next-character-number field for the entry being looked at is not blank, the process proceeds to step **240**. During this step, a search is made in the supplementary table (see FIG. 7) to see if this next-character number has been previously used. If syllable strings starting with this next-character-number have not previously been formed, the circuit proceeds to step **242** during which time the circuit is set up to cause the character of the input word indicated by the next-character-number field to be selected. From step **242**, the process proceeds back to step **220** to cause the previously described character to be coded and read into the first position of the argument register. The circuit then proceeds as previously indicated to find and record possible table entries starting with this letter and succeeding letters.

If during step **240** it is found that syllables starting with the character in question have already been found and entered in the table, or that there are no possible syllable patterns starting with this particular character, then there is no need to perform this search and record operation again. The process therefore proceeds to step **224** during which the chain address for this next-character-number as found in the supplementary table, is read into the chain-address field of the current table entry. From step **244**, the circuit proceeds back to step **234** to find a new character at which syllables may begin.

Referring now to FIG. 3B, it is seen that when the last table entry is found during step **238**, the circuit proceeds to step **250**. Up until this point, the circuit has been searching for possible syllables and storing these syllables with their relative frequency when found. The remainder of step **212** (FIG. 1) is concerned with forming these possible syllables into syllable patterns and computing various statistics for these patterns.

During step **250**, a table entry is selected and looked at. The first time step **250** is performed, the table entry looked at is the first entry in the table shown in FIG. 6. Later, other criteria are used for selecting the particular table entry to be looked at. During step **250**, the used-bit field of the selected table entry is looked at to determine if it contains a one bit. A one bit in this field means that the particular table entry in question has been previously looked at and recorded and that the required statistics relating to it have already been computed. Therefore, if the used-bit for the table entry is a one, the process proceeds to step **252** to select the succeeding table entry. From step **252**, the process returns to step **250** to look at this table entry and determine whether its used-bit is a one.

If the used bit of the selected table entry is a zero, it means that this table entry has not yet been used and that it should be investigated further. When the used bit is a zero, the circuit therefore proceeds from step **250** to step **254**. During step **254**, the circuit investigates the chain-address field of the table entry to determine if it has a chain address of zero. It will be remembered that if an entry has a chain address of zero, it means that there are no acceptable syllable patterns starting with the letter following the last letter of the syllable for that entry and therefore that the syllable for that entry is not a proper one for the word being investigated. Therefore, if an entry has a chain address of zero, it need not be investigated further, and the process proceeds to step **255** to determine if the entry has a mark in its level termination field.

A zero bit in the LTM field (i.e., no mark) means that the succeeding entry starts with the same letter as this entry and is therefore a possible syllable for the pattern being formed. Therefore, if there is a zero in the LTM field of the entry the process proceeds to step **252** to select the next table entry. If, on the other hand, there is a mark in the LTM field of the entry (i.e. the LTM field does not contain a zero) this means that the entry is the last one beginning with the particular letter and, since it has a chain address of zero, that the syllable pattern being formed cannot be completed (i.e. that it is not a possible syllable pattern for the word in question). The process therefore proceeds directly to step **264** to determine if all possible syllable patterns have been formed. Step **264** will be discussed later.

If during step **254** it is found that the entry does not have a chain address of zero, then this table entry may be a proper one for the word being investigated, and the process proceeds to step **256**. During this step, the syllable, its frequency, and its address are stored in various registers. From this step, the circuit proceeds to step **258**. During step **258**, a determination is made as to whether the chain address for the entry is equal to two. If the chain address is equal to two, then the end of the word has been found and a complete possible syllable pattern has been formed. If the chain address is not equal to two, then the circuit proceeds back to step **252**. However, when step **252** is entered in this manner, it is the address indicated by the chain-address field of the previously selected entry which is now selected rather than the next address, as was previously the case.

It can be seen that the process described above causes all the syllables in a chain to be read, in sequence, into a storage register, which register is referred to as a word

register. The last syllable in the chain will have a chain address of two. Therefore, when the entry for the last syllable is being operated upon, the circuit proceeds from step **258** to step **260**.

During step **260**, various statistics for the possible syllable pattern previously determined are generated. While various statistics might be employed in determining the most probable hyphenation pattern, it has been found that the most accurate results are obtained when the statistic used is the ratio of the average frequency to the square root of the sum of the squares of the deviations of the individual syllable frequencies from the average. Therefore, during step **260**, the above described ratio is computed for a selected syllable pattern. When this ratio has been computed, the process proceeds to step **262** during which the computed ratio is stored in the word register with the syllable pattern it is associated with.

The process then proceeds to beforementioned step **264** during which a determination is made as to whether all possible syllable patterns have been formed (i.e. a determination is made as to whether the ratio just computed was for the last of the possible syllable patterns). If there are still possible syllables in the table which have not been used, then the process proceeds to step **266** during which a used bit is read into the used bit field of the next-to-the-last entry looked at and a word position is selected in the word register to store a new possible syllable pattern. If the process enters step **264** from step **255**, the word position selected during step **266** is the same position previously used whereas if step **264** is entered from step **262**, the following word position is selected during step **266**. From step **266**, the process proceeds back to step **252** to select a new table entry which may be the first syllable of a new possible syllable pattern.

When, during step **264**, it is determined that all possible syllable patterns have been formed, the process proceeds to step **270** (FIG. 1). Referring to FIG. 1, it is seen that during step **270** the preferred pattern is selected. FIG. 4 is a detailed flow diagram of the individual steps which are performed during the select-preferred-pattern operation **270**. Referring to FIG. 4, it is seen that the first step of the select-preferred-pattern operation is to select a word from the word register. The word selected by this step (step **272**) is the word in the first word position of the word register. Having decided that the most significant statistic for selecting the preferred syllable pattern is the ratio of the average frequency to the square root of the sum of the squares of the deviations of the frequencies of the individual syllables from the average frequency, the selecting of the preferred pattern becomes a relatively straight-forward matter. The particular criteria which have been found to be the most effective is that the preferred pattern be that with a ratio greater than one. Where there is more than one possible syllable pattern with a ratio greater than one, the preferred pattern is that with the fewest syllables. Where these two criteria are not sufficient to identify a preferred syllable pattern, the preferred pattern is that with the highest ratio.

FIG. 4 shows the steps which are performed to implement the above criteria. During step **274**, the ratio field of the position in the word register selected during step **272** is investigated to determine if it contains a ratio greater than one, less than one, or blank. If there is a possible syllable pattern in the selected position of the word register, the ratio field wil not be blank. Therefore, a blank indication means that all possible syllable patterns have been investigated and is used as an indication that step **270** of the process has been completed. If during step **274** it is found that the ratio in the ratio field is less than or equal to one but is not a blank, then, according to the criteria indicated above, the pattern yielding this ratio is not a possible syllable pattern for the word in question, and the process proceeds to step **276**. During step **276**, the word following the word which was just looked at is selected. From step **276**, the process proceeds back to step **274** to lock at the ratio field of the now selected word.

If, during step **274**, it is found that the ratio in the ratio field of the selected word position is greater than one and this is the first word found with such a ratio, a register is set to indicate that the word register position presently being investigated contains the preferred word. This register will be referred to as the preferred-word register. The setting of the preferred-word register is effected during step **278**. From step **278**, the process proceeds to step **276** to cause the next position in the word register to be selected.

If during step **274**, it is found that the ratio field of the selected position contains a ratio greater than one, and it is further determined that this is not the first such ratio greater than one that has been found, the process proceeds from step **274** to step **280**. During step **280**, a comparison is made between the number of syllables in the word presently being looked at and the number of syllables in the word contained at the memory register position indicated by the preferred-word register. If the word at the memory position indicated by the preferred-word register has the lesser number of syllables, then this is still the preferred word, and the process proceeds to step **276** to investigate the next possible syllable pattern. If the word presently being investigated has the lesser number of syllables, then this is the preferred word, and the process proceeds to step **278** to set the word register position of this word into the preferred-word register. If the word presently being investigated and the word at the address indicated in the preferred-word register both have the same number of syllables, then the process proceeds to step **282** to determine which has the highest ratio. If the word presently being investigated has the highest ratio, then the process proceeds to step **278** to set the register position of this word into the preferred-word register. If the ratio of the word in the preferred-word register is greater than or equal to that of the word presently being investigated, then this is still the preferred word, and the process proceeds to step **276** to investigate a new word pattern.

If during step **274**, it is found that the ratio field contains a blank, then, as has been previously indicated, all possible syllable patterns have been investigated and, since it has been found that the above described procedure will lead to at least one syllable pattern with a ratio greater than one, that a preferred syllable pattern has been selected. The process is therefore ready to proceed to step **290** (FIG. 1) and read out this preferred pattern. Since the preferred pattern in the word register is stored in the reduced character set, it is necessary, when reading this prefered pattern into the output register, to read the letters from the input register into the output register rather than from the word register. The read-out operation **290** therefore involves the reading of characters from the input register to the output register under control of the hyphenation pattern indicated in the word register.

GENERAL CIRCUIT DESCRIPTION

Having seen from the flow diagrams of FIGS. 1–4, the process for performing automatic word hyphenation in accordance with the teachings of this invention, the circuitry for performing these functions shown in FIGS 5A–5T will now be investigated. Before describing in detail the various elements shown in this figure, the major elements of the circuit will be identified, and the inter-relation of these major elements to perform the automatic hyphenation function will be briefly described. To assist in distinguishing the major elements to be described now from the remainder of the circuit, heavier lines have been used in drawing these elements, and heavier characters have been used for the reference numerals designating these elements.

Input words are initially applied a letter at a time through lines **300** (FIG. 5A) to input-buffer register **302**. If an overflow condition is not detected by overflow-

detection circuit **304** (FIG. 5C) before an end-of-word signal is applied to line **306** from the input circuit, a signal is generated on line **308** to reset input-buffer register **302**. An overflow is indicated at the end of a word by a signal on line **309** from overflow-detection circuit **304**. As each input character is applied to input-buffer register **302**, a signal is also applied through line **310** (FIG. 5C) to increment letter-per-word (*l/w*) counter **312**.

The elements described above are the basic elements in performing steps **202** and **204** of FIG. 1. If overflow-detection circuit **304** (FIG. 5C) generates a signal on line **309** indicating that an overflow has occurred and letter-per-word counter **312** indicates that there are two or less characters in input-buffer register **302**, AND gate **314** generates an output signal which ultimately results in a do-not-hyphenate signal on output line **316** from OR gate **318**. If *l/w* counter **312** indicates that there are four or less letters in input-buffer **302** at the same time that an end-of-word indication is received on line **306**, AND gate **320** is fully conditioned to generate an output signal through OR gate **318** to do-not-hyphenate line **316**.

If the overflow did not occur on the first or second character and the word is more than four characters long, AND gate **322** is fully conditioned, causing an output signal on test-for-number-of-vowels line **324**.

It will be noticed that output line **324** bears the legend SS–**50**. Legends such as SS, XX or CL, XX appear throughout FIGS. 5A–5T. These legends refer to inputs or outputs respectively, to and from clock circuit **326** (FIG. 5T), which clock controls the sequencing and timing of the various operations performed by the circuit. The operation and function of this clock will not be considered in this section, but will be considered in detail with reference to the Appendix in the "Detailed Circuit Description" and "Detailed Operations Description" sections. For purposes of the present discussion, it will be assumed that proper timing and sequencing of operations have occurred without specific mention being made thereof.

FIGS. 5G–5J show a decoding circuit **328** for counting the number of vowel groups contained in the word stored in input-buffer register **302** according to the criteria for counting vowel groups previously stated. These criteria are schematically indicated in the flow diagram of FIG. 2. Selected letters of the input word are applied to vowel-group-counting circuit **328** from input-buffer register **302** (FIG. 5A) through OR gates **330** and lines **332**. If it is found by vowel-group-counting circuit **328** that the word contained in input-buffer register **302** contains less than two vowel groups, an output signal appears on line **334** (FIG. 5J) which is applied through OR gate **318** (FIG. 5C) to do-not-hyphenate line **316**. A signal on line **316** is applied to the printing device (not shown) to cause the word which it has been determined cannot be hyphenated to be omitted from the line of text when it is printed. If, on the other hand, it is found that the word in input-buffer register **302** contains two or more vowel groups, vowel-group-counting circuit **328** generates an output signal on line **336** (FIG. 5J) which causes step **210** (FIG. 1) of the automatic hyphenation process to be initiated.

Referring now to FIG. 5A, it is seen that the circuit includes an associative memory **338** and an argument register **340**. Associative memory **338** has an argument portion **341** which contains a short list of words which are encountered with some degree of frequency and which would be erroneously hyphenated if hyphenated in accordance with the statistical procedure of this invention. Function portion **342** of the associative memory contains the proper hyphenation pattern for each of these words. Since the list is a short one, it may be stored in a small memory and rapidly scanned. Lines **344** connect the output from function portion **342** of associative memory **338** to an output register **346** (FIG. 5L) AND gates **350** and **354**, (FIG. 5A), having output lines **348** and **352** respectively, are provided for detecting the result of the associate operation.

To perform step **210** of the process (to determine if the word in input-buffer register **302** is in the list of most frequent errors) the word in input-buffer register **302** is read into argument register **340**. An association is then performed. If the association is successful, the proper hyphenation pattern is generated on lines **344** and applied to output register **346**. An indication is the given on line **348** from AND gate **350** that the hyphenation procedure has been completed. If, on the other hand, the word is not found in the table of associative memory **338**, a signal is generated on output line **352** from AND gate **354**, causing the circuit to proceed to step **212**, the form-syllable-patterns step in the process.

In addition to applying the contents of input-buffer register **302** (FIG. 5A) to vowel-group-counting circuit **328**, lines **332** also apply the characters of the input word, a character at a time, through coder **358** (FIG. 5B) to selected positions in argument register **360**. The code conversion effected by coder **358** is shown in FIG. 8. The reduced character-set characters stored in argument register **360** are compared with acceptable syllable patterns in the reduced character set stored in an associative memory **362**. The relative frequency of occurrance of each syllable pattern is stored in associative memory **362** with the syllable pattern.

Therefore, to perform step **220** (FIG. 3A), a selected character of the input word is applied through OR gates **330**, lines **332**, and coder **358** to argument register **360**. The syllable pattern then stored in argument register **360** is compared with acceptable-syllable pattern stored in associative memory **362**.

Referring now to FIG. 5F, table entries of the type shown in FIG. 6 and supplementary table entries of the type shown in FIG. 7 are stored in a high-speed magnetic-core memory **364**. Inputs are applied to core storage **364** through memory-buffer register (MBR) **368**. The six possible fields in the memory-buffer register are shown in FIG. 5F. These fields correspond to the six nonaddress fields in the table of FIG. 6. Line **370** from associative memory **362** (FIG. 5B) applies syllable and frequency data to the appropriate fields in memory-buffer register **368**. The address at which information contained in memory-buffer register **368** is to be stored in core storage **364** or from which information is to be read from core storage **364** into memory-buffer register **368** is controlled by memory-address register (MAR) **372** (FIG. 5E). Other important components during this phase of the operation are used-flip-flop **374** (FIG. 5B) and gating circuit **376**.

If during step **220** (FIG. 3A) a matching is found, the circuit is conditioned to read the matched-on-possible syllable and its associated frequency through lines **370** into memory-buffer register **368**. If at the same time letter-per-word counter **312** (FIG. 5C) contains a count of seven, gate **378** (FIG. 5F) is conditioned to gate a one bit from generator **380** into the level-termination mark field of MBR **368**. From MBR **368**, the information is read into core storage **364** at the address indicated in MAR **372**. At this point, one of three things can occur, depending on the state of used-flip-flop **374** (FIG. 5B) and on the count in letter-per-word counter **312** (FIG. 5C). If used flip-flop **374** is in its OFF state and letter-per-word counter **312** has a count of less than seven in it, gating circuit **376** (FIG. 5B) generates an output signal on line **382** causing the circuit to return to state **220** (FIG. 3A) to cause the next character to be read into argument register **360**. If the used flip-flop is in its OFF state and the count in letter-per-word counter **312** is equal to seven, then gating circuit **376** generates an output signal on line **384** which switches the circuit to state **234** (FIG. 3A). The circuitry for performing operations during this step will be described later. Finally, if used flip-flop **374** is in

its ON state, gating circuit **376** generates an output signal on line **385** which switches the circuit to state **224** (FIG. 3A).

The principal elements employed during step **224** are table-address counter (TAC) **386** (FIG. 5D) and previous-address register (PAR) **390** (FIG. 5E). Anytime information is read into MAR **372**, the address presently in MAR is transferred into PAR **390**. TAC **386** is incremented after each table entry is made so that at any given time it contains either the address at which a table entry is being made or the address at which a table entry is about to be made.

To understand how step **224** (FIG. 3A) is performed, it should be remembered that when used-flip-flop **374** is in its ON state, it means that all possible syllable patterns starting with a particular letter have been found and that any syllable pattern now found will start with a letter following the last letter of a previously found syllable. Any syllable found at this time should therefore be chain addressed to the syllable preceding it which has a next-character number which is the same as the first letter of the syllable found. As will be seen later, under these circumstances the last access to memory **364** was to find the character at which a new search for syllables should begin. Any syllable found starting with that character should be chain addressed back to the address from which this next-character number was retrieved. Therefore, when the address in TAC **386** is transferred into MAR **372** to cause the syllable found to be properly entered during step **222** (FIG. 3A), the address at which it is desired to place the chain address is transferred into PAR **390**.

Therefore, during step **224**, the first operation is to transfer the address in PAR **390** into MAR **372**. Circuitry is then energized to cause the address in TAC **386** to be transferred into the chain-address field of MBR **368**. The desired chain address is therefore properly entered into memory **364**.

From step **224**, the circuit proceeds to step **226**, which at this point, involves the making of a supplementary table entry. The major elements for performing this function are supplementary-address counter (SAC) **392** (FIG. 5D), next-character-number register **394** (FIG. 5F), and previously mentioned table-address counter **386** (FIG. 5D).

As a result of previous operations, at the start of step **226**, register **394** contains the next-character number used in locating the syllable just entered, and TAC contains the address of this syllable. In order to make the supplementary table entry MBR **368** (FIG. 5F) is first cleared and then the contents of next-character register **394** are read into the next-character-number field of MBR and the contents of TAC **386** are read into the chain-address field of this register. Since SAC **392** contains the address at which a supplementary table entry is to be made, the address in SAC is transferred into MAR and the information in MBR **368** is then transferred into memory **364** at the address indicated by MAR. A supplementary table entry of the form shown in FIG. 7 is in this way created.

When the supplementary table entry has been made, gating circuit **396** (FIG. 5B) is used to determine whether the entry was made for a syllable which is the last syllable of a word or whether the syllable entered was a six-letter syllable. If either of these conditions exist, an output signal is derived on line **398** causing the circuit to proceed to step **234**. If neither of these conditions exist, an output signal is derived on line **400**, causing the circuit to return to step **220**.

The function indicated in box **228** of FIG. 3A, namely the determination as to whether a six-letter syllable or the end of word has been stored in argument register **360** when a mismatch condition is detected during step **220**, is performed by gating circuit **402** (FIG. 5B). If neither a six-letter syllable nor the end of word is in argument register **360**, gating circuit **402** generates an output signal on line **404** causing the circuit to return to state **220**. If, however, either a six-letter syllable is in argument register **360** or the end of the word has been reached, gating circuit **402** generates an output signal on line **406** causing the circuit to proceed to step **230**.

Step **230** is performed by table-entered-flip-flop **408** (FIG. 5A) and gating circuit **410**. Table-entered-flip-flop **408** is switched to its ON state whenever a table entry is made. If, during step **230**, this flip-flop is in its ON state, it means that a table entry has been made starting with the first letter then in argument register **360**. If a table entry has been made, gating circuit **410** generates an output signal on line **412**, causing the circuit to proceed to step **226** (FIG. 3A). During step **226**, gate **378** (FIG. 5F) would be conditioned to write a level-termination mark into the proper field of MBR **368** in the manner previously described. Since no table entry is made during step **230** no supplementary table entry is made during step **226** when it is entered from step **230**. If, during step **230**, it is found that a table entry has not been made (flip-flop **408** is in its OFF state), then gating circuit **410** generates an output signal on line **414** causing the circuit to perform step **232**.

Since no possible syllable pattern was found starting with the letter indicated in the next-character-number field of the table entry contained at the address indicated in MAR **372**, it is now desired to write a zero into the chain-address field of this table entry. The major elements for performing this step are zero-signal-code generator **416** (FIG. 5E) and gates **418**. During step **232**, gates **418** are conditioned to pass the code for the zero signal into the chain-address field of MBR **368**. From MBR, this zero is transferred into the chain-address field of the desired table entry. After step **232** has been performed, the circuit proceeds to write the proper chain address, in this case a zero, into an entry in the supplementary table in a manner previously described. Since no entry is made starting with the particular letter contained in the first position of argument register **360**, a level-termination mark is not written during step **226** if this step follows step **232**.

When, during step **226**, it is found that either a six-letter syllable or the end of the word is contained in argument register **360**, the circuit proceeds to step **234**. The major element for performing this step is next-character-address counter **420** (FIG. 5D). This counter always contains the address in memory **364** of the table entry, the next-character-number field of which contains the character number of the next character which it will be attempted to begin syllables with. During step **234**, the address in next-character-address counter **420** is transferred to MAR **372**. The entry contained at this address is then transferred into MBR **368**.

From step **234**, the circuit proceeds to step **236**. The major elements for performing this step are blank decoder **422** (FIG. 5F) which has the contents of the next-character-number field of MBR **368** applied to it through lines **424** and gating circuit **426** which samples the output from blank decoder **422**. To perform step **236**, the content of the next-character-number field of MBR **368** are applied through lines **424** to blank decoder **422**. If the decoder indicates that the contents of this field are blank, gating circuit **426** generates an output signal on line **428** causing the circuit to proceed to step **238**. If, on the other hand, blank decoder **422** indicates that the contents of this next-character-number field are not blank, gating circuit **426** generates an output signal on line **430**, causing the circuit to proceed to step **240**.

It will be remembered that if the next-character-number field of a table entry is blank, it means that this is a possible last syllable for the word; however, there may be other possible last syllables, and this therefore may not be the last table entry. Therefore, two things are done during step **238**. First, a chain address of two is written into the current table entry; and second, an in-

vestigation is made to determine if this is the last table entry. The major elements for performing the first of these steps are coded-character "2," generator **432** (FIG. 5E) and gates **434**. Compare-and-gating circuit **436** (FIG. 5D) is the major element for performing the second step.

In operation, gates **434** are conditioned to permit the coded-character "2" to be read into the chain-address field of the table entry then stored in MBR **368**. This table entry is then read back into core storage **364**. A comparison is then made in circuit **436** to determine if TAC **386** (FIG. 5D) and next-character-address counter **420** have the same count in them. If they do, as will be seen later, this means that the table entry looked at is the last table entry and compare-and-gating circuit **436** generates an output signal on line **438** causing the circuit to proceed to step **250**. Step **250** will be described later. If, on the other hand, the last table entry has not been found, compare-and-gating circuit **436** generates an output signal on line **440**, causing next-character-address counter **420** to be incremented one position. The circuit then proceeds back to step **234** to find a new next-character number.

Referring back to FIG. 3A, it is seen that if, during step **236**, it is found that the next-character-number field is not blank, the circuit proceeds to step **240** to determine if a syllable string starting with the character indicated in the next-character-number field of the table entry presently in MBR **368** has previously been found. This requires that a search be made in the supplementary table (see FIG. 7). Major elements used in performing step **240** are compare-circuit **442** (FIG. 5F), gating- circuit **444**, register **446** (FIF 5D, compare-circuit **448**, and gating-circuit **450**. Previously mentioned supplementary-address counter **392** and next-character-number register **394** are also employed.

In operation, the first sub-step is to read the next character number being applied by MBR **368** to lines **424** into next-character-number register **394**. The address of the start of the supplementary table is then applied to supplementary-address counter **392** (FIG. 5D), and from this register to MAR **372** (FIG. 5E). The next-character-number field of the supplementary table entry being investigated is then compared in compare circuit **442** (FIG. 5F) with the next-character-number register **394**. If these next-character numbers are found to be equal, gating circuit **444** generates an output signal on line **452** causing the circuit to proceed to step **244** (FIG. 3A). If the two next-character-number fields are not equal, gating circuit **444** generates an output signal on line **454**. A signal on line **454** does not, however, necessarily mean that the desired next-character number stored in next-character-number register **394** is not in the supplementary table. A further test must be made to determine if all entries in the supplementary table have been looked it. Therefore, the signal on line **454** causes a comparison to be made between the address in supplementary-address counter **392** (FIG. 5D) and the address of the last entry in supplementary-address counter **392** which has previously been transferred into register **446**. This comparison is made in compare-circuit **448**. If the entries are not equal, indicating that a search of the supplementary table has not been completed, gating circuit **450** generates an output signal on line **456** which causes the address in supplementary-address counter **392** to be incremented one position, resulting in a new supplementary-table entry being read into MBR **368**. If compare-circuit **448** indicates that the entries are equal, this means that the next-character number stored in the next-character-number register **394** is not in the supplementary table, and gating circuit **450** generates an output signal on line **458** causing the circuit to proceed to step **242** (FIG. 3A).

As indicated previously, if the next-character number is in the supplementary-address table, the circuit proceeds to step **244** (FIG. 3A) to write the chain address into the current table entry. The major elements foar performing this step are chain-address register **460** (FIG. 5E) which receives an input from the chain-address field of MBR through lines **462** and previously described next-character-address counter **420** (FIG. 5D). To perform step **244**, the chain-address field of the proper supplementary-table entry is read through lines **462** into chain-address register **460**. The address in next-character-address counter **420** is then read into MAR **372**. Finally, the chain-address in chain-address register **460** is read into the chain-address field of the table entry, the address of which has just been read into MAR. When step **244** has been completed, next-character-address counter **420** is incremented one position, and the circuit returns to step **234**.

As indicated previously, if the next-character number contained in next-character-number register **394** is not in the supplementary table, gating circuit **450** (FIG. 5D) generates an output signal on line **458** causing the circuit to go to step **242** (FIG. 3A). During this step, a determination is made as to where to start looking for the next syllable string. This step is performed by the circuit of FIGS. 5A–5T in an indirect manner, which will not be described at this point. Suffice to say at this point that the effect of this step is to cause step **220** to commence with the character indicated in the next-character-number field of the entry at the address indicated in next-character-address counter **420**. During this step, the next-character-address counter is also incremented by one, and many of the flip-flops and registers in the circuit are appropriately reset.

When it is determined that the last table entry has been made during step **238**, the circuit is ready to form the possible syllables into syllable patterns. The circuit therefore proceeds to step **250** (FIG. 3B) to read a table entry. The first time this operation is performed, the first address at which an entry is made in the table is read into TAC **386** (FIG. 5D), and this address is transferred into MAR **372**. The contents of this address are then transferred into MBR **368**. The manner in which later entries are selected to be read into MBR **368** during step **250** will be described later. Having read the table entry into MBR **368**, the circuit proceeds to determine whether the entry has a used bit of one. The elements for performing this function include decoder **470** (FIG. 5F) and gating-circuit **472**. If the used bit is found by decoder **470** to be a zero, gating circuit **472** generates an output signal on line **474** causing the circuit to proceed to step **254** (FIG. 3B). If the used bit is found by decoder **470** to be a one, gating circuit **472** generates an output signal on line **476** causing the circuit to proceed to step **252** to select a new table entry. The reason for selecting a new table entry if the used bit in an entry is a one is that, as indicated in a previous section, the significance of a one in the used-bit field of an entry is that this has previously been used in forming syllable patterns. The function of the used bit is therefore to prevent the same operation from being repeated.

Step **252** is implemented by incrementing TAC **385** (FIG. 5D). From step **252**, the circuit proceeds back to step **250** to read TAC into MAR **372**.

If, during step **250**, the used bit is found to be zero, this means that this table entry has not previously been used, and the circuit proceeds to step **254**. The major elements employed in steps **254**, **255**, **256** and **258** of the operation are zero-detector **480** (FIG. 5E), two-detector **482**, gating circuit **484**, AND gate **485**, gating circuit **486**, flip-flop **487**, word registers **488** (FIGS. 5R–5S), frequency register **490** FIG. 5M) and zero-detector **512** (FIG. 5F). The input to both zero-detector **480** and two-detector **482** are lines **462** from the chain-address field of MBR **368**. The input to zero-detector **512** is output line **520** from the LTM field of MBR **368**. Gating circuit **484** allows address information from MAR **372** and

status and field information from the appropriate fields of MBR **368** to be passed through lines **492**, **494** and **496** respectively to previously selected storage positions in word register **488** and frequency register **490**. The manner in which positions are selected in word register **488** and frequency register **490** will be described later.

In operation, steps **254**, **255**, **256**, and **258** of FIG. 3B are performed at substantially the same time. The signal applied by line **462** to zero-detector **480** causes an output which forms one input to gating circuit **486** if a zero is detected. If a zero is not detected, the output from detector **480** conditions the gates in gating circuit **484** to pass the syllable address, the syllable, and its frequency into selected positions in word registers **488** and frequency register **490**. The outputs from both zero-detector **480** and two-detector **482** are applied to gating circuit **486**. The output from the zero side of zero-detector **480** and the not-zero side of zero-detector **512** are connected as the inputs to AND gate **485**. Output line from this AND gate is connected to switch flip-flop **487** to its ON state. This flip-flop is therefore in its ON state only when the chain-address field of the entry contain a zero and the LTM field of the entry contains a mark.

After an entry has either been made or attempted (if the output from zero-detector **480** is a zero, no entries are made) gating circuit **486** is sampled. If zero-detector **480** indicates that the chain address is a zero and flip-flop **487** is in its OFF state indicating that there is no mark in the LTM field, gating circuit **486** generates an output signal on line **499** causing the circuit to return through step **252** to step **250**. TAC is incremented prior to this so that a new table entry is selected. If detectors **480** and **482** indicate that there is neither a zero nor a two in the chain-address field of the table entry, gating circuit **486** generates an output signal on line **500**, which causes the address in the chain-address field to be read into MAR. Therefore, when the circuit subsequently returns to step **250**, the table entry read will be that at the address indicated in the chain-address field. This is the manner in which a string of syllable patterns is formed in word register **488**. If flip-flop **487** is in its ON state indicating that there is a zero in the chain-address field of the entry and a mark in the LTM field, gating circuit **482** generates an output signal on line **501** causing the circuit to proceed to step **264**. Finally, if detector **482** indicates that the chain-address field contains a 2, gating circuit **486** generates an output signal on line **502** causing the circuit to proceed to step **260**.

During step **260**, the ratio of the average frequency for the formed syllable pattern to the square root of the sum of the squares of the deviations of the individual frequencies from the average frequency is computed. The major elements in performing this function are frequency register **490** (FIG. 5M) and calculate-circuit **504**. In operation, when all of the syllables for a given syllable pattern have been found and their frequencies entered into frequency register **490**, these frequencies are applied to calculate-circuit **504**, which circuit computes the desired ratio and applies it to output lines **506**. The ratio on lines **506** is stored in the ratio register for the particular word in word register **488** (FIG. 5S). The transferring of the ratio from compute-circuit **504** to word register **488** is accomplished during step **262** in the flow diagram of FIG. 3B.

Steps **264** and **266** of the flow diagram of FIG. 3B are performed more or less together. The major elements employed in performing these steps are column-select counter **508** (FIG. 5M), gating circuit **510**, zero-detector **512**, (FIG. 5F), and gating circuit **514**. Column-select counter **508** indicates in which column of word register **488** a particular syllable should be placed. The number in this counter will therefore always be one more than the number of syllables in the particular syllable pattern.

Before describing how the circuit operates to perform steps **264** and **266**, the theory of operation of these steps will be explained. When an acceptable syllable pattern is found, or when it is determined that the syllable pattern being formed is not a possible syllable pattern the circuit then attempts to find alternative syllable patterns starting at any particular point in the word. The circuit therefore first attempts to see if there are alternate last syllables which may be used in conjunction with the syllable pattern formed up to that point. Failing in this, it proceeds back through the word, looking for a place at which a new possible syllable pattern may be begun. When the circuit gets back to possible first syllables and finds that there are no alternative first syllables which have not already been investigated, this phase of the operation is completed and the circuit is ready to proceed to step **272** (FIG. 4).

The above procedure is implemented in the circuit of FIGS. 5A–5T by initially decrementing column-select counter **508** to bring the column position of the last syllable entered in word register **488** into this counter. Gating circuit **510** then determines whether the count then in column-select counter **508** is zero. As will be seen later, if this count is zero, then all possible syllable patterns have been formed, and the circuit is ready to select a preferred pattern. In this situation, gating circuit **510** generates an output signal on line **516**. If column-select counter **508** does not contain a count of zero, gating circuit **510** generates an output signal on line **518**. A signal on line **518** causes the entry at the address indicated for the syllable indicated by column-select counter **508** to be read from memory **364** (FIG. 5F) into memory-buffer register (MBR) **368**. A used bit is placed in the used-bit field of this entry at this time in a manner to be described later. The level-termination-mark field of this entry is then applied through line **520** to zero-detector **512**. If there is a level-termination mark in this field, it means that this is the last syllable starting with the first letter of this syllable and that to find a new starting point for additional syllable patterns, the preceding syllable must be looked at. An output signal on line **522** from gating circuit **514** implements the above operation by causing column-select counter **508** to be decremented one position and the previously described sequence of operations to be repeated. It can be seen that if it were the level-termination field of a possible first syllable which was being looked at at this time, the last step described would cause the column-select counter to be decremented to zero, indicating that all possible syllable patterns have been formed as would in fact be the case. If the level-termination-mark field of the entry looked at does not contain a mark, gating circuit **514** generates an output signal on line **524**, causing the proper word position in word register **588** to be selected and causing a new word pattern to be formed (i.e. causing the circuit to return to step **250**) starting with the syllable presently being looked at.

As indicated previously, if all possible syllable patterns have been formed, the circuit generates an output signal on line **516** from gating circuit **510**, causing the circuit to go to step **270** (FIG. 1) to select the preferred pattern. Referring to FIG. 4, it is seen that the first operation in selecting the preferred pattern is to select the first word from word register **488**. The particular word being selected in word register **488** is determined by word-select counter **528** (FIG. 5P). Therefore, step **272** is easily effected by merely resetting word-select counter **528** to a count of one.

The next step, step **274** (FIG. 4), is to look at the ratio field in the selected word of word register **488**. The major elements in performing this function are blank-detector **530** (FIG. 5P), compare-to-one circuit **532**, first-time-through flip-flop **535**, and gating circuit **534**. Flip-flop **535** is set to its ON state during step **272** and is reset when the first ratio greater than one is detected. Information is applied to blank-detector **530** and comparison-circuit **532** from the selected one of the ratio fields through OR gate **536**.

In operation, it will be remembered that a blank in the ratio field of the selected word means that all possible words have been investigated and that the circuit is ready to read out the preferred pattern (step **290**, FIG. 1). When this situation occurs, gating circuit **534** generates an output signal on line **538**. If during step **274** it is found that the ratio field is less than one but is not blank, gating circuit **534** generates an output signal on line **540** causing the circuit to advance to step **276**. During step **276**, word-select counter **528** is incremented by one. Therefore, when the circuit then returns to step **274**, the ratio field of the word following that which was previously looked at is investigated.

If it is found that the ratio field contains a ratio greater than one and first-time-through flip-flop **535** indicates that this is the first time such a ratio has been located, gating circuit **534** generates an output signal on line **542**. This signal is effective to reset first-time-through flip-flop **535** and to cause the circuit to proceed to step **278**.

The major element in performing step **278** is preferred-word-register **544** (FIG. 5P). During step **278**, the word in word-select counter **528** is transferred into preferred-word register **544**. The significance of this transfer will be apparent later. From step **278**, the circuit proceeds to step **276**, advancing word-select counter **528** in the process and returns to step **274**.

If compare-to-one detector **532** detects that the ratio in the ratio field is greater than one and first-time-through flip-flop **535** has been reset, then gating circuit **534** generates an output signal on line **546** causing the circuit to proceed to step **280** (FIG. 4). Comparison-and-gating circuit **548** (FIG. 5P) is the element employed in performing step **280**. The number-of-syllable fields from the word indicated by word-select counter **528** and the number of syllables from the word indicated by preferred-word register **544** are applied to circuit **548** and an output is derived on line **550** if it is found that the syllable pattern of the word indicated by word-select counter **528** has less syllables than that indicated by the preferred-word register. Since this is now the preferred word, a signal on line **550** causes the address in word-select counter **528** to be transferred to preferred-word register **544** (i.e. causes step **278** to be performed). If the syllable pattern at the address indicated in the preferred-word register has fewer syllables, this is still the preferred pattern, and an output signal is generated on line **552**, causing the word-select counter **528** to be incremented one position to allow the ratio field of the following word to be investigated. Finally, if the two words looked at have the same number of syllables, an output signal is derived on line **554** from circuit **548**, causing the circuit to proceed to step **282**.

Compare-and-gating circuit **556** (FIG. 5Q) is the element employed in performing step **282** (FIG. 4). The ratios contained in the ratio field of the words indicated by the word-select counter and a preferred-word register are applied to circuit **556**, and an output signal is derived on line **558** from this circuit if the higher ratio appears in the word indicated by the word-select counter. This causes the word address in word-select counter **528** to be transferred into preferred-word register **544**. If the ratio for the word indicated by the preferred word register is equal to or greater than the word indicated by the word-select counter, an output signal is derived on line **560** causing word-select counter **528** to be incremented by one, and the circuit to return to step **274**.

When, during step **274** (FIG. 4) gating circuit **534** (FIG. 5P) generates an output signal on line **538** indicating that a blank has been detected in the ratio field of the word in the word register then being looked at, the address in the word register of the most probable syllable pattern is contained in preferred-word register **544**. The circuit is therefore ready to proceed to step **290** (FIG. 1) to read out the preferred pattern.

The principal elements employed in performing step **290** if the operation are coded-word-hyphenation registers **564** (FIG. 5K), scan-control circuit **566**, interpret-and-control circuits **568** (FIG. 5L), gating circuit **570**, and hyphen generator **572**. Output register **346** is also employed during this phase of the operation. There is an individual coded-word-hyphenation register for each possible syllable of a word, and it is assumed that there are enough of these registers to take care of the longest word which may be encountered. Since it has been assumed that syllables more than six characters long will not be encountered, each of the coded-word-hyphenation registers contains only six character positions.

In operation, just before step **290** is begun, the syllable pattern in the word indicated by preferred-word register **544** is transferred into coded-word-hyphenation registers **564**. The syllables transferred into coded-word hyphenation registers **564** are in the reduced character set. Scan-control circuit **566** causes these characters to be applied in succession to interpret-and-control circuit **568**. When a character is detected in a particular position in a coded-word-hyphenation register, a signal is applied through line **574** to gating circuit **570**. The signal on line **574** causes gating circuit **570** to gate the corresponding character in input-buffer register **302** (FIG. 5A) through lines **576** (FIG. 5L) to output register **346**. Successive characters of a syllable are in this manner read into output register **346**. When the scan encounters a blank position in a particular one of the coded-word-hyphenation registers, this means that the end of a syllable has been reached. When this occurs, interpret-and-control circuit **568** generates an output signal on line **578**, permitting a hyphen from hyphen generator **572** to be gated through lines **576** into the appropriate position in output register **346**. Hyhpens are in this manner appled to the proper positions in the output register. If instead of reaching a blank in a hyphenation register, the circuit instead reads out six characters from this register, the scan-control circuit gives an overflow indication to interpret-and-control circuit **568**, resulting in a signal being applied to gating circuit **570** over line **580** to gate a hyphen into the proper position in output register **346**. The word with its preferred hyphenation pattern is in this way read into output register **346**. When interpret-and-control circuit **568** detects a blank in the scanned register position and scan-control circuit **566** gives an indication that this is the first character of a syllable, circuit **568** generates an end-of-operation signal.

DETAILED CIRCUIT DESCRIPTION

As indicated previously, the key element in controlling the operation of the circuit shown in FIGS. 5A–5T is clock **326** (FIG. 5T). This circuit consists of 128 single-shot multivibrators, each of which generates an output signal on a line **600** when it is in its ON state (note that four of these signals are not used), and an output signal on a line **602** when it times out (i.e. returns from its ON to its OFF state). Each of the single-shot multivibrators is switched from its OFF to its ON state by a signal applied to it over a line **604**. In most cases, the turning off of one single-shot multivibrator causes the turning on of another single-shot multivibrator. Therefore, 95 lines of the **602** are shown as being fed back to become lines **604**. Due to the large number of inputs to and outputs from clock circuit **326** and to the fact that these inputs and outputs come from and go to all portions of the circuit shown in FIGS. 5A–5T, no attempt has been made to show the actual connections between the clock circuit and the other elements of the circuit. Instead, the inputs to the clock circuit have been broken off at the particular elements generating them, and the outputs from the clock circuit have been broken off at the particular elements which these outputs are applied to. An appendix starting on page A1 of the specification shows in chart form the turn-on condition for

each of the single shots, the functions which are performed while the single shot is in its ON state, the functions which are performed when this single shot times out, and the next step which the clock circuit proceeds to. The sequence of operations in the circuit may be easily followed by use of the chart in the appendix.

Referring now to FIG. 5A, it is seen that in addition to data-input line **300** and reset line **605**, input-buffer register **302** has two inputs which have not previously been mentioned. These are control lines **606** from input-control ring **608** and control lines **610** from output-control ring **612**. The input and the output-control rings are registers having a number of positions equal to the number of character positions in input-buffer register **302**. Output-control ring **612** also has an overflow position at its lower end. Each of these registers generates an output on only one of its output lines at any given time. Line **308** from overflow detection circuit **304** (FIG. 5C) is connected as one input to OR gate **611** (FIG. 5A). The other input to this OR gate is line **613**. A signal is applied to line **613** when power is applied to the circuit and at the end of each cycle of operation. Output line **605** from OR gate **611** is connected to reset input-buffer register **302**, and to reset input-control ring **608**. Input-control ring **608** is decremented by a signal applied to line **614**. The control signals applied to line **614** are from the external circuitry.

Output-control ring **612** is reset by a signal applied to line **616**. A signal is applied to reset line **616** when power is applied to the circuit and at the end of each cycle of operation. A signal is also applied to line **616** when either single-shot **0**, **33***w*, **50** or **120** of clock circuit **326** is in its ON state. Output-control ring **612** is decremented by a signal applied to line **618**. A signal is applied to line **618** when either single-shot **1***b*, **57**, **58**, **59**, **65**, **66**, **67**, **71**, **72**, **73**, or **78** is in its ON state. Output-control ring **612** is incremented by a signal applied to line **620**. A signal is applied to line **620** when either single-shot **1***a*, **33***x*, **51**, **51***a*, **53**, **55**, **63**, **76**, **122**, or **126** is in its ON state.

The signal on a line **606** from input-control ring **608** determines where data applied to input lines **300** will be stored in input-buffer register **302**. Similarly, a signal on an output line **610** from output-control ring **612** determines where in input-buffer register **302** data will be derived to be applied to output lines **622**. Lines **622** are connected as the inputs to OR gates **330**. One additional input to OR gates **330** is derived from gates **624**, the conditioning input to which is derived from the overflow position of output-control ring **612** and the information input to which is blank-code generator **626**. Blank generator **626** therefore applies a blank code to OR gate **330** when output-control ring **612** has been decremented beyond the first letter in input-buffer register **302**.

Output line **322** from OR gate **330** is connected as the input to blank decoder **628** (FIG. 5B), as the information input to gates **630**, as information input to decoder **632** (FIG. 5G) in vowel-count circuit **328**, and as the information input to gates **634** (FIG. 5L) in gating circuit **570**. Output line **636** from blank decoder **628** is connected through capacitor **638** to the ON-side input of end-of-word flip-flop **640**. Capacitor **638** converts the level output from blank decoder **628** into a pulse level suitable for switching flip-flop **640**. The reset input to end-of-word flip-flop **640** is line **642**. A signal is applied to line **642** when power is applied to the circuit, at the end of each cycle of operation, and when SS–**33***t* of clock **326** is in its ON state. Output line **644** from the ON side of end-of-word flip-flop **640** is connected as one input to AND gate **646** in gating circuit **396**, as one input to AND gate **648**, as one input to AND gate **650** in gating circuit **402**, as one input to AND gate **651** (FIG. 5E), and as a control input to coder **358** (FIG. 5B). Output line **652** from the OFF side of end-of-word flip-flop **640** is connected as one input to AND gate **654** in gating circuit **396**, as one input to AND gate **656**, and as one input to AND gate **657** (FIG. 5C). The other input to AND gates **648** and **656** is line **658**. A pulse appears on line **658** when SS–**1***a* transfers from its ON to its OFF state. As indicated previously, such a pulse is designated by the single-shot number followed by the Greek letter tau (τ). Output line **659** from AND gate **648** is conected to switch SS–**1***b* to its ON state and output line **661** from AND gate **656** is connected to switch SS–**2** to its ON state.

Output lines **622** (FIG. 5A) from input-buffer register **302** are also connected as the information inputs to gates **660**. A conditioning input is applied to these gates by line **668**. A signal appears on line **668** when single-shot **115** is in its ON state. Output lines **664** from gates **660** are connected as the information inputs to argument register **340**. An associate signal applied to line **666** causes the contents of argument register **340** to be compared against the argument portion **341** of associative memory **338**. A signal is applied to line **666** when SS–**116** is in its ON state. The signal applied to line **668** when SS–**115** is in its ON state also applied to associative memory **338** to switch all the control circuits therein to their match state and to the ON-side input of match flip-flop **670**. After an association has been made in associative memory **338**, a test signal is applied to line **672**. If there has been no match during the associate operation, this signal passes through associative memory **638**, causing an output signal on line **674**, which signal switches flip-flop **670** to its OFF state. A test signal is applied to line **672** when SS–**117** is in its ON state. ON-side output line **676** from flip-flop **670** is connected as one input to AND gate **350**. OFF-side output line **678** from flip-flop **670** is connected as one input to AND gate **354**. The other input to AND gates **350** and **354** is line **680**. A signal appears on line **680** when SS–**117** switches from its ON to its OFF state. An output from AND gate **350** on line **348** indicates that the circuit has found the desired syllable pattern and that the end of the operating cycle has been reached. An output from AND gate **354** on line **352** causes SS–**0** to be switched to its ON state. As indicated previously, lines **344** connect the function portion **342** of associative memory **338** to output register **346** (FIG. 5L).

As was indicated previously, the information input to gates **630** (FIG. 5B) is lines **332**. The conditioning input to these gates is line **682**. A signal appears on line **682** when either SS–**1** or SS–**1***c* is in its ON state. Output lines **684** from gates **630** are connected as one input to OR gates **686**. The other inputs to OR gates **686** are ouput lines **638** from gates **690**. The information input to gates **690** are the output lines **692** from asterisk-code generator **694**. The conditioning input to gates **690** is line **696**. A signal appears on line **696** when either SS–**0***a* or **1***e* is in its ON state. Output lines **698** from OR gates **686** are applied as the principal input to coder **358**. The code conversion which is effected in coder **358** is shown in FIG. 8. It is noted in this figure that a different code is used for a finel "e" and a final "g" than is used for these letters when they are not the final letter of a word. Line **644** from end-of-word flip-flop **640** supplies the additional input to coder **358** to enable it to perform the code conversion properly when one of these two characters is involved.

Out lines **700** from coder **358** are connected as the information inputs to argument register **360**. Since the longest syllable to be matched on by the circuit contains six characters, argument register **360** need have a capacity of only seven characters. The reason for the seventh character is that there are some syllables which may appear only at the beginning or only at the end of a word. Therefore, an asterisk is inserted in argument register **360** before the first syllable is applied to the register and is

put into argument register **360** at the end of the word. If a possible first or last syllable of a word were to be six characters long, a seventh position would be needed for the asterisk. The position in argument register **360** at which an output from coder **358** is stored is determined by a signal on an output line **702** from input-control ring **704**. As with the previously described control rings, control ring **704** generates an output on only one of its output lines **702** at any given time. Control ring **704** is reset to its first position by a signal applied to line **706**. A signal is applied to reset-line **706** when SS–**0** or SS–**33***z* is in its ON state. Control ring **704** is incremented by a signal applied to line **708**. A signal is applied to line **708** when either SS–**0***b* or SS–**1***d* is in its ON state.

Argument register **360** is reset by a signal applied to line **710**. A signal is applied to line **710** when power is applied to the circuit, at the end of each cycle of operation, and when SS–**33***z* is in its ON state. A comparison is made between the contents of argument register **360** and the contents of associative memory **362** (it being remembered that acceptable syllable patterns and their relative frequency are stored in memory **362**), when a signal is applied to associate line **712**. A signal is applied to associate line **712** when SS–**2** is in its ON state.

A signal applied to line **714** causes all of the control circuits in associative memory **362** to be set to their match condition. The signal on line **714** is also applied to reset match flip-flop **716** to its OFF state. A signal is applied to line **714** when SS–**1** is in its ON state. The condition of the control circuits in associative memory **362** is tested by a signal applied to line **718**. If all of the control circuits in associative memory **362** are in their no-match state when a signal is applied to line **718**, a signal appears on line **720**, causing match flip-flop **716** to be switched to its ON state. A signal is applied to line **718** when SS–**3** is in its ON state.

Output line **722** from the OFF side of match flip-flop **716** is connected as one input to AND gate **724**. Output line **726** from the ON side of match flip-flop **716** is connected to one input of AND gates **650**, **728**, and **730** of gating circuit **402**. The final input to AND gates **650** and **724** and the second input to AND gates **728** and **730** is line **732**. A signal appears on line **732** when SS–**3** switches from its ON to its OFF state. The final input to AND gates **728** and **730** will be described later. Output line **734** from AND gate **724** is connected to switch SS–**20** to its ON state. Output lines **736** and **738** from AND gates **650** and **728** respectively are connected as the two inputs to OR gates **740**. Output line **406** from OR gate **740** is connected to switch SS-**4** to its ON state. Output line **404** from AND gate **730** is connected to switch SS–**0***b* to its ON state.

If the signal applied to test line **718** should find one of the match indicators in associative memory **362** switched to its match state, the associated syllable and frequency are read out through lines **370** to the syllable and frequency fields of memory-buffer register (MBR) **368** (FIG. 5F). In addition, a signal is also applied to line **742** to fully condition AND gate **657**. Output line **743** from AND gate **657** is connected as the conditioning input to gates **744** (FIG. 5D). The information input to gates **744** is derived through lines **746** from next-character-number counter **748**. During the reading of characters into argument register **360** (FIG. 5B), next-character-number counter **748** always contains the number of the character following the character which was just read into the argument register. Counter **748** is reset to a count of one by a signal applied to line **750**. A signal is applied to line **750** when either SS–**0** or SS–**3***w* is in its ON state. Counter **748** is incremented by a signal applied to line **752**. A signal is applied to line **752** when either SS–**1** or SS–**33***x* is in its ON state. Output lines **754** from gates **744** are connected as one input to OR gates **756** (FIG. 5F). The other input to these OR gates will be described later. Output lines **758** from OR gates **756** are connected as the input to the next-character-number field of MBR **368**.

Looking now at FIG. 5C, it is seen that the end-of-word (i.e. space) signal on line **306** from the external circuit is applied through delay **760** and line **762** to the reset-to-zero input of letter-per-word counter **312**. The signal on line **306** is also applied to one input of AND gates **764** and **766** of overflow-detection circuit **304**. A signal on line **768** causes *l/w* counter **312** to be set to a count of one. A signal is applied to line **768** when SS–**0** is in its ON state. Signals on lines **770** from gates **772** (FIG. 5F) cause the contents of the next-character-number field in MBR **368** to be stored in *l/w* counter **312** when a conditioning signal is applied to gates **772** through line **774**. A signal is applied to line **774** when SS–**33***t* is in its ON state. Letter-per-word counter **312** is decremented by a signal applied to line **776**. A signal is applied to line **776** when SS–**33***y* is in its ON state.

Output lines **778** from *l/w* counter **312** are connected as the input to decoder **780**. When the contents of *l/w* counter **312** are equal to seven, decoder **780** generates an output signal on line **782** which is applied as the final input to AND gate **728** (FIG. 5B), as one input to AND gate **784** (FIG. 5B) in gating circuit **376**, as one input to AND gate **786** (FIG. 5E), and as one input to AND gate **788** (FIG. 5B) in gating circuit **396**. When *l/w* counter **312** contains a count other than seven, decoder **780** generates an output signal on line **790**, which is connected as the final input to AND gate **730** (FIG. 5B), as one input to AND gate **792** in gating circuit **376**, and as the second input to AND gate **654** in gating circuit **306**. The final input to AND gates **646**, **654**, and **788** in gating circuit **396** is line **794**. A pulse appears on line **794** when SS–**31** switches from its ON to its OFF state. Output lines **796** and **798** from AND gates **646** and **788** respectively are connected as the inputs to OR gates **800**. Output line **398** from OR gate **800** is connected to switch SS–**32** to its ON state. Output line **400** from AND gate **654** is connected to switch SS–**0***b* to its ON state.

Output line **802** from the OFF side of used flip-flop **374** (FIG. 5B) is connected as a second input to AND gates **784** and **792** of gating circuit **376**. Used flip-flop **374** is switched to its OFF state by a signal applied to line **804**. A signal is applied to line **804** when power is turned on to the circuit, at the end of each cycle of operation, and when SS–**23** is in its ON state. Used flip-flop **374** is switched to its ON state by a signal applied to line **806**. A signal is applied to line **806** when SS–**33***t* is in its ON state. Output line **808** from the ON side of used flip-flop **374** is connected as one input to AND gate **810** in gating circuit **376** and as one input to AND gate **812** in gating circuit **410**. The final input to AND gates **784**, **792**, and **810** is line **814**. A pulse appears on line **814** when SS–**22** switches from its ON to its OFF state. Output line **384** from AND gate **784** is connected to switch SS–**32** to its ON state. Output line **382** from AND gate **792** is connected to switch SS–**0***b* to its ON state. Output line **385** from AND gate **810** is connected to switch SS–**23** to its ON state.

A second input to AND gate **812** (FIG. 5A) of gating circuit **410** is derived from OFF-side output line **816** of table-entered flip-flop **408**. Output line **818** from the ON side of this flip-flop is connected as one input to AND gate **820** of gating circuit **410**. Line **818** is also connected as one input to AND gate **822** (FIG. 5E). Table-entered flip-flop **408** is reset to its OFF state by a signal applied to line **824**. A signal is applied to line **824** when power is turned on to the circuit, at the end of each cycle of operation, and when SS–**33***t* is in its ON state. Flip-flop **408** is switched to its ON state by a signal applied to line **826**. A signal appears on line **826** when SS–**22** is in its ON state. The final input to AND gates **812** and **820** of gating circuit **410** is derived from line **828**. A pulse appears on line **828** when SS–**4** switches from its ON to its OFF state. Output line **412** from AND gate **820** is applied to switch

SS–31 to its ON state. Output line **414** from AND gate **812** is connected to switch SS–5 to its ON state.

In addition to output lines **782** and **790**, decoder **780** (FIG. 5C) has output lines **830**, **832**, and **833**. Output line **830** is connected as one input to AND gate **320** and through inverter **834** as one input to AND gate **322**. A signal appears on line **830** when a count of one, two, three or four appears in *l/w* counter **312**. Output line **832** is connected as one input to AND gate **314**. Output line **832** has a signal applied to it when a count of one or two is in *l/w* counter **312**. Output line **833** is connected as one input to AND gate **836** and through inverter **838** to one input of AND gate **840**. Line **833** has a signal on it when *l/w* counter **312** has a count of one in it. The other input to AND gates **836** and **840** is line **842**. Line **842** has a pulse applied to it when SS–33*y* is switched from its ON to its OFF state. Output line **844** from AND gate **836** is connected to switch SS–33*z* to its ON state. Output line **846** from AND gate **840** is connected to switch SS–33*x* to its ON state.

Overflow-detection circuit **304** (FIG. 5C) has three inputs. The first of these inputs is the previously described end-of-word line **306**. The second input to overflow-detection circuit **304** is bit line **310**, which, it will be remembered, has a signal applied to it each time a character is applied to input-buffer register **302** (FIG. 5A). A signal on line **310** is applied to increment character counter **848**. A signal on line **310** is also applied through delay **850** and line **852** to one input of AND gate **854**, and through OR gate **856** and line **858** to the increment input of *l/w* counter **312**. The other input to OR gate **856** is line **860**. A signal is applied to line **860** when SS–1 is in its ON state. The final input to overflow-detection circuit **304** is "spaces/line" lines **862**. Lines **862** are connected as the input to line-size register **864**. Before the start of the print operation, signals are applied to lines **862** indicating the number of spaces which are to be permitted in a line of text, and this information is stored in line-size register **864**.

Output lines **866** and **868** from character counter **848** and line-size register **864**, respectively, are connected to the two inputs of compare circuit **870**. Compare circuit **870** generates an output signal on line **872** when the two inputs to it are equal. Line **872** is connected as the other input to AND gate **854**. Output line **874** from AND gate **854** is connected to the ON-side input of an overflow flip-flop **876**. Overflow flip-flop **876** is reset to its OFF state by a signal applied to line **878**. A signal is applied to line **878** when power is turned on and at the end of each operating cycle. Output line **880** from the OFF side of overflow flip-flop **876** is connected as the second input to AND gate **766**. Output line **308** from AND gate **766** is connected to reset input-buffer register **302** (FIG. 5A) and input-control ring **608**. Output line **882** from the ON side of flip-flop **876** is connected as the second input to AND gates **314** and **764**. Output line **309** from AND gate **764** is connected to stop the flow of input data to input-buffer register **302** and is also connected as an input to AND gates **320**, **322**, and **886**. Output line **888** from AND gate **314** is connected to the ON-side input of flip-flop **890**. Line **892** is connected to the OFE-side input of this flip-flop. A signal appears on line **892** when power is applied to the circuit and at the end of each cycle of operation. Output line **894** from the OFF-side of flip-flop **890** is connected as the final input to AND gate **322** and output line **896** from the ON side of flip-flop **890** is connected as the final input to AND gate **886**. Output line **324** from AND gate **322** is connected to switches SS–**50** to its ON state. Output lines **898** and **900** from AND gates **320** and **886** respectively are connected as two inputs to OR gate **318**. As indicated previously, output line **316** from OR gate **318** is connected to external circuitry to indicate that the word in input-buffer register **302** should not be hyphenated.

Referring now to FIG. 5G, it is seen that, as previously indicated, a character from input-buffer register **302** is applied through lines **332** to decoder **632** in vowel-count circuit **328**. Decoder **632** generates an output signal on line **902** if the character applied to it is an "a," on output line **903** if the character applied to it is a "b," on output line **904** if the character applied to it is a "c," on output line **905** if the character is a "d," on output line **906** if the character is an "e," on output line **907** if the character is a "g," on output line **908** if the character is an "h," on output line **909** if the character is an "i," on output line **910** if the character is a "j," on output line **912** if the character is a "k," on output line **913** if the character is an "l," on output line **914** if the character is an "o," on output line **915** if the character is a "p," on output line **916** if the character is an "s," on output line **917** if the character is a "t," on output line **918** if the character is a "u," on output line **919** if the character is an "x," on output line **920** if the character is a "y," and on output line **921** if the character is a blank.

Output line **902** for the letter "a" is connected as one input to vowel-detector OR gate **924**, as one input to OR gate **926**, through inverter **928** and line **930**, as one input to AND gate **932**, and as one input to AND gate **934**. Output line **903** for the letter "b" is connected as one input to OR gate **936**. Line **904** for the letter "c" is connected as one input to OR gate **938**. Line **905** for the letter "d" is connected as one input to OR gate **940**, as one input to AND gate **942**, as one input to OR gate **944**, and as a second input to OR gate **936**. Line **906** for the letter "e" is connected as a second input to OR gate **924**, and as one input to AND gate **946**. Line **907** or the letter "g" is connected as a second input to OR gate **938** and as a third input to OR gate **936**. Line **908** for the letter "h" is connected as a third input to OR gate **938**. Line **909** for the letter "i" is connected as a third input to OR gate **924** and as one input to AND gate **948**. Line **910** for the letter "j" is connected as a fourth input to OR gate **936**. Line **912** for the letter "k" is connected as a fifth input to OR gate **936**. Line **913** for the letter "l" is connected as one input to AND gate **950** and through inverter **952** and line **954** as one input to AND gate **956**. Line **914** for the letter "o" is connected as a fourth input to OR gate **924** and as a second input to OR gate **926**. Line **915** for the letter "p" is connected as a sixth input to OR gate **936**. Line **916** for the letter "s" is connected as a second input to OR gate **940**, as one input to AND gate **958** and as a fourth input to OR gate **938**. Line **917** for the letter "t" is connected as the second input to OR gate **944** and as the seventh and final input to OR gate **936**. Line **918** for the letter "u" is connected as a fifth input to OR gate **924** and as the third input to OR gate **926**. Line **919** for the letter "x" is connected as the fifth and final input to OR gate **938**. Line **920** for the letter "y" is connected as a sixth input to OR gate **924**, and as an input to AND gate **960**. Blank line **921** is connected as the seventh and final input to OR gate **924**, as a third and final input to OR gate **940**, as an input to AND gates **962**, **964**, **965**, and **966**, and through inverter **968** and line **970**, as an input to AND gates **972**, **974**, **976**, and **978**.

Output line **980** from OR gate **924** is connected through inverter **982** and line **984** to one input of AND gates **986** and **988** (FIG. 5H). Output line **990** from OR gate **940** (FIG. 5G) is connected through inverter **992** and line **994** to one input of AND gates **996** and **998**. Output line **1000** from OR gate **938** is connected as one input to AND gate **1002**. Output line **1004** from OR gate **944** is connected as one input to AND gate **1006**. Output line **1008** from OR gate **936** is connected as one input to AND gate **1010**. Output line **1012** from OR gate **926** is connected as one input to AND gates **1014** and **1016**.

Referring to FIG. 5H, it is seen that the number of vowels in the word is counted by number-of-vowels counter **1018**. This counter is reset to a count of zero by a signal applied to line **1020**. A signal is applied to line

1020 when power is applied to the circuit and at the end of each operating cycle. Counter 1018 is incremented by a signal applied to output line 1022 from OR gate 1024. The inputs to OR gate 1024 will be described later. Counter 1018 generates an input signal on line 1026 when it contains a count of less than two, and an output signal on line 1028 when it contains a count of two. Line 1026 is connected as one input to AND gates 1030, 1032, 1034, 1036, and 1038, and as a second input to AND gates 1016, 965, 974, 966, 978, 942, 958, 964, 962, and 996. Equal-to-two line 1028 is connected as one input to AND gates 1040, 1042, 1044, 1046, 1048, 1050, 1052, 1054, and 1056.

The other input to AND gate 1014 (FIG. 5I) is line 1058. A signal appears on line 1058 when SS–52 is in its ON state. The other input to AND gates 946, 948, 960, 1016, 1056, 962, an 986 is line 1060. A pulse appears on line 1060 when SS–52 is switched from its ON to its OFF state. Output line 1062 from AND gate 986 (FIG. 5G) is connected to switch SS–51*a* to its ON state. Output line 1064 from AND gate 962 is connected as one input to not-hyphenable OR gate 1066. Output line 1068 from AND gate 1014 (FIG. 5I) is connected as an input to OR gate 1024. Output line 1070 from AND gate 1056 is connected as an input to OR gate 1072. Output line 1074 from AND gate 1054 is connected as the second input to OR gate 1072. Output lines 1076, 1078, and 1080 from AND gates 1052, 1050, and 1048 respectively are also connected as inputs to OR gate 1072. Output line 1082 from AND gate 1016 (FIG. 5I) is connected to switch SS–51 to its ON state. Output line 1084 from AND gate 960 is connected to switch SS–78 to its ON state. Output line 1086 from AND gate 948 is connected to switch SS–75 to its ON state and output line 1088 from AND gate 946 is connected to switch SS–53 to its ON state.

Line 1090 is connected as the second input to AND gate 998 (FIG. 5G). A signal is applied to line 1090 when SS–54 is in its ON state. Output line 1092 from AND gate 998 is connected as an input to OR gate 1024. Line 1094 is connected as an input to AND gates 1054 (FIG. 5I), 942 (FIG. 5G), 958, 964, and 996. A signal appears on line 1094 when SS–54 switches from its ON to its OFF state. Output line 1096 from AND gate 996 is connected to switch SS–51*a* to its ON state. Output line 1098 from AND gate 964 is connected to switch SS–71 to its ON state. Output line 1100 from AND gate 958 is connected to switch SS–55 to its ON state. Output line 1102 from AND gate 942 is connected to switch SS–63 to its ON state.

Line 1104 forms one input to AND gate 972 (FIG. 5G). A signal appears on line 1104 when SS–56 is in its ON state. Output line 1106 from AND gate 972 is connected as an input to OR gate 1024. Line 1108 is connected as an input to AND gates 1052 (FIG. 5I), 965 (FIG. 5G), and 974. A pulse appears on line 1108 when SS–56 switches from its ON to its OFF state. Output line 1110 from AND gate 965 is connected to switch SS–57 to its ON state. Output line 1112 from AND gate 974 is connected to switch SS–52 to its ON state.

Line 1114 is connected as an input to AND gate 1002 (FIG. 5G). A signal appears on line 1114 when SS–60 is in its ON state. Output line 1116 from AND gate 1002 is connected as an input to OR gate 1024. Line 1118 is connected as an input to AND gates 1034 (FIG. 5I) and 1050. A signal appears on line 1118 when SS–60 switches from its ON to its OFF state. Output line 1120 from AND gate 1034 is connected as an input to not-hyphenable OR gate 1066.

Line 1122 is connected as an input to AND gate 976 FIG. 5G). A signal is applied to line 1122 when SS–64 is in its ON state. Output line 1124 from AND gate 976 is connected as an input to OR gate 1024. Line 1126 is connected as an input to AND gates 1048, 966, and 978. A signal appears on line 1126 when SS–64 switches from its ON to its OFF state. Output line 1128 from AND gate 966 is connected to switch SS–65 to its ON state. Output line 1130 from AND gate 978 is connected to switch SS–52 to its ON state.

Line 1132 is connected as one input to AND gate 1006. A signal appears on line 1132 when SS–68 is in its ON state. Output line 1134 from AND gate 1006 is connected as an input to OR gate 1024. Line 1136 is connected as an input to AND gates 1046, and 1036. A pulse appears on line 1136 when SS–68 switches from its ON to its OFF state. Output line 1138 from AND gate 1046 is connected as the final input to OR gate 1072. Output line 1140 from OR gate 1072 is connected as an input to OR gate 1142. Output line 1144 from AND gate 1036 is connected as an input to not-hyphenable OR gate 1066.

Line 1146 is connected as an input to AND gates 950 (FIG. 5J) and 956. A pulse appears on line 1146 when SS–72 times out. Output line 1148 from AND gate 950 is connected to switch SS–73 to its ON state. Output line 1150 from AND gate 956 is connected as an input to not-hyphenable OR gate 1066.

Line 1152 is connected as an input to AND gate 1010 (FIG. 5J). A signal appears on line 1152 when SS–75 is in its ON state. Output line 1154 from AND gate 1010 is connected as an input to OR gate 1024. Line 1156 is connected as an input to AND gates 1032 and 1044. A signal appears on line 1156 when SS–74 switches from its ON to its OFF state. Output line 1158 from AND gate 1032 is connected as an input to OR gate 1066. Output line 1160 from AND gate 1044 is connected as an input to OR gate 1142.

Line 1162 is connected as an input to OR gate 1024 (FIG. 5H). A signal appears on line 1162 when SS–75 is in its ON state. Line 1164 is connected as an input to AND gates 1030 and 1042. A pulse appears on line 1164 when SS–75 switches from its ON to its OFF state. Output line 1166 from AND gate 1030 is connected to switch SS–76 to its ON state. Output line 1168 from AND gate 1042 is connected as an input to OR gate 1142.

Line 1170 is connected as an input to AND gates 932 and 934. A signal appears on line 1170 when SS–77 switches from its ON to its OFF state. Output line 1172 from AND gate 932 is connected to switch SS–51*a* to its ON state. Output line 1174 from AND gate 934 is connected as an input to OR gate 1142.

Line 1176 is connected as an input to AND gate 988 (FIG. 5H). A signal appears on line 1176 when SS–79 is in its ON state. Output line 1178 from AND gate 988 is connected as an input to OR gate 1024. Line 1180 is connected as an input to AND gates 1040 and 1038. A pulse is applied to line 1180 when SS–79 switches from its ON to its OFF state. Output line 1182 from AND gate 1038 is connected to switch SS–51 to its ON state. Output line 1184 from AND gate 1040 is connected as an input to OR gate 1142. Output line 336 from OR gate 1142 is connected to switch SS–115 to its ON state.

Referring now to next-character-address counter 420 (FIG. 5D), it is seen that an input is applied to this counter through lines 1190. The address of the first table storage position in core storage 364 is read into next-character-address counter 420 when power is turned on to the circuit and at the end of each cycle of operation. The count in the next-character-address counter is incremented by a signal applied to line 1192. A signal is applied to line 1192 when SS–33*b* or SS–33*r* is in its ON state. Output lines 1194 from counter 420 are connected as the information input to gates 1196 and as one input to compare-circuit 1198 of comparison-and-gating circuit 436. Line 1200 is connected as the conditioning input to gates 1196 and as one input to OR gate 1202. A signal is applied to line 1200 when SS–32, 33*j*, or 33*q* is in its ON state. Output lines 1204 from gates 1196 are connected as one input to OR gates 1206 (FIG. 5E).

A suitable address is also stored in starting-point register 1208 (FIG. 5D) and supplementary-starting-point register 1210 when power is turned on to the circuit and

at the end of each cycle of operation. The input to starting-point register **1208** is on lines **1212**. The address stored in this register is the address one position behind the address at which it is desired to begin storing a table of the type shown in FIG. 6 in core storage **364**. The input to supplementary-starting-point register **1210** is on lines **1214**. The signal on these lines contains the address at which it is desired to start storing a supplementary table of the type shown in FIG. 7 in core storage **364**.

Output lines **1216** from starting-point register **1208** are connected as the information input to gates **1218**. The conditioning input to these gates is on line **1220**. A signal is applied to line **1220** when SS–**0** or **85***a* is in its ON state. Output lines **1222** from gates **1218** are connected as one set of inputs to OR gates **1224**. The other inputs to these OR gates will be described later. Output lines **1226** from OR gates **1224** are connected as the information inputs to table-address counter (TAC) **386**. As indicated previously, TAC **386** is the register in which the next table address to be looked at is determined.

TAC **386** is incremented by a signal applied to line **1228**. A signal is applied to line **1228** when SS–**20**, **85**b, **87***b*, or **88** is in its ON state. Output lines **1230** from TAC **386** are connected as the other set of inputs to compare-circuit **1198** in compare-and-gating circuit **436**, and as the information inputs to gates **1232** and **1234**. Line **1236** is connected as the conditioning input to gates **1232** and as one input to OR gate **1202**. A signal is applied to line **1236** when either SS–**21** or SS–**86** is in its ON state. The conditioning input to gates **1234** is line **1238**. A signal is applied to line **1238** when either SS–**25** or SS–**28** is in its ON state. Output lines **1240** from gates **1232** are connected as one set of inputs to OR gates **1206**. is derived on line **1250**, which line is connected as one set of inputs to OR gates **1244**.

The two inputs to compare-circuit **1198** (FIG. 5D) in compare-and-gating circuit **436** have been previously indicated. If the inputs to this circuit are equal, compare-circuit **1198** generates an output signal on line **1246**, which line is connected as one input to AND gate **1248**. If the inputs to compare-circuit **1198** are not equal, an output is derived on line **1250**, which line is connected as one input to AND gate **1252**. The second input to AND gates **1248** and **1252** is line **1254**. A signal is applied to line **1254** when SS–**33***v* switches from its ON to its OFF state. Output line **438** from AND gate **1248** is connected to switch SS–**85** to its ON state. Output line **440** from AND gate **1252** is connected to switch SS–**33***b* to its ON state.

Referring back to supplementary-starting-point register **1210**, output lines **1256** from this register are connected as the information input to gates **1258**. The conditioning input to these gates is line **1260**. A signal is applied to line **1260** when either SS–**0** or SS–**33***e* is in its ON state. Output lines **1262** from gates **1258** are connected as one set of inputs to OR gates **1264**. The other set of inputs to these OR gates will be described later. Output lines **1266** from OR gates **1264** are connected as the information inputs to supplementary-address counter (SAC) **392**. As indicated previously, this is the register in which the address at which an entry is to be made in the supplementary table is determined. The count in supplementary-address counter **392** is incremented by a signal applied to line **1268**. A signal is applied to line **1268** when either SS–**30***a* or **33***p* is in its ON state.

Output lines **1270** from supplementary-address counter **392** are connected as the information inputs to gates **1272** and **1274** and as one input to compare-circuit **448**. The conditioning input to gates **1274** is line **1276**. A signal is applied to line **1276** when SS–**33***d* is in its ON state. Output lines **1278** from gates **1274** are connected as the information inputs to register **446**. Output lines **1280** from register **446** are connected as the information input to gates **1282** and as the second input to compare-circuit **448**. The conditioning input to gates **1282** is line **1284**. A signal is applied to line **1284** when SS–**33***n* is in its ON state. Output lines **1286** from gates **1282** are connected as the other set of inputs to OR gates **1264**.

Referring now to previously mentioned compare circuit **448** (FIG. 5D), when the two inputs to this compare circuit are equal, an output is derived on line **1288**, which line is connected as one input to AND gate **1290** in gating circuit **450**. When the inputs applied to compare-circuit **448** are not equal, an output is derived on line **1292** which line is one input to AND gate **1294**. The other input to AND gates **1290** and **1294** is line **1296**. A pulse appears to line **1296** when SS–**33***o* switches from its ON to its OFF state. Output line **458** from AND gate **1290** is connected to switch SS–**33***q* to its ON state. Output line **456** from AND gate **1294** is connected to switch SS–**33***p* to its ON state.

Referring back to gates **1272** (FIG. 5D), line **1298** is the conditioning input to these gates and is also one input to OR gate **1202**. A signal is applied to line **1298** when either SS–**30** or SS–**33***f* is in its ON state. Output lines **1300** from gates **1272** are connected as one set of inputs to OR gates **1206**.

Referring now to core storage **364** (FIG. 5F), it is seen that it is caused to perform a memory cycle (either read or write) when a signal is applied to line **1302**. A signal is applied to line **1302** when either SS–**6**, **22**, **24**, **26**, **31**, **33**, **33***g*, **33***k*, **33***m*, **33***s*, **33***v*, **87**, **102**, or **104** is in its ON state. Whether a memory cycle of core storage **364** will be a read cycle or a write cycle is determined by memory-status flip-flop **1304**. This flip-flop is switched to its ON or write state by a signal applied to line **1306**. A signal is applied to line **1306** when SS–**5**, **21**, **25**, **33***l*, **3***u*, or **103** is in its ON state. Flip-flop **1304** is switched to its OFF or read state by the signal applied to line **1308**. A signal is applied to line **1308** when SS–**23**, **32**, **33***f*, **33***r*, **86**, or **101** is in its ON state. Output line **1310** from the ON side of flip-flop **1304** is applied to core storage **364** to cause it to operate in its write mode. Output line **1312** from the OFF side of flip-flop **1304** is applied to core storage **364** to cause it to operate in its read mode.

As indicated previously, an entry is read into core storage **364** from MBR **368** or is read out of core storage into MBR. This register is cleared by a signal applied to line **1314**. A signal is applied to line **1314** when SS–**7** or **27** is in its ON state. A signal is also applied to this line when power is turned on to the circuit and at the end of each cycle of operation.

The first field in MBR **368** is the syllable field. The manner in which information is read into this field has been previously described. Output lines **1316** from this field are connected as the information input to gates **1318** (FIG. 5E) in gating circuit **484**. The next field in MBR **368** is the frequency field. The manner of storing information in this field has likewise been previously described. The information contained in this field is applied through output lines **1320** to the information input of gates **1322** in gating circuit **484**.

The next field in MBR **368** is the level-termination-mark field (LTM field). An input is applied to this field from gates **378** through lines **1324**. The information input to these gates is from coded-character-one generator **380** through lines **1326**. The conditioning input to gates **378** is from OR gate **1328** (FIG. 5E) through line **1330**. The inputs to OR gate **1328** are lines **1332**, **1334**, and **1336** from AND gates **786**, **651**, and **822** respectively. Previously mentioned lines **782**, **644**, and **818** form one input to AND gates **786**, **651**, and **822** respectively. The other input to AND gates **651** and **786** is line **1338**. A signal appears on line **1338** when SS–**20** is in its ON state. The other input to AND gate **822** is on line **1340**. A signal appears on line **1340** when SS–**4** is in its ON state.

As indicated previously, output line **520** from the LTM field of MBR **368** is connected as the input to zero-detector **512**. When there is a mark in this field, detector

512 generates an output signal on line 1342, which signal is applied as one input to AND gate 1344 of gating circuit 514 and as one input to AND gate 485 (FIG. 5E). When there is no mark in the LTM field, detector 512 generates an output signal on equal line 1345, which signal is applied as one input to AND gate 1346 and as one input to AND gate 1343 in gating circuit 486 (FIG. 5E). The other input to AND gates 1344 and 1346 is line 1347. A signal is applied to line 1347 when SS–103 times out. Output line 522 from AND gate 1344 is connected to switch SS–100 to its ON state, and output line 524 from AND gate 1346 is connected to switch SS–104 to its ON state.

The next field in memory-buffer register 368 is the next-character-number (NCN) field. The input to this field has been previously described. Output line 424 from this field is connected as one input to compare-circuit 442, as the information input to gates 1348, as the information inputs to previously described gates 772, and as the input to blank decoder 422. The conditioning input to gates 1348 is line 1350. A signal appears on line 1350 when either SS–25 or SS–33*c* is in its ON state. Output lines 1352 from gates 1348 are connected as the information input to next-character-number register 394. Output lines 1354 from next-character-number register 394 are connected as the second input to compare-circuit 442 and as the information input to gates 1356. The conditioning input to gates 1356 is line 1358. A signal appears on line 1358 when SS–29 is in its ON state. Output lines 1360 from gates 1356 are connected as the other set of inputs to previously described OR gates 756.

The two inputs to compare-circuit 442 (FIG. 5F) have been described in the previous paragraph. When these inputs are equal, a signal appears on equal-line 1362, which line forms one input to AND gate 1364 in gating circuit 444. When the two inputs applied to compare-circuit 442 are not equal, a signal appears on line 1366, which line forms one input to AND gate 1368 of gating circuit 444. The other input to AND gates 1364 and 1368 is line 1370. A pulse appears on line 1370 when SS–33*h* switches from its ON to its OFF state. Output line 452 from AND gate 1364 is connected to switch SS–33*i* to its ON state, and output line 454 from AND gate 1368 is connected to switch SS– 33*o* to its ON state.

Blank decoder 422 (FIG. 5F) determines whether the next-character-number field of MBR 368 contains a character or not. If there is no character in this field, an output signal appears on line 1372, which line forms one input to AND gate 1374 of gating circuit 426. If a character is in the NCN field, decoder 422 generates an output signal on line 1376, which line forms one input to AND gate 1378 of gating circuit 426. The other input to AND gates 1374 and 1378 is line 1380. A pulse appears on line 1380 when SS–33*a* switches from its ON to its OFF state. Output line 428 from AND gate 1374 is connected to switch SS–33*u* to its ON state, and output line 430 from AND gate 1378 is connected to switch SS–33*c* to its ON state.

The next field in MBR 368 is the chain-address field. Inputs are applied to this field through OR gates 1244 and lines 1382. One input to OR gates 1244 is from gates 434 through lines 1384. The information input to gates 434 is derived from character-two-code generator 432 through lines 1386. The conditioning input to gates 434 is line 1388. A signal is applied to line 1388 when SS–33*u* is in its ON state. A second input to OR gates 1244 is derived from gates 418 through lines 1390. The information input to gates 418 is from zero-character-code generator 416 through lines 1392. The conditioning input to gates 418 is line 1394. A signal appears on line 1394 when either SS–5 or SS–8 is on its ON state. A third input to OR gates 1244 is from gates 1396 through lines 1398. The conditioning input to gates 1396 is line 1400. A signal appears on line 1400 when SS–33*l* is in its ON state. The information input to gates 1396 will be described later. The final input to OR gates 1244 on lines 1242 has been previously described.

Output line 462 from the chain-address field of MBR 368 is connected as the input to two-detector 482 (FIG. 5E), as the input to zero-detector 480, and as the information input to gates 1402 and 1404. The conditioning input to gates 1402 is line 1406. A signal appears on lines 1406 when SS–33*i* is in its ON state. Output lines 1408 from gates 1402 are connected as the input to chain-address register 460. Output lines 1410 from chain-address register 460 are the information input to previously described gates 1396. Two-detector 482 generates an output signal on equal line 1412 when the chain-address field of MBR 368 contains a 2. Line 1412 is connected as one input to AND gate 1414 in gating circuit 486. Two detector 482 generates an output signal on line 1416 when a count other than 2 appears in the chain-address field of MBR. Line 1416 is connected as one input to AND gate 1418 in gating circuit 486 and as one input to AND gate 1420. Zero-detector 480 generates an output signal on equal line 1422 when a zero is in the chain-address field of MBR. Line 1422 is connected as one input to AND gate 485 and as a second input to AND gate 1343 in gating circuit 486. Zero-detector 480 generates an output signal on line 1426 when a count other than zero is in the chain-address field of MBR. Line 1426 is connected as one input to AND gate 1428 and as a second input to AND gate 1418. The final input to AND gates 485 and 1428 is line 1432. A signal appears on line 1432 when SS–88 is in its ON state. Output line 498 from AND gate 485 is connected to switch flip-flop 487 to its ON state. Line 1424 is connected to switch flip-flop 487 to its OFF state. A signal appears on line 1424 when SS–85*a* is in its ON state. Output line 1427 from the ON side of flip-flop 487 is connected as one input to AND gate 1429 in gating circuit 486 and to AND gate 1431. Output line 1433 from the OFF side of flip-flop 487 is connected as a third input to AND gate 1343 and as an input to AND gate 1435. The final input to AND gates 1414, 1418, 1343, and 1429 is line 1430. A pulse appears on line 1430 when SS–88 switches from its ON to its OFF state. The final input to AND gates 1431 and 1435 is line 1437. A signal appears on line 1437 when SS–104 times out. Output line 499 from AND gate 1343 is connected to switch SS–86 to its ON state. Output line 500 from AND gate 1418 is connected to switch SS–88*a* to its ON state. Output line 501 from AND gate 1429 is connected to switch SS–100 to its ON state. Output line 502 from AND gate 1414 is connected to switch SS–89 to its ON state. Output line 1439 from AND gate 1431 is connected to switch SS–85*a* to its ON state and output line 1441 from AND gate 1435 is connected to switch SS–105 to its ON state.

Output line 1434 from AND gate 1428 (FIG. 5E) is connected as the conditioning input to gate 1322, 1318, and 1436 in gating circuit 484, as the second input to AND gate 1420, and as the increment-by-one input to number-of-syllable counter 1670 (FIG. 5M). Output line 1440 from AND gate 1420 is connected as the conditioning input to gates 1404 and as another input to OR gate 1202. Output lines 1442 from gates 1404 are connected as another set of inputs to OR gates 1206.

In addition to the inputs previously mentioned, lines 1444 and 1446 are also inputs to OR gate 1202 (FIG. 5D). Line 1444 is also the conditioning input to gates 1448. A signal appears on line 1444 when SS–23 is in its ON state, and a signal appears on line 1446 when SS–101 is in its ON state. Output line 1450 from OR gate 1202 is connected as the conditioning input to gates 1452 (FIG. 5E). The information input to gates 1452 is output lines 1454 from MAR 372. Output lines 1456 from gates 1452 are connected through delay-circuit 1458 and lines 1460 as the input to previous-address register (PAR) 390. Output lines 1462 from PAR 390 are connected as the information input to gates 1448. Output lines 1464 from gates 1448 are connected as another input to OR gates

1206. The final input to OR gates 1206 is output line 1466 from gates 1468 (FIG. 5T). The inputs to this gate will be described later. Output lines 1470 from OR gates 1206 are connected through delays 1472 and lines 1474 as the input to MAR 372.

In addition to being connected as the information input to gates 1452, output lines 1454 from MAR 372 are connected as the address input to core storage 364, as the information input to gates 1436 of gating circuit 484, and as the information input to gates 1476 (FIG. 5D). The conditioning input to these gates will be mentioned later. Output lines 1478 from gates 1476 are the other set of inputs to OR gates 1224.

The last field in MBR 368 is the used-bit field. Output line 1480 from gates 1482 (FIG. 5F) are the input to this field. The information inputs to gates 1482 are output lines 1484 from character-one generator 1486. The conditioning input to gates 1482 is line 1488. A signal appears on line 1488 when SS–103 is in its ON state.

Output line 1490 from the used-bit field of MBR 368 is connected as the input to decoder 470. When a zero is in the used-bit field, decoder 470 generates an output signal on line 1492, which line forms one input to AND gate 1494 of gating circuit 472. When a "1" is in the used-bit field, decoder 470 generates an output signal on line 1496. Line 1496 is one input to AND gate 1498 and is also one input to AND gate 1500 (FIG. 5D). The other input to AND gate 1500 is line 1502. A signal appears on line 1502 when SS–87*a* is in its ON state. Output line 1504 from AND gate 1500 is the conditioning input to gates 1476. The other input to gates 1494 (FIG. 5F) and 1498 is line 1506. A pulse appears on line 1506 when SS–87*a* switches from its ON to its OFF state. Output line 474 from AND gate 1494 is connected to switch SS–88 to its ON state, and output line 476 from AND gate 1498 is connected to switch SS–87*b* to its ON state.

Output lines 492 from gates 1436 of gating circuit 484 (FIG. 5E) are connected as the information inputs to gates 1510A–1510N (FIG. 5R) of word registers 488. It will be understood that while only three positions of this register are shown in FIG. 5R, this register would in fact contain as many positions as might be required. Output line 494 from gates 1318 are connected as the information input to gates 1512A–1512N. Output lines 496 from gates 1322 (FIG. 5E) are connected as the information inputs to gates 1521–152N (FIG. 5M).

Referring now to FIG. 5M, it is seen that column-select counter 508 is reset to a count of one by a signal applied to line 1530. A signal is applied to line 1530 when power is turned on to the circuit, at the end of each cycle of operation, and when either SS–85*a* or SS–89 is in its ON state. Column-select counter 508 is incremented one position by a signal applied to line 1532 and is decremented one position by a signal applied to line 1534. A signal is applied to line 1532 when either SS–88*a* or 94 is in its ON state. A signal is applied to line 1534 when SS–100 is in its ON state. Output lines 1536 from column-select counter 508 are connected as one input to compare-circuit 1538 and as the input to decoder 1540. Decoder 1540 generates an output signal on line 1542 when a count of zero is in column-select counter 508 and an output on line 1544 when a count other than zero is in this counter. Lines 1542 and 1544 are connected as inputs to AND gates 1546 and 1548 respectively of gating circuit 510. The other input to AND gates 1546 and 1548 is on line 1549. A pulse appears on line 1549 when SS–100 switches from its ON to its OFF state. Output line 516 from AND gate 1546 is connected to switch SS–107 to its ON state and output line 518 from AND gate 1548 is connected to switch SS–101 to its ON state.

Decoder 1540 generates an output signal on line 1551 when a count of one is in column-select counter 508, an output on line 1552 when there is a count of 2 in this counter, and an output on line 155N when a count of N is in the counter. Lines 1551–155N are connected as the conditioning inputs to gates 1521–152N respectively and through cable 1550 as the conditioning inputs to (a) gates (1561A–1561N)-(156NA–156NN) (FIG. 5R) respectively; (b) gates (1571A–1571N)-(157NA–157NN) respectively; and (c) gates (1581A–1581N)-(158NA–158NN) (FIG. 5T) respectively.

Output lines 1591–159N from gates 1521–152N respectively are connected as the information input to frequency registers 1601–160N respectively of frequency register 490. Frequency registers 1601–160N are reset by a signal applied to line 1600. A signal is applied to line 1600 when power is turned on to the circuit and at the end of each cycle of operation. A signal is also applied to line 1600 when either SS–85 or 99 is in its ON state. Output lines 1611–161N from frequency registers 1601–160N respectively are connected as the information inputs to gates 1621–162N and 1631–163N respectively. The conditioning input to gate 1621–162N is line 1620. A signal appears on line 1620 when SS–89 is in its ON state. The conditioning input to gates 1631–163N is the previously described output lines 1551–155N from decoder 1540. Output lines 1641–164N from gates 1621–162N are connected as the inputs to adder 1640. Output lines 1651–165N from gates 1631–163N are connected as the inputs to OR gate 1650.

Output line 1660 from adder 1640 (FIG. 5M) is connected as the information input to gates 1662. The conditioning input to gates 1662 and gates 1664 is line 1666. A signal appears on line 1666 when SS–90 is in its ON state. The information input to gates 1664 is output lines 1668 from number-of-syllable counter 1670. Number-of-syllable counter 1670 is reset to a count of zero by a signal on line 1672. A signal is applied to line 1672 when power is turned on and at the end of each cycle of operation. A signal is also applied to line 1672 when SS–85*a* is in its ON state. Number-of-syllable counter 1670 is incremented by a signal applied to previously described line 1434.

In addition to being applied as an input to gates 1664, output lines 1668 from number-of-syllable counter 1670 are also applied as the second input to compare circuit 1538 and as the information input to gates 1678 and 1680. When the two inputs to compare circuit 1538 are equal, a signal appears on line 1682, which line is one input to AND gate 1684. When the two inputs to compare-circuit 1538 are not equal, a signal appears on line 1686, which line acts as one input to AND gate 1688. The other input to AND gates 1684 and 1688 is line 1690. A pulse appears on line 1690 when SS–93 switches from its ON to its OFF state. Output line 1692 from AND gate 1684 is connected to switch SS–95 to its ON state. Output line 1694 from AND gate 1688 is connected to switch SS–94 to its ON state.

Output lines 1696 and 1698 from gates 1662 (FIG. 5M) and 1664 respectively are connected as the inputs to divider 1700. Output lines 1702 from divider 1700 are connected as the information inputs to gates 1704 and gates 1706 (FIG. 5N). Output lines 1708 from OR gate 1650 is connected as the information input to gates 1710. The other input to gates 1704 and 1710 is line 1712. A signal appears on line 1712 when SS–91 is in its ON state. Output lines 1714 and 1716 from gates 1704 and 1710 respectively are connected as the inputs to subtractor 1718. Output line 1720 from subtractor 1718 is connected as the information input to gates 1722. The conditioning input to gates 1722 is line 1724. A signal is applied to line 1724 when SS–92 is in its ON state. Output lines 1726 from gates 1722 are connected as the input to squarer 1728 (FIG. 5N).

Output lines 1730 from squarer 1728 are connected as the information input to gates 1732. Output lines 1734 from intermediate-storage unit 1736 are connected as the information input to gates 1738. The conditioning input to gates 1732 and 1738 is in line 1740. A signal appears on line 1740 when SS–93 is in its ON state. Output lines

1742 and 1744 from gates 1732 and 1738 respectively are connected as the inputs to adder 1746. Output lines 1748 from adder 1746 are connected as the information inputs to gates 1750 and 1752. The conditioning input to gates 1750 is line 1754. A signal appears on line 1754 when SS–94 is in its ON state. Output lines 1756 from gates 1750 are connected as the information input to intermediate-store 1736. Intermediate-store 1736 is reset by a signal to line 1758. A signal is applied to line 1758 when power is applied to the circuit and at the end of each cycle of operation. A signal is also applied to line 1758 when SS–85*a* is in its ON state.

Line 1760 is connected as the conditioning input to gates 1680 and 1752. A signal is applied to line 1760 when SS–95 is in its ON state. Output lines 1762 and 1764 from gates 1680 and 1752 respectively are connected as the inputs to divider 1766. Output lines 1768 from divider 1766 are connected as the information input to gates 1770. The conditioning input to gates 1770 is line 1772. A signal is applied to line 1772 when SS–96 is in its ON state. Output lines 1774 from gates 1770 are connected as the inputs to square rooter 1776. Output lines 1778 from square rooter 1776 are connected as the inputs to gates 1780. The conditioning input to gates 1706 and 1780 is line 1782. A signal appears on line 1782 when SS–97 is in its ON state. Output lines 1784 and 1786 from gates 1706 and 1780 respectively are connected as the inputs to divider 1788. Output line 1790 from divider 1788 is connected as the information input to gates 1792. The conditioning inputs to gates 1792 is line 1794. A signal appears on line 1794 when SS–98 is in its ON state. Output lines 506 from gates 1792 are connected as the information inputs to gates 1796A–1796N (FIG. 5S) of word register 488.

Referring now to FIG. 5P, it is seen that word-select counter 528 is reset to a count of one by a signal applied to line 1798. A signal is applied to line 1798 when power is turned on to the circuit and at the end of each cycle of operation. A signal is also applied to line 1798 when SS–85 or SS–107 is in its ON state. Word-select counter 528 is incremented by a signal applied to line 1800. A signal is applied to line 1800 when SS–105 or SS–111 is in its ON state. There is an output line 1802 from word-select counter 528 for each word in word register 488. Lines 1802A–1802N are connected as the conditioning inputs to gates 1804A–1804N (FIG. 5Q) respectively (only 1804A and 1804B are shown in the figure) as the conditioning input to gates 1806A–1806N (FIG. 5R), as the conditioning input to gates 1510A–1510N, as the conditioning input to gates 1512A–1512N, as the conditioning input to gates 1796A–1796N (FIG. 5S), and as one input to AND gates (1581A–1581N)-(158NA–158NN) (FIG. 5T) respectively.

Output lines 1808A–18180N from gates 1510A–1510N are connected as the information inputs to gates 1571A–157NA, 1571B–157NB . . . 1571N–157NN. Output lines 1809A–1809N from gates 1512A–1512N respectively are connected as the information inputs to gates 1561A–156NA, 1561B–156NB . . . 1561N–156NN respectively. Output lines 1811A–1811N from gates 1561A–1561N are connected as the information input to first-syllable registers 1821A–1821N. Output lines 1812A–1812N from gates 1562A–1562N are connected as inputs to second-syllable registers 1822A–1822N. There are similar connections for the remaining syllable registers in each of the words of word registers 488 with output lines 181NA–181NN from gates 156NA–156NN being connected as the inputs to syllable N registers 182NA–182NN respectively. Similarly, output lines 1831A–1831N from gates 1571A–1571N are connected as the inputs to address registers 1841A–1841N. Output lines 1832A–1832N from gates 1572A–1572N are connected as the inputs to address registers 1842A–1842N respectively. Similar inputs are present for the remaining address registers in each of the words with output lines 183NA–183NN from gates 157NA–157NN being connected as the inputs to address registers 184NA–184NN.

Referring now to FIG. 5N, it is seen that the information input to gates 1678 is output lines 1668 from number-of-syllables counter 1670. The conditioning input to these gates is line 1850. A signal is applied to line 1850 when SS–98 is in its ON state. Output lines 1852 from these gates are connected as the information inputs to gates 1806A–1806N (FIG. 5R). Output lines 1854A–1854N from gates 1806A–1806N respectively are connected as the inputs to number-of-syllable registers 1856A–1856N. The final input to the word registers is output lines 1858A–1858N (FIG. 5S), from gates 1796A–1796N, which are connected as the inputs to ratio registers 1860A–1860N respectively. The reset input to the number of syllable, syllable, address, and ratio registers in word registers 488 is line 1861. A signal is applied to line 1861 when SS–85 is in its ON state.

Output lines 1862A–1862N from number-of-syllablye registers 1854A–1854N (FIG. 5R) are connected as the information inputs to gates 1864A–1864N and 1866A–1866N (FIG. 5P). It should be noted that only the gates for the A and B words are shown for this portion of the circuit. However, this circuitry is exemplary of that which would appear for the other words and is shown as an example of the type of circuitry which would be employed. Output lines 1868A–1868N from gates 1864A–1864N are connected as the inputs to OR gate 1870 in compare-and-gating circuit 548. Output lines 1872A–1872N from gates 1866A–1866N are connected as the inputs to OR gate 1874. Output lines 1876 and 1878 from OR gates 1870 and 1874 respectively are connected as the inputs to compare circuit 1880. If the inputs applied to compare-circuit 1880 by OR gate 1874 (this would be the ratio at the address indicated by preferred-word register 544 during step 112), (FIG. 4) is greater than the ratio applied by OR gate 1870, compare-circuit 1880 generates an output signal on line 1882, which line is connected as one input to AND gate 1884. If the number of syllables in the two words being compared as equal, compare-circuit 1880 generates on output signal on line 1886, which is applied as one input to AND gate 1888. If the number of syllables applied by OR gate 1870 is greater than the number applied by OR gate 1874, compare-circuit 1880 generates an output signal on line 1890 which is applied as one input to AND gate 1892. Line 1894 is the other input to AND gates 1884, 1888, and 1892. A signal appears on line 1894 when SS–109 times out. An output signal on line 550 from AND gates 1892 causes SS–110 to be switched to its ON state. An output signal on line 554 from AND gate 1888 causes SS–112 to be switched to its ON state. An output signal on line 552 from AND gate 1884 causes SS–111 to be switched to its ON state.

Cables 1896A–1896N from syllable registers (1812A–182NA)-(1821N–182NN) (FIGS. 5R–5S) respectively are connected as the information inputs to gates 1898A–1898N (FIG. 5Q) (only two of which are shown in the figure). Output lines 1900A–1900N are connected through OR gates 1902 and lines 1904 as the inputs to registers 1911–191N of coded-word-hyphenation registers 564 (FIG. 5K).

Output lines 1921A–192NA from address registers 1841A–184NA are connected as the information inputs to gates 1931A–193NA (FIG. 5T) respectively. Output lines 1921B–192NB from address registers 1841B–184NB are connected as the information inputs to gates 1931B–193NB respectively. The outputs from the address registers of the other words in word registers 488 are similarly connected with output lines 1921N–192NN from address registers 1841N–184NN being connected as the inputs to gates 1931N–193NN. Output lines 1941A–194NA from AND gates 1581A–158NA (FIG. 5T) are connected as the conditioning inputs to gates 1931A–193NA. Output lines 1941B–194NB from AND

gates 1581B–158NB are connected as the conditioning inputs to gates 1931B–193NB. There are similar connections for the remaining words in word register 488 with output lines 1941N–194NN from AND gates 1581N–158NN being connected as the conditioning inputs to gates 1931N–193NN. Output lines 1951A–195NA from gates 1931A–193NA are connected as the inputs to OR gate 1960A. Output line 1951B–195NB from gates 1931B–193NB are connected as the inputs to OR gate 1960B. There are similar connections for the remaining words in word register 488 with output lines 1951N–195NN from gates 1931N–193NN being connected as the inputs to OR gates 1960N. Output lines 1962A–1962N from OR gates 1960A–1960N are connected as the inputs to OR gate 1964. Output lines 1966 from OR gates 1964 are connected as the information inputs to previously mentioned gates 1468. Line 1968 is the conditioning input to gates 1468. A signal appears on lines 1968 when SS–101 is in its ON state. As indicated previously, output lines 1466 from gates 1468 is connected as an input to OR gate 1206 (FIG. 5E), which leads into MAR 372.

Output lines 1970A–1970N from ratio registers 1860A–1860N (FIG. 5S) of word registers 488 are connected as the information inputs to gates 1804A–1804N (FIG. 5Q), 1972A–1972N, and 1974A–1974N (only two of each of the gates are shown in the figures). Output lines 1976A–1976N from gates 1804A–1804N are connected as the inputs to OR gate 536 (FIG. 5P). Output lines 1978 from OR gates 536 are connected as the input to blank detector 530 and compare-to-one detector 532. When the ratio applied to blank detector 530 is not a blank, an output signal appears on line 1980, which line is connected as one input to AND gate 1982 of gating circuit 534. When the ratio field applied through OR gate 536 to blank detector 530 contains a blank, an output signal appears on line 1984, which line is connected as one input to AND gates 1986 and 1988. When the ratio in the field applied to compare-to-one circuit 532 is less than or equal to one, a signal appears on line 1990, which line is connected as a second input to AND gate 1982. When the ratio applied to compare-to-one detector 532 is greater than one, an output signal is derived on line 1992. Line 1992 is connected as one input to AND gates 1994 and 1996 in gating circuit 534. A second input to AND gate 1994 is output line 1998 from the OFF side of first-time-through flip-flop 535. Another input to AND gate 1996 is output line 2000 from the ON side of first-time-through flip-flop 535. Flip-flop 535 is reset to its OFF state by a signal applied to line 2002. A signal is applied to this line when power is applied to the circuit and at the end of each cycle of operation. A signal is also applied to this line when SS–110 is in its ON state. Flip-flop 535 is switched to its ON state by a signal applied to line 2004. A signal is applied to line 2004 when SS–107 is in its ON state. The final input to AND gates 1982, 1988, 1994, and 1996 is line 2006. A pulse is applied to line 2006 when SS–108 switches from its ON to its OFF state. Output line 538 from AND gate 1988 is connected to switch SS–120 to its ON state. Output line 540 from AND gate 1982 is connected to switch SS–111 to its ON state. Output line 546 from AND gate 1994 is connected to switch SS–109 to its ON state and output line 542 from AND gate 1996 is connected to switch SS–110 to its ON state.

As indicated previously one input to AND gate 1986 (FIG. 5P) is blank-output line 1984 from blank detector 530. The other input to this AND gate is line 2008. A signal is applied to line 2008 when SS–108 is in its ON state. Output line 2010 from AND gate 1986 is connected as one input to AND gates 2012A–2012N (only two of which are shown in FIG. 5P). The other input to AND gates 2012A–2012N are output lines 2014A–2014N respectively from preferred-word register 514. The input to preferred-word register 544 is output lines 2016 from gates 2108. The information inputs to gates 2018 are output lines 1802 from word-select counter 528. The conditioning input to gate 2018 is line 2020. A signal is applied to line 2020 when SS–110 is in its ON state. Preferred-word register 544 is reset to a count of one by a signal applied to line 2022. A signal is applied to line 2022 when power is applied to the circuit and at the end of each cycle of operation. A signal is also applied to line 2022 when SS–107 is in its ON state. Output lines 2024A–2024N from AND gates 2012A–2012N respectively are connected as the conditioning inputs to the beforementioned gates 1898A–1898N (only two of which are shown in FIG. 5Q).

In addition to the connections previously mentioned, output lines 1802A–1802N from word-select counter 528 (FIG. 5P) are connected as one input to AND gates 2026A–2026N and 2028A–2028N respectively (only two of which are shown in FIG. 5P). Similarly, output lines 2014A–2014N from preferred-word register 544 are connected as one input to AND gates 2030A–2030N and 2032A–2032N (only two of which are shown in FIG. 5P). The other input to AND gates 2026A–2026N and 2030A–2030N is line 2034. A signal is applied to line 2034 when SS–109 is in its ON state. The other input to AND gates 2028A–2028N and 2032A–2032N is line 2036. A signal appears on line 2036 when SS–112 is in its ON state. Output lines 2038A–2038N from AND gates 2026A–2026N are connected as the conditioning inputs to gates 1864A–1864N, and output lines 2040A–2040N from AND gates 2030A–2030N are connected as the conditioning inputs to gates 1866A–1866N (only two of each of these lines and gates being shown in FIG. 5P).

Output lines 2042A–2042N from AND gates 2028A–2028N are connected as the conditioning inputs to gates 1972A–1972N (FIG. 5Q) and output lines 2044A–2044N from AND gates 2032A–2032N are connected as the conditioning inputs to gates 1974A–1974N (only two of each of the lines and each of the gates being shown in FIG. 5Q). Output lines 2046A–2046N from gates 1972A–1972N are connected as the inputs to OR gate 2048 of gating-and-compare circuit 556. Output lines 2050A–2050N from gates 1974A–1974N are connected as the inputs to OR gate 2052. Output lines 2054 and 2056 from OR gates 2048 and 2052 respectively are connected as the inputs to compare-circuit 2058. If the ratio represented by the signal applied through OR gate 2048 to compare-circuit 2058 is greater than the ratio applied through OR gate 2052, compare-circuit 2058 generates an output signal on line 2060, which line forms one input to AND gate 2062. If the ratio represented by the signal applied to OR gate 2048 is equal to or less than that represented by the signal applied through OR gate 2052, compare circuit 2058 generates an output signal on line 2064 which is applied as one input to AND gate 2066. The other input to AND gates 2062 and 2066 is line 2068. A signal is applied to line 2068 when SS–112 switches from its ON to its OFF state. It will be remembered that gating-and-compare circuit 556 is involved in the performance of step 282 of FIG. 4. The significance of the inputs and outputs from this gate can be judged accordingly. Output line 558 from AND gate 2062 is connected to switch SS–110 to its ON state, and output line 560 from AND gate 2066 is connected to switch SS–111 to its ON state.

Referring now to FIG. 5K, it is seen that scan-control circuit 566 includes a group-control ring 2070 and a unit-control ring 2072. Signals on lines 2074 and 2076 reset group-control ring 2070 and unit-control ring 2072 respectively to their first position. A signal is applied to lines 2074 and 2076 when power is applied to the circuit, at the end of each cycle of the circuit, and when SS–120 is in its ON state. In addition, a signal is applied to line 2076 when either SS–126 or SS–128 is in its ON state. Group-control ring 2070 is incremented by a signal applied to line 2078, and unit-control ring 2072 is incremented by a signal applied to line 2080. A signal is applied to line 2078 when either SS–124 or SS–128 is in its ON state, and a signal is applied to line 2080 when SS–122 is in its ON state.

Output lines **2079A–2079N** from group-control ring **2070** are connected as the inputs to AND gates (**2081A–2086**)–(**2081N–2086N**) respectively. Output lines **2091–2096** from unit-control ring **2072** form the other input to AND gates (**2081A–2081N**)–(**2086A–2086N**) respectively. Line **2091** is also connected as one input to AND gate **2100** (FIG. 5L) and through inverter **2102** and line **2104** as one input to AND gates **2106** and **2108**. Output line **2096** from unit-control ring **2072** is also connected as one input to AND gate **2110** and through inverter **2112** and line **2114** as one input to AND gate **2116**. Output lines **2121A–2126A** from AND gates **2081A–2086A** are connected as the scan control inputs to coded-word-hyphenation register **1911**. Output lines **2121B–2126B** from AND gates **2081B–2086B** are connected as the scan-control inputs to coded-word-hyphenation register **1912**. There are similar inputs to the remaining coded-word-hyphenation registers with output lines **2121N–2126N** from AND gates **2081N–2086N** being connected as the scan-control inputs to coded-word-hyphenation register **191N**.

Output lines **2131–213N** from coded-word-hyphenation registers **1911–191N** respectively are connected as the inputs to OR gates **2140** (FIG. 5L) in control-and-gating-circuit **568**. Output lines **2142** from OR gates **2140** are connected as the inputs to blank decoder **2144**. When the position being scanned in the coded-word-hyphenation registers contains a blank, blank decoder **2144** generates an output signal on equals line **2146**, which line is connected as a second input to AND gates **2100**, **2106**, and **2108**. When the position in the coded-word-hyphenation registers being scanned contains some code character other than a blank, blank decoder **2144** generates an output signal on line **2148**, which line is connected as an input to AND gates **2110**, **2116**, and **2150**. The other input to AND gates **2100**, **2106**, **2110**, and **2116** is line **2152**. A pulse is applied to line **2152** when SS–**121** switches from its ON to its OFF state. An output signal on line **2154** from AND gate **2116** causes SS–**122** to be switched to its ON state. An output signal on line **2156** from AND gate **2110** causes SS–**124** to be switched to its ON state. An output signal on line **2158** from AND gate **2100** causes SS–**129** to be switched to its ON state, and an output signal on line **2160** from AND gate **2106** causes SS–**128** to be switched to its ON state.

The other input to AND gates **2108** and **2150** is line **2162**. A signal is applied to line **2162** when SS–**121** is in its ON state. Output line **578** from AND gate **2108** is connected as one input to OR gate **2164** in gating circuit **570**. The other input to this OR gate is line **580**. A signal appears on line **580** when SS–**125** is in its ON state. Output line **2166** from OR gate **2164** is connected as the conditioning input to gates **2168**. The information input to gates **2168** are output lines **2170** from hyphen-generator **572**. Output line **574** from AND gate **2150** is connected as the conditioning input to gates **634**. As indicated previously, the information input to gates **634** are the lines **332** which contain characters from input-buffer register **302** (FIG. 5A). Output lines **2170** and **2172** from gates **634** and **2168** respectively are connected as two sets of inputs to OR gates **2174**. The other set of inputs to these OR gates are output lines **2176** from gates **2178**. The information input to gates **2178** are output lines **2180** from blank-code generator **2182**. The conditioning input to gates **2178** is line **2184**. A signal appears on line **2184** when SS–**130** is in its ON state. Output line **576** from OR gate **2174** is connected as the information input to output register **346**. Output lines **2186** from input-control-ring **2188** determine where in output register **346** data applied to lines **576** will be stored. Only one of the lines **2186** is conditioned in any given time. Input-control-ring **2188** is reset to its initial position by a signal applied to line **2190**. A signal is applied to line **2190** when power is turned on, at the end of each operating cycle, and when SS–**120** is in its ON state. Input-control-ring **2188** is incremented by a signal applied to line **2192** and is decremented by a signal applied to line **2194**. A signal is applied to line **2192** when either SS–**122**, **124**, **126**, or **128** is in its ON state. A signal is applied to line **2194** when SS–**129** is in its ON state. Characters are read out of output register, a character at a time on line **2196** under control of the printing device (not shown).

DETAILED OPERATION

The description of the operation of the circuit shown in FIGS. 5A–5T will generally be with reference to a specific example in which the proper hyphenation pattern for the word "Mercantile" is determined. Prior to the beginning of the operation to be described, a signal is applied to the reset input of each flip-flop, register, and counter in the circuit so that all of the circuit elements are in their reset condition at the beginning of the operating cycle. As indicated previously, these reset inputs are generated when power is applied to the circuit and at the end of each operating cycle. Once all of the registers have been reset, the following registers are set as indicated:

Line-size register **864** (FIG. 5C) in overflow-detection circuit **304** is set by signals applied to lines **862** to indicate the total number of spaces which the line is to contain. Next-character-address counter **420** (FIG. 5D) is set by signals applied to lines **1190** to the first-table address in core storage **364**. Starting-point register **1208** is set by signals applied to lines **1212** to an address-one address position below that at which the table is to begin. Supplementary-starting-point register **1210** is set by signals applied to lines **1214** to the address at which the supplementary table in core storage **364** is to begin. Assuming that the table in core storage **364** is to begin at address **100** and the supplementary table at address **500**, the address **100** is stored in next-character address counter **420**, the address **099** in starting-point register **1208**, and the address **500** in supplementary-starting-point register **1210**.

Steps 202 and 204

As a word is applied to the printing device (not shown), each character of the word is applied through lines **300** to input-buffer register **302** (FIG. 5A). Since input-control ring **608** is initially reset to its first position, the first character applied to input-buffer register **302** is stored in the first position of this register. A signal is applied by the external circuitry to line **614** after this character is applied to the circuit incrementing the input-control ring so that the second character applied to the input-buffer register is stored in its second position. Succeeding characters are stored in succeeding positions of the input-buffer register under control of the input-control ring. As each character is applied to the input-buffer register, a signal is applied to bit-line **310** (FIG. 5C) causing *l/w* counter **312** and character counter **848** to be incremented. When the end of a word (i.e., a space) is detected, a signal is applied to line **306**. If, at this time, the count in character counter **848** is less than the count in line-size register **864**, then overflow flip-flop **876** is still in its reset condition, generating an output signal on line **880** to partially condition AND gate **766**. The signal on line **306** fully conditions AND gate **766**, causing an output signal on line **308**, which is applied through OR gate **611** (FIG. 5A) and line **605** to reset input-buffer register **302** and input-control ring **608**. After a short delay in delay **760**, the end-of-word signal on line **306** is also applied to reset *l/w* counter **312**. This means that each word in a line is stored in the input-buffer register, and the number of characters in this word counted, but that this information is not used unless the permissible number of characters in the line is reached before the end of the word is reached.

When the count is counter **848** (FIG. 5C) becomes equal to the count in line-size register **864** while a word is being loaded into input-buffer register **302**, compare circuit **870** generates an output signal on line **872**, which partially conditions AND gate **854**. The bit signal on line **310**, which caused character counter **848** to increment

to the required count, is delayed in delay **850** and applied through line **852** to the other input of AND gate **854**. AND gate **854** being fully conditioned at this time generates an output signal on line **874** which switches overflow flip-flop **876** to its ON state. Flip-flop **876** being in its ON state causes an output signal to appear on line **882**, which signal partially conditions AND gate **314**. If, at this time, decoder **780** generates an output signal on line **832** indicating that a count of one or two is in *l/w* counter **312**, this means that the overflow occurred on the first or second character of the word, and that the word is therefore not to be hyphenated. The combined occurrence of signals on lines **832** and **882** fully conditions AND gate **314** to generate an output signal on line **888** which signal switches flip-flop **890** to its ON state. The sequence of operations just described effectively performs step **202** of the flow diagram of FIG. 1 (i.e., test to determine if the overflow occurred on the first or second character).

The circuit then continues to load characters into input-buffer register **302** until a signal on line **306** indicates that the end of the word has been reached. Since overflow flip-flop **876** is now in its ON state, AND gate **764** is fully conditioned by the end-of-word signal on line **306**, causing an output signal on line **309** which is applied to stop the input-character-generating circuit. The signal on line **309** is also applied to partially condititon AND gates **320**, **322**, and **886**. If the overflow occurred on the first or second character, flip-flop **890** is generating an output signal on line **896** at this time to fully condition AND gate **886**, causing an output signal on line **900** which is applied through OR gate **318** to do-not-hyphenate line **316**. This completes step **202** (FIG. 1) of the operation. If, at this time, *l/w* counter **312** contains a count of 1, 2, 3, or 4, output line **830** from decoder **780** has a signal on it which fully conditions AND gate **320** to generate an output signal on line **898**. The signal on line **898** is applied through OR gate **318** to do-not-hyphenate line **316**. The above operation performs step **204** of the flow diagram of FIG. 1 by causing a do-not-hyphenate signal to be generated if the word at the end of the line contains four or less characters.

If the overflow did not occur on the first or second character, then flip-flop **890** is still in this reset condition, generating an output signal on line **894** which is applied to the second input to AND gate **322**, and if the *l/w* counter **312** contains more than four letters, there is no signal on line **830**, and inverter **834** applies the third input to AND gate **322**, fully conditioning this AND gate to generate an output signal on line **324**. The signal on line **324** switches SS–**50** of clock circuit **326** to its ON state. This causes the test-for-number-of-vowels operation (step **206** of FIG. 1) to be initiated.

Test for number of vowels

The procedure for counting the number of vowel groups in a word is shown in FIG. 2. It will be remembered from previous discussion that each vowel or group of two vowels is counted as a single vowel group with the following exceptions: (1) An "i" followed by an "a" is counted as two vowels; (2) a "y" is counted as a vowel if it follows a consonant; (3) a final "e" is not counted as a vowel unless it is preceded by a stop character (b, d, g, j, k, p, t) followed by an "l"; (4) an "e" before a final "d" is counted only if the letter preceding the "e" is a "t" or a "d"; (5) and an "e" before a final "s" is counted only if the preceding letter is s, c, g, h, or x.

The CL–**50** signal which occurs when SS–**50** is in its ON state is applied through line **616** to reset output-control ring **612** (FIG. 5A) to its first position. This conditions input-buffer register **302** to apply the character contained in its first character position, for the example chosen an "m," through the appropriate lines **622** and OR gates **330** to lines **332**. The signals on lines **332** are applied to decoder **632** (FIG. 5G) in vowel-count circuit **328**. As may be seen from column 77 of the appendix when SS–**50** switches to its OFF state, it causes SS–**52** to be switched to its ON state.

The resulting signal on line **1058** is applied as one conditioning input to AND gate **1014** (FIG. 5I) in vowel-count circuit **328**. If the character applied to decoder **632** at this time is an "a," an "o," or a "u," OR gate **926** applies a second input to AND gate **1014** through line **1012** at this time. It will be remembered that these three vowels are always counted as a single vowel group, regardless of what character or characters may follow them. Therefore, output line **1068** from AND gate **1014** is connected through OR gate **1024** and line **1022** to increment number-of-vowel counter **1018**.

When SS–**52** switches from its ON to its OFF state, a signal is applied through line **1060** to one input of AND gates **986**, **962**, **1056**, **1016**, **960**, **948** and **946**. Assume first that the character applied to decoder **632** is an "a," an "o," or a "u," as previously stated. Then, there is a signal on line **1012** applying a second input to AND gate **1016**. If a count of less than two is in number-of-vowel counter **1018**, a third input to this AND gate is derived from line **1026**. AND gate **1016** is therefore fully conditioned to generate an output signal on line **1082** which switches SS–**51** to its ON state. Referring to the appendix, it is seen that when SS–**51** is in its ON state, a signal is applied to line **620** (FIG. 5A) to increment output-control ring **612**. This causes the character following the vowel just looked at to be applied to decoder **632**. However, it will be remembered that according to the rules previously set out, even if this character is a vowel, it should not be counted. Therefore, there is no need to look at this character. To implement this, when SS–**51** switches from its ON to its OFF state, it causes SS–**51***a* to be switched to its ON state. SS–**51***a* being in its ON state causes another signal to be applied to line **620** to increment output control ring **612** one more position. When SS–**51***a* switches from its ON to its OFF state, it causes SS–**52** to be switched to its ON state. This causes the character, two characters advanced from the last character looked at, to be investigated to determine if it should be counted as a vowel group.

If the character applied to decoder **632** is an "a," "o," or "u," and the incrementing of counter **1018** during CL–**52** time steps this counter to a count of two, there is no need to look for further vowels, since it has already been determined that the word is hyphenable. The resulting signal on line **1028** therefore conditions AND gate **1056** to generate an output signal on line **1070** during 52τ time, which signal is applied through OR gate **1072**, line **1140** and OR gate **1142** to proceed-to-next step line **336**.

Assume that the character now being applied to decoder **632** is not a vowel, is not a "y," and is not a blank. There will therefore be no output from OR gate **924** (FIG. 5G) and inverter **982** will therefore generate an output signal on line **984**, which output is applied as one input to AND gate **986**, the other input to this AND gate being the 52τ signal on line **1060**. Output line **1062** from AND gate **986** is connected to switch SS–**51***a* to its ON state. It will be remembered that when SS–**51***a* is in its ON state, it causes a signal to be applied to line **620** to increment output-control ring **612** one position. This results in the next character being applied to decoder **632**. As before, when single shot **51A** switches to its OFF state, single-shot **52** is switched ON to cause a new character to be investigated.

Assume that the character now being applied to decoder **632** is a "y." It will be remembered that this character is counted as a vowel group if it is preceded by a consonant. The resulting signal out of decoder **632** on line **920** is applied as one input to AND gate **960**. The other input to AND gate **960** is the 52τ line **1060**. AND gate **960** being fully conditioned causes an output signal to appear on line **1084**, which signal switches single-shot **78** to its ON state. From the appendix, it is

seen that when single-shot **78** is in its ON state, a signal is applied to line **618** to decrement output-control ring **612** one position. This causes the character preceding that previously looked at, in this case the character preceding the "y," to be applied to decoder **632**. When single-shot **78** switches from its ON to its OFF state, it causes single-shot **79** to be switched to its ON state. When single-shot **79** is in its ON state, it causes a signal to be applied through line **1176** to one input of AND gate **988** (FIG. 5H). The other input to AND gate **988** is the aforementioned line **984** from inverter **982**, which line has a signal on it when the character applied to decoder **632** is not a vowel, a "y," or a blank, or in other words when the character applied to decoder **632** is a consonant. If AND gate **988** is fully conditioned, it generates an output signal on line **1178** which is applied through OR gate **1024** and line **1022** to increment number-of-vowel counter **1018**. When SS–**79** switches from its ON to its OFF state, a signal is applied through line **1180** to one input of AND gates **1038** and **1040**. If, at this time, number-of-vowel counter **1018** is generating an output signal on line **1026** indicating that less than two vowel groups have been found, then the search for vowels must be continued, and AND gate **1038** is fully conditioned to generate an output signal on line **1182**, causing single-shot **51** to be switched to its ON state. If two vowel groups have been found, then the resulting signal on line **1028** is applied to fully condition AND gate **1040**. The resulting output signal on line **1184** is applied through OR gate **1140** to line **336**. The signal on line **336** causes the circuit to proceed to the next step in the operation.

If the character applied to decoder **632** is an "i," a signal is applied through line **909** to one input of AND gate **948**, the other input to this AND gate being the 52_T line **1060**. AND gate **948** being fully conditioned generates an output signal on line **1086**, which switches single-shot **75** to its ON state. It will be remembered that the character "i" is counted as one vowel group and that an additional vowel group is counted if the "i" is followed by an "a." When SS–**75** is in its ON state, a signal is applied through line **1162**, OR gate **1024**, and line **1022** to increment number-of-vowel counter **1018**. When SS–**75** switches from its ON to its OFF state, a signal is applied to line **1164** to condition AND gates **1030** and **1042**. If two vowel groups have already been counted, it does not make any difference whether the "i" is followed by an "a" or not. Therefore, if a signal appears on line **1028** at this time indicating that a count of two is in number-of-vowel counter **1018**, AND gate **1042** is fully conditioned, generating an output signal on line **1168**, which is passed through OR gate **1142** to line **336** to cause the circuit to proceed to the next step. If, however, there is a count of less than two in vowel-group counter **1018**, the resulting signal on line **1026** is applied to fully condition AND gate **1030**, resulting in an output signal on line **1166** which switches single-shot **76** to its ON state. Single-shot **76** being in its ON state causes a signal to be applied through line **620** to increment output-control ring **612** one position. This causes the character following the "i" to be applied to decoder **632**. When single-shot **76** switches from its ON to its OFF state, single-shot **77** is switched to its ON state. No functions are performed during the period that single-shot **77** is in its ON state. However, when single-shot **77** switches from its ON to its OFF state, a signal is applied through line **1170** to one input of AND gates **932** and **934**. If, at this time, the character applied to decoder **632** is not an "a," there is no signal on line **902** from decoder **632**, and inverter **928** generates an output signal on line **930**, which fully conditions AND gate **932**. The resulting output signal on line **1172** causes single-shot **51***a* to be switched to its ON state, causing the search for additional vowel groups to be continued. If the character applied to decoder **632** at this time is an "a," a signal is applied through line **902** to fully condition AND gate **934**. Since an "i" followed by an "a" is counted as two vowel groups, the required number of vowel groups has been found and the circuit may proceed on to the next step. Therefore, the output from AND gate **934** on line **1174** is applied through OR gate **1142** to line **336**.

If the character applied to decoder **632** is an "e," the resulting output signal on line **906** from decoder **632** is applied as a conditioning input to AND gate **946**. The other input to this AND gate is the 52_T line **1060**. Output line **1088** from AND gate **646** causes single-shot **53** to be switched to its ON state. As may be seen from FIG. 2, the first step following the detection of an "e" being applied to decoder **632** is to look at the next character. Therefore, when SS–**53** is in its ON state, a signal is applied to line **620** to increment output-control ring **612** one position. When SS–**53** switches from its ON to its OFF state, a signal is generated which switches SS–**54** to its ON state. SS–**54** being in its ON state causes a signal to be applied through line **1090** to one input of AND gate **998** (FIG. 5G). The other input to AND gate **998** is output line **994** from inverter **992**, which inverter generates an output signal when the character applied to decoder **632** is not a "d," an "s," or a blank. Referring to FIG. 2, it is seen that if the character following the "e" is not one of the three characters indicated above, the "e" should be counted as a vowel group. Therefore, output line **1092** from AND gate **998** is connected through OR gate **1024** and line **1022** to increment number-of-vowel counter **1018**.

When SS–**54** switches from its ON to its OFF state, a signal is applied through line **1094** to one input of AND gates **1054**, **942**, **958**, **964** and **996**. If inverter **992** is generating an output signal on line **994** at this time, a second input is applied to AND gate **996**. A third unit to this AND gate is the not-equal-to-two output line **1026** from number-of-vowel counter **1018**. Output line **1096** from AND gate **996** is connected to switch SS–**51***a* to its ON state. This means that if the next character is not a "d," an "s," or a blank, so that the "e" is counted as a vowel group, but the vowel-group counter still contains a count less than two, the search is continued for additional vowel groups. If, at this time, the count in number-of-vowel group counter **1018** is equal to two, an output signal appears on line **1028** to fully condition AND gate **1054**. This results in an output signal on line **1074** which is applied throunh OR gate **1072**, line **1140** and OR gate **1142** to line **336** to cause the circuit to proceed to the next step in the operation.

If at this time the character applied to decoder **632** is a "d," a signal is applied through line **905** to a second input of AND gate **942**, the third input to this AND gate being the not-equal-to-two line **1026**. Output line **1102** from AND gate **942** is connected to switch SS–**63** to its ON state. It will be remembered that an "e" before a final "d" is counted only if the letter preceding the "e" is a "t" or a "d." It is therefore necessary to look at the character following the "d" to determine if it is a blank and therefore whether the "d" is the final character of the word. During step **63** a signal is applied to line **620** to increment output-control ring **612**. This causes the character following the "d" to be applied to decoder **632**. When SS–**63** times out (i.e. switches from its ON to its OFF state), SS–**64** is switched to its ON state. When SS–**64** is in its ON state, a signal is applied through line **1122** to one input of AND gate **976**. If the character then being applied to decoder **632** is not a blank there is no signal on line **921** at this time, and inverter **968** therefore generates a signal on line **970** which is applied as the other input to AND gate **976**. Output line **1124** from AND gate **976** is connected through OR gate **1024** and line **1022** to increment number-of-vowel counter **1018**. The "e" is in this way counted as a vowel if the "d" following it is not the final letter of a word.

When SS–**64** times out, a signal is applied through line

1126 to one input of AND gates 1048, 966 and 978. The other input to AND gate 1048 is equal to two line 1028 from number-of-vowel counter 1018. The output from AND gate 1048 is connected through OR gates 1072 and 1142 to line 336 to cause the circuit to proceed to the next step in the operation. If the character following the "d" is not a blank and the count in number-of-vowel counter 1018 is less than two, then all inputs to AND gate 978 are fully conditioned at this time, causing an output signal on line 1130 which switches SS–52 to its ON state. This causes a determination to be made as to whether the character following the "d" should be counted as a vowel group. Finally, if the character following the "d" is a blank, and there is a count of less than two in number-of-vowel counters 1018, all inputs to AND gate 966 are fully conditioned, causing an output signal on line 1128 which switches SS–65 to its ON state.

It is now desired to look at the character preceding the "e" to see whether it is in a "t" or a "d". Since the character presently being looked at is the character following the "d," it is necessary to look back three characters at this time. SS–65 being in its ON state causes a signal to be applied to line 618 to decrement output-control ring 612 one position. When SS–65 times out, it causes SS–66 to be switched to its ON state. SS–66 being in its ON state causes a second signal to be applied to line 618 to decrement output-control ring 612 the second position. When SS–66 times out, it causes SS–67 to be switched to its ON state. When SS–67 is in its ON state, a third signal is applied to line 618 to decrement output-control ring 612 to the desired position. When SS–67 times out, it switches SS–68 to its ON state. When SS–68 is in its ON state, a signal is applied through line 1132 to one input of AND gate 1006. The other input to AND gate 1006 is output line 1004 from OR gate 944. A signal appears on line 1004 if either a "d" or a "t" is applied to decoder 632 at this time. If AND gate 1006 is fully conditioned, a signal appears on line 1134, which signal is applied through OR gate 1024 and line 1022 to increment number-of-vowel counter 1018. It should be noted at this point that if the circuit has gone this far, it means that there is only a count of one in counter 1018 and, since the "d" was the final letter of the word, that there can be no further vowel groups in this word. Therefore, according to the criteria previously indicated, the word in question is not hyphenable. Therefore, when SS–68 times out, a signal is applied through line 1136 to AND gates 1036 and 1046. The other input to AND gate 1036 is not-equal-to-two line 1026. Output line 1144 from AND gate 1036 is connected through OR gate 1066 to not-hyphenable line 334. The other input to AND gate 1046 is equal-to-two line 1028. Output line 1138 from AND gate 1046 is connected through OR gates 1072 and 1142 to proceed-to-next-step line 336.

Returning now to the time at which SS–54 times out, if the character then applied to decoder 632 (i.e. the character following the "e") is an "s," all inputs to AND gate 958 (FIG. 5G) are present and an output signal is derived on line 1100 to switch SS–55 to its ON state. When SS–55 is in its ON state, a signal is applied to line 620 to increment output-control ring 612, causing the character following the "s" to be applied to decoder 632. It will be remembered that if this charatcer is a blank, if the "s" is a final "s," the "e" is counted only if it is preceded by an "s," "c," "g," "h," or "x." When SS–55 times out, it switches SS–56 to its ON state. When SS–56 is in its ON state, a signal is applied through line 1104 to AND gate 972. The other input to AND gate 972 is the not-blank line 970 from inverted 968. If the "s" is not a final "s," the "e" should be counted as a vowel group. Therefore, output line 1106 from AND gate 972 is connected through OR gate 1024 and line 1022 to increment vowel counter 1018.

When SS–56 times out, a signal is applied through line 1108 to one input of AND gates 1052, 965 and 974. If there is a count of two in number-of-vowel-group counter 1018 at this time, the resulting signal on line 1028 fully conditions AND gate 1052 to generate an output signal on line 1076, which is transmitted through OR gates 1072 and 1142 to proceed-to-next step line 336. If, at this time, there is a count of less than two in vowel-group counter 1018, line 1026 is applying a second input to AND gate 974. The third input to this AND gate is not blank line 970 from inverter 968, which line has a signal on it if the character following the "s" is not a blank (i.e. if the "s" is not the last letter of the word). When AND gate 974 is fully conditioned, it generates an output signal on line 1112 which causes SS–52 to be switched to its ON state. This results in a determination being made as to whether the character following the "s" should be counted as a vowel group. Finally, if the character following the "s" is a blank and there is a count of less than two in counter 1018, signals on lines 921 and 1026 fully condition AND gate 965 to generate an output signal on line 1110, which signal is applied to switch SS–57 to its ON state.

From FIG. 2 it is seen that if the "s" is the final letter of the word, it is necessary to look back three characters to see if the character preceding the "e" is an "s," "c," "g," "h," or "x." When SS–57 is in its ON state, a signal is applied to line 618 to decrement output-control ring 612 one position. When SS–57 times out, SS–58 is switched to its ON state. SS–58 being in its ON state causes a second signal to be applied to line 618 to decrement output-control ring 612 a second position. When SS–58 times out, it switches SS–59 to its ON state. SS–59 being in its ON state causes a third signal to be applied to line 618 to decrement output-control ring 612 to the desired position. When SS–59 times out, it causes SS–60 to be switched to its ON state. SS–60 being in its ON state causes a signal to be applied through line 1114 to one input of AND gate 1002. The other input to AND gate 1002 is output line 1000 from OR gate 938. The inputs to OR gate 938 are the "c," "g," "h," "s," and "x" output lines from decoder 632. If the character then applied to decoder 632 is one of these characters, AND gate 1002 generates an output signal on line 1116 which is applied through OR gate 1024 and line 1022 to increment number-of-vowel counter 1018. When SS–60 times out, a signal is applied through line 1118 to one input of AND gates 1034 and 1050. If counter 1018 has a count of two in it as this time, line 1028 applies a second input to AND gate 1050, causing an output signal on line 1078, which is applied through OR gates 1072 and 1142 to proceed-to-next-step output line 336. If, on the other hand, there is a count of less than two in counter 1018, the resulting signal on line 1026 fully conditions AND gate 1034. Since all characters in the word have been looked at at this time, if there is still a count of less than two in counter 1018, it means that the word is not hyphenable. Therefore, output line 1120 from AND gate 1034 is connected through OR gate 1066 to not-hyphenable-output line 334.

Returning now to the time when SS–54 times out, if the character following the "e" is a blank (i.e. if the "e" is the last letter of the word), a second conditioning signal is applied to AND gate 964 by line 921. If there is a count of less than two in counter 1018, a third conditioning input is applied to this AND gate by line 1026. AND gate 964 being fully conditioned causes an output signal on line 1098 which is applied to switch SS–71 to its ON state. Referring to FIG. 2, it is seen that when a blank follows the "e," the next step is to look back two characters to see if this character is an "l". When SS–71 is in its ON state, a signal is applied to line 618 to decrement output-control ring 612 one position. When SS–71 times out, SS–72 is switched to its ON state. When SS–72 is in its ON state, a second signal is applied to line 618 to cause output-control ring 612 to be decremented to the desired letter position. When SS–72 times out, a sig-

nal is applied through line **1146** to one input of AND gates **950** and **956**. From FIG. 2 it is seen that if the character two positions preceding the "e" is not an "l," the "e" is not counted as a vowel group and, since this is the last letter of the word and there is a count of less than two in vowel-group counter **1018** at this time, the word is not hyphenable. Therefore, output line **1150** from AND gate **956**, which gate is fully conditioned if the character being looked at is not an "l," is connected through OR gate **1066** to not-hyphenable line **334**. If there is a signal on line **913** at this time indicating that the character being applied to decoder **632** is an "l," AND gate **950** is fully conditioned to generate an output signal on line **1148** which causes SS–**73** to be switched to its ON state.

It is now desired to look at the character preceding the "l" to determine if it is a stop character. Therefore, when SS–**73** is in its ON state, a signal is applied to line **618** to decrement output-control ring **612** one position. When SS–**73** times out, it causes SS–**74** to be switched to its ON state. When SS–**74** is in its ON state, a signal is applied through line **1152** to one input of AND gate **1010** (FIG. 5J). The other input to AND gate **1010** is output line **1008** from OR gate **936**, the inputs to this OR gate being the lines **903**, **905**, **907**, **910**, **912**, **915** and **917** from decoder **632** for the stop characters "b," "d," "g," "j," "k," "p," and "t" respectively. Output line **1154** from AND gate **1010** is connected through OR gate **1024** and line **1022** to increment counter **1018** one position. When SS–**74** times out, a signal is applied through line **1156** to AND gates **1032** and **1044**, the other inputs to these AND gates being not-equal-to-two line **1026** and equal-to-two line **1028** respectively. For reasons previously indicated, output line **1158** from AND gate **1032** is connected through OR gate **1066** to not-hyphenable line **334**, and output line **1160** from AND gate **1044** is connected through OR gate **1142** to proceed to next-step-output line **336**.

Returning now to the time at which SS–**52** times out, there is one possible condition which has yet not been discussed. Line **1060**, to which a signal is applied at this time, is also connected as one input to AND gate **962** (FIG, 5G), the other inputs to this AND gate being blank line **921** from decoder **632** and not-equal-to-two line **1026** from counter **1018**. If AND gate **962** is fully conditioned, it means that the end of the word has been reached without having found two vowel groups. Therefore, according to criteria previously specified, the word is not hyphenable. Output line **1064** from AND gate **962** is therefore connected through OR gate **1066** to not-hyphenable line **334**.

STEP 210 TO DETERMINE IF WORD IS IN TABLE OF MOST FREQUENT ERRORS

Since the word "Mercantile," which is being investigated for this example, has three vowel groups according to the criteria previously stated, an output signal will ultimately be derived on line **336** to cause the circuit to switch SS–**115** to its ON state. When SS–**115** is in its ON state, a signal is applied to line **668** (FIG. 5A) to (a) condition gates **660** to apply the word stored in input-buffer register **302** to argument register **340**; (b) reset the match indicators in the control portion of associative memory **338** to the match side; and (c) set flip-flop **670** to its OFF state. As indicated previously, the argument portion **341** of associative memory **338** contains the words which would be improperly hyphenated if it were attempted to hyphenate them in accordance with the probability scheme of this invention, and function portion **342** of this memory contains the proper hyphenation patterns for these words.

When SS–**115** times out, a signal is applied to switch SS–**116** to its ON state. When SS–**116** is in its ON state, a signal is applied to line **666** to cause the word stored in argument register **340** to be associated with the words stored in argument portion **341** of associative memory **338**. As mismatches are detected during this associate operation, the match indicators for these entries are reset to their mismatch side. The match indicator for a matched entry is left on its match side. When SS–**116** times out, it applies a signal to switch SS–**117** to its ON state.

When SS–**117** is in its ON state, a test signal is applied to line **672**. If a match has been had during the associate operation, this signal causes the appropriate entry in function portion **342** to be read out through lines **344** into output register **346** (FIG. 5L). If no match was found during the associate operation, the signal applied to line **672** passes through the control circuit to line **674** to switch flip-flop **670** to its ON side. When SS–**117** times out, a signal is applied through line **680** to one input of AND gates **350** and **354**. If a match was found during the associate operation, flip-flop **670** is still in its OFF state, causing an output signal to appear on line **676**, which fully conditions AND gates **350** to generate an output signal on line **348**. An output signal on line **348** means that the desired hyphenation pattern has been found and that the circuit has reached the end of its operation. If no match was found during the associate operation, flip-flop **670** is switched to its ON state and an output signal appears on line **678**. AND gate **354** is therefore fully conditioned at this time to generate an output signal on line **352** which switches SS–**0** to its ON state. The switching of SS–**0** to its ON state causes the circuit to start performing step **212** (FIG. 1) of the operation.

FORM SYLLABLE PATTERNS (STEP 212, FIG. 1)

Since the word "Mercantile" is hyphenable in accordance with the teachings of this invention, it is not found in associative memory **338**, and the circuit proceeds to step **212** (FIG. 1) to form syllable patterns. When SS–**0** is in its ON state, a signal is applied to line **750** to reset next-character-number counter, **748** (FIG. 5D) to a count of one, a signal is applied to line **768** to reset *l/w* counter **312** (FIG. 5C) to a count of one, a signal is applied to line **616** to reset output-control ring **612** (FIG. 5A) to its first position, a signal is applied to line **706** to reset argument-input-control ring **704** (FIG. 5B) to its first position, a signal is applied to line **1220** to condition gates **1218** (FIG. 5D) to transfer the address stored in starting-point register **1208** into table-address counter (TAC) **386**, and a signal is applied to line **1260** to condition gates **1258** to transfer the address in supplementary-starting-point register **1210** into supplementary-address counter **392**.

When SS–**0** times out, a signal is applied to switch SS–**0***a* to its ON state. Since some syllables may only begin a word and other syllables may only end a word, it is necessary that there be some indication as to the beginning and the end of the word in argument register **360**. For this reason, an asterisk (*) is inserted in the argument register at the beginning of the word and at the end of the word. Therefore, when SS–**0***a* is in its ON state, a signal is applied to line **696** (FIG. 5B) to condition gates **690** to pass the code representing an asterisk (*) through OR gates **686** and coder **358** to the position in argument register **360** indicated by input-control ring **704**. Since input-control ring **704** was just set to its first position, this is the position in which the asterisk is stored. When SS–**0***a* times out, a signal is applied to switch SS–**0***b* to its ON state. Since a character has been stored in the first position of argument register **360**, it is now necessary to step input-control ring **704** so that the first character of the word applied to the argument register will be stored in the next position. This is accomplished by applying a signal to increment-line **708** when SS–**0***b* is in its ON state. When SS–**0***b* times out, a signal is applied to switch SS–**1** to its ON state.

Having performed the preliminary steps indicated above, the circuit is now ready to perform step **220** indicated in FIG. 3A. When SS–**1** is in its ON state, a signal is applied to line **682** (FIG. 5B) to gate the character in input-buffer register **302**, indicated by output control ring

612, in this case the first character of the word, through gates 630, lines 684, OR gates 686, line 698, coder 358, and lines 700 into the position in argument register 360 indicated by input-control ring 704. For the first letter of the word "Mercantile," it is seen from FIG. 8 that the coder would cause the character "N" to be stored in the second position in argument register 360. When SS–1 is in its ON state, the following operations are also performed: (a) a signal is applied to line 714 (FIG. 5B) to set all the match indicators in associative memory 362 to their match state and to reset match-flip-flop 716 to its OFF state. (b) A signal is applied to line 752 to step next-character-number counter 748 (FIG. 5D) plus 1. (c) A signal is applied to line 860 which signal is passed through OR gates 856 and line 858 to increment *l/w* counter 312 (FIG. 5C) by one. The last two operations stated above have the effect of causing next-character counter 748 to contain the number of the character following the last character applied to argument register 360 and of causing *l/w* counter 312 to indicate a number of characters one greater than that applied to argument register 360. The significance of these two facts will become apparent later. When SS–1 times out, it causes SS–1*a* to be switched to its ON state.

Referring to FIG. 8, it is seen that the coded character "V" is used for an "e," and the coded character "T" for a "g," except when these characters appear at the end of a word. When these characters appear at the end of a word, an "e" is recorded as an "E" and a "g" as a "G." Since, when a character is applied to coder 358, there is no way of determining whether it is a final character or not, the operations to be described now are required to be sure that the proper entry is made for an "e" or a "g" in a word. When SS–1*a* is in its ON state, a signal is applied to line 620 to increment output-control ring 612 (FIG. 5A) one position. The resulting output on a line 622 is applied through OR gate 330 and lines 332 to blank decoer 628 (FIG. 5B). If the character is a blank, indicating that the preceding character is the end of the word, blank decoder 628 generates an output signal on line 636 which is converted into a pulse by capacitor 638 to flip end-of-word flip-flop 640 to its ON state. When SS–1*a* times out, a signal is applied through line 658 to one input of AND gates 648 and 656. If the character just looked at is not a blank, as would be the case after the first letter of the word "Mercantile" is transferred into argument register 360, end-of-word flip-flop 640 is still in its OFF state at this time, causing a signal to be applied through line 652 to the other input of AND gate 656. The resulting output signal on line 661 is applied to switch SS–2 to its ON state. This will cause an associate operation to be performed in a manner to be described presently.

However, in order to complete the description of the loading phase of the operation at this point, assume that the operation has reached a point where the final "e" of the word "Mercantile" has been transferred into argument register 360. Since it has not yet been determined that this is the last letter of the word, this "e" is stored in argument register 360 as the coded character "V." Now, during CL–1*a* a blank character is applied to blank decoder 628, and end-of-word flip-flop 640 is switched to its ON state. Therefore, when SS–1*a* times out, flip-flop 640 is applying a signal to line 644 to fully condition AND gate 648. The resulting output signal on line 659 causes SS–1*b* to be switched to its ON state.

When SS–1*b* is in its ON state, a signal is applied to line 618 to decrement output-control ring 612 to the position of the last letter in the word, in this case the "e." When SS–1*b* times out, a signal is applied to switch SS–1*c* to its ON state. When SS–1*c* is in its ON state, a signal is again applied to line 682 to condition gates 630 to apply this last character through OR gate 686 and coder 358 to the appropriate position in argument register 360. However, at this time, end-of-word flip-flop 640 is still in its ON state, causing a signal to be applied through line 644 to coder 358. The coder is therefore advised of the fact that this is the final "e" of a word and stores the coded character "E" rather than the coded character "V" in argument register 360. If the final letter of the word is a character other than "e" or a "g," the signal on line 644 has no effect on the coder, and the same character is stored in the argument register as would otherwise be stored. When SS–1*c* times out, a signal is applied to switch SS–1*d* to its ON state. When SS–1*d* is in its ON state, a signal is applied to increment input-control ring 704 for argument register 360 to condition the position following that in which the last character for the word is stored. When SS–1*d* times out, a signal is applied to switch SS–1*e* to its ON state. When SS–1*e* is in its ON state, a signal is applied to line 696 to condition gates 690 to pass the code for the asterisk (*) from code generator 694 through lines 688, OR gates 686, lines 698, coder 358, and lines 700 to the position in argument register 360 indicated by input-control ring 704, in this case to the position following that in which the last letter of the word is stored. In this way, the required asterisk (*) is stored at the end of the word. When SS–1*e* times out, a signal is applied to switch SS–2 to its ON state.

When SS–2 is switched to its ON state, either by a signal on line 661 when the last character read into argument register 360 is not the last letter of the word, or by SS–1*e* timing out when the last character read into argument register 360 is the last character of a word, a signal is applied to line 712 (FIG. 5B) to cause an association to be made between the syllable stored in argument register 360 and the acceptable syllables stored in associative memory 362. If no acceptable syllable is found, then all of the match indicators in the control portion of associative memory 362 are reset. If a match is found, the match indicator for that entry is left set. When SS–2 times out, a signal is applied to switch SS–3 to its ON state.

When SS–3 is in its ON state, a signal is applied to line 718 to test the match indicators in the control portion of associative memory 362. If there has been a failure to achieve a match, as would be the case, for example, when only the asterisk (*) and the "N" for the word "Mercantile" are in argument register 360, the signal applied to line 718 passes through the control circuit to line 720 to set match flip-flop 716 to its ON state. When this occurs, a signal is applied to output line 726 which partially conditions AND gates 650, 728 and 730. When SS–3 times out, a signal is applied through line 732 to a second input of these AND gates. When only the asterisk (*) and the "N" are in argument register 360, *l/w* counter 312 has a count of only two therein, and decoder 780 is therefore generating an output signal on not-equal-to-seven line 790, which fully conditions AND gate 730 to generate an output signal on line 404, which switches SS–0*b* to its ON state. From previous discussion, it will be remembered (referring to the chart in the appendix) that when SS–0*b* is in its ON state, the operation of reading a new character into argument register 360 is commenced. It can be seen that during succeeding time intervals, the *l/w* counter and the next-character-number counter are each incremented, and an association is performed to see if the letters now in argument register 360 form a possible syllable.

From FIG. 6, it is seen that "NV" (i.e. the reduced character set for the first two letters of the word "Mercantile") is a possible syllable. Therefore, during clock-time two (CL–2) when an association is made on this possible syllable, a match is indicated, and a match circuit in associative memory 362 is left in its match state. During CL–3 when a test signal is applied to line 718, a match condition is now found, causing the syllable and its associated frequency to be read out of associate memory 362 through lines 370 into the syllable and frequency fields of memory-buffer register (MBR) 368 (FIG. 5F). The detection of a match also causes a signal to be applied

through line **742** to AND gate **657**. Since end-of-word-flip-flop **640** is in its OFF state at this time, there is a signal on line **652** which fully conditions AND gate **657** to generate a signal on line **743** which conditions gates **744** to apply the next-character number now stored in next-character-number counter **748** (FIG. 5D) through gates **744** and lines **754**, OR gates **756** (FIG. 5F), and lines **758** to the next-character-number field of MBR **368**. Since counter **748** is always incremented to contain the number of the character following the last character which was applied to argument register **360**, the number **3** is read at this time into the next-character-number field. When SS–**3** times out, the signal applied to line **732** now finds AND gate **724** (FIG. 5B) conditioned by a signal applied to output line **722** from the OFF side of match flip-flop **716**. The resulting output signal on line **734** is applied to switch SS–**20** to its ON state. As will be seen presently, SS–**20** being in its ON state causes the entry which was just made into MBR **368** to be stored in the table in core storage **364**. However, before describing this particular operation, the other possible conditions which may exist when SS–**3** times out will be investigated.

As was indicated previously, *l/w* counter **312** (FIG. 5C) always contains a count one greater than the number of characters in argument register **360**. Therefore, when a six-letter syllable has been stored in argument register **360**, it having previously been decided that syllables in excess of six letters will not be considered, *l/w* counter **312** has a count of seven in it. When there is a count of seven in *l/w* counter **312**, decoder **780** generates an output signal on equal-to-seven line **782**. This line is connected as one of the conditioning inputs to AND gate **728** (FIG. 5B), the other conditioning inputs to this AND gate being 3τ line **732** and no-match line **726** from flip-flop **716**. This means that when there is no match but a six-letter syllable is in argument register **360**, AND gate **728** generates an output signal on line **738** which is applied through OR gate **740** to line **406** to switch SS–**4** to its ON state. A similar situation exists when there is a no-match indication on line **726** from flip-flop **716** and an end-of-word indication on output line **644** from end-of-word-flip-flop **640**. Under these conditions, AND gate **650** is fully conditioned when a signal is applied to 3τ line **732**, causing an output signal on line **736** which is applied through OR gate **740** to line **406** to switch SS–**4** to its ON state. It will be remembered from previous discussion and by reference to the flow chart of FIG. 3A that under either of the conditions indicated above, a sequence of operations is commenced to find a new letter at which to start looking for possible syllables. The functions which are performed when SS–**4** is switched to its ON state will be described later.

As indicated previously, when an acceptable syllable pattern is found in associative memory **362**, the syllable, its relative frequency, and the appropriate next-character number from next-character-number counter **748** are stored in the proper fields in memory-buffer register **368** (FIG. 5F). AND gate **724** (FIG. 5B) is then caused to generate an output signal on line **734** which switches SS–**20** to its ON state. The switching of SS–**20** to its ON state causes step **222**, the create table-entry step, of FIG. 3A to commence. When SS–**20** is in its ON state, a signal is applied to line **1228** (FIG. 5D) to increment table-address counter **386** by one. Since the number originally stored in starting-point register **1208** and previously transferred to TAC **386** is one less than the first address position of the table in core storage **364**, the incrementing of TAC during CL–**20** brings the first address in the table into TAC. When CL–**20** is in its ON state, a signal is also applied through line **1338** (FIG. 5E) to one input of AND gates **651** and **786**. If end-of-word-flip-flop **640** is in its ON state, the resulting signal on line **644** is applied to fully condition AND gate **651**. If the count in *l/w* counter **312** is equal to seven, a signal is applied through line **782** to condition the other input to AND gate **786**. It can be seen that when either of these conditions occur, this will be the last entry in the table starting with the particular letter in question. Therefore, output lines **1334** and **1332** from AND gates **651** and **786** respectively are connected through OR gate **1328** to condition gates **378** to store a one from character generator **380** in the level-termination field of MBR **368**.

When SS–**20** times out, a signal is applied to switch SS–**21** to its ON state. When SS–**21** is in its ON state, a signal is applied to line **1236** (FIG. 5D) to condition gates **1232** to apply the address in TAC **386** to MAR **372**, and to condition gates **1452** to pass the address in MAR into previous address register (PAR) **390**. Delay **1472** prevents the new address from being read from TAC into MAR while the previous address in MAR is being read into PAR. When SS–**21** is in its ON state, a signal is also applied to line **1306** (FIG. 5F) to switch memory-status-flip-flop **1304** to its OFF or write state. When CL–**21** switches from its ON to its OFF state, a signal is applied to switch SS–**22** to its ON state. When SS–**22** is in its ON state, a signal is applied to line **1302** to cause a memory cycle of core storage **364**. Since memory-status-flip-flop **1304** is in its write state, this causes the syllable, frequency and next-character-number information stored in MBR **368** to be recorded in core storage **364** at the address indicated in MAR **372**. For the first syllable of the word, this storage is made at the address **100**. Also, at this time, a signal is applied to line **826** (FIG. 5A) to cause table-entered-flip-flop **408** to be switched to its ON state. This means that a table entry has been made starting with the first letter contained in argument register **360**.

As indicated previously, used-flip-flop **374** (FIG. 5B) is switched to its ON state when a complete set of entries has been made starting with a particular letter and the circuit is seeking a new letter to start looking for possible syllables at. Therefore, when only the first syllable has been entered into the table in core storage **364**, used flip-flop **374** is in its reset or OFF state. At this time, there is also a signal on line **790** indicating that the count in *l/w* counter **312** (FIG. 5C) is less than seven. Therefore, when a signal is applied to line **814** (FIG. 5B) when SS–**22** times out, it finds AND gate **792** of gating circuit **376** fully conditioned, causing an output signal on line **382** which switches SS–**0***b* to its ON state. From the appendix and from the flow diagram of FIG. 3A, it will be seen that the switching of SS–**0***b* to its ON state causes the circuit to apply an additional character to argument register **360** in an attempt to find additional possible syllables starting with the first letter of the word.

Referring to FIG. 6, a table entry is made at address **100** for the syllable "NV," and an entry is made at address **101** for the syllable "NVL." Attempts to find additional possible syllables starting with the first letter of the word are unsuccessful. When six characters have been entered into argument register **360** and a mismatch indication occurs; line **782** from decoder **780** is supplying a conditioning input to AND gate **728** (FIG. 5B), and match flip-flop **716** is applying a second conditioning input to this AND gate through line **726**. Therefore, when SS–**3** times out applying a signal to line **732**, AND gate **728** generates an output signal on line **738** which is applied through OR gate **740** to switch SS–**4** to its ON state.

When SS–**4** is in its ON state, a signal is applied through line **1340** (FIG. 5E) to one input of AND gate **822**. Since table-entered-flip-flop **408** is in its ON state (this flip-flop was switched to its ON state when the table entry was made for the possible syllable "NV"), a signal appears on on-side-output line **818** to fully condition AND gate **822**, causing an output signal on line **1336** which is applied through OR gate **1328** to condition gates **378** to apply a numeral one from generator **386** to the level-termination-mark field of MBR **368**. When SS–**4** times out, a signal is applied through line **828** (FIG. 5A) to one input of AND gates **812** and **820** in gating-circuit **410**. Since the table-entered-flip-flop is in its ON state, a signal is applied

through line **818** to fully condition AND gate **820**, causing an output signal on line **412** which switches SS–**31** to its ON state. When SS–**31** is in its ON state, a signal is applied to line **1302** to cause a memory cycle to be performed. It will be remembered that memory-status-flip-flop **1304** was switched to its write state when the table entry was made. There has been no signal since then to switch it to its read state. Therefore, the memory cycle during CL–**31** time is a write cycle causing the level-termination mark which was read into the level-termination field of MBR **368** during CL–**4** time to be stored in the appropriate field position of the last table entry, in this case the table entry for the possible syllable "NVL" at address **101** (see FIG. 6). When SS–**31** times out, a signal is applied through line **794** (FIG. 5B) to the AND gates in gating-circuit **396**. Since a count of seven is in *l/w* counter **312** at this time, decoder **780** generates an output signal on line **782** which fully conditions AND gate **788** to generate an output signal on line **798** which is applied through OR gate **800** to line **398** to switch SS–**32** to its ON state.

From FIG. 3A, it is seen that it is now desired to find the next letter with which possible syllables may begin. As a first step in this operation, when SS–**32** is in its ON state, a signal is applied to line **1308** (FIG. 5F) to set memory-status flip-flop **1304** to its OFF or read state. A signal is also applied to line **1200** (FIG. 5D) to condition gates **1196** to pass the address in next-character-address counter **420** through OR gates **1206** and delay **1472** to MAR **372**. It will be remembered that the address of the first position in the table of FIG. 6, in this case the address **100**, was initially stored in address counter **420**. Therefore, this is the address which is now applied to MAR. It is noted that the signal on line **1200** also causes the address which was in MAR to be read into PAR **390**. When SS–**32** times out, a signal is applied to switch SS–**33** to its ON state. When SS–**33** is in its ON state, a signal is applied to line **1302** to cause a memory cycle, in this case to cause the entry at the first position in the table of FIG. 6 to be read into MBR **368**. When SS–**33** times out, it causes SS–**33***a* to be switched to its ON state. Nothing occurs, during the time that SS–**33***a* is in its ON state. However, when **33***a* times out, a signal is applied through line **1380** (FIG. 5F) to one input of AND gates **1374** and **1378**. It will be remember that if the next-character-number field of an entry is blank, this means that this entry contains a possible last syllable for the word. At the present stage of the operation, however, entry **100** is the one which is in MBR **368**, and the next-character-number field of this entry contains the character 3. Therefore, blank decoder **422** finds the contents of the next-character-number field not equal to a blank and generates an output signal on line **1376** which fully conditions AND gate **1378** to generate an output signal on line **430**, which is applied to switch SS–**33***c* to its ON state.

When SS–**33***c* is in its ON state, a signal is applied to line **1350** (FIG. 5F) to condition gates **1348** to apply the character in next-character-number field of MBR **368** to next-character-number register **394**. When SS–**33***c* times out, a signal is applied to switch SS–**33***d* to its ON state. When SS–**33***d* is in its ON state, a signal is applied to line **1276** (FIG. 5D) to condition gates **1274** to apply the contents of supplementary-address counter **392** to register **446**. When SS–**33***d* times out, a signal is applied to switch SS–**33***e* to its ON state. When SS–**33***e* is in its ON state, a signal is applied to line **1260** to condition gates **1258** to pass the contents of supplementary-starting-point register **1210** into supplementary-address counter **392**. When SS–**33***e* times out, a signal is applied to switch SS–**33***f* to its ON state. When SS–**33***f* is in its ON state, a signal is applied to line **1298** to condition gates **1272** to apply the address in supplementary-address counter **92** to MAR **372** and to condition gates **1452** to apply the address which was in MAR to PAR **390**. A signal is also applied to line **1308** (FIG. 5F) at this time to switch memory-status-flip-flop **1304** to its OFF or read state. When SS–**33***f* times out, a signal is applied to switch SS–**33***g* to its ON state. When SS–**33***g* is in its ON state, a signal is applied to line **1302** to cause a memory cycle to be performed, in this case to cause the contents of the first address in the supplementary table to be read out into MBR **368**. When SS–**33***g* times out, a signal is applied to switch SS–**33***h* to its ON state. Nothing occurs during CL–**33***h* time. However, when SS–**33***h* times out, a signal is applied through line **1370** (FIG. 5F) to condition AND gates **1364** and **1368**. The other inputs to these AND gates are the outputs from compare-circuit **442** which circuit is comparing the next-character-number which was stored in next-character-number register **394** with the next-character number in the next-character-number field of memory buffer register **368**, this number being the next-character number in the first entry of the supplementary table. The purpose of this operation is to determine whether a search has previously been made starting with the character represented by this next-character number. It will be remembered that if such a search has been made, then there is no need to repeat it, and the proper chain address may be read into the table entry which gave rise to this next-character number. However, in this case, there has been no entry into the supplementary table so that compare-circuit **442** generates an output signal on not-equal line **1366**, fully conditioning AND gate **1368** to generate an output signal on line **454**. A signal on line **454** is applied to switch SS–**33***o* to its ON state.

Nothing occurs in the circuit during CL–**33***o* time. However, when SS–**33***o* times out, a signal is applied through line **1296** (FIG. 5D) to one input of AND gates **1290** and **1294**. The other inputs to AND gates **1290** and **1294** are the outputs from compare-circuit **448**, which circuit is comparing the address which was in SAC (i.e. the address at which the last entry was made in the supplementary table) with the address presently in SAC. Since, at the present stage of the operation, both of these registers contain the address **500**, compare-circuit **448** generates an output signal on equal line **1288**, fully conditioning AND gate **1290** to generate an output signal on line **458**, which signal is applied to switch SS–**33***q* to its ON state.

The circuit is now ready to determine the starting point for the next syllable string, to perform step **242** of FIG. 3A. When SS–**33***q* is in its ON state, a signal is again applied to line **1200** (FIG. 5D) to condition gates **1196** to apply the address in next-character-address counter **420** to MAR **372**. The address **100** is therefore stored in MAR **372**. When SS–**33***q* times out, a signal is applied to switch SS–**33***r* to its ON state. When SS–**33***r* is in its ON state, a signal is applied to line **1192** to increment next-character-address counter **420** one position. During CL–**33***r* time a signal is also applied to line **1308** (FIG. 5F) to switch memory-status flip-flop **1304** to its read state. When SS–**33***r* times out, a signal is applied to switch SS–**33***s* to its ON state. When SS–**33***s* is in its ON state, a signal is applied to line **1302** to cause a memory cycle during which the entry at address **100** is read into MBR **368**. When SS–**33***s* times out, a signal is applied to switch SS–**33***t* to its ON state. When SS–**33***t* is in its ON state, a signal is applied to line **824** (FIG. 5A) to reset table entered flip-flop **408** to its OFF state, a signal is applied to line **806** (FIG. 5B) to set used flip-flop **374** to its ON state, a signal is applied to line **642** to set used flip-flop state, a signal is applied to line **642** to reset end-of-word flip-flop **640** to its OFF state (in this example it still is in its OFF state and this acts is a mere perfunctory one) and a signal is applied to line **774** (FIG. 5F) to condition gates **772** to apply the contents of the next-character-number field in MBR **368** through lines **770** to *l/w* counter **312** (FIG. 5C). It should be noted that in this instance the *l/w* counter is merely being used as a register and not as a *l/w* counter. When SS–**33***t* times

out, a signal is applied to switch SS–33*w* to its ON state. When SS–33*w* is in its ON state, a signal is applied to line **616** to reset output-control ring **612** to its first position, and a signal is applied to line **750** (FIG. 5D) to reset next-character-number counter **748** to a count of one. These are preliminary steps to the setting of the output-control ring to the position indicated by the next-character number which is now stored in *l/w* counter **312**.

When SS–33*w* times out, a signal is applied to switch SS–33*x* to its ON state. When SS–33*x* is in its ON state, a signal is applied to line **752** (FIG. 5D) to increment next-character-number counter **748**, and a signal is applied to line **620** (FIG. 5A) to increment output-control ring **612** one position. When SS–33*x* times out, a signal is applied to switch SS–33*x* to its ON state. When SS–33*y* is in its ON state, a signal is applied to line **776** to decrement *l/w* counter **312** one position. When SS–33*y* times out, a signal is applied through line **842** (FIG. 5C) to AND gates **836** and **840**. Since the count originally in *l/w* counter **312** was 3, decrementing it a single time drops this count to only 2. Therefore, there is no signal on line **833** from decoder **780** at this time, and inverter **838** applies a signal to AND gate **840** to fully condition this AND gate, causing an output signal on line **846** which switches SS–33*x* to its ON state.

The incrementing of the output-control ring and the next-character-number counter, and the decrementing of the *l/w* counter continue until the *l/w* counter has been decremented to a count of one. When this occurs, the output-control ring and the next-character-number counter will have been incremented to the desired position. At this time, when SS–33*y* times out, there is a signal on line **833** fully conditioning AND gate **836** to generate an output signal on line **844** which switches SS–33*z* to its ON state. When SS–33*z* is in its ON state, a signal is applied to line **710** (FIG. 5*B*) to reset argument register **360**, and a signal is applied to line **706** to reset argument-input-control ring **704** to its first position. When SS–33*z* times out, a signal is applied to switch SS–1 to its ON state. The circuit is now ready to look for possible syllables starting with the third character of the word, in this case the reduced character set letter "L."

As may be seen from FIG. 6, there are no possible syllables beginning with this letter. Therefore, successive letters are applied to argument register **360** and successive attempts made to match until six characters have been read into argument register **360**. When SS–2 times out at this time, the signal applied to line **732** again finds AND gate **728** (FIG. 5B) fully conditioned causing an output signal on line **406** which switches SS–4 to its ON state. Since no table entry was made starting with this letter, table-entered flip-flop **408** is still in its reset condition, and AND gate **822** 5E) is therefore not conditioned. No operations are therefore performed during CL–4 time. When SS–4 times out, the signal applied to line **828** (FIG. 5A) now finds AND gate **812** fully conditioned, causing an output signal on line **414** which switches SS–5 to its ON state. It should be noted that output line **808** from the ON side of used flip-flop **374** is also a conditioning input to AND gate **812**. However, if the table-entered flip-flop is in its OFF state, the used flip-flop should always be on its ON state. The reason for this is that the only situation when the used flip-flop is in its OFF state during CL–4 time is when possible first syllables of the word are being looked for, and, if the word is to be hyphenable at all, it must have at least one possible first syllable, and the tabel-entered flip-flop will be in its ON state.

At his point, the circuit is performing step **232** of FIG. 3A. When SS–5 is in its ON state, a signal is applied to line **1394** (FIG. 5E) to condition gates **418** to route the zero signal from generator **416** into the chain-address field of MBR **368** A signal is also applied to line **1306** (FIG. 5F) to switch memory-status-flip-flop **1304** to its write mode. When SS–5 times out, a signal is applied to switch SS–6 to its ON state. When SS–6 is in its ON state, a signal is applied to line **1302** to cause a memory cycle to be performed. In this instance during the memory cycle, the "O" stored in the chain-address field of MBR is written into the entry whose address is in MAR **372** at that time. In this case, this is the entry **100**. When SS–6 times out, a signal is applied to switch SS–7 to its ON state.

At his time, the operation of making the appropriate supplementaryt-able entry is commenced. When SS–7 is in its ON state, a signal is applied to line **1314** (FIG. 5F) to reset MBR **368**. When SS–7 times out, a signal is applied to switch SS–8 to its ON state. When SS–8 is in its ON state, a signal is again applied to line **1394** to condition gates **418** to apply to the "0" code to the chain address field of MBR **368**. When SS–8 times out, a signal is applied to switch SS–29 to its ON state. When SS–29 is in its ON state, a signal is applied to line **1358** (FIG. 5F) to apply the next-character number stored in next-character-number register **394** to the next-character-number field of MBR **368**. When SS–29 times out, a signal is applied to switch SS–30 to its ON state. When SS–30 is in its ON state, a signal is applied to line **1298** (FIG. 5D) to condition gates **1272** to apply the address in supplementary-address counter **392** to MAR. The address in MAR is also stored in PAR during this time. The address at which the next supplementary-table entry is to be made, in this case the address **500**, is in this way read into MAR. When SS–30 times out, a signal is applied to switch SS–30*a* to its ON state. When SS–30*a* is in its ON state, a signal is applied to line **1268** to increment supplementary-address counter **392** by one. When SS–30*a* times out, a signal is applied to switch SS–31 to its ON state. When SS–31 is in its ON state, a signal is applied to line **1302** to cause a memory cycle, in this case to cause the supplementary-table entry which is stored in the next-character number and the chain-address fields of MBR **368** to be entered into the proper position in the supplementary table. When SS–31 times out, a signal is applied to line **794** (FIG. 5B), which, since there is a count of seven in *l/w* counter **312** at this time, finds AND gate **788** fully conditioned by the signal on line **782** to generate an output signal on line **798** which is applied throuhg OR-gate **800** to line **398** to cause SS–32 to switch to its ON state. When SS–32 is switched to its ON state, the circuit proceeds to find the next character at which it may begin looking for possible syllables, in this case the fourth character which is contained in the next-character-number field of entry **101**. As before, an unsuccessful check is made to see if this character is contained in the supplementary table before the circuit proceeds to find syllables starting with this letter.

Assume now that a match has been had on the syllable containing the reduced character set letters "CV" and that this syllable with its realtive frequency and its next-character number have been stored in memory position **102** (see FIG. 6) of core storage **364** in a manner previously described. Under these conditions, when SS–22 times out, causing a signal to be applied to line **814** (FIG. 5B), used flip-flop **374** is in its ON state (having been set to its ON state during a preceding 33*t* time when a search was being made for a new character at which to begin looking for possible syllables) and AND gate **810** is therefore fully conditioned, causing an output signal on line **385** which switches SS–23 to its ON state. The step now being performed is step **224** in FIG. 3A, namely the writing of a chain address. Looking at CL–21 time in the appendix section, it is noted that when TAC was transferred to MAR in order to put the proper address in MAR at which to store the new table entry, the address which was in MAR, the address from which the next-character number which gave rise to the present search was derived, was transferred into PAR **390**

(FIG. 5E). When SS–23 is in its ON state, a signal is applied through lines **1444** to gates **1448** to cause the address in PAR to be transferred into MAR **372**. The signal on line **1444** also causes the address in MAR to be transferred to PAR. The effect of this is to restore the address from which the next-character number was derived to MAR. During CL–23 time, a signal is also applied to line **804** (FIG. 5B) to reset used-flip-flop **374** to its OFF state, and a signal is applied to line **1308** (FIG. 5F) to set memory-status-flip-flop **1304** to its OFF or read state. When SS–23 times out, a signal is applied to switch SS–24 to its ON state. When SS–24 is in its ON state, a signal is applied to line **1302** to cause a memory cycle to be performed, in this case to cause the entry at the address indicated by MAR, the entry at address **101**, to be read into MBR **368**. When SS–24 times out, a signal is applied to switch SS–25 to its ON state. When SS–25 is in its ON state, a signal is applied to line **1350** (FIG. 5F) to condition gates **1348** to read the next-character number in the next-character-number field of this entry into next-character-number register **394**. As will be seen presently, this register will be used in forming the next supplementary-table-address entry to be made. During CL–25, a signal is also applied to line **1238** (FIG. 5D) to condition gates **1234** to apply the address in TAC **386** to the chain-address field of MBR **368**. This address is the address at which the last entry was made in the table, in this case the address **102**. Finally, during CL–25 time, a signal is applied to line **1306** to switch memory-status-flip-flop **1304** to its ON or write state. When SS–25 times out, a signal is applied to switch SS–26 to its ON state. When SS–26 is in its ON state, a signal is applied to line **1302** to cause a memory cycle to be performed, in this case to cause the entry at address **101** to be re-entered with the chain-address field now containing the address **102**. When SS–26 times out, a signal is applied to switch SS–27 to its ON state.

The circuit is now ready to perform step **226** to make an appropriate supplementary table entry. When SS–27 is in its ON state, a signal is applied to line **1314** (FIG. 5F) to reset MBR **368** to a blank. When SS–27 times out, a signal is applied to switch SS–28 to its ON state. When SS–28 is in its ON state, a signal is again applied to line **1238** (FIG. 5D) to condition gates **1234** to apply the address in TAC **386**, in this case the address **102**, to the chain-address field of MBR. When SS–28 times out, a signal is applied to switch SS–29 to its ON state. When SS–29 is in its ON state, a signal is applied to line **1358** (FIG. 5F) to condition gates **1356** to pass the next-character number stored in next-character-number register **394** into the next-character-number field of MBR. The two entries thus made in MBR **368** combine to form the second entry in the supplementary table shown in FIG. 7. When SS–29 times out, a signal is applied to switch SS–30 to its ON state. When SS–30 is in its ON state, a signal is applied to line **1298** (FIG. 5D) to condition gates **1272** to apply the address in supplementary-address counter **392** to MAR **372** (and also to transfer the address in MAR to PAR). When SS–30 times out, a signal is applied to switch SS–30*a* to its ON state. When SS–30*a* is in its ON state, a signal is applied to line **1268** to step supplementary-address counter **392** by one. When SS–30*a* times out, a signal is applied to switch SS–31 to its ON state. When SS–31 is in its ON state, a signal is applied to line **1302** to cause a memory cycle during which the supplementary entry is stored in the supplementary table. When SS–31 times out, a signal is applied to line **794** (FIG. 5B). Since end-of-word flip-flop **640** is not in its ON state at this time and there is a count of less than seven in *l/w* counter **312**, AND gate **654** of gating circuit **396** is fully conditioned at this time to generate an output signal on line **400**, which is applied to switch SS–0*b* to its ON state. This causes additional characters to be read into argument register **360** and additional possible syllables starting with the reduced character set letter "C" to be found in a manner previously described.

The circuit proceeds to store entries in the manner indicated above until the syllable having the reduced character set "TVLE" is applied to argument register **360**. Since the "E" in this syllable is the final "E" of the word, during CL–a time, end-of-word-flip-flop **640** is switched to its ON state in a manner previously described. Therefore, during CL–3 time when a test signal is applied to line **718** (FIG. 5B) the syllable and its frequency are applied through line **370** to the syllable and frequency fields in MBR **368**, but the signal applied to line **742** finds AND-gate **657** deconditioned, there being no signal applied to line **652** by end-of-word-flip-flop **640**, and the next-character-number field of MBR is therefore left blank. As indicated previously, a blank next-character-number field is interpreted to mean that the entry containing it is for the last syllable of a word. From this point on, the operations for this entry are the same as those previously described.

The circuit then proceeds to store the remaining entries in the table of FIG. 6 in a manner similar to that previously described.

After having written the entry at address **111** for the possible syllable "LE" and having found no further syllables starting with the letter "l," the circuits looks at the next-character-address field of entry **106** and seeks to find the possible syllables beginning with the character "e." The first attempt to do this during CL–3 time results in a mismatch and, since end-of-word-flip-flop **640** is in its ON state, when SS–3 times out, AND gate **650** (FIG. 5B) is fully conditioned, causing an output signal which switches SS–4 to its ON state. The circuit then proceeds to write a chain address of "0" at table **106** in a manner previously described to make a supplementary table entry for the next-character number **10**, and to proceed to step **234** (FIG. 3A) to attempt to find a new next-character number. At this time, next-character-address counter **420** (FIG. 5D) contains the address **107**, and during CL–32 and **33** time, the table entry at this address is read out into MBR **368** in a manner previously described. When SS–33 times out, a signal is applied to switch SS–33*a* to its ON state. When SS–33*a* times out, a signal is applied to line **1380** (FIG. 5F) to condition AND gates **1374** and **1378** of gating-circuit **426**. Since the syllable at address **107** is the last syllable of a word, the next-character field of this entry is a blank, and blank decoder **422** therefore generates an output signal on line **1372** to fully condition AND gate **1374** to generate an output signal on line **428** which switches SS–33*u* to its ON state.

The circuit is now ready to perform step **238** of FIG. 3A. When SS–33*u* is in its ON state, a signal is applied through line **1388** (FIG. 5E) to condition gates **343** to store the code for the character 2 contained in generator **432** in the chain-address field of MBR **368**. A signal is also applied to line **1306** to switch memory-status-flip-flop **1304** to its ON or write mode. When SS–33*u* times out, a signal is applied to switch SS–33*u* to its ON state. When SS–33*v* is in its ON state, a signal is applied to line **1302** to cause a memory cycle which stores the chain address of 2 at the address indicated in MAR, in this case at the address **107**. At this time, TAC **386** contains the address at which the last table entry was made, in this case the address **111**, and next-character-address counter **420** contains the address of the entry, the next-character-address field of which was just looked at. When these two registers contain the same count, the next-character-number field of all entries has been looked at and the circuit is ready to form syllable patterns. The contents of these two registers are applied to compare-circuit **1198** (FIG. 5D). Since they are not equal at this time, com-

pare-circuit **1198** generates an output signal on line **1250** which is applied as one input to AND gate **1252**. When SS–33*v* times out, a signal is applied through line **1254** as the other input to AND gate **1252**, causing an output signal on line **440**, which switches SS–33*b* to its ON state. When SS–33*b* is in its ON state, a signal is applied to line **1192** to step next-character-address counter **420**. This counter now contains the address **108**. When SS–33*b* times out, a signal is applied to switch SS–**32** to its ON state.

During clock times **32** and **33**, the table entry stored at address **108** is read out into MBR **368**. Since this entry contains the next-character number 9, when SS–33*a* times out, blank decoder **422** (FIG. 5F) is generating an output signal on line **1376** which conditions AND gate **1378** to generate an output signal on line **430** to switch SS–33*c* to its ON state. During CL–33*c*, 33*d*, 33*e*, 33*f*, and 33*g*, the next-character number 9 is stored in next-character-number register **394**, and the first entry in the supplementary table is read into MBR **368**, all in a manner previously described. Since the first entry in the supplementary table does not contain the next-character number 9, compare-circuit **442** (FIG. 5F) is generating an output signal on line **1366** when a signal is applied to line **1370** when SS–33*h* times out, causing AND gate **1368** to generate an output signal on line **454** which switches SS–33*o* to its ON state. As described previously, this causes the next entry in the supplementary table to be applied to MBR **368**. The above operations are repeated until the sixth entry in the supplementary table is applied to MBR **368**. The next-character-number field of this entry contains a 9, and compare-circuit **442** therefore generates an output signal on equal-line **1362** when SS–33*h* times out, causing AND gate **1364** to be fully conditioned to generate an output signal on line **452**, which switches SS–33*i* to its ON state.

The circuit is now ready to perform the write-chain-address-incurrent-table-entry step **244** of FIG. 3A. When SS–33*i* is in its ON state, a signal is applied to line **1406** (FIG. 5E) to condition gates **1402** to pass the chain address then stored in MBR **368**, in this case the chain address **111**, into chain-address register **460**. When SS–33*i* times out, a signal is applied to switch SS–33*j* to its ON state. When SS–33*j* is in its ON state, a signal is applied through line **1200** (FIG. 5D) to condition gates **1196** to apply the address in next-character-address counter **420**, in this case the address **108**, to MAR. When SS–33*i* times out, a signal is applied to switch SS–33*j* to its ON state. When SS–33*k* is in its ON state, a signal is applied to line **1302** to cause a memory cycle which results in the entry at the address indicated in MAR, the entry at address **108**, to be read into MBR **368**. When SS–33*k* times out, a signal is applied to switch SS–33*l* to its ON state. When SS–33*l* is in its ON state, a signal is applied through line **1400** (FIG. 5E) to condition gates **1396** to apply the chain address which was stored in chain-address register **460** to the chain-address field of MBR **368**. During CL–33*l* time, a signal is also applied to line **1306** to switch memory-status-flip-flop **1304** to its ON or write mode. When SS–33*l* out, a signal is applied to switch SS–33*m* to its ON state. When SS-33*m* is in its ON state, a signal is again applied to line **1302** to cause a memory cycle which results in the entry in MBR, this entry including the chain address which was derived from the supplementary table, being read into address **108** in core storage **364**. When SS–33*m* times out, a signal is applied to switch SS–33*n* to its ON state. When SS–33*n* is in its ON state, a signal is applied through line **1284** (FIG. 5D) to condition gates **1282** to apply the address in register **446** to supplementary-address counter **392**. This step is necessary to restore SAC to the address of the last entry which was made in the supplementary table so that any additional entry in this table which may be required will be made in the proper position. When SS–33*n* times out, a signal is applied to switch SS–33*b* to its ON state. As indicated previously, this causes a signal to be applied to line **1192** to increment next-character-address counter **420**, in this case incrementing it to the address **109**. When SS–33*b* times out, a signal is applied to switch SS–**32** to its ON state, causing a new search to commence for a character at which new syllables may be formed.

Operations similar to those described above are performed when the next-character fields of entries **109** and **110** are looked at. When the next-character-number field of entry **111** is looked at, it is found to be blank. and SS–33*u* is switched to its ON state in a manner previously described. When SS–33*u* is in its ON state, a 2 is stored in the chain-address field of this entry in a manner previously described, and the circuit to proceed to clock time 33*v*. When SS–33*v* times out, a signal is applied through line **1254** (FIG. 5D) to AND gates **1248** and **1252**. Since the next-character-number field of the last entry in the table is being looked at at this time, both next-character-address counter **420** and table-address counter **386** contain the same address, in this case the address **111**, and compare circuit **1198** in compare-and-gating circuit **436** therefore generates an output signal on equals-line **1246** which fully conditions AND gate **1248** to generate an output signal on line **438**, which causes SS–**85** to be switched to its ON state.

The circuit has now completed the storing of possible syllables in the table in core storage **364** and is ready to form possible syllable patterns. When SS–**85** is in its ON state, a signal is applied to line **1861** (FIG. 5R) to reset all of the number of syllable, syllable address, and ratio registers in word registers **488**. A signal is also applied to line **1600** (FIG. 5M) to reset all the frequency registers **1601**–**160N** in frequency registers **490**, and to line **1798** (FIG. 5P) to reset word-select counter **528** to its first position. When SS–**85** times out, a signal is applied to switch SS–85*a* to its ON state. When SS–85*a* is in its ON state, a signal is applied to line **1220** (FIG. 5D) to condition gates **1218** to apply the address in starting-point register **1208** to TAC **386**, a signal is applied to line **1424** (FIG. 5E) to switch flip-flop **487** to its OFF state, a signal is applied to line **1530** (FIG. 5M) to reset column-select counter **508** to its first position, a signal is applied to line **1672** to reset number-of-syllables counter **1670** to a count of one, and a signal is applied to line **1758** to reset intermediate-store register **1736**. When SS–85*a* times out, a signal is applied to switch SS–85*b* to its ON state. When SS–85*b* is in its ON state, a signal is applied to line **1228** (FIG. 5D) to step TAC **386** by one. This effectively puts the address of the first entry in the table of FIG. 6, the address **100**, in TAC. When SS–85*b* times out, a signal is applied to switch SS–**86** to its ON state. When SS–**86** is in its ON state, a signal is applied to line **1236** to condition gates **1232** to transfer the address in TAC **386** to MAR **372**. As with all other similar transfers, the address in MAR is transferred into PAR. This effectively sets the address **100** into MAR. During CL–**86**, a signal is also applied to line **1308** (FIG. 5F) to switch memory-status flip-flop **1304** to its OFF or read state. When SS–**86** times out, a signal is applied to switch SS–**87** to its ON state. When SS–**87** is in its ON state, a signal is applied to memory-cycle line **1302** to cause the contents of the entry at the address indicated in MAR, in this case the entry at address **100**, to be read into MBR **368**. When SS–**87** times out, a signal is applied to switch SS–87*a* to its ON state. When SS–87*a* is in its ON state, a signal is applied through line **1502** (FIG. 5D) to AND gate **1500**. However, since there is no used-bit in the entry in MBR at this time, AND gate **1500** is not fully conditioned and no operations are performed during this clock time. When SS–87*a* times out, a signal is applied through line **1506** (FIG. 5F) to AND gates **1494** and **1498**. Since the used-bit in MBR at this time is a 0, decoder **470** generates an output signal on 0 line **1492**, which fully conditions AND gate **1494** to generate an output signal on line **474**, which switches

SS–88 to its ON state. The purpose of the operation just described is to determine if the entry being investigated has been used to form previous syllable patterns.

Referring to FIG. 3B, it is seen that a determination must now be made as to whether the entry has a chain address of 0 or not. If the entry has a chain address of 0, this means that there are no acceptable syllables which may start with the letter following its last letter and therefore that it itself is not an acceptable syllable for the word in question. When SS–88 is in its ON state, a signal is applied to line **1228** (FIG. 3D) to step TAC **386** by one. A signal is also applied through line **1432** (FIG. 5E) to one input of AND gates **485** and **1428**. Since the entry at address **100** contains no level termination mark, there is no signal on line **1342** from zero-detector **512**, and flip-flop **487** remains in its OFF state. The chain address of 0 in the entry at address **100** means that the other input to AND gate **1428** on line **1426** from zero-detector **480** is not present. Nothing further happens during CL–88 time. When SS–88 times out, a signal is applied through line **1430** (FIG. 5E) to one input of AND gate **1343**. The other inputs to this AND gate are conditioned by output line **1422** from the equals side of zero-detector **480**, output line **1345** from the equals output of detector **512** and output line **1433** from the OFF side of flip-flop **487**. Output line **499** from AND gate **1343** is connected to switch SS–86 to its ON state. It will be remembered that TAC was incremented by one during CL–88 time so that it now contains the address **101**. Therefore, during CL–86 time, the entry at this address is read into MBR and the circuit proceeds through the steps just described until CL–88 time is again arrived at.

At CL–88 time, TAC **386** is again incremented by a signal applied to line **1228**, and a signal is again applied through line **1432** to condition AND gates **1428** and **485**. However, since the chain-address field of entry **101** is not blank, it contains the address **102**, zero-detector **480** is generating an output signal on not-equal-to-zero line **1426** at this time, fully conditioning AND gate **1428** to generate an output signal on line **1434**, which is applied to condition AND gates **1318**, **1322**, and **1436**. The absence of a signal on equal-to-zero line **1422** prevents AND gate **485** from being fully conditioned and flip-flop **487** remains in its OFF state. Gates **1318** being conditioned allow the syllable contained in MBR **368**, in this case the syllable "NVL," to be applied through line **494** to gates **1512A** (FIG. 5R). Since word-select counter **528** (FIG. 5P) is in its first position, there is an output signal on lines **1802A** which conditions gates **1512A** to pass the syllable through lines **1809A** to gates **1561A**. Since column-select counter **508** (FIG. 5M) is also in its first position at this time, decoder **1540** is generating an output signal on line **1551**, which is applied through cable **1550** to condition gates **1561A** (FIG. 5R) to store the syllable in syllable register **1821A**. Similarly, gates **1436** being conditioned allow the address in MAR **372**, the address **101**, to be applied through lines **492**, condition gates **1510A** (FIG. 5R), lines **1808A** conditioned gates **1571A**, and lines **1831A** to address register **1841A**. The first syllable and its address are in this manner stored in the first-syllable and address registers of word registers **488**. Gates **1322** being conditioned permit the frequency for this syllable contained in MBR **368** to be applied through lines **496**, conditioned gates **1521** (FIG. 5M) and lines **1591** to the first-frequency register **1601**. The output signal on line **1434** is also applied to increment number-of-syllables counter **1670** (FIG. 5M) to a count of one. Finally, the signal on line **1434** is applied as one input to AND gate **1420** (FIG. 5E) the other input to this AND gate being the output on not-equal-line **1416** from two-detector **482**. Since the chain address in the chain-address field of entry **101** is not a 2, AND gate **1420** is fully conditioned at this time to generate an output signal on line **1440**, which is applied to condition gates **1404** to pass the address in the chain-address field of MBR **368** into MAR **372**. As always, the address in MAR is also read into PAR. The effect of the above operations is to store the first syllable with its frequency and address in appropriate positions in the word registers and frequency registers and to set the circuit to continue forming the syllable pattern at a syllable which may follow the first syllable. When SS–88 times out, a signal is applied through line **1430** to one input of AND gate **1418**. Since the chain-address field of MBR **368** contains neither a 0 nor a 2, detectors **480** and **482** apply the other required inputs to this AND gate, causing an output signal to be generated on line **500**, which switches SS–88*a* to its ON state. When SS–88*a* is in its ON state, a signal is applied to line **1532** (FIG. 5M) to increment column-select counter **508** to its second position. When SS–88*a* times out, a signal is applied to switch SS–87 to its ON state.

Since the address **102** is now in MAR **372**, the return to CL–87 causes this entry to be read into MBR **368** and causes the succeeding operations indicated by step **87***a* and **87***b* to be performed on this entry. TAC is stepped during CL–88 time and, since the chain address of this entry is 0, no entry is made and the circuit returns to CL–86 time.

From the appendix and previous discussion, it is seen that since TAC now contains the address **103**, the entry at this address is now investigated and, since it has a chain address other than 0 or 2, the syllable, "CVN," is stored in syllable register **1822A**, the address **103** is stored in address register **1922A**, and the frequency is stored in frequency register **1602**. The chain address **105** is read into MAR, and the number-of-syllables counter and column-select counter are each incremented by one. The circuit then returns to CL–87 time to complete the forming of the syllable pattern.

The entry at address **105**, the syllable "TV," chain addresses to the entry at **111**, the syllable "LE." Therefore, as may be seen from FIG. 9 which shows the entries which are made in the word registers, the fourth time that CL–88 time comes up, the syllables at addresses **101**, **102**, and **105** have already been stored in the appropriate positions in the word register and the frequency register, and MBR **368** contains the entry stored at address **111**. Since this entry has a chain address of 2, AND gate **1428** is still conditioned at CL–88 time causing gates **1318**, **1322**, and **1436** to be conditioned to store the syllable, its frequency, and its address in appropriate positions of the word and frequency registers, and a signal is applied through line **1434** to step number-of-syllables counter **1670** to a count of four. However, 2-detector **482** is now generating an output signal on line **1412**, and AND gate **1420** is therefore not conditioned to store the chain address **2** in MAR. When SS–88 times out, the signal applied to line **1430** finds AND gate **1414** conditioned by the output signal on line **1412** from detector **482** and generates an output signal on line **502** which is applied to switch SS–89 to its ON state.

Referring to FIG. 3B, it is seen that the operation now to be performed is that of calculating the ratio of the average frequency for the syllables of the first possible syllable pattern to the square root of the sum of the squares of the standard deviation of these frequencies from the average. Calculate-circuit **504** of FIGS. 5M–5N operates to perform this function. When SS–89 is in its ON state, a signal is applied to line **1530** (FIG. 5M) to reset column-select counter **508** to its first position, and a signal is applied to line **1620** to condition gates **1621–162N** to pass the frequencies stored in frequency registers **1601–160N** (note that only the first four of the registers actually contain information at this time) through lines **1641–164N** to adder **1640**. When SS–89 times out, a signal is applied to switch SS–90 to its ON state. When SS–90 is in its ON state, a signal is applied through lines **1666** to condition gates **1662** to pass the sum of the frequencies as arrived at by adder **1640** through lines **1696** to the dividend input of divider **1700** and to condition gates **1664** to pass the number-of-syllables count, in this

case 4, from counter **1670** through lines **1698** to the divisor input of divider **1700**. The average frequency is therefore computed in divider **1700**. When SS–**90** times out, a signal is applied to switch SS–**91** to its ON state. When SS–**91** is in its ON state, a signal is applied to line **1712** to condition gates **1704** to pass the average frequency through lines **1714** to one input of subtractor **1718** and to condition gates **1710** to pass the frequency contained in frequency register **1601** (gates **1631** are now conditioned by the output signal on line **1551** since column-selector counter was reset to its first position at CL–**89** time) through lines **1716** to the other input of subtractor **1718**. Subtractor **1718** computes the deviation of the first frequency in register **1601** from the average frequency. When SS–**91** times out, a signal is applied to switch SS–**92** to its ON state. When SS–**92** is in its ON state, a signal is applied through line **1740** to condition gates **1732** to pass the deviation computed in subtractor **1718** through lines **1726** to squarer **1728** (FIG. 5N). The square of the deviation for the first frequency is in this way computed. When SS–**92** times out, a signal is applied to switch SS–**93** to its ON state. When SS–**93** is in its ON state, a signal is applied through line **1740** to condition gates **1732** to pass the square of the deviation just computed in squarer **1728** through lines **1742** to one input of adder **1746** and to condition gates **1738** to pass the contents of intermediate store **1736** through lines **1744** to the other input of adder **1746**. Since, at this time, intermediate store **1736** is blank, the effect of this operation is to store the square computed in squarer **1728** in adder **1746**.

When SS–**93** times out, a signal is applied through line **1690** (FIG. 5M) to one input of AND gates **1684** and **1688**. Since number-of-syllables counter **1670** contains a count of 4 at this time and column-select counter **508** is set to its first position at this time, the two inputs to compare-circuit **1538** are not equal, and compare-circuit **1538** therefore generates an output signal on line **1686** which fully conditions AND gate **1688** to generate an output signal on line **1694**, which signal switches SS–**94** to its ON state.

When SS–**94** times out, a signal is applied to switch through line **1532** to increment column-select counter **508** to its second position, and a signal is applied through lines **1754** (FIG. 5N) to condition gates **1750** to pass the contents of adder **1746**, in this case the square of the standard deviation of the first frequency from the average, into intermediate store **1736**.

When SS–**94** times out, a signal is applied to switch SS–**91** to its ON state. Since the column-select counter is now in its second position, during CL–**91** and CL–**92** time, the square of the deviation of the frequency of the second syllable from the average is computed and stored in quarter **1728**. During CL–**93** time, this square deviation is applied to one input of adder **1746**, and the previously computed square deviation stored in intermediate store **1736** is applied to the other input of this adder. The resulting sum in this adder is the sum of the squares of these two frequencies. Since the count in the column-select counter and the number-of-syllables counter is still not equal, this sum is stored in intermediate store **1736**, and the circuit proceeds to compute the square of the deviation of the frequency of the third syllable from the average, to add this to the previously computed sum, and to compute the square of the deviation of the frequency of the fourth syllable from the average and to add this to the previously computed sum. When CL–**93** times out at this time, adder **1746** contains the sum of the square of the standard deviations of all the frequencies from the average, and both column-select counter **508** and number-of-syllables counter **1670** contain a count of 4. Therefore, compare circuit **1538** generates an output signal on line **1682**, which fully conditions AND gate **1684** to generate an output signal on line **1692**, which signal switches SS–**95** to its ON state.

When SS–**95** is in its ON state, a signal is applied through line **1760** (FIG. 5N) to condition gates **1752** to apply the sum of the squares of the standard deviations contained in adder **1746** through lines **1764** to the dividend input of divider **1766** and to condition gates **1680** to apply the number of syllables contained in number-of-syllables counter **1670** through lines **1762** to the divisor input of this divider. The result of this operation is to compute the average square deviation. When SS–**95** times out, a signal is applied to switch SS–**96** to its ON state. When SS–**96** is in its ON state, a signal is applied through line **1772** to condition gates **1770** to apply the average square deviation through lines **1774** to square-rooter **1776**. The square root of the sum of the squares of the average deviation is computed in square-rooter **1776**. When SS–**96** times out, a signal is applied to switch SS–**97** to its ON state. When SS–**97** is in its ON state, a signal is applied to line **1782** to condition gates **1706** to apply the average frequency contained in divider **1700** through lines **1702** and lines **1784** to the dividend input of divider **1788** and to condition gates **1780** to pass the square root of the sum of the squares of the average deviation through lines **1786** to the divisor input of this divider. When SS–**97** times out, a signal is applied to switch SS–**98** to its ON state.

When SS–**98** is in its ON state, a signal is applied through line **1794** to condition gates **1792** to apply the computed ratio through lines **506** and conditioned gates **1796**A (FIG. 5S) (word-select counter **528** is still in its first position) to ratio register **1860**A. A signal is also applied through lines **1850** (FIG. 5N) to condition gates **1678** to apply the number of syllables contained in number-of-syllable register **1670** through lines **1852** and conditioned gates **1806**A (FIG. 5R) to number-of-syllables register **1856**A. All of the desired information with reference to the first possible syllable pattern is therefore stored in the word-one position of word register **488**. When SS–**98** times out, a signal is applied to switch SS–**99** to its ON state. When SS–**99** is in its ON state, a signal is applied to line **1600** (FIG. 5M) to reset frequency registers **1601**–**160**N. When SS–**99** times out, a signal is applied to switch SS–**100** to its ON state.

The circuit is now ready to perform step **264** of the operation, namely to determine whether all possible syllable patterns have been formed or whether there are additional possible syllable patterns. When SS–**100** is in its ON state, a signal is applied to line **1534** (FIG. 5M) to decrement column-select counter **508** by one position. At this time, the column-select counter is set to 4. Therefore, when it is decremented, it has a count of 3 therein. When SS–**100** times out, a signal is applied through line **1549** to one input of AND gate **1548**. Since column-select counter **508** has a count of 3 in it (i.e. a count other than 0), line **1544** from decoder **1540** has a signal on it at this time which fully conditions AND gate **1548** to generate an output signal on line **518** which switches SS–**101** to its ON state. As will be seen later, if column-select counter has a count of 0 in it at this time, then all possible syllable patterns have been formed and the circuit is ready to select the preferred pattern.

During clock times **101** through **105**, the circuit is seeking a new table entry at which to begin forming syllable patterns. When SS–**101** is in its ON state, a signal is applied through line **1968** (FIG. 5T) to condition gates **1468**. Since word-select counter **528** is still in its first position and column-select counter **508** is in its third position, AND gate **1583**A (not shown) of the gates **1581**A–**158**NA (FIG. 5T), is conditioned, generating an output signal on line **1943**A which conditions gate **1933**A to pass the address of the third syllable of the syllable pattern which was formed in word **1** through line **1953**, OR gate **1960**A, line **1962**A, OR gate **1964**, and line **1966** to the information input of gates **1468**. This address is applied through gates **1468** and lines **1466** to MAR **372**. The address **105** is in this manner read into MAR. When SS–**101** is in its ON state, a signal is also applied to line **1308** (FIG. 5F) to switch memory-status-flip-flop **1304** to its OFF or read state. When SS–**101** times out, a signal

is applied to switch SS–102 to its ON state. When SS–102 is in is ON state, a signal is applied to line 1302 to cause the entry at address 105 to be read into MBR 368. When SS–102 times out, a signal is applied to switch SS–103 to its ON state. When SS–103 is in its ON state, a signal is applied to line 1306 to switch memory-status-flip-flop 1304 to its ON or write state, and a signal is applied through line 1488 to condition gates 1482 to apply a one from generator 1486 through lines 1480 to the used-bit field of MBR 368. When SS–103 times out, a signal is applied through line 1347 to one input of AND gates 1344 and 1346. Since there is no level termination mark in the level-termination field of entry 105, zero-detector 512 generates an output signal on line 1345 which fully conditions AND gate 1346 to generate an output signal on line 524 which switches SS–104 to its ON state. When SS–104 is in its ON state, a signal is applied to line 1302 to cause the entry at address 105 to be read back in with a used bit in its used-bit field. When SS–104 times out, a signal is applied through line 1437 (FIG. 5E) to one input of AND gate 1435. Since flip-flop 487 is in its OFF state at this time, a second input is being applied to AND gate 1435 through line 1433. Output line 1441 from AND gate 1435 is connected to switch SS–105 to its ON state. When SS–105 is in its ON state, a signal is applied to line 1800 (FIG. 5P) to increment word-select counter 528 to its second position. When SS–105 times out, a signal is applied to switch SS–85*a* to its ON state.

The circuit now proceeds to store the syllables "NVL" and "CVN" in the first two syllable positions of the second word in word register 488 and to store the relative frequencies of these two syllables in frequency registers 1601 and 1602 in a manner identical to that previously described. The address 105 is again stored in MAR after these operations and after CL–87 time when this entry is read into MBR 368, the circuit proceeds to CL–87*a* time. During CL–87*a* time, a signal is applied to line 1502 (FIG. 5D) to condition AND gate 1500. Since there is now a used bit of 1 in the used-bit field of the entry at address 105, decoder 470 generates an output signal on line 1496 which is applied to fully condition AND gate 1500 to generate an output signal on line 1504 which conditions gates 1476 to apply the address in MAR, the addres 105, through lines 1478 and OR gate 1224 to TAC 386. When SS–87*a* times out, the signal applied to line 1506 (FIG. 5F) finds AND gate 1498 fully conditioned by the signal on line 1496 from decoder 470 and generates an output signal on line 476 which switches SS–87*b* to its ON state. When SS–87*b* is in its ON state, a signal is applied to line 1228 to step TAC 386 by one, to increment this counter to the address 106. When SS–87*b* times out, a signal is applied to switch SS–86 to its ON state.

The circuit now attempts to determine if the entry at address 106, the syllable "TVL," is a possible syllable which may be used with the two syllables already stored. Since this syllable has a chain address of 0, it is not an acceptable syllable, and the circuit proceeds to determine if the syllable stored at address 107 is an acceptable syllable. As is seen from FIG. 9, this is an acceptable syllable in the pattern, and the circuit proceeds to store the second possible syllable pattern and to compute the desired ratio therefore in a manner previously described.

When the column-select counter is now decremented by one, it is the second syllable, the syllable "CVN" stored at address 103, which has a used bit inserted in its used-bit field, and the circuit proceeds to form the remaining two syllable patterns shown in FIG. 9 in a manner substantially the same as that already described. In the process, a used bit is also inserted in the used-bit field of the syllable "V" stored at address 108.

When the last syllable pattern has been stored, the circuit, as always, proceeds to CL–100 time. During this time, column-select counter 508 (FIG. 5M) is decremented to a count of 2 (i.e. the last possible syllable pattern has 3 syllables). Since the column-select counter does not have a count of 0 in it, the circuit proceeds to switch SS–101 to its ON state. During CL–101 time, the address 104, the address of the second syllable in the last possible syllable pattern, is read into MAR and the memory is set to its read status. During SS–102 time, this entry is read into MBR 368. During CL–103 time, a one bit is inserted in the used field of MBR, and the memory is set to its write status. When SS–103 times out, the signal applied to line 1347 (FIG. 5F) finds AND gate 1344 fully conditioned by the output signal on not-equal-to-zero line 1342 from detector 512, this being true since there is a level-termination mark in the level-termination field of entry 104. AND gate 1344 therefore generates an output signal on line 522 to switch SS–100 to its ON state. This results in column-select counter 508 being decremented to a count of 1. Since there is still not a count of 0 in this counter, the circuit again proceeds through CL–101 to 103 time, this time reading the entry at address 101, the entry for the syllable "NVL," into MBR 368. Since this entry also has a mark in its level-termination field, AND gate 1344 is again caused to generate an output signal on line 522, which again switches SS–100 to its ON state. This time when column-select counter 508 (FIG. 5M) is decremented, it is decremented to a count of 0. Therefore, when SS–100 times out, decoder 1540 is generating an output signal on line 1542, and AND gate 1546 is fully conditioned to generate an output signal on line 516 which switches SS–107 to its ON state. As has been indicated previously, this means that the circuit has formed all possible syllable patterns and is now ready to select the preferred pattern.

Before describing the select-preferred-pattern step of the operation, one additional condition should be considered. In the word "Mercantile" there is no possible syllable having both a level termination mark in its LTM field and a chain address of zero. When an entry having both a level termination mark and a chain addres of zero is encountered, the signal is applied to line 1432 (FIG. 5E) during CL–88 time fully conditions AND gate 485 to generate an output signal on line 498 which signal switches flip-flop 487 to its ON state. When SS–88 times out, the signal applied to line 1430 finds AND gate 1429 in gating circuit 486 conditioned by the output signal on ON-side output line 1427 from flip-flop 487 causing an output signal on line 501 which switches SS–100 to its ON state. This effectively performs step 255 (FIG. 3B) of the operation. The effect of this operation is to indicate that the syllable pattern being formed cannot be completed and that a ratio should not be computed for this pattern.

The circuit proceeds through clock times 100–103 in a manner identical to that previously described with column-select counter 508 (FIG. 5M) being decremented by one and a used bit being written into the used-bit field of the last syllable stored in the word register. If the syllable which has the zero chain address and the level termination mark is a possible first syllable for the word, then no further syllable patterns can be formed, and, when SS–100 times out, the circuit proceeds to the select-preferred-pattern step in the manner indicated above (also see appendix).

If the syllable is not a possible first syllable for the word, then, when SS–103 times out, the circuit either returns to CL–100 time or goes on to CL–104 time, depending on whether the preceding syllable has a level termination mark or not (see appendix). If the circuit proceeds to CL–104 time, when SS–104 times out a signal is applied through line 1437 (FIG. 5E) to one input of AND gate 1431. Since flip-flop 487 was previously set to its ON state, this AND gate is fully conditioned by the output signal on ON-side output line 1427 from flip-flop 487 causing an output signal on line 1439 which switches SS–85*a* to its ON state. It will be noted that since the operation performed during CL–105 time was not performed, the word-select counter is not incremented and is therefore still set at the same position it was set at when the unsuc-

cessful effort to form a syllable pattern was attempted. The new syllable pattern which is formed when SS–85*a* is switched to its ON state is therefore stored in this word position.

Select preferred pattern

Since all the ratios are now stored in the registers **1860A–1860D** (FIG. 5S) of word register **488**, the circuit is now ready to select the preferred pattern, i.e. the circuit is ready to perform step **270** of (FIG. 1). As indicated previously, at the beginning of this step, SS–**107** is switched to its ON state. When SS–**107** is in its ON state, a signal is applied to line **1798** (FIG. 5P) to reset word-select counter **528** to its first position, a signal is applied to line **2022** to reset preferred-word register **544** to its first position, and a signal is applied to line **2004** to set first-time-through flip-flop **535** to its ON state. When SS–**107** times out, a signal is applied to switch SS–**108** to its ON state.

The circuit is now ready to perform step **274** (FIG. 4). When SS–**108** is in its ON state, a signal is applied through line **2008** (FIG. 5P) to condition AND gate **1986**. However, since word-select register **528** is in its first position, gate **1804A** (FIG. 5Q) is conditioned to apply the ratio in ratio register **1860A** through OR gate **536** to blank-detector **530** and, since there is a ratio in this register, blank-detector **530** is not generating an output at this time so that AND gate **1986** is not fully conditioned and nothing happens during CL–**108** time. Referring to FIG. 9, it is seen that the ratio in ratio register **1860A** is less than 1. Therefore, compare-to-one circuit **532** is generating an output signal on less-than-or-equal-to line **1990** which is applied to partially condition AND gate **1982** in gating circuit **534**. A second conditioning input to the AND gate is supplied through not-blank line **1980** from detector **530**. When SS–**108** times out, a signal is applied through line **2006** to the other input of this AND gate, causing an output signal on line **540**, which switches SS–**111** to its ON state.

When SS–**111** is in its ON state, a signal is applied through line **1800** to increment word-select counter **528** by one. This counter now contains the address of the second word. When SS–**111** times out, a signal is applied to switch SS–**108** to its ON state. Since the ratio in ratio register **1860B** is also not a blank, again nothing happens during CL–**108** time. However, the ratio in ratio register **1860B** is greater than 1; therefore, compare-to-one circuit **532** is generating an output signal on line **1992** to partially condition AND gates **1994** and **1996**. Since first-time-through flip-flop **535** is still in its ON state, there is a signal on its ON-side output line **2000** which fully conditions AND gate **1996** when SS–**108** times out, causing an output signal on line **542** which switches SS–**110** to its ON state. When SS–**110** is in its ON state, a signal is applied to line **2002** to switch first-time-through flip-flop **535** to its OFF state, and a signal is applied to line **2020** to condition gates **2018** to transfer the word address in word-select counter **528** to preferred-word register **544**. The preferred-word register, therefore, contains the address of the second word. When SS–**110** times out, a signal is applied to switch SS–**111** to its ON state. As before, when SS–**111** is in its ON state, word-select counter **528** is incremented, and the circuit proceeds to look at the ratio field of the next word in the word register.

Since neither of the remaining words has a ratio greater than 1, nothing happens during either of these steps, and after the ratio field of the last of these words is looked at, the circuit returns to CL–**108** time. At this time, the word-select counter has a count of 5 in it. Since the ratio field of this word is blank, at CL–**108** time both inputs to AND gate **1986** are conditioned, causing this AND gate to generate an output signal on line **2010** which is applied to partially condition AND gates **2012A–2012N**. The preferred-word register is at this time set to its second position, causing an output signal on line **2014B**, which fully conditions AND gate **2012B** to generate an output signal on line **2024B**, which signal conditions gates **1898B** (FIG. 5Q) to pass the syllable pattern contained in syllable registers **1821B–182NB** (FIG. 5R) into coded-word-hyphenation registers **1911–191N** (FIG. 5R). The output signal on line **1964** from blank-detector **530** also partially conditions AND gate **1988** (FIG. 5P). When SS–**108** times out, a signal is applied to line **2006** to fully condition this AND gate, causing an output signal on line **538** which switches SS–**120** to its ON state. When SS–**120** is in its ON state, the read-out operation commences.

While in the example just described, there was only one possible syllable pattern having a ratio greater than 1, and, as was indicated previously, there will always be at least 1 such ratio, situations will arise where there are 2 or more syllable patterns having a ratio greater than 1. To illustrate the sequence of events which occur in such a situation, assume that the word "Mercantile" had a fifth possible syllable pattern and that this pattern had a ratio greater than 1. Therefore, when the word-select counter **1800** was advanced to this word position, the fifth word position in the word register, and SS–**108** times out, compare-to-one circuit **532** is generating an output signal on greater-than-one line **1992** to partially condition AND gate **1994** (FIG. 5P). Since first-time-through flip-flop **535** was reset to its OFF state when the second word was found to have a ratio greater than 1, there is an output signal on line **1998** from the OFF side of this flip-flop fully conditioning AND gate **1994** to generate an output signal on line **546** which causes SS–**109** to be switched to its ON state.

The circuit is now set to perform step **280** (FIG. 4). When SS–**109** is in its ON state, a signal is applied through line **2034** (FIG. 5P) to condition one input of AND gate **2030B**. The other input to this AND gate is output line **2014B** from preferred-word register **544**, which line, since register **544** is set to its second position, is up at this time. The resulting output on line **2040B** is applied to condition gates **1866B** to pass the number of syllables in this case 3, of the preferred word through lines **1872B**, OR gates **1874**, and lines **1878** to the preferred-word input of compare-circuit **1880**. The signal on line **2034** also conditions an AND gate **2026E** (not shown), the other input to which is the output line **1802E** from word-select counter **528**. This results in the number of syllables for the fifth word being applied through a line **1868E**, OR gates **1870**, and lines **1876** to the selected-word input of compare-circuit **1880**. As indicated previously, when two possible syllable patterns have ratios greater than 1, the one with the least number of syllables is the preferred pattern. Therefore, if the pattern in the second position, the pattern indicated by preferred-word register **544**, has less syllables, compare-circuit **1880** generates an output signal on line **1882** which, when SS–**109** times out, fully conditions AND gate **1884** to generate an output signal on line **552** which switches SS–**111** to its ON state, causing the second word to be left in the preferred-word register and the circuit to proceed to look at the next word. If, on the other hand, the pattern in word register **5** has a lesser number of syllables, for example two syllables, then compare-circuit **1880** generates an output signal on line **1890** which, when SS–**109** times out, fully conditions AND gate **1892** to generate an output signal on line **550** which causes SS–**110** to be switched to its ON state. From the appendix and previous discussion, it can be seen that this causes the word then in the word-select counter, in this case the fifth word, to be transferred to the preferred-word register and then causes the circuit to advance to the next-word register to see what its ratio is. Finally, if it is found that the two words applied to compare-circuit **1880** have the same number of syllables, in this case 3, an output is derived on line **1886** which, when SS–**109** times out, fully conditions AND

gate **1888** to generate an output signal on line **554** which causes SS–**112** to be switched to its ON state.

It will be remembered from previous discussion that under these conditions the pattern having the highest ratio is the preferred pattern. When SS–**112** is in its ON state, the circuit is performing step **282** (FIG. 4). At this time, a signal is applied to line **2036**, which signal partially conditions AND gate **2032B** (FIG. 5P) and **2028E** (not shown). The other input to AND gate **2032B** is the output line **2014B** from preferred-word register **544**, and the other input to AND gate **2028E** is the output line **1802E** (not shown) from word-select counter **528**. Output line **2044B** from AND gate **2032B** is connected to condition gates **1974B** (FIG. 5Q) to apply the ratio in register **1860B** through lines **2050B**, OR gates **2052** and lines **2056** to the preferred-word input of compare-circuit **2058**. Output line **2042E** from AND gate **2028E** is connected to condition gates **1972E** to pass the ratio contained in register **1860E** through lines **2046E** (none of the previously described elements are shown in the figure), OR gates **2048** (FIG. 5Q), and lines **2054** to the word input of compare-circuit **2058**. If the fifth word, the word indicated by word-select counter **528** has the higher ratio, compare-circuit **2058** generates an output signal on line **2060** which, when SS–**112** times out causing a signal to appear on line **2068**, fully conditions AND gate **2062** to generate an output signal on line **558** which switches SS–**110** to its ON state. As indicated previously, when this occurs the address in word-select counter **528**, the address **5**, is transferred into preferred-word register **544**, and the circuit then proceeds to look at the next word in the word register. If, on the other hand, the ratio of the word in the word register is less than or equal to the ratio of the preferred word, the second word in word register **488**, compare-circuit **2058** generates an output signal on line **2064** which, when SS–**112** times out causing a signal on line **2068**, fully conditions AND gate **2066** to generate an output signal on line **560** which signal is applied to switch SS–**111** to its ON state. Under these conditions, the entry in the preferred-word register is left alone and the circuit proceeds to look at the next word position in word register **488**.

Read-out pattern

When SS–**120** is switched to its ON state, this indicates that a preferred pattern has been selected and that the circuit is ready to read out this preferred pattern (i.e. to perform step **290** of FIG. 1). It will be remembered that the last operation performed at CL–**108** time before the circuit switched SS–**120** to its ON state, was to transfer the preferred syllable pattern from word register **488** (FIG. 5R) to coded-word-hyphenation register **564** (FIG. 5K). Each of these syllables is stored in an individual register **1911**–**191N**. When SS–**120** is in its ON state, a signal is applied to line **616** (FIG. 5A) to reset output-control ring **612** to its first position, a signal is applied to line **2190** (FIG. 5L) to reset input-control ring **2188** for the output register to its first position, a signal is applied to line **2074** (FIG. 5K) to reset group-control ring **2070** to its first position, and a signal is applied to line **2076** to reset unit-control ring **2072** to its first position. When SS–**120** times out, a signal is applied to switch SS–**121** to its ON state.

The first time that SS–**121** is in its ON state, AND gate **2081A** is fully conditioned by the signals applied to it on line **2079A** from group-control ring **2070** and line **2091** from unit-control ring **2072**. The resulting output signal on line **2021A** causes the reduced character set character "N" stored in the first position of register **1911** to be applied through lines **2131** and OR gates **2140** to blank decoder **2144**. Since this character is not a blank, decoder **2144** generates an output signal on line **2148** which partially conditions AND gate **2150**. This AND gate is fully conditioned by a signal applied to line **2162** when SS–**121** is in its ON state. The resulting output signal on line **574** is applied to condition gates **634** to pass the character stored in the first position of input-buffer register **302** (FIG. 5A) through lines **2170** (FIG. 5L), OR gates **2174**, and lines **576** to the first position of output register **346**. The character "m" is in this way stored in output register **346**. When SS–**121** times out, a signal is applied through line **2152** (FIG. 5L) to one input of AND gate **2116**. Since unit-control ring **2072** is in its first position, there is no signal on line **2096**, and inverter **2112** therefore applies a second conditioning signal through line **2114** to this AND gate. The third input to AND gate **2116** is derived from not-blank line **2148** from blank decoder **2144**. AND gate **2116** being fully conditioned causes an output signal to appear on line **2154**, which signal switches SS–**122** to its ON state.

When SS–**122** is in its ON state, a signal is applied to line **620** (FIG. 5A) to increment output-control ring **612** to its second position, causing the character "e" to be applied to lines **332**, a signal is applied to line **2192** (FIG. 5K) to increment input-control ring **2188** to its second position, and a signal is applied to line **2080** (FIG. 5K) to increment unit-control ring **2072** to its second position. When SS–**122** times out, a signal is applied to again switch SS–**121** to its ON state.

At this time, AND gate **2082A** is fully conditioned, causing the reduced character set letter "v" to be applied to blank decoder **2144** and ultimately resulting in the character "e" being stored in the second position of output register **346**. The third letter of the first syllable of "Mercantile," the character "r," is, a similar manner, stored in output register **346**.

During the next CL–**121** time, AND gate **2084A** is fully conditioned, causing a blank character to be applied to blank decoder **2144**. Decoder **2144** therefore generates an output signal on equal-line **2146** which partially conditions AND gate **2108**. Since unit-control ring **2072** is not in its first position, there is no signal on line **2091**, and inverter **2102** is generating an output signal on line **2104** which is applied as the second input to AND gate **2108**. The signal applied to line **2162** when SS–**121** is in its ON state fully conditions this AND gate to generate an output signal on line **578** which is applied through OR gate **2164** and line **2166** to condition gates **2168** to pass the coded character for a hyphen from generator **572** through lines **2170**, gates **2168**, lines **2172**, OR gates **2174**, and lines **576** to the fourth position in output register **346**. The signal on equal-line **2146** is also applied as one input to AND gate **2106**, a second input to this AND gate being output line **2104** from inverter **2102**. When SS–**121** times out, a signal is applied to line **2152** to fully condition AND gate **2106**, causing an output signal on line **2160** which switches SS–**128** to its ON state.

When SS–**128** is in it ON state, a signal is applied to line **2078** (FIG. 5K) to increment group-control ring **2070** to its second position, a signal is applied to line **2076** to reset unit-control ring **2072** to its first position, and a signal is applied to line **2192** (FIG. 5L) to increment input-control ring **2188** to its fifth position. It is noted that output-control ring **612** is not incremented at this time so that it is still sitting over the fourth letter of the word, the letter "c." When SS–**128** times out, a signal is applied to again switch SS–**121** to its ON state. From previous discussion, it can be seen that the second syllable of the word, the syllable "can" is stored in the output register in a manner identical to that described for the first syllable, and a hyphen is inserted in the output register at the end of this syllable. The storing of the last syllable of the word, the syllable "tile," is accomplished in an identical manner.

Before describing the sequence of events which occur after the last syllable has been stored, mention should be made of the sequence of operations which occur when a six-letter syllable is encountered in one of the registers **1911**–**191N**, a situation which does not occur in the word "Mercantile." Under these circumstances, when SS–**121**

is in its ON state, blank decoder **2144** is still generating an output signal on line **2148** and the character is stored in output register **346** in the manner previously described. However, when SS–**121** times out, unit-control ring **2172** is generating an output signal on line **2096** which, in conjunction with the signal on line **2148**, and the signal applied to line **2152**, fully conditions AND gate **2110** to generate an output signal on line **2156**, which signal is applied to switch SS–**124** to its ON state. When SS–**124** is in its ON state, a signal is applied to line **2078** to increment group-control ring **2070**, and a signal is applied to line **2192** to increment input-control ring **2188**. Input-control ring **2188** is therefore set over the character position following that in which the last character was stored. When SS–**124** times out, a signal is applied to switch SS–**125** to its ON state. When SS–**125** is in its ON state, a signal is applied through line **580** (FIG. 5L) and OR gate **2164** to condition gates **2168** to apply the hyphen from hyphen generator **572** through lines **576** to the position in output register **346** indicated by input-control ring **2188**, in this case the position following that in which the last character was stored. When SS–**125** times out, a signal is applied to switch SS–**126** to its ON state. When SS–**126** is in its ON state, a signal is applied to line **620** to increment output-control ring **612**, a signal is applied to line **2192** to again increment input-control ring **2188**, and a signal is applied to line **2076** to reset unit-control ring **2072** to its first position. The circuit is therefore completely reset to look at the next syllable. When SS–**126** times out, a signal is applied to again switch SS–**121** to its ON state.

Returning to the example with the word "Mercantile," when a blank is detected after the last character of the last syllable, a hyphen is stored in the output register in a manner previously described. The circuit then proceeds to CL–**128** time during which the group-control ring is incremented, the unit-control ring reset and input-control ring **2188** is stepped one position. The circuit then returns to CL–**121** time. At this time, unit-control ring **2072** is in its first position and the character selected in coded-word-hyphenation registers **564** is a blank. Therefore, neither AND gate **2108** nor AND gate **2150** is fully conditioned at CL–**121** time and this time period is effectively idle. When SS–**121** times out, the signals applied to lines **2146**, **2091**, and **2152** fully condition AND gate **2100** to generate an output signal on line **2158**, which switches SS–**129** to its ON state. When SS–**129** is in its ON state, a signal is applied to line **2194** to decrement input-control ring **2188** by one. This sets the input-control ring over the position in output register **346** at which a hyphen was stored after the last syllable of the word. When SS–**129** times out, a signal is applied to switch SS–**130** to its ON state. When SS–**130** is in its ON state, a signal is applied to line **2184** to condition gates **2178** to pass a blank from blank generator **2182** through lines **2176**, OR gates **2174**, and lines **576** to output register **346**. The hyphen inserted at the end of the word is in this way eradicated. When SS–**130** times out, a signal is generated which indicates that the function of this circuit has been completed.

While the invention has been particularly shown and described with reference to a preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
0	117_T	Reset next-character-number counter 748 (FIG. 5D) to one (i.e. apply a signal to line 750). Reset *l/w* counter 312 (FIG. 5C) to one (i.e. apply a signal to line 768). Reset output-control ring 612 (FIG. 5A) to first position (i.e. apply signal to line 616). Reset argument-input-control ring 704 (FIG. 5B) to position one (i.e. apply a signal to line 706). Transfer contents of starting-point register 1208 (FIG. 5D) to TAC 386 (i.e. apply a signal to line 1220 to condition gates 1218). Apply contents of supplementary-starting-point register 1210 (FIG. 5D) to supplementary-address counter 392 (i.e. apply a signal through line 1260 to gates 1258).	Switch SS-0*a* to its ON state	CL-0*a*.
0*a*	0_T	Route * from generator 694 (FIG. 5B) into argument register 360 (i.e. apply a signal through line 696 to condition gates 690).	Switch SS-0*b* to its ON state	CL-0*b*.
0*b*	$0a_T$ or 3_T or 22_T or 31_T	Increment argument-input-control ring 704 (FIG. 5B) by one (i.e. apply a signal to line 708).	Switch SS-1 to its ON state	CL-1.
1	$0b_T$ or $33z_T$	Route selected character from input-buffer register 302 (FIG. 5A) to argument register 360 (i.e. apply conditioning signal through line 682 to gates 630). Set match indicators in associative memory 362 (FIG. 5B) to their match side, and set match flip-flop 716 to its OFF state (i.e. apply a signal to line 714). Step *l/w* counter (FIG. 5C) by one (i.e. apply a signal through line 860 to O R gate 856). Step next-character-number counter 748 (FIG. 5D) by one (i.e. apply a signal to line 752).	Switch SS-1*a* to its ON state	CL-1*a*.
1*a*	1_T	Increment output-control ring 612 (FIG. 5A) by one (i.e. apply a signal to line 620). (Note end-of-word flip-flop 640 (FIG. 5B) is switched to its ON state if selected character is a blank.)	Switch on SS-1*b* if end-of-word flip-flop is in its ON state (i.e. condition AND gate 648). Switch on SS-2 if end-of-word flip-flop is in its OFF state (i.e. condition AND gate 656).	CL-1*b* or CL-2.
1*b*	$1a_T$	Decrement output-control ring 612 (FIG. 5A) by one (i.e. apply a signal to line 618).	Switch on SS-1*c*.	CL-1*c*.
1*c*	$1b_T$	Route selected character from input-buffer register 302 into argument register 360 (i.e. apply a signal through line 682 (FIG. 5B) to gates 630).	Switch on SS-1*d*	CL-1*d*.
1*d*	$1c_T$	Increment argument-input-control ring 704 (FIG. 5B) (i.e. apply a signal to line 708).	Switch on SS-1*e*	CL-1*e*.
1*e*	$1d_T$	Route * to argument register 360 (i.e. apply a signal through line 696 (FIG. 5B) to condition gates 690).	Switch on SS-2	CL-2.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
2	$1a_T$ or $1e_T$	Perform an association between the contents of argument register 360 and the contents of associative memory 362 (i.e. apply a signal to line 712 (FIG. 5B)).	Switch on SS-3	CL-3.
3	2_T	If no match, switch match flip-flop 716 (FIG. 5B) to its ON state; if a match, read syllable and frequency of occurrence from associative memory 362 into appropriate fields in MBR 368 (FIG. 5F) and route contents of next-character-number counter 748 (FIG. 5D) into next-character-number field of MBR 368 (i.e. apply a signal to test line 718).	If a match is found, switch SS-20 to its ON state (i.e. condition AND gate 724 (FIG. 5B)). If no match and *l/w* counter equals 7, or if no match and end-of-word flip-flop 640 is in its ON state, switch SS-4 to its ON state (i.e. condition AND gate 650 or 728). If no match is found and *l/w* counter is not equal to 7, switch on SS-0*b* to its ON state (i.e. condition AND gate 730).	CL-20 or CL-4 or CL-0*b*.
4	3_T	Put one in level-termination field of MBR 368 if TE flip-flop 408 is in its ON state (i.e. apply a conditioning signal through line 1340 (FIG. 5E) to AND gate 822).	If TE trigger 408 is reset, switch SS-5 to its ON state (i.e. condition AND gate 812 (FIG. 5A)). If TE flip-flop is in its ON state, switch SS-31 to its ON state (i.e. condition AND gate 820).	CL-5 or CL-31.
5	4_T	Route 0 into chain-address field of MBR 368 (i.e. apply conditioning signal through line 1394 (FIG. 5E) to gates 418). Set memory to write status (i.e. apply signal through line 1306 (FIG. 5F) to memory-status flip-flop 1304).	Switch SS-6 to its ON state	CL-6.
6	5_T	Perform memory cycle (i.e. apply a signal to line 1302 (FIG. 5F)).	Switch on SS-7	CL-7.
7	6_T	Reset MBR 368 (FIG. 5F) to blank (i.e. apply a signal to line 1314).	Switch SS-8 to its ON state	CL-8.
8	7_T	Route a 0 to chain-address field of MBR 368 (i.e. apply a conditioning signal through line 1394 (FIG. 5E) to gates 418).	Switch on SS-29	CL-29.
20	3_T	Increment TAC 386 by 1 (i.e. apply a signal to line 1228 (FIG. 5D)). Put 1 into level-termination field of *l/w* counter equals seven or if EOW flip-flop 640 is in its ON state (i.e. condition AND gates 651 (FIG. 5E) and 786).	Switch on SS-21	CL-21.
21	20_T	Transfer TAC to MAR and transfer MAR to PAR (i.e. apply a conditioning signal to line 1236 (FIG. 5D)). Set memory to write status (i.e. apply signal to line 1306 (FIG. 5F) to switch memory-status flip-flop 1304 to write side).	Switch on SS-22	CL-22.
22	21_T	Do memory cycle (i.e. apply a signal to line 1302 (FIG. 5F). Set TE flip-flop 408 to its ON side (i.e. apply a signal to line 826 (FIG. 5A)).	If used flip-flop 374 is in its ON state, switch SS-23 to its ON state (i.e. condition AND gate 810 (FIG. 5B)). If used flip-flop is in its OFF state and *l/w* counter not equal to seven, switch SS-0*b* to its ON state (i.e. condition AND gate 792). If used flip-flop in OFF state and *l/w* counter equal to 7, switch SS-32 to its ON state (i.e. condition AND gate 784).	CL-23 or CL-0*b* or CL-32.
23	22_T	Transfer PAR 390 to MAR 372 and transfer MAR 372 to PAR 390 (i.e. apply a conditioning signal through line 1444 (FIG. 5B) to gates 1448, etc.). Switch used flip-flop 374 to its OFF state (i.e. apply a signal to line 804 (FIG. 5B)). Set memory to its read status (i.e. apply a signal through line 1308 to the OFF side of memory-status flip-flop 1304).	Switch on SS-24	CL-24.
24	23_T	Do a memory read cycle (i.e. apply a signal to line 1302 (FIG. 5F)).	Switch SS-25 to its ON state	CL-25.
25	24_T	Transfer contents of next-character-number field of MBR 368 to next-character-number register 394 (i.e. apply a conditioning signal through line 1350 (FIG. 5F) to gates 1348). Transfer TAC 386 to chain-address field of MBR (i.e. apply conditioning signal through line 1238 (FIG. 5D) to gates 1234). Set memory to write status (i.e. apply signal through line 1306 to ON side of memory-status flip-flop 1304).	Switch on SS-26	CL-26.
26	25_T	Do memory cycle (i.e. apply signal to line 1302 (FIG. 5F) to write chain-address into table).	Switch on SS-27	CL-27.
27	26_T	Reset MBR 368 to blank (i.e. apply a signal to line 1314 (FIG. 5F)).	Switch SS-28 to its ON state	CL-28.
28	27_T	Transfer TAC to CA field of MBR 368 (i.e. apply conditioning signal through line 1238 (FIG. 5D) to gates 1234).	Switch on SS-29	CL-29.
29	8_T or 28_T	Transfer next-character-number register 394 to next-character-number field of MBR 368 (i.e. apply conditioning signal through line 1358 (FIG. 5F) to gates 1356).	Switch on SS-30	CL-30.
30	29_T	Transfer SAC 392 to MAR 372 and transfer MAR to PAR (i.e. apply conditioning signal through line 1298 (FIG. 5D) to gates 1272).	Switch on SS-30*a*	CL-30*a*.
30*a*	30_T	Increment SAC 392 by one (i.e. apply a signal to line 1268 (FIG. 5D)).	Switch SS-31 to its ON state	CL-31.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
31	30a_T or 4$_T$	Do memory write cycle (i.e. apply a signal to line 1302 (FIG. 5F)).	If *l/w* counter equal 7 or end-of-word flip-flop 640 is in ON state, switch SS-32 to its ON state (i.e. apply conditioning signal to AND gate 788 or to AND gate 646). If *l/w* counter not equal to 7 and end-of-word flip-flop in OFF state, switch SS-0*b* to its ON state (i.e. apply conditioning signal to AND gate 654 (FIG. 5B)).	CL-32 or CL-0*b*.
32	22$_T$ or 31$_T$ or 33b_T	Transfer next-character-address counter 420 to MAR 372 and transfer MAR to PAR (i.e. apply a conditioning signal through line 1200 (FIG. 5D) to gates 1196, etc.). Set memory to read status (i.e. apply signal through line 1308 (FIG. 5F) to OFF side of memory-status flip-flop 1304).	Switch SS-33 to its ON state	CL-33.
33	32$_T$	Do memory read cycle (i.e. apply a signal to line 1302 (FIG. 5F).	Switch SS-33*a* to its ON state	CL-33*a*.
33*a*	33$_T$	Idle	If next-character field of MBR is not blank, switch SS-33*c* to its ON state (i.e. condition AND gate 1378 (FIG. 5F). If next character field of MBR is blank, switch SS-33*u* to its ON state (i.e. condition AND gate 1374).	CL-33*c* or CL-33*u*.
33*b*	33a_T or 33c_T	Increment next-character-address counter 420 by one (i.e. apply a signal to line 1192 (FIG. 5D)).	Switch SS-32 to its ON state	CL-32.
33*c*	33a_T	Transfer next-character-number field of MBR 368 to next-character-number register 394 (i.e. apply conditioning signal through line 1350 (FIG. 5F) to gates 1348).	Switch SS-33*d* to its ON state	CL-33*d*.
33*d*	33c_T	Transfer SAC 392 to compare-register 446 (i.e. apply conditioning signal through line 1276 (FIG. 5D) to gates 1274).	Switch SS-33*e* to its ON state	CL-33*e*.
33*e*	33d_T	Transfer supplementary-starting-point register 1210 to SAC 392 (i.e. apply conditioning signal through line 1260 (FIG. 5D) to gates 1258).	Switch SS-33*f* to its ON state	CL-33*f*.
33*f*	33e_T or 33p_T	Transfer SAC 392 to MAR 372 and transfer MAR to PAR (i.e. apply conditioning signal through line 1298 (FIG. 5D) to gates 1272, etc.). Set memory to read status (i.e. apply signal through line 1308 (FIG. 5F) to switch memory-status flip-flop 1304 to OFF state).	Switch SS-33*g* to its ON state	CL-33*g*.
33*g*	33f_T	Do memory read cycle (i.e. apply signal to line 1302 (FIG. 5F)).	Switch on SS-33*h*	CL-33*h*.
33*h*	33g_T	Idle	If character numbers applied to compare-circuit 442 are equal, switch SS-33*i* to its ON state (i.e. condition AND gate 1364 (FIG. 5F)). If character numbers applied to compare-circuit 442 are not equal, switch SS-33*o* to its ON state (i.e. condition AND gate 1368).	CL-33*i* or CL-33*o*.
33*i*	33h_T	Transfer CA field of MBR 368 to chain-address register 460 (i.e. apply conditioning signal through line 1406 (FIG. 5E) to gates 1402).	Switch SS-33*j* to its ON state	CL-33*j*.
33*j*	33i_T	Transfer next-character-address counter 420 to MAR 372 and transfer MAR to PAR (i.e. apply conditioning signal through line 1200 (FIG. 5D) to gates 1196, etc.).	Switch SS-33*k* to its ON state	CL-33*k*.
33*k*	33j_T	Do memory read cycle (i.e. apply signal to line 1302 (FIG. 5F)).	Switch SS-33*l* to its ON state	Cl-33*l*.
33*l*	33k_T	Transfer chain-address register 460 to CA field of MBR 368 (i.e. apply conditioning signal through line 1400 (FIG. 5E) to gates 1396). Set memory to write status (i.e. apply signal through line 1306 (FIG. 5F) to set memory-status flip-flop 1304 to write state).	Switch SS-33*m* to its ON state	CL-33*m*.
33*m*	33l_T	Do memory write cycle (i.e. apply signal to line 1302 (FIG. 5F)).	Switch SS-33*n* to its ON state	CL-33*n*.
33*n*	33m_T	Transfer register 446 to SAC 392 (i.e. apply conditioning signal through line 1284 (FIG. 5D) to gates 1282).	Switch SS-33*b* to its ON state	CL-33*b*.
33*o*	33h_T	Idle	If addresses applied to compare unit 448 are not equal, switch SS-33*p* to its ON state (i.e. apply conditioning signal to AND gate 1294 (FIG. 5D)). If addresses applied to compare circuit 448 are equal, switch SS-33*q* to its ON state (i.e. apply conditioning signal to AND gate 1290)).	CL-33*p* or CL-33*q*
33*p*	33o_T	Increment SAC 392 by one (i.e. apply signal to line 1268 (FIG. 5D)).	Switch SS-33*f* to its ON state	CL-33*f*.
33*q*	33o_T	Transfer next-character-address counter 420 to MAR 372 and transfer MAR to PAR (i.e. apply conditioning signal through line 1200 (FIG. 5D) to gates 1196, etc.).	Switch SS-33*r* to its ON state	CL-33*r*.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
33*r*	33*q*T	Increment next-character-address counter by one (i.e. apply signal to line 1192 (FIG. 5D)). Set memory to read status (i.e. apply signal through line 1308 (FIG. 5F) to switch memory-status flip-flop 1304 to its OFF state).	Switch SS-33*s* to its ON state	CL-33*s*.
33*s*	33*r*T	Do memory read cycle (i.e. apply signal to line 1302 (FIG. 5F)).	Switch SS-33*t* to its ON state	CL-33*t*.
33*t*	33*s*T	Transfer next-character field of MBR 368 to *l/w* counter 321 (i.e. apply conditioning signal through line 774 (FIG. 5F) to gates 772)). Switch TE flip-flop 408 to its OFF state (i.e. apply a signal to line 824 (FIG. 5A)). Switch used flip-flop 374 to its ON state (i.e. apply a signal to line 806 (FIG. 5B)). Switch EOW flip-flop 640 to its OFF state (i.e. apply a signal to line 642 (FIG. 5B)).	Switch SS-33*w* to its ON state	CL-33*w*.
33*u*	33*a*T	Write a 2 into CA field of MBR 368 (i.e. apply a conditioning signal through line 1388 (FIG. 5E) to gates 434). Set memory to write status (i.e. apply a signal through line 1306 (FIG. 5F) to ON side of memory-status flip-flop 1304).	Switch SS-33*v* to its ON state	CL-33*v*.
33*v*	33*u*T	Do memory write cycle (i.e. apply signal to line 1302 (FIG. 5F)).	If inputs to compare-circuit 1198 (i.e. next-character-address counter 420 and TAC 386) are not equal, switch SS-33*b* to its ON state (i.e. apply conditioning signal to AND gate 1252 (FIG. 5D)). If addresses applied to compare-circuit 1198 are equal, switch SS-85 to its ON state (i.e. condition AND gate 1248).	CL-33*b* or CL-85.
33*w*	33*t*T	Reset output-control ring 612 to its first position (i.e. apply a signal to line 615 (FIG. 5A)). Reset next-character-number counter 748 to a count of one (i.e. apply a signal to line 750 (FIG. 5D)).	Switch SS-33*x* to its ON state	CL-33*x*.
33*x*	33*w*T or 33*y*T	Increment next-character-number counter 748 by 1 (i.e. apply a signal to line 752 (FIG. 5D)). Increment ouput-control ring 612 by 1 (i.e. apply a signal to line 620 (FIG. 5A)).	Switch SS-33*y* to its ON state	CL-33*y*.
33*y*	33*x*T	Decrement *l/w* counter by 1 (i.e. apply a signal to line 776 (FIG. 5C)).	If *l/w* counter is not equal to 1, switch SS-33*x* to its ON state (i.e. condition AND gate 840 (FIG. 5C)). If *l/w* counter equal to 1, switch SS-33*z* to its ON state (i.e. condition AND gate 836).	CL-33*x* or CL-33*z*.
33*z*	33*y*T	Reset argument register 360 to blank (i.e. apply a signal to line 710 (FIG. 5B)). Reset input-control ring 704 to its first position (i.e. apply a signal to line 706 (FIG, 5B)).	Switch SS-1 to its ON state	CL-1.
50	Overflow did not occur on 1st or 2d character and word more than 4 characters long. (Output on line 324 from AND gate 322 (FIG. 5C)).	Reset output-control ring 612 to first position (i.e. apply a signal to line 616 (FIG. 5A).	Switch on SS-52	CL-52.
51	52*r* or 79*r*	Increment output-control ring 612 by 1 (i.e. apply signal to line 620 (FIG. 5A)).	Switch on SS-51*a*	CL-51*a*.
51*a*	51*r* or 52*r* or 77*r* or 54*r*	Increment output-control ring 612 by 1 (i.e. apply signal to line 620 (FIG. 5A)).	Switch on SS-52	CL-52.
52	51*a*r or 50*r* or 64*r* or 56*r*	If letter is a, o, or u, add 1 to vowel-group counter 1018 (i.e. condition AND gate 1014 (FIG. 5I)).	If letter is a, o, or u, and vowel count is <2, switch on SS-51 (i.e. condition AND gate 1016 (FIG. 5I)). If letter is not a vowel, a y, or a blank, switch or on SS-51*a* (i.e. condition AND gate 986 (FIG. 5G). If letter is a y, switch on SS-78 (i.e. condition AND gate 960). If #VGC has count of 2, go to step 210 (FIG. 1) (i.e. condition AND gate 1056). If letter is an i, switch on SS-75 (i.e. condition AND gate 948). If letter is an e, switch on SS-53 (i.e. condition AND gate 946). If character is a blank and count in #VGC is <2, word is not hyphenable (i.e. condition AND gate 962).	CL-51 or CL-51*a* or CL-78 or CL-115 (i.e. step 210) CL-75 or CL-53 or end of operation (i.e. word is not hyphenable).
53	52*r*	Increment ouput-control ring by 1 (i.e. apply signal to line 620 (FIG. 5A)).	Switch on SS-54	CL-54.
54	53*r*	If character following the e is not a, d, s, or blank, increment #VGC by 1 (i.e. condition AND gate 998 (FIG. 5G)).	If character is not d, s, or blank and count in #VGC<2, switch on SS-51*a* (i.e. condition AND gate 996 (FIG. 5G)). If character is a d, switch on SS-63 (i.e. condition AND gate 942). If #VGC has count of 2, go to step 210 (i.e. condition AND gate 1054) (FIG. 5I). If character is an s, switch on SS-55 (i.e. condition AND gate 958). If character following e is a blank, switch on SS-71 (i.e. condition AND gate 964).	CL-51*a* or CL-63 or CL-115 (i.e. step 210) or CL-55 or CL-71.
55	54*r*	Increment output-control ring 612 by 1 (i.e. apply signal to line 620 (FIG. 5A).	Switch on SS-56	CL-56.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
56	55τ	If s following e is not last letter, add 1 to #VGC (i.e. condition AND gate 972 (FIG. 5G)).	If #VGC has count of 2, go to step 210 (i.e. condition AND gate 1052 (FIG. 5I)). If #VGC has count of <2 and character following s is not a blank, switch on SS-52 (i.e. condition AND gate 974). If #VGC has count of <2 and character following s is a blank, switch on SS-57 (i.e. condition AND gate 965).	CL-115 (i.e. step 210) or CL-52 or CL-57.
57	56τ	Decrement output-control ring 612 by 1 (i.e. apply a signal to line 618 (FIG. 5A)).	Switch on SS-58	CL-58.
58	57τ	Decrement output-control ring 612 by 1 (i.e. apply a signal to line 618 (FIG. 5A)).	Switch on SS-59	CL-59.
59	58τ	Decrement output-control ring 612 by 1 (i.e. apply a signal to line 618 (FIG. 5A)).	Switch on SS-60	CL-60.
60	59τ	If character preceding e is a, c, g, h, s, or x, and 1 to #VGC (i.e. condition AND gate 1002 (FIG. 5G)).	If #VGC has count of 2, go to next step (i.e. condition AND gate 1050 (FIG. 5I)). If #VGC has count of <2, word is not hyphenable (i.e. condition AND gate 1034).	CL-115 (i.e. step 210) or end of operation (i.e. word not hyphenable.)
63	54τ	Increment output-control ring 612 by 1 (i.e. apply signal to line 620 (FIG. 5A)).	Switch on SS-64	CL-64.
64	63τ	If character not a blank, add one to #VGC (i.e. condition AND gate 976 (FIG. 5G)).	If #VGC is equal to two, go to step 210 (i.e. condition AND gate 1048 (FIG. 5I)). If count in #VGC is less than two and character is not a blank, switch on SS-52 (i.e. condition AND gate 978 (FIG. 5G)). If count in #VGC is less than two and character is a blank, switch on SS-65 (i.e. condition AND gate 966).	CL-115 (i.e. step 210) or CL-52 or CL-65.
65	64τ	Decrement output-control ring 612 by 1 (i.e. apply signal to line 618 (FIG. 5A)).	Switch on SS-66	CL-66.
66	65τ	Decrement output-control ring 612 by 1 (i.e. apply signal to line 618 (FIG. 5A)).	Switch on SS-67	CL-67.
67	66τ	Decrement output-control ring 612 by 1 (i.e. apply signal to line 618 (FIG. 5A)).	Switch on SS-68	CL-68.
68	67τ	If character is a d or a t, add 1 to #VGC (i.e. condition AND gate 1006 (FIG. 5II)).	If count in #VGC is equal to two, go to step 210 (i.e. condition AND gate 1046 (FIG. 5J)). If count in #VGC is less than two, word not hyphenable (i.e. condition AND gate 1036 (FIG. 5H)).	CL-115 (i.e. step 210) or end of operation (i.e. word not hyphenable).
71	54τ	Decrement output-control ring 612 by 1 (i.e. apply a signal to line 618 (FIG. 5A)).	Switch on SS-72	CL-72.
72	71τ	Decrement output-control ring 612 by 1 (i.e. apply a signal to line 618 (FIG. 5A)).	If character is not an l, word is not hyphenable (i.e. condition AND gate 956 (FIG. 5J)). If character is an l, switch on SS-73 (i.e. condition AND gate 950).	End of operation (i.e. word not hyphenable) or CL-73.
73	72τ	Decrement output-control ring 612 by one (i.e. apply signal to line 618 (FIG. 5A)).	Switch on SS-74	CL-74.
74	73τ	If character is a stop character (i.e. b, d, g, j, k, p, or t) add 1 to #VGC (i.e. condition AND gate 1010 (FIG. 5J)).	If count in #VGC is less than two, word not hyphenable (i.e. condition AND gate 1032 (FIG. 5J)). If count in #VGC equal to two, go to step 210 (i.e. condition AND gate 1044).	End of operation (i.e. word not hyphenable.) or CL-115 (i.e. step 210).
75	52τ	Add 1 to #VGC (i.e. apply a signal to OR gate 1024 (FIG. 5II)).	If count in #VGC is less than two, switch on SS-76 (i.e. condition AND gate 1030 (FIG. 5J)). If count in #VGC equal to two, switch on SS-115 (i.e. condition AND gate 1042).	CL-76 or CL-115 (i.e. step 210).
76	75τ	Increment output-control ring 612 by 1 (i.e. apply a signal to line 620 (FIG. 5A)).	Switch on SS-77	CL-77.
77	76τ	Idle	If character is an a, switch on SS-115 (i.e. condition AND gate 934 (FIG. 5J)). If character is not an a, switch on SS-51a (i.e. condition AND gate 932).	CL-115 (i.e. step 210) or CL-51a.
78	52τ	Decrement output-control ring 612 by 1 (i.e. apply a signal to line 618 (FIG. 5A)).	Switch on SS-79	CL-79.
79	78τ	If character is not a vowel, a y, or a blank, add 1 to #VGC (i.e. condition AND gate 988 (FIG. 5H)).	If count in #VGC equal to two, switch on SS-115 (i.e. condition AND gate 1040 (FIG. 5J)). If count in #VGC less than two, switch on SS-51 (i.e. condition AND gate 1038 (FIG. 5H)).	CL-115 (i.e. step 210) or CL-51a.
85	33υτ	Reset syllable registers, address registers, number-of-syllable registers, and ratio registers in word registers 488 (i.e. apply a signal to line 1861 (FIG. 5R)). Reset frequency registers 1601-160N (i.e. apply a signal to line 1600 (FIG. 5M)). Reset word-select counter 528 to a count of 1 (i.e. apply a signal to line 1798 (FIG. 5P)).	Switch SS-85a to its ON state	CL-85a.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
85*a*	85*τ* or 105*τ*	Transfer address in starting-point register 1208 to TAC 386 (i.e. apply a signal through line 1220 (FIG. 5D) to condition gates 1218). Reset column-select counter 508 to its first position (i.e. apply a signal to line 1530 (FIG. 5M)). Reset number-of-syllable counter 1670 to a count of 1 (i.e. apply a signal to line 1672 (FIG. 5M)). Reset intermediate store 1736 to a count of 0 (i.e. apply a signal to line 1758 (FIG. 5N).) Reset flip-flop 487 to its OFF state (i.e. apply signal to line 1424 (FIG. 5E)).	Switch SS–85*b* to its ON state	CL–85*b*.
85*b*	85*aτ*	Increment TAC 386 by 1 (i.e. apply a signal to line 1228 (FIG. 5D)).	Switch SS–86 to its ON state	CL–86.
86	85*bτ* or 87*bτ* or 88*τ*	Transfer TAC 386 to MAR 372 and transfer MAR to PAR (i.e. apply a signal through line 1236 (FIG. 5D) to condition gates 1232, etc.). Set memory to read status (i.e. apply a signal to line 1308 (FIG. 5F) to switch memory-status flip-flop 1304 to its OFF state).	Switch SS–87 to its ON state	CL–87.
87	86*τ* or 88*aτ*	Perform memory cycle, read entry from core storage into MBR 368 (apply a signal to line 1302 (FIG. 5F)).	Switch on SS–87*a*	CL–87*a*.
87*a*	87*τ*	If used bit in MBR 368 is present, transfer MAR to TAC (i.e. apply a conditioning signal through line 1502 (FIG. 5D) to AND gate 1500).	If used bit equals 0, switch SS–88 to its ON state (i.e. apply conditioning signal to AND gate 1494 (FIG. 5F)). If used bit equals 1, switch SS–87*b* to its ON state (i.e. apply conditioning signal to AND gate 1498).	CL–88 or CL–87*b*.
87*b*	87*aτ*	Increment TAC 386 by 1 (i.e. apply a signal to line 1228 (FIG. 5D)).	Switch SS–86 to its ON state	CL–86.
88	87*aτ*	Increment TAC 386 by 1 (i.e. apply a signal to line 1228 (FIG. 5D)). If chain address is 0 and there is mark in LTM field, switch flip-flop 487 to its ON state (i.e. apply conditioning signal through line 1432 (FIG. 5E) to AND gate 485). If chain address not 0 and not 2, transfer chain address in MBR to MAR 372. If chain address not 0, perform above step and the following steps: (a) Transfer contents of syllable field to MBR and contents of MAR 372 to selected-syllable and address registers in word registers 488. (b) Transfer frequency field of MBR to selected-frequency register 1601–160N. (c) Step number-of-syllable counter 1670 by 1 (i.e. apply conditioning signal through line 1432 to AND gate 1428).	If chain address not 2 and not 0, switch SS–88*a* to its ON state (i.e. apply conditioning signal to line 1418 (FIG. 5E)). If chain-address field contains a 2, switch SS–89 to its ON state (i.e. apply conditioning signal to AND gate 1414)). If chain-addressed field contains a 0, and LTM field does not contain a mark, switch SS–86 to its ON state (i.e. apply conditioning signal to AND gate 1343). If chain-address field contains a 0 and there is a mark in the LTM field, swtich SS–100 to its ON state (i.e. apply conditioning signal to AND gate 1429).	CL–88*a* or CL–89 or CL–86 or CL–100.
88*a*	88*τ*	Step column-select counter 508 by (i.e. apply signal to line 1532 (FIG. 5M)).	Switch SS–87 to its ON state	CL–87.
89	88*τ*	Form sum of frequencies in frequency registers 1601–160N (i.e. apply conditioning signal through line 1620 (FIG. 5M) to gates 1621–162N). Reset column-select counter 508 to first position (i.e. apply a signal to line 1530 (FIG. 5M)).	Switch SS–90 to its ON state	CL–90.
90	89*τ*	Transfer contents of adder 1640 to dividend input of divider 1700 and transfer contents of number-of-syllable counter 1670 to devisor input of divider 1700 (i.e. apply conditioning signal through line 1666 (FIG. 5M) to gates 1662 and 1664).	Switch SS–91 to its ON state	CL–91.
91	90*τ* or 94*τ*	Transfer contents of divider 1700 to 1 input of subtractor 1718 and transfer contents of selected frequency register 1601–160N to other imput of subtractor 1718 (i.e. apply conditioning signal through line 1712 (FIG. 5M) to gates 1704 and 1710).	Switch SS–92 to its ON state	CL–92.
92	91*τ*	Apply output of subtractor 1718 to squarer 1728 (i.e. apply conditioning signal through line 1724 (FIG. 5M) to gates 1722).	Switch SS–93 to its ON state	CL–93.
93	92*τ*	Apply contents of squarer 1728 to 1 input of adder 1746 and apply contents of intermediate store 1736 to other input of adder 1746 (i.e. apply conditioning signal through line 1740 (FIG. 5N) to gates 1732 and 1738).	If counts applied to compare-circuit 1538 are not equal (i.e. if the count in column-select counter 508 is not equal to the count in number-of-syllable counter 1670) switch SS–94 to its ON state (i.e. apply a conditioning signal to AND gate 1668 (FIG. 5M)). If the counts applied to compare-circuit 1538 are equal, switch SS–95 to its ON state (i.e. apply a conditioning signal to AND gate 1684).	CL–94 or CL–95.
94	93*τ*	Transfer contents of adder 1746 to intermediate store 1736 (i.e. apply conditioning signal through line 1754 (FIG. 5N) to gates 1750). Set column-select counter 508 by 1 (i.e. apply signal to line 1532 (FIG. 5M)).	Switch SS–91 to its ON state	CL–91.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
95	93$_T$	Transfer contents of adder 1746 to dividend input of divider 1766 and transfer contents of number-of-syllable counter 1670 to devisor input of divider 1766 (i.e. apply conditioning signal through line 1760 (FIG. 5N) to gates 1752 and 1680).	Switch SS-96 to its ON state	CL-96.
96	95$_T$	Transfer contents of divider 1766 to square rooter 1776 (i.e. apply conditioning signal through line 1772 (FIG. 5N) to gates 1770).	Switch SS-97 to its ON state	CL-97.
97	96$_T$	Transfer contents of square rooter 1776 to divisor input of divider 1788. Transfer contents of divider 1700 to dividend input of divider 1788 (i.e. apply conditioning signal through line 1782 (FIG. 5N) to gates 1780 and 1706).	Switch SS-98 to its ON state	CL-98.
98	97$_T$	Transfer contents of divider 1788 to ratio register of selected word in word register 488 (i.e. apply conditioning signal through line 1794 (FIG. 5N) to gates 1792). Transfer contents of number-of-syllable counter 1670 to number-of-syllable register of selected word in word register 488 (i.e. apply conditioning signal through line 1850 (FIG. 5N) to gates 1678).	Switch SS-99 to its ON state	CL-99.
99	98$_T$	Reset frequency registers 1601-160N (i.e. apply signal to line 1600 (FIG. 5M)).	Switch SS-100 to its ON state	CL-100.
100	99$_T$ or 103$_T$	Decrement column-select counter 508 by 1 (i.e. apply signal to line 1534 (FIG. 5M)).	If column-select counter not equal to 0, switch SS-101 to its ON state (i.e. apply conditioning signal to AND gate 1548 (FIG. 5M)). If column-select counter has count of 0, switch SS-107 to its ON state (i.e. apply conditioning signal to AND gate 1546).	CL-101 or CL-107.
101	100$_T$	Transfer contents of selected-address register in word registers 488 to MAR 372 (i.e. apply conditioning signal through line 1968 (FIG. 5T) to gates 1468). Transfer contents of MAR to PAR (i.e. apply signal to line 1446 (FIG. 5D)). Set memory to read status (i.e. apply signal through line 1308 (FIG. 5F) to switch memory-status flip-flop 1304 to its OFF state).	Switch SS-102 to its ON state	CL-102.
102	101$_T$	Do a memory read cycle (i.e. apply a signal to line 1302 (FIG. 5F)).	Switch SS-103 to its ON state	CL-103.
103	102$_T$	Set memory to write status (i.e. apply a signal through line 1306 (FIG. 5F) to switch memory-status flip-flop 1304 to its ON state). Put 1 in used field of MBR 368 (i.e. apply conditioning signal through line 1488 (FIG. 5F) to gates 1482).	If LTM field in MBR contains a mark, switch SS-100 to its ON state (i.e. apply a conditioning signal to AND gate 1344 (FIG. 5F)). If LTM field does contain a mark, switch SS-104 to its ON state (i.e. apply a conditioning signal to AND gate 1346).	CL-100 or CL-104.
104	103$_T$	Do a memory write cycle (i.e. apply a signal to line 1302 (FIG. 5F)).	If flip-flop 487 is in its OFF state, switch SS-105 to its ON state (i.e. apply conditioning signal to AND gate 1435 (FIG. 5E)). If flip-flop 487 is in its ON state, switch SS-85a to its ON state (i.e. apply conditioning signal to AND gate 1431)).	CL-105 or CL-85a.
105	104$_T$	Increment word-select counter 528 by 1 (i.e. apply a signal to line 1800 (FIG. 5P)).	Switch SS-85a to its ON state	CL-85a.
107	100$_T$	Reset word-select counter 528 to its first position (i.e. apply a signal to line 1798 (FIG. 5P)). Set preferred-word register 544 to 1 (i.e. apply a signal to line 2022 (FIG. 5P)). Set first-time-through flip-flop 535 to its ON state (i.e. apply a signal to line 2004).	Switch SS-108 to its ON state	CL-108.
108	107$_T$ or 111$_T$	If selected ratio field is blank, transfer contents of syllable registers of preferred word to coded-word-hyphenation registers 1911-191N of coded-word-hyphenation registers 564 (i.e. apply a conditioning signal through line 2008 (FIG. 5P) to AND gate 1986).	If the contents of the selected-ratio field are greater than 1 and first-time-through flip-flop 535 is in its OFF state, switch SS-109 to its ON state (i.e. apply conditioning signal to AND gate 1994 (FIG. 5P)). If the ratio in the selected-ratio register is greater than 1 and first-time-through flip-flop 535 is in its ON state, switch SS-110 to its ON state (i.e. apply a conditioning signal to line 1996). If the ratio in the selected-ratio field is equal to or less than 1 but is not a blank, switch SS-111 to its ON state (i.e. apply a conditioning signal to AND gate 1982). If the selected-ratio register contains a blank, switch SS-120 to its ON state (i.e. apply a conditioning signal to AND gate 1988).	CL-109 or CL-110 or CL-111 or CL-120.
109	108$_T$	Apply contents of number-of-syllable register of word indicated in preferred-word register and of word indicated in word-select counter to compare-circuit 1880 (i.e. apply a conditioning signal through line 2034 (FIG. 5P) to AND gates 2026 and 2030).	If word in word-select register has less syllables, switch SS-110 to its ON state (i.e. apply conditioning signal to AND gate 1892 (FIG. 5P)). If word indicated by preferred-word register has fewer syllables, switch SS-111 to its ON state (i.e. apply conditioning signals to AND gate 1884). If words applied to compare-register 1880 have same number of syllables, switch SS-112 to its ON state (i.e. apply conditioning signal to AND gate 1888).	CL-110 or CL-111 or CL-112.

Single shot No.	Turn-on condition	Functions performed while on	Functions performed when single-shot times out	Next clock time
110	108_T or 109_T or 112_T	Switch first-time-through flip-flop 535 to its OFF state (i.e. apply a signal to line 2002 (FIG. 5P)). Transfer contents of word-select counter 528 to preferred-word register 544 (i.e. apply conditioning signal through line 2020 (FIG. 5P) to gates 2018).	Switch SS-111 to its ON state	CL-111.
111	108_T or 109_T or 110_T or 112_T.	Increment word-select counter 528 by 1 (i.e. apply a signal to line 1800 (FIG. 5P)).	Switch SS-108 to its ON state	CL-108.
112	109_T	Transfer contents of ratio register of word indicated by preferred-word register to 1 input of compare-circuit 2058 and transfer contents in ratio field of word indicated by word-select counter to other input of compare-circuit 2058 (i.e. apply conditioning signal through line 2036 (FIG. 5P) to AND gates 2028 and 2032).	If ratio for word indicated by word-select counter is greater, switch SS-110 to its ON state (i.e. apply a conditioning signal to AND gate 2062 (FIG. 5Q)). If ratio of word indicated by preferred-word register is greater, switch SS-111 to its ON state (i.e. apply a conditioning signal to AND gate 2066).	CL-110 or CL-111
115	A signal on output line 326 from OR gate 1142 (i.e. 52_T or 54_T or 56_T or 60_T or 64_T or 68_T or 74_T or 75_T or 77_T or 79_T).	Transfer contents of input-buffer register 302 into argument register 340 (i.e. condition AND gates 660 (FIG. 5A)). Set match indicators in associative memory 338 to match side (i.e. apply a signal to line 668). Set flip-flop 670 to its OFF side (apply signal to line 668 (FIG. 5A)).	Switch SS-116 to its ON state	CL-116.
116	115_T	Perform association between contents of argument register 340 and contents of portion 341 of associative memory 338 (i.e. apply a signal to line 666 (FIG. 5A)).	Switch on SS-117	CL-117.
117	116_T	Test contents of match indicators in associative memory 338 (i.e. apply signal to line 672 (FIG. 5A) and store proper hyphenation pattern in output register 346 (FIG. 5Z) after a match is found. If a match is not found, switch flip-flop 670 to its ON state.	If match found, signal end of operation (i.e. condition AND gate 350 (FIG. 5A)). If match not found, switch on SS-0 (i.e. condition AND gate 354).	End of operation (i.e. proper hyphenation pattern) or CL-0 (i.e. step 212).
120	108_T	Reset output-control ring 612 to its first position (i.e. apply a signal to line 616 (FIG. 5A)). Reset group-control ring 2070 to its first position (i.e. apply a signal to line 2074 (FIG. 5K)). Reset unit-control ring 2072 to its first position (i.e. apply a signal to line 2076 (FIG. 5K)). Reset input-control ring 2188 to its first position (i.e. apply a signal to line 2190 (FIG. 5L)).	Switch SS-121 to its ON state	CL-121.
121	120_T or 121_T or 126_T or 128_T.	If selected character is not a blank, transfer selected character in input-buffer register into selected position of output register 346 (i.e. apply a conditioning signal to AND gate 2150 (FIG. 5L) causing an output signal on line 574 to condition gates 634). If selected character in coded-word-hyphenation register is a blank and character is not first character of a syllable, transfer hyphen from hyphenation generator 572 to selected position of output register 346 (i.e. condition AND gate 2108 (FIG. 5L) to generate an output signal on line 578 which conditions gates 2168).	If selected character is not a blank and unit-control ring is not in its sixth position, switch SS-122 to its ON state (i.e. apply a conditioning signal to AND gate 2116 (FIG. 5L)). If selected character is not a blank and unit-control ring is in its sixth position, switch SS-124 to its ON state (i.e. apply a conditioning signal to AND gate 2110). If selected character is a blank and unit-control ring is not in its first position, switch SS-128 to its ON state (i.e. apply a conditioning signal to AND gate 2106). If selected character is a blank and unit-control, ring is in its first position, switch SS-129 to its ON state (i.e. apply conditioning signal to AND gate 2100).	CL-122 or CL-124 or CL-128 or CL-129.
122	121_T	Increment output-control ring 612 by 1 (i.e. apply a signal to line 620 (FIG. 5A)). Increment input-control ring 2188 by 1 (i.e. apply a signal to line 2192 (FIG. 5L)). Increment unit-control ring 2072 by 1 (i.e. apply a signal to line 2080 (FIG. 5K)).	Switch SS-121 to its ON state	CL-121.
124	121_T	Increment input-control ring 2188 by 1 (i.e. apply a signal to line 2192 (FIG. 5L)). Increment group-control ring 2070 by 1 (i.e. apply a signal to line 2078 (FIG. 5K)).	Switch SS-125 to its ON state	CL-125.
125	124_T	Gate hyphen character into selected position of output register 2188 (i.e. apply conditioning signal through line 580 (FIG. 5K) and OR gate 2164 to gates 2168).	Switch SS-126 to its ON state	CL-126.
126	125_T	Increment output-control ring 612 to 1 (i.e. apply a signal to line 620 (FIG. 5A)). Increment input-control ring 2188 by 1 (i.e. apply a signal to line 2192 (FIG. 5L)). Reset unit-control ring to first position (i.e. apply a signal to line 2076 (FIG. 5K)).	Switch SS-121 to its ON state	CL-121.
128	121_T	Increment input-control ring 2188 by 1 (i.e. apply a signal to line 2192 (FIG. 5L)). Increment group-control ring 2070 by 1 (i.e. apply a signal to line 2078 (FIG. 5K)). Reset unit-control ring 2072 to its first position (i.e. apply a signal to line 2076 (FIG. 5K)).	Switch SS-121 to its ON state	CL-121.
129	121_T	Decrement input-control ring 2188 by 1 (i.e. apply a signal to line 2194 (FIG. 5L)).	Switch SS-130 to its ON state	CL-130.
130	129_T	Transfer blank from generator 2182 to selected position in output register 346 (i.e. apply conditioning signal through line 2184 (FIG. 5L) to gates 2178).	Signal end of operation	End of operation.

What is claimed is:

1. Text composing apparatus for automatically hyphenating a word comprising the combination of:

means for storing an input word;

means for storing a table denoting the relative frequencies with which hyphens occur with respect to characters that constitute or represent letters in a representative large quantity of known words having known occurrence frequencies;

means operable under the control of said input word storing means and said table storing means for making a letter-by-letter analysis of at least a selected portion of the input word to determine which ones, if any, of all the stored hyphen and character patterns are of possible application to the input word;

means operable under the control of said analyzing means for ascertaining which, if any, of the possible hyphen and character patterns represents the most probable hyphenation of the word or selected portion thereof being considered; and

means under control of said ascertaining means for manifesting a particular hyphenation of the word in accordance with said most probable hyphen and character pattern.

2. Text composing apparatus for automatically hyphenating a word comprisnig the combination of:

means for storing an input word;

means for converting the stored input word to a reduced character set;

means for storing the reduced character set;

means under control of said reduced character set storing means for determining the most probable syllable pattern for the word as represented by the reduced character set; and

means under control of said syllable pattern determining means for manifesting a particular hyphenation of the word in accordance with said most probable syllable pattern.

3. A method of operating a data processing system having data storage means and data manifesting means for automatically hyphenating a word, said method comprising the steps of:

operating a portion of said storage means as a reference table for denoting the relative frequencies with which hyphens occur with respect to characters that constitute or represent letters in a representative large quantity of known words having known occurrence frequencies;

operating said system under the control of said reference table to determine from a letter-by-letter analysis of at least a selected portion of an input word which ones, if any, of all the hyphen and character patterns stored in said reference table are of possible application to such input word;

operating a portion of said storage means as a pattern table for storing the possible hyphen and character patterns thus determined;

further operating said system in response to the data stored in said pattern table to determine which one, if any, of the possible hyphen and character patterns represents the most probable hyphenation of the word or selected portion thereof being considered; and

operating said manifesting means to manifest a particular hyphenation of the word in accordance with said most probable hyphen and character pattern.

4. A method of operating a data processing system having data storage means and data manifesting means for automatically hyphenating a word, said method comprising the steps of:

operating said system in accordance with a predetermined character conversion algorithm for converting the word to a reduced character set in which only certain characters are utilized to represent the letters of the actual word;

operating a portion of said storage means as a reference table for denoting the relative frequencies with which syllables represented by said characters occur in a representative large quantity of known words having known occurrence frequencies;

operating said system under the control of said reference table to determine all possible syllable patterns utilizing characters in the reduced character set for said word;

operating a portion of said storage means as a pattern table for storing the possible syllable patterns determined;

further operating said system in response to the data stored in said pattern table to determine which one, if any, of the possible syllable patterns represents the most probable hyphenation of the word; and

operating said manifesting means to manifest said most probable hyphenation of the word.

References Cited

UNITED STATES PATENTS

3,222,650	12/1965	Lazarus	340—172.5
3,248,705	4/1966	Dammann et al.	340—172.5
3,267,431	8/1966	Greenberg et al.	340—172.5 X
3,299,408	1/1967	Wang et al.	340—172.5
3,307,154	2/1967	Garth et al.	340—172.5
3,309,674	3/1967	Lemay	340—172.5
3,328,764	6/1967	Sorensen et al.	340—172.5
3,333,248	7/1967	Greenberg et al.	340—172.5

PAUL J. HENON, Primary Examiner