United States Patent [19]

Brodrueck

[11] **3,845,710**

[45] **Nov. 5, 1974**

[54] **PRINT CONTROL LOGIC CIRCUITRY FOR ON-THE-FLY PRINTERS**

[75] Inventor: **Detlef A. Brodrueck,** Northbrook, Ill.

[73] Assignee: **Teletype Corporation,** Skokie, Ill.

[22] Filed: **Oct. 2, 1973**

[21] Appl. No.: **402,864**

[52] **U.S. Cl.** **101/93.14,** 340/172.5

[51] **Int. Cl.** **B41j 7/08**

[58] **Field of Search** ..... 101/93 C; 340/172.5

[56] **References Cited**

UNITED STATES PATENTS

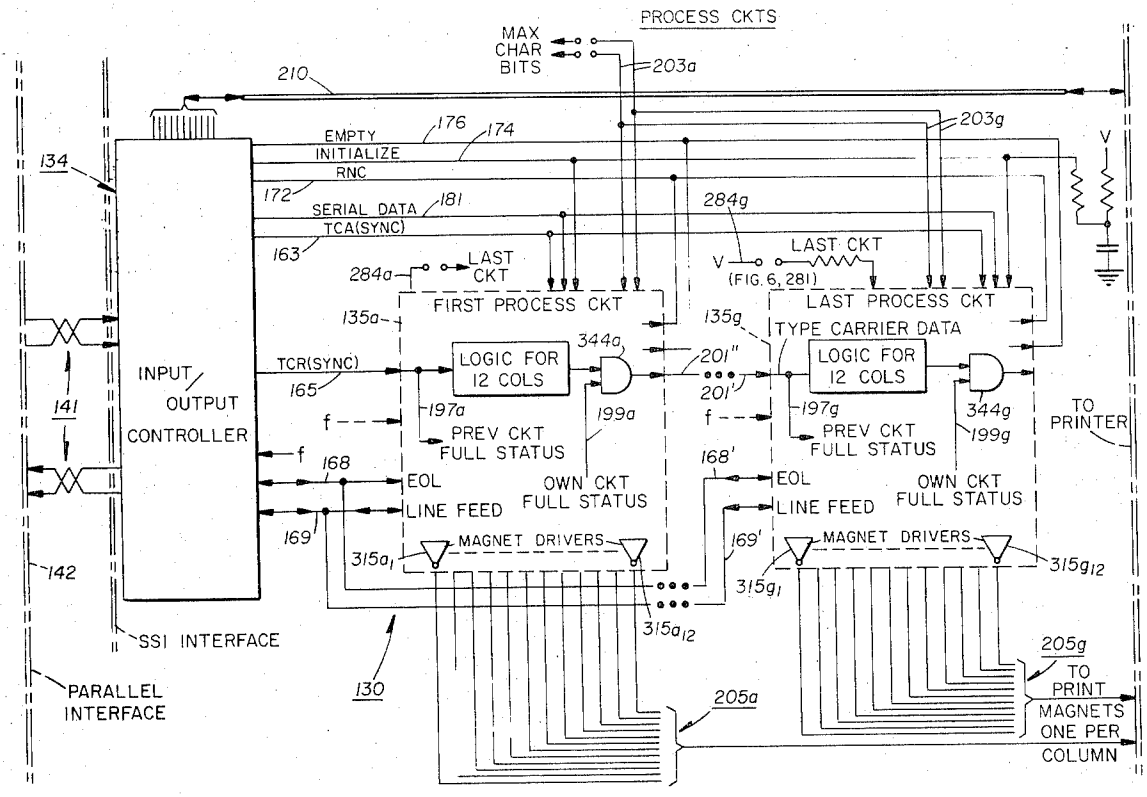
2,993,437	7/1961	Demer et al.	101/93 C
3,289,576	12/1966	Bloom et al.	101/93 C
3,430,210	2/1969	Foure et al.	101/93 C X
3,463,081	8/1969	Levine	101/93 C
3,633,496	1/1972	Kearns	101/930
3,728,688	4/1973	Elmhurst et al.	101/93 C X
3,732,408	5/1973	Bettin	101/93 C X

Primary Examiner—Robert E. Pulfrey
Assistant Examiner—Edward M. Coven
Attorney, Agent, or Firm—K. R. Bergum; J. L. Landis

[57] **ABSTRACT**

Print control logic circuitry for use in high speed "on-the-fly" impact printers, wherein at least one, but preferably a plurality of independent, but interconnected data processing circuit modules, each including a separate free-running memory with logic controlled dual line storage areas, effects sequential character printing in a given line as the print data is being received, and simultaneous therewith allows for the temporary storage of print data for the next adjacent line, when required. Through the utilization of modular construction, each circuit module may advantageously be of identical design, and formed on a single integrated circuit chip, preferably using inexpensive MOSFET'S exhibiting relatively slow switching speeds, with no sacrifice in printer performance.

30 Claims, 15 Drawing Figures



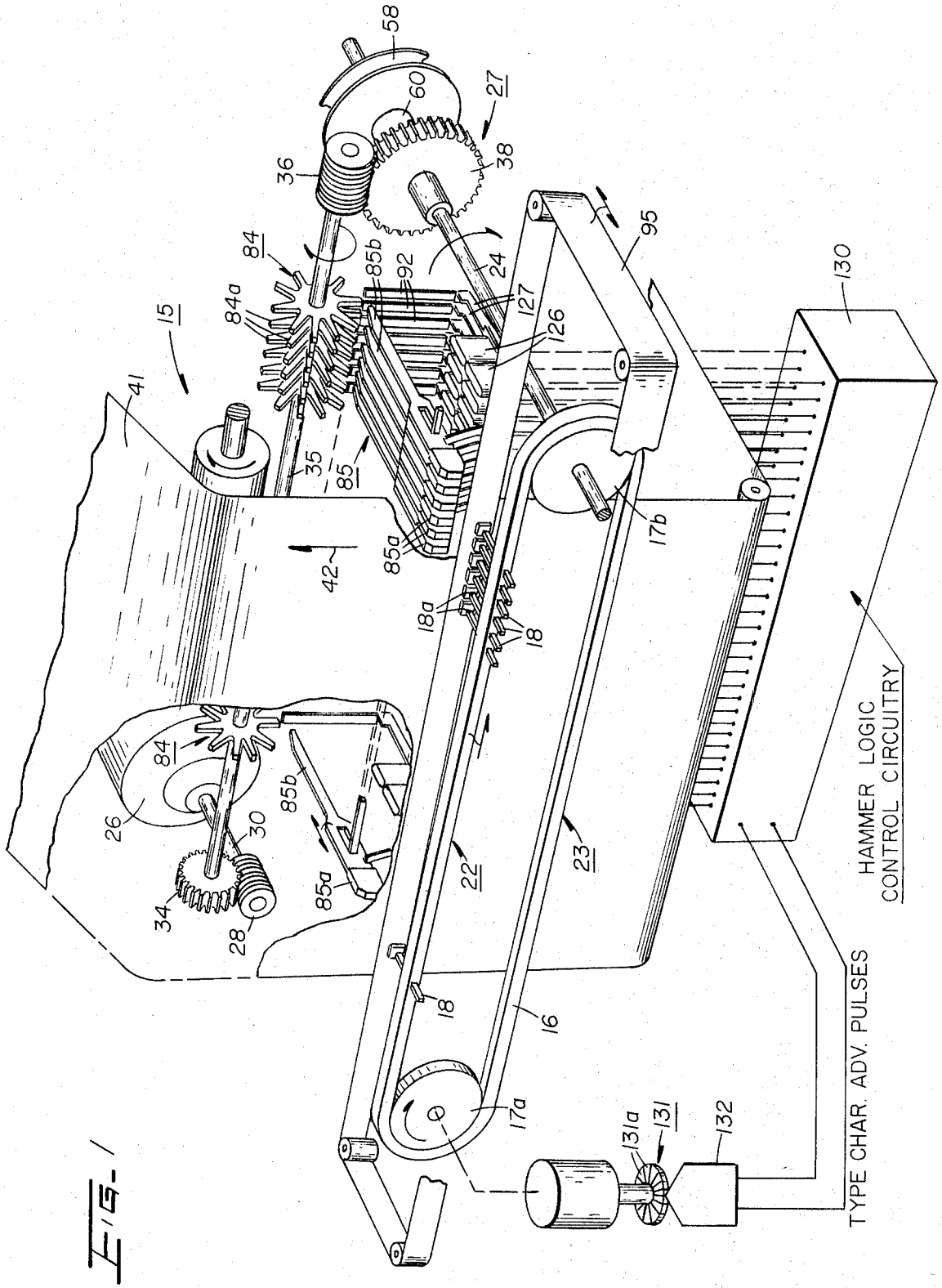

58
60
27
38
36
84
84a
85b
92
24
127
126
95
130
15
41
35
85
85a
42
18a
18
17b
84
26
30
85b
85a
34
28
22
23
HAMMER LOGIC
CONTROL CIRCUITRY
18
16
17a
131a
131
132
TYPE CHAR. ADV. PULSES

FIG. 1

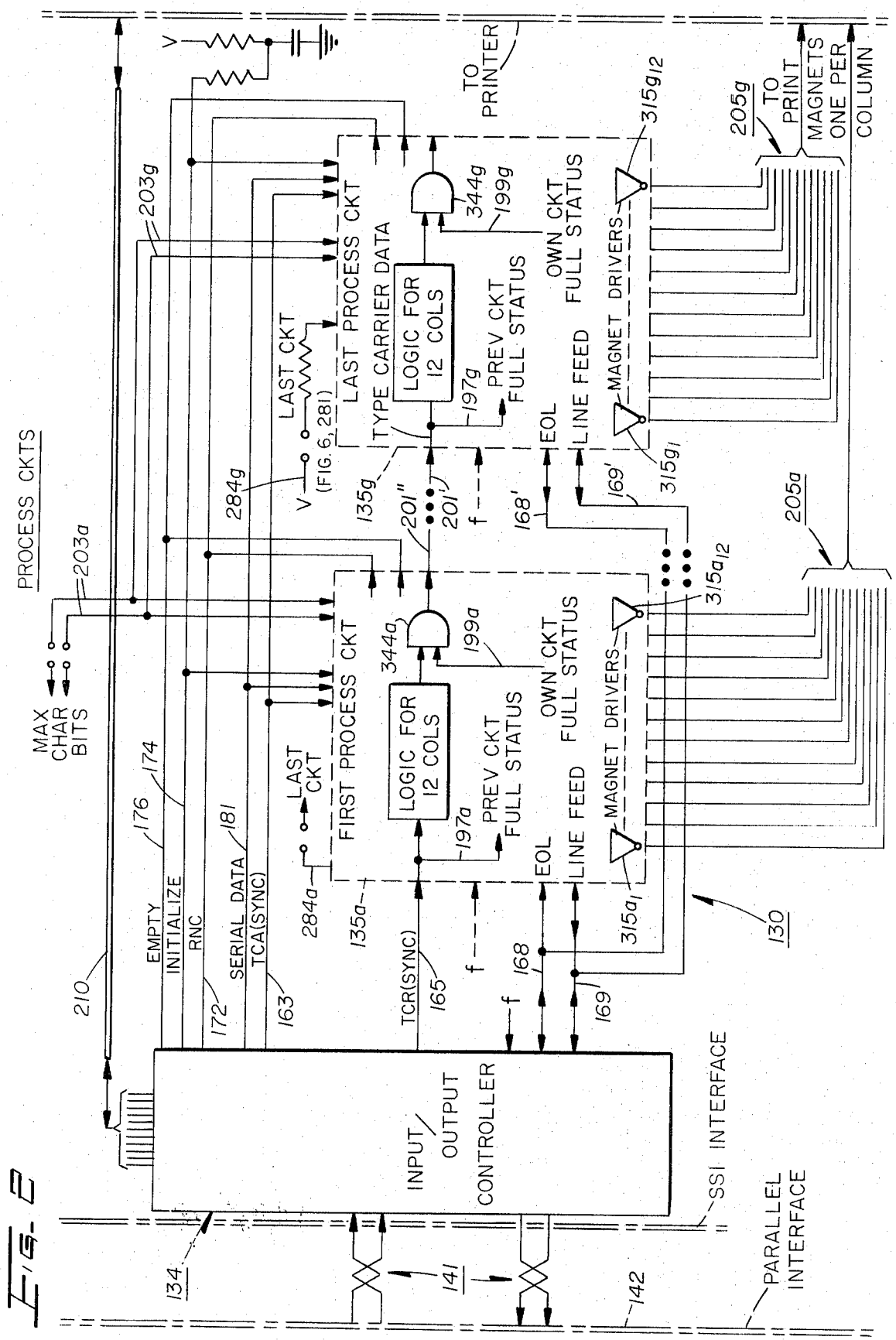

PROCESS CKTS
MAX CHAR BITS
203a
203g
210
EMPTY
INITIALIZE
RNC
SERIAL DATA
TCA(SYNC)
TCR(SYNC)
174
176
181
172
163
165
FIRST PROCESS CKT
LAST PROCESS CKT
LAST CKT
(FIG. 6, 281)
284a
284g
135a
135g
LOGIC FOR 12 COLS
197a
197g
PREV CKT FULL STATUS
344a
344g
199a
199g
OWN CKT FULL STATUS
EOL
LINE FEED
MAGNET DRIVERS
315a1
315a12
315g1
315g12
168
168'
169
169'
201'
201''
205a
205g
TO PRINT MAGNETS ONE PER COLUMN
TO PRINTER
130
INPUT/OUTPUT CONTROLLER
SSI INTERFACE
PARALLEL INTERFACE
134
141
142

FIG. 2

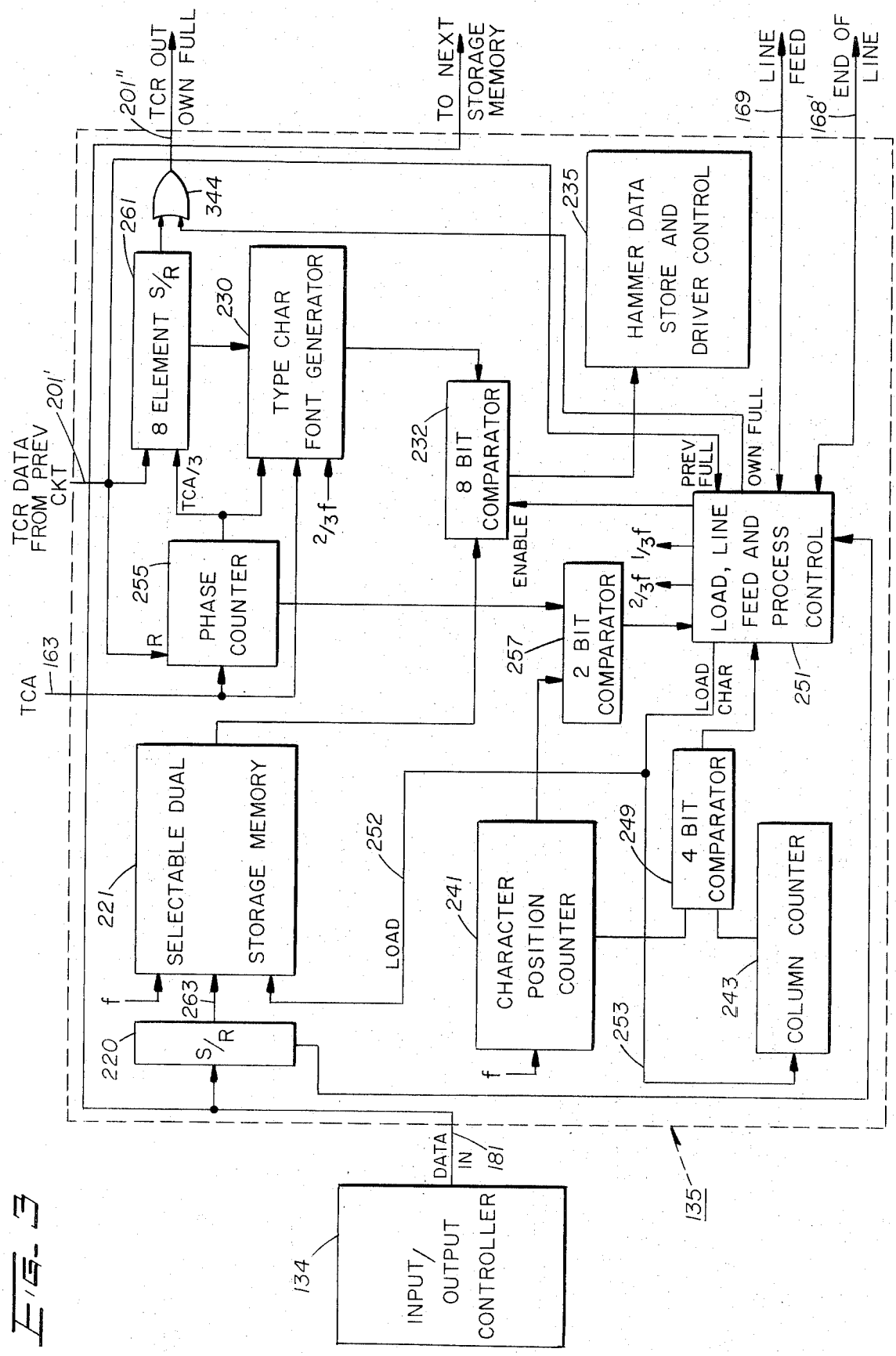

TO NEXT STORAGE MEMORY
201" TCR OUT OWN FULL
169 LINE FEED
168' END OF LINE
344
261
8 ELEMENT S/R
230
TYPE CHAR FONT GENERATOR
235
HAMMER DATA STORE AND DRIVER CONTROL
201' TCR DATA FROM PREV CKT
TCA/3
2/3f
232
8 BIT COMPARATOR
PREV FULL
OWN FULL
ENABLE
255
R
PHASE COUNTER
2/3f 1/3f
LOAD, LINE FEED AND PROCESS CONTROL
TCA 163
257
2 BIT COMPARATOR
LOAD CHAR
251
221
SELECTABLE DUAL STORAGE MEMORY
252
LOAD
241
CHARACTER POSITION COUNTER
249
4 BIT COMPARATOR
f
263
220
S/R
253
243
COLUMN COUNTER
DATA IN 181
135
134
INPUT/ OUTPUT CONTROLLER

FIG. 3

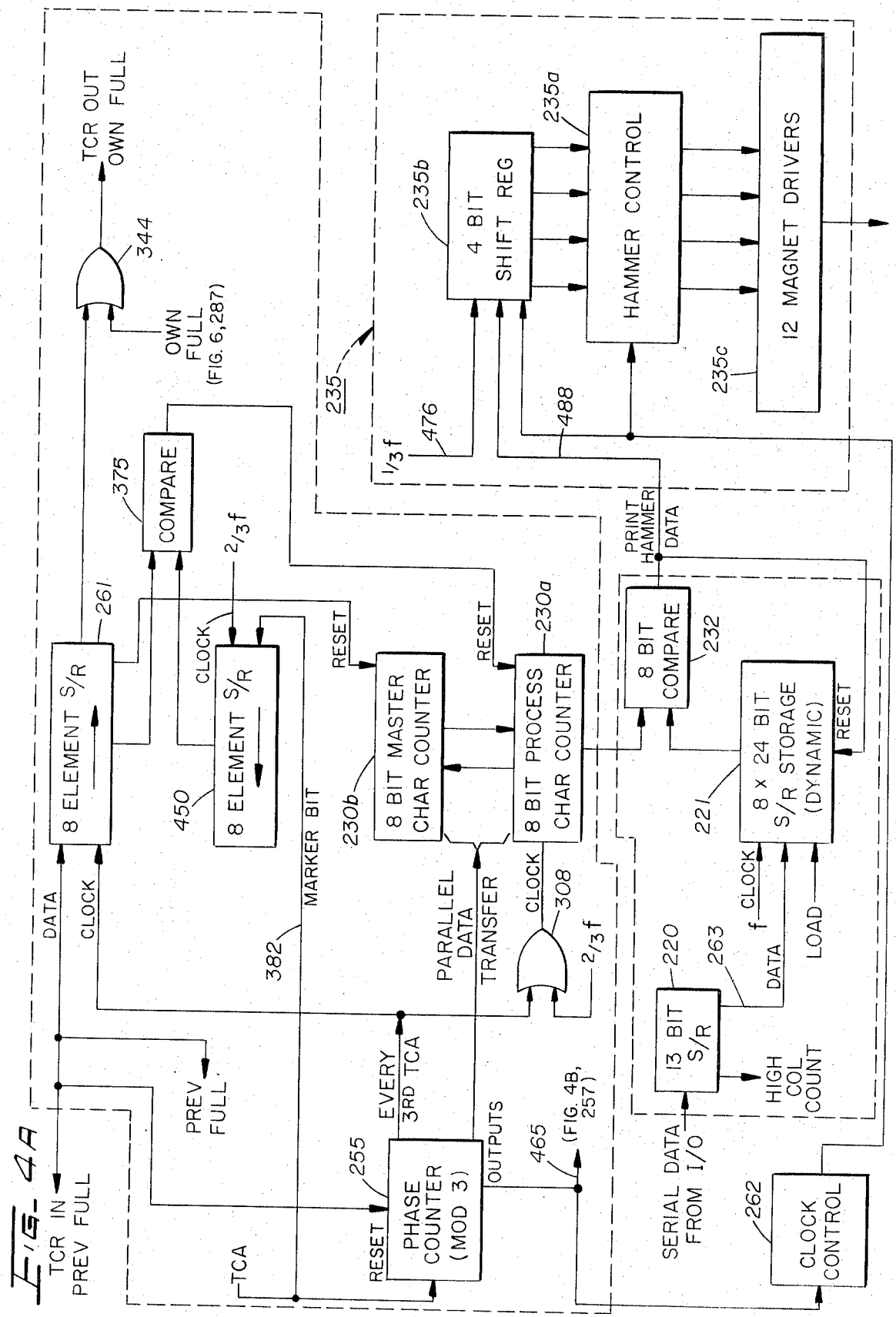

TCR IN
PREV FULL
TCA
RESET
PHASE COUNTER (MOD 3)
255
EVERY 3RD TCA
OUTPUTS
465
(FIG. 4B, 257)
PREV FULL
DATA
CLOCK
8 ELEMENT S/R
261
MARKER BIT
382
CLOCK
8 ELEMENT S/R
450
2/3f
COMPARE
375
OWN FULL (FIG. 6, 287)
344
TCR OUT OWN FULL
RESET
8 BIT MASTER CHAR COUNTER
230b
RESET
8 BIT PROCESS CHAR COUNTER
230a
PARALLEL DATA TRANSFER
CLOCK
308
2/3f
235
1/3f
476
4 BIT SHIFT REG
235b
HAMMER CONTROL
235a
12 MAGNET DRIVERS
235c
488
PRINT HAMMER DATA
8 BIT COMPARE
232
8 x 24 BIT S/R STORAGE (DYNAMIC)
221
RESET
f CLOCK
DATA
LOAD
13 BIT S/R
220
263
HIGH COL COUNT
SERIAL DATA FROM I/O
CLOCK CONTROL
262

FIG. 4A

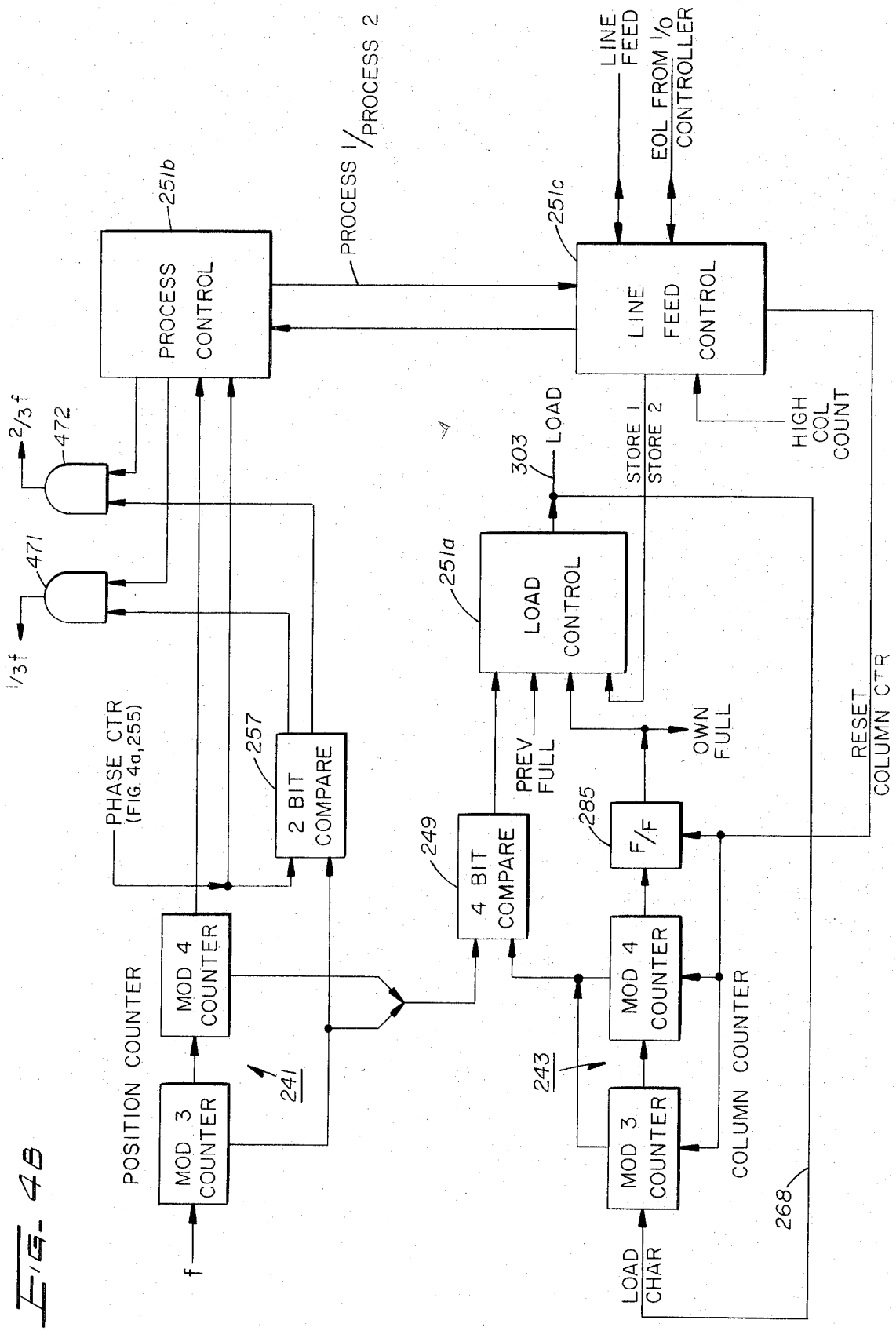

PROCESS CONTROL
251b
PROCESS 1/PROCESS 2
LINE FEED CONTROL
251c
LINE FEED
EOL FROM I/O CONTROLLER
2/3 f
472
471
1/3 f
303
LOAD
STORE 1
STORE 2
HIGH COL COUNT
LOAD CONTROL
251a
PHASE CTR (FIG. 4a, 255)
257
2 BIT COMPARE
PREV FULL
OWN FULL
249
4 BIT COMPARE
285
F/F
RESET COLUMN CTR
POSITION COUNTER
MOD 4 COUNTER
MOD 3 COUNTER
241
243
COLUMN COUNTER
LOAD CHAR
268
f

FIG. 4B

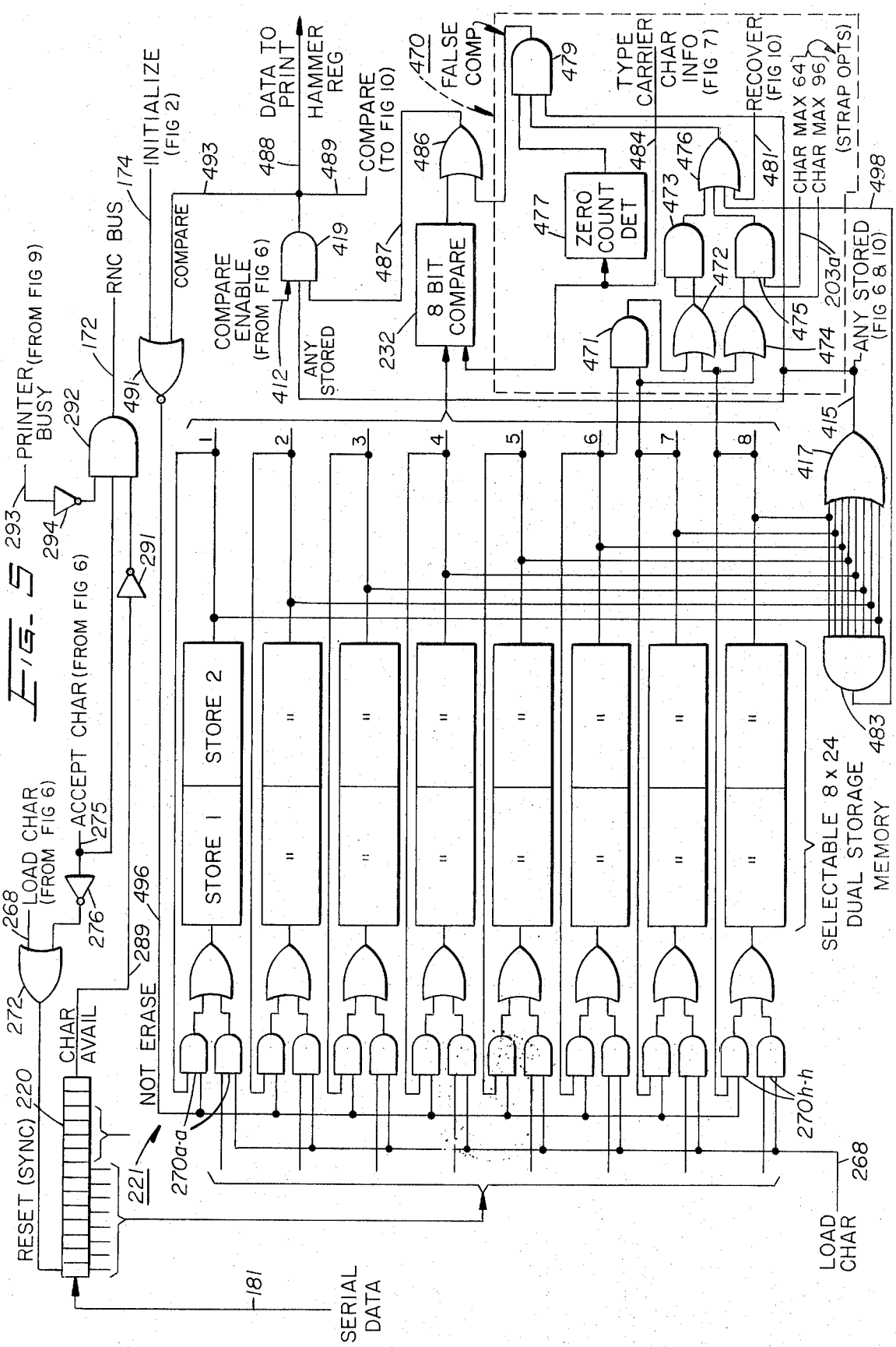
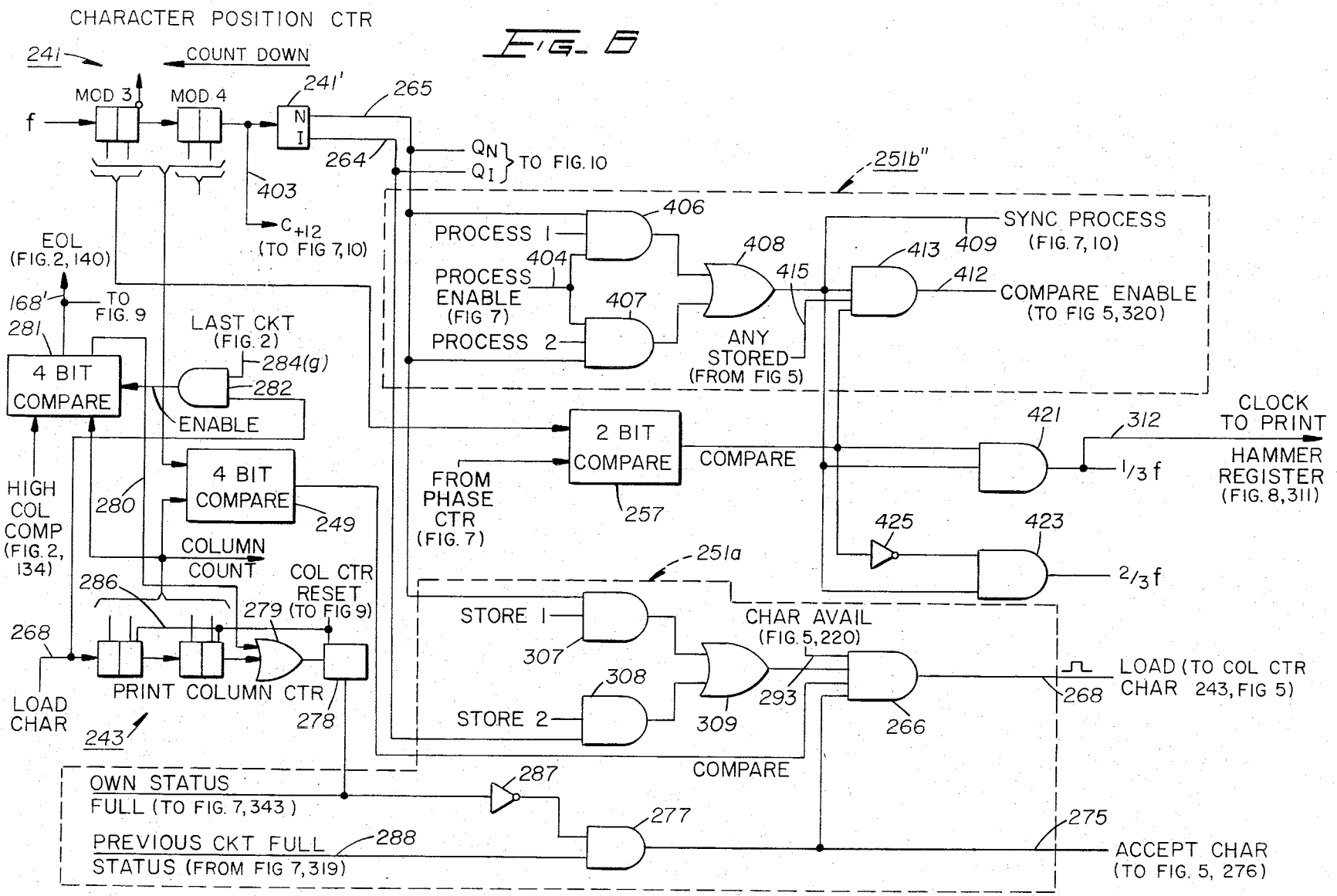


FIG. 5
PRINTER BUSY (FROM FIG 9)
293
294
291
RNC BUS
174
INITIALIZE (FIG 2)
172
292
491
COMPARE
493
DATA TO PRINT
488
HAMMER REG
489
COMPARE (TO FIG 10)
COMPARE ENABLE (FROM FIG 6)
412
ANY STORED
419
487
232
8 BIT COMPARE
486
470
FALSE COMP
479
477
ZERO COUNT DET
484
TYPE CARRIER
CHAR INFO (FIG 7)
RECOVER (FIG 10)
481
CHAR MAX 64
CHAR MAX 96
(STRAP OPTS)
498
473
476
472
475
474
203a
471
ANY STORED (FIG 6 & 10)
415
417
483
ACCEPT CHAR (FROM FIG 6)
275
LOAD CHAR (FROM FIG 6)
268
276
289
496
272
CHAR AVAIL
NOT ERASE
RESET (SYNC) 220
221
270a-a
270h-h
STORE 1
STORE 2
1
2
3
4
5
6
7
8
SELECTABLE 8 x 24 DUAL STORAGE MEMORY
LOAD CHAR
268
SERIAL DATA
181

FIG. 8


CHARACTER POSITION CTR
241
COUNT DOWN
MOD 3
MOD 4
f
241'
N
I
265
264
403
Q_N
Q_I
TO FIG. 10
C_{+12}
(TO FIG 7,10)
251b"
406
PROCESS 1
404
PROCESS ENABLE (FIG 7)
PROCESS 2
407
408
415
ANY STORED (FROM FIG 5)
413
409
SYNC PROCESS (FIG. 7, 10)
412
COMPARE ENABLE (TO FIG 5,320)
EOL (FIG. 2, 140)
168'
TO FIG. 9
281
4 BIT COMPARE
LAST CKT (FIG. 2)
284(g)
282
ENABLE
HIGH COL COMP (FIG. 2, 134)
280
4 BIT COMPARE
249
COLUMN COUNT
2 BIT COMPARE
257
COMPARE
FROM PHASE CTR (FIG. 7)
421
312
CLOCK TO PRINT HAMMER REGISTER (FIG. 8, 311)
1/3 f
425
423
2/3 f
251a
286
279
COL CTR RESET (TO FIG 9)
268
LOAD CHAR
243
PRINT COLUMN CTR
278
STORE 1
307
STORE 2
308
309
CHAR AVAIL (FIG. 5, 220)
293
266
COMPARE
LOAD (TO COL CTR CHAR 243, FIG 5)
268
287
OWN STATUS FULL (TO FIG. 7, 343)
277
PREVIOUS CKT FULL STATUS (FROM FIG 7, 319)
288
275
ACCEPT CHAR (TO FIG. 5, 276)

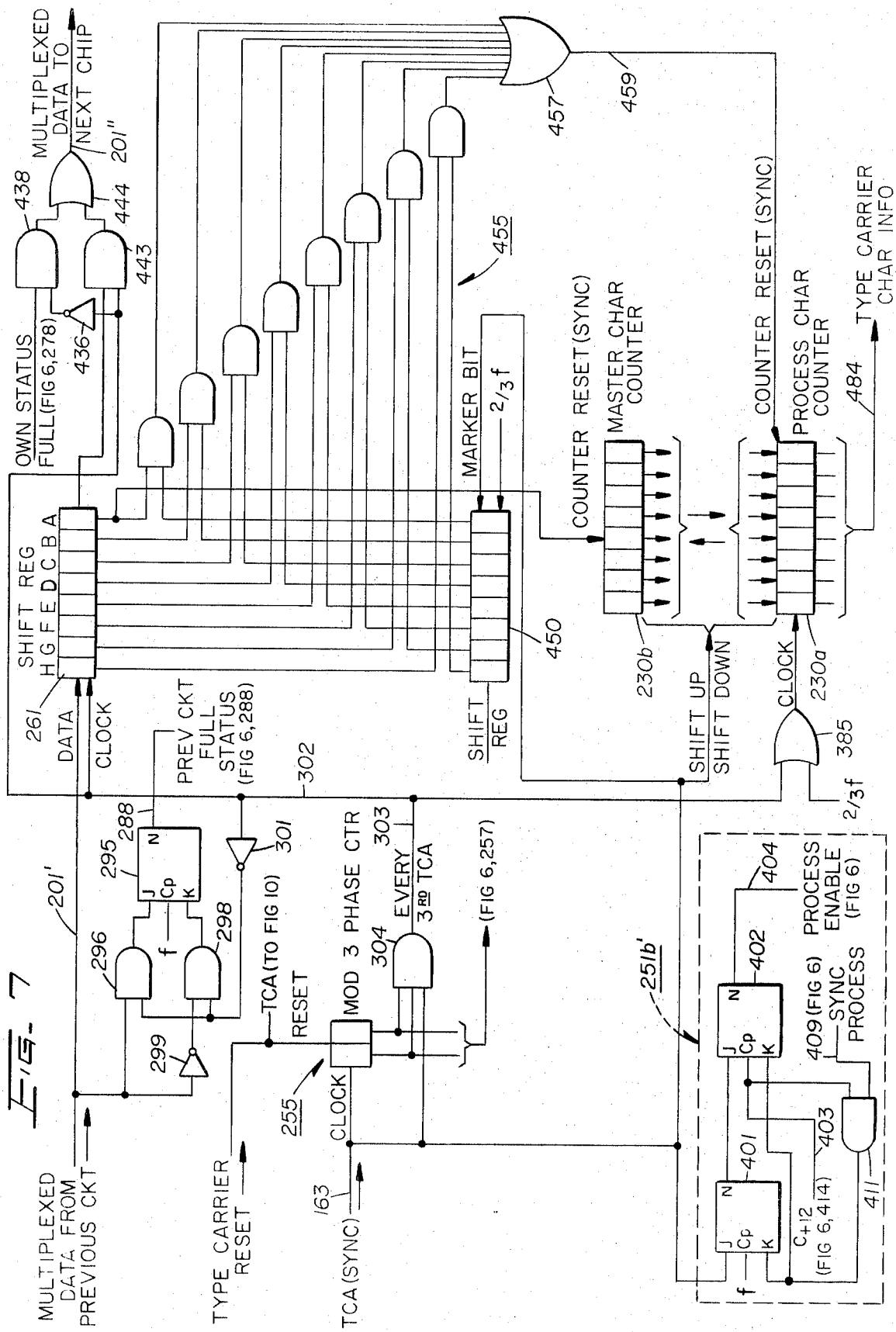

MULTIPLEXED DATA FROM PREVIOUS CKT
TYPE CARRIER RESET
TCA (SYNC)
163
255
CLOCK
RESET
TCA (TO FIG 10)
MOD 3 PHASE CTR
EVERY 3RD TCA
303
304
(FIG 6,257)
201'
295
288
296
298
299
301
302
PREV CKT FULL STATUS (FIG 6,288)
261
DATA
CLOCK
SHIFT REG H G F E D C B A
OWN STATUS FULL (FIG 6,278)
436
438
443
444
MULTIPLEXED DATA TO NEXT CHIP
201"
457
459
455
MARKER BIT
2/3 f
SHIFT REG
450
COUNTER RESET (SYNC)
MASTER CHAR COUNTER
230b
SHIFT UP
SHIFT DOWN
COUNTER RESET (SYNC)
PROCESS CHAR COUNTER
230a
CLOCK
385
2/3f
484
TYPE CARRIER CHAR INFO
251b'
PROCESS ENABLE (FIG 6)
404
402
409 (FIG 6) SYNC PROCESS
401
403
411
C+12 (FIG 6,414)
f

FIG. 7

FIG. 8

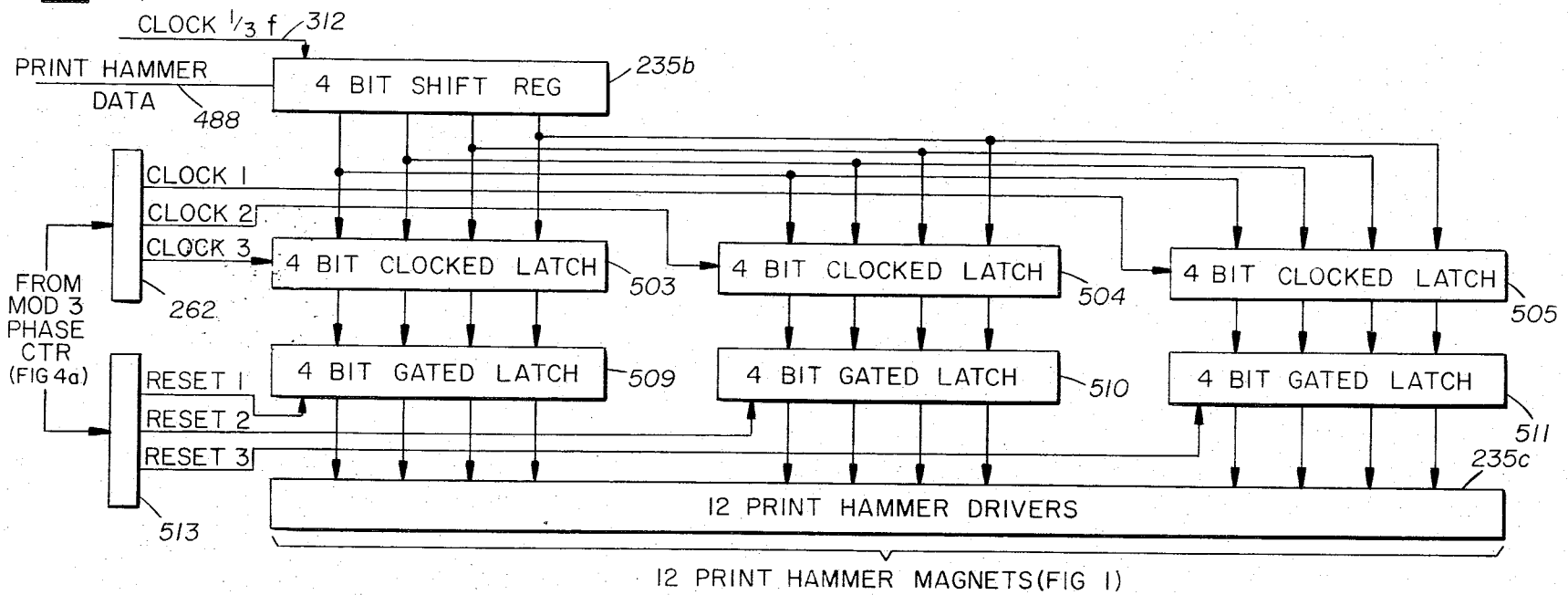
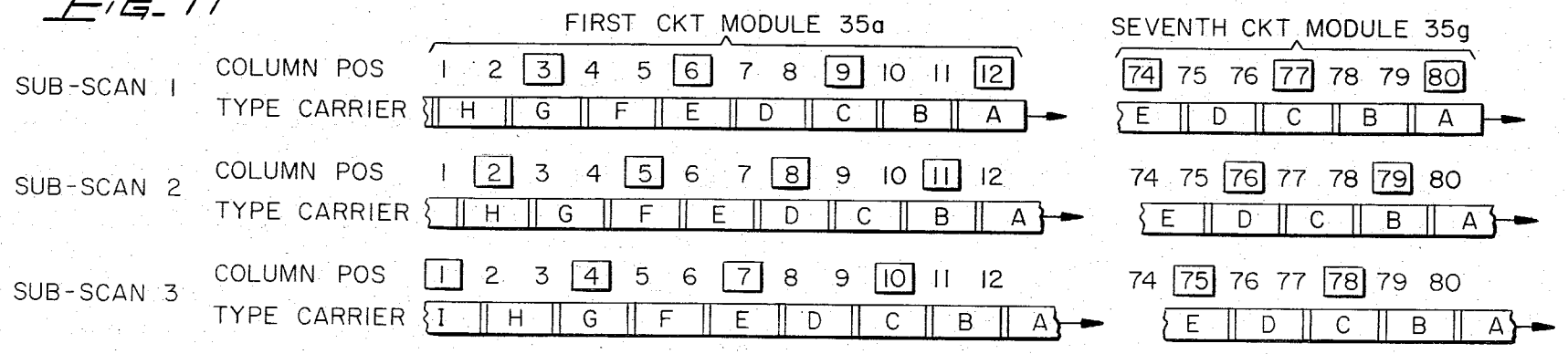

CLOCK 1/3 f
312
PRINT HAMMER DATA
488
4 BIT SHIFT REG
235b
CLOCK 1
CLOCK 2
CLOCK 3
262
FROM MOD 3 PHASE CTR (FIG 4a)
4 BIT CLOCKED LATCH
503
4 BIT CLOCKED LATCH
504
4 BIT CLOCKED LATCH
505
RESET 1
RESET 2
RESET 3
513
4 BIT GATED LATCH
509
4 BIT GATED LATCH
510
4 BIT GATED LATCH
511
235c
12 PRINT HAMMER DRIVERS
12 PRINT HAMMER MAGNETS (FIG 1)

FIG. 11


FIRST CKT MODULE 35a
SEVENTH CKT MODULE 35g
SUB-SCAN 1
COLUMN POS 1 2 3 4 5 6 7 8 9 10 11 12
TYPE CARRIER H G F E D C B A
74 75 76 77 78 79 80
E D C B A
SUB-SCAN 2
COLUMN POS 1 2 3 4 5 6 7 8 9 10 11 12
TYPE CARRIER H G F E D C B A
74 75 76 77 78 79 80
E D C B A
SUB-SCAN 3
COLUMN POS 1 2 3 4 5 6 7 8 9 10 11 12
TYPE CARRIER I H G F E D C B A
74 75 76 77 78 79 80
E D C B A

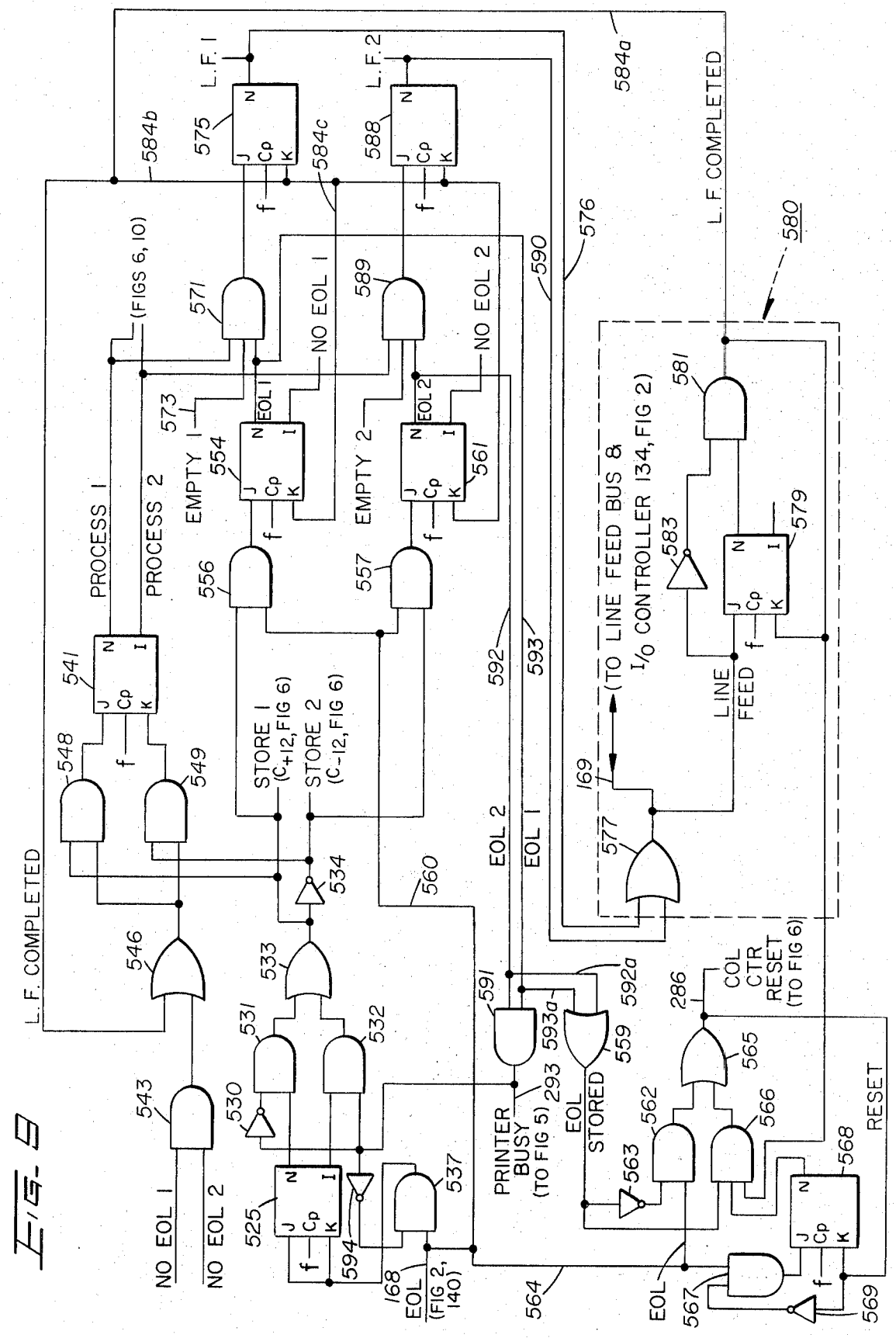

L.F. 1
L.F. 2
575
588
584a
584b
584c
L.F. COMPLETED
580
(FIGS 6, 10)
571
589
NO EOL 1
NO EOL 2
590
576
PROCESS 1
PROCESS 2
EMPTY 1
EMPTY 2
573
554
561
556
557
EOL 1
EOL 2
581
583
579
(TO LINE FEED BUS & I/O CONTROLLER 134, FIG 2)
LINE FEED
592
593
541
548
549
STORE 1 (C+12, FIG 6)
STORE 2 (C-12, FIG 6)
169
577
534
560
546
533
591
592a
593a
559
286
COL CTR RESET (TO FIG 6)
531
532
565
543
530
PRINTER BUSY (TO FIG 5)
293
EOL STORED
562
566
563
568
RESET
525
537
594
168
EOL (FIG 2, 140)
564
567
569
NO EOL 1
NO EOL 2

FIG. 9

FIG. 10

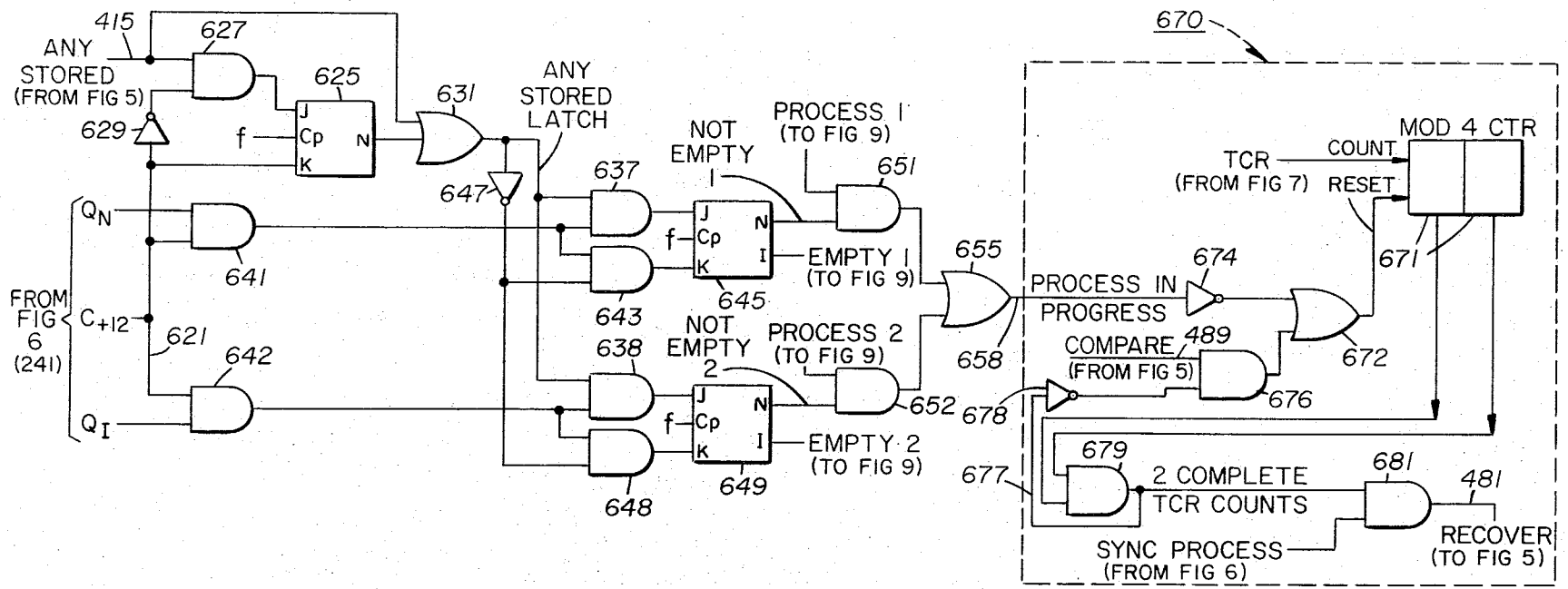

415
ANY STORED (FROM FIG 5)
627
625
629
f
J Cp K N
631
ANY STORED LATCH
647
637
NOT EMPTY 1
PROCESS 1 (TO FIG 9)
651
J Cp K N I
f
EMPTY 1 (TO FIG 9)
645
655
643
FROM FIG 6 (241)
Q_N
641
C_{+12}
621
642
638
NOT EMPTY 2
PROCESS 2 (TO FIG 9)
652
658
678
Q_I
649
EMPTY 2 (TO FIG 9)
648
670
MOD 4 CTR
COUNT
TCR (FROM FIG 7)
RESET
671
674
PROCESS IN PROGRESS
672
COMPARE (FROM FIG 5)
489
676
679
677
2 COMPLETE TCR COUNTS
681
481
SYNC PROCESS (FROM FIG 6)
RECOVER (TO FIG 5)

FIG. 12

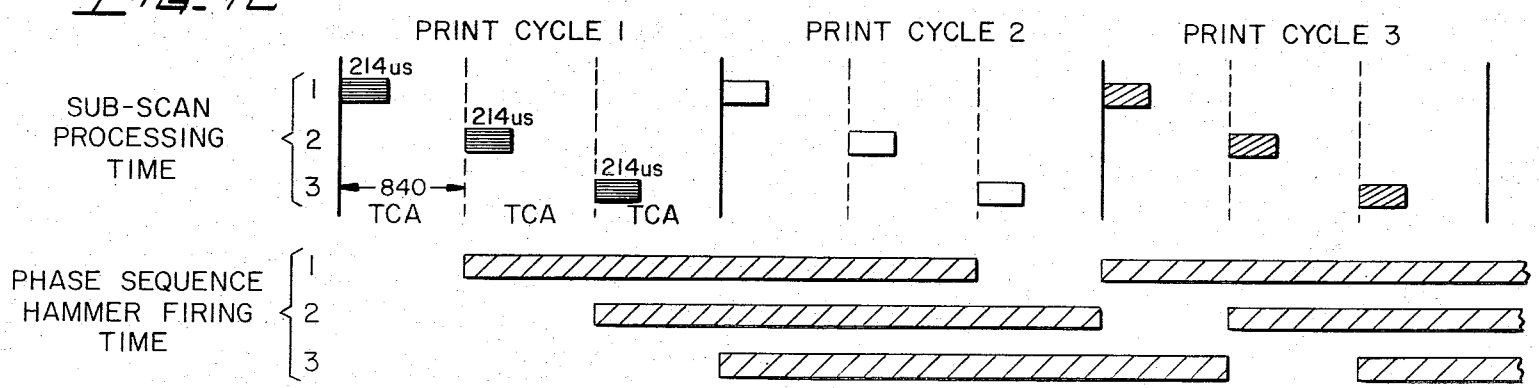

PRINT CYCLE 1
PRINT CYCLE 2
PRINT CYCLE 3
SUB-SCAN PROCESSING TIME
1
2
3
214us
214us
214us
840
TCA
TCA
TCA
PHASE SEQUENCE HAMMER FIRING TIME
1
2
3

FIG. 13

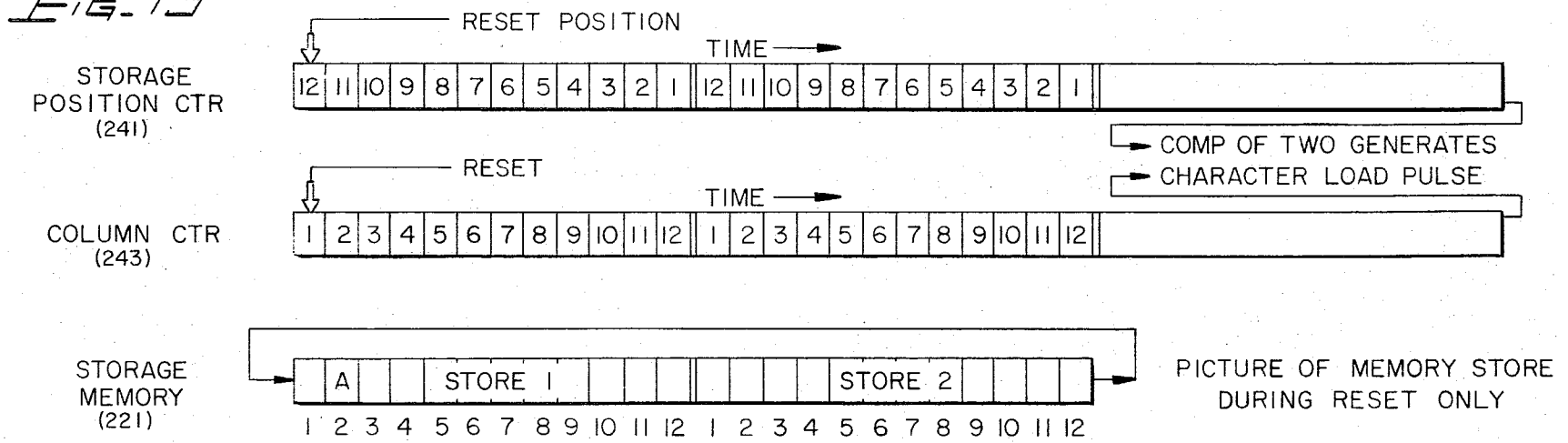
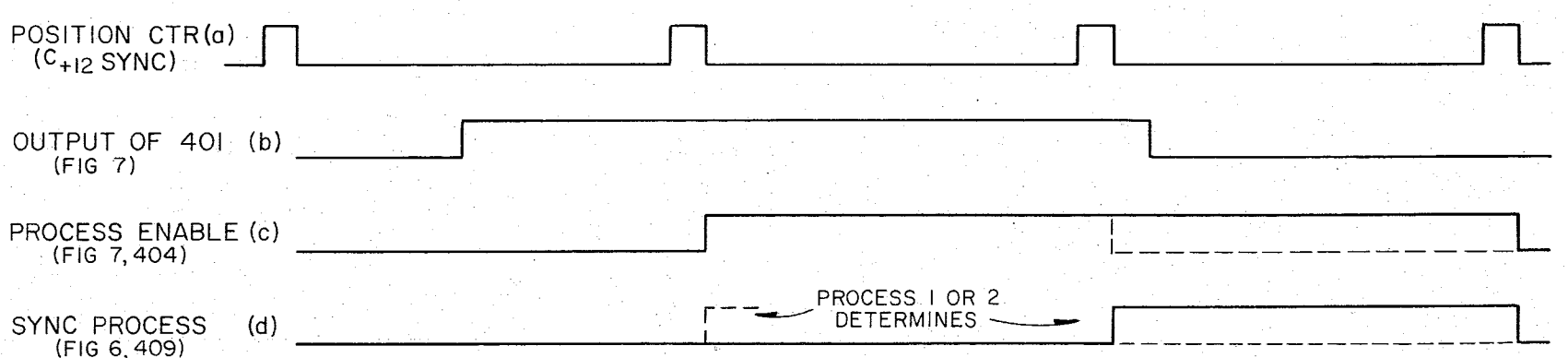

RESET POSITION
TIME
STORAGE POSITION CTR (241)
12 11 10 9 8 7 6 5 4 3 2 1 12 11 10 9 8 7 6 5 4 3 2 1
COMP OF TWO GENERATES
CHARACTER LOAD PULSE
RESET
TIME
COLUMN CTR (243)
1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
STORAGE MEMORY (221)
A
STORE 1
STORE 2
1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11 12
PICTURE OF MEMORY STORE DURING RESET ONLY

FIG. 14


POSITION CTR (a) (C_{+12} SYNC)
OUTPUT OF 401 (b) (FIG 7)
PROCESS ENABLE (c) (FIG 7, 404)
SYNC PROCESS (d) (FIG 6, 409)
PROCESS 1 OR 2 DETERMINES

PRINT CONTROL LOGIC CIRCUITRY FOR ON-THE-FLY PRINTERS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to printer apparatus and, more particularly, to logic control circuitry for selectively and sequentially controlling the line printing of type characters on-the-fly in such apparatus utilizing a continuously moving, endless type carrier.

2. Description of the Prior Art

So-called on-the-fly printers, with which the present invention is primarily concerned, are of two basic impact types. One type utilizes a rotating type character drum, and the other more recent, and generally preferred type, utilizes an endless type character chain or belt such as is described in Huntoon-Kearney U.S. Pat. No. 3,742,848, herein incorporated by reference.

In all of these versions, the type supporting member is constructed and positioned so as to carry one or more fonts of type characters past an array of print-inducing members, such as hammers, which are selectively actuated under the control of logic circuitry at the proper times, so as to effect the printing of the desired characters at the proper positions along successive print lines of an indexable imprint character receiving medium, such as paper in either roll or sheet stock form.

One significant advantage of the "belt" or "chain" type printer over the drum printer is that the two former versions do not present troublesome problems with respect to vertical registration of the printed characters (i.e., accurate alignment of the printed characters with respect to a horizontal reference line). In drum printers vertical registration is critically dependent on the timing of hammer actuation as the characters pass vertically thereby. This problem is essentially obviated in an endless carrier printer because both the carrier and the array of hammers are juxtaposed in accurate parallel, horizontal relationship across the width dimension of the paper.

Endless carriers also normally have the type characters mounted thereon in such a way that they are readily replaceable individually or as an entire font. In addition, both multiple identical and dissimilar fonts, having the same or different font lengths, may also be employed in endless carrier impact printers. This greatly increases the versatility of the printer for specialized printing applications. Such character font versatility is not possible in drum printers because the circumferentially disposed font of characters in every row must be identical.

In addition, as an endless type carrier allows the utilization of a wider spacing between type characters than between hammers, referred to herein as a pitch relationship therebetween, the so-called problem of "ghosting" may be readily eliminated. Such a type character-hammer relationship is described in the commonly assigned copending application of Egon S. Babler, Ser. No. 268,236, now U.S. Patent 3,795,187, herein incorporated by reference. Ghosting (or shadow printing) is a phenomenon which occurs as a result of a print hammer, in pressing the paper against the type character to be printed, also creating sufficient pressure to produce a slight impression on the paper of the edge of a type character adjacent to the character being printed. Such ghosting or shadow effects are particularly troublesome in drum printers because the lateral spacing of the type characters in adjacent rows on the drum must necessarily be equal to the relatively close spacing of the associated print hammers.

Accordingly, while chain or belt type printers afford a number of significant advantages over the drum printer, this does not mean that there are no serious problems involved in obtaining precise control over type character-hammer registry while the former is moving "on-the-fly." Such registry is very important, of course, if printed character impressions are to be not only uniformly spaced, but to exhibit distinct, sharp character line formations as required for esthetically pleasing and easily read copy.

Compounding the timing problems involved in endless type carrier printers is the fact that with the spacing between adjacent type characters normally being purposely chosen to be greater than the spacing that exists between adjacent hammers, it becomes readily apparent that at any given point in time during a print cycle, only a fractional number of the hammers will simultaneously be in exact registry with an equal number of type characters; preferably every third hammer as described in the above-cited Babler application. As such, the hammer logic control circuitry must be capable of temporarily storing the incoming data to be printed on a given line in a manner that will allow the sequential read out and utilization of that information only as the proper type characters on the carrier are brought into alignment with the proper hammers (aligned with the print columns) for each line to be printed.

Thus, the logic control circuit must be capable of serially receiving encoded input data to be printed, but thereafter printing that data in a sequential rather than serial manner. Temporary memory storage of the input data is necessary, of course, in order that every hammer, at some point in time during the printing of each line, be aligned with and capable of effecting the printing of a character corresponding to every type characters of a given font (or fonts) passing thereby.

Such a time-delayed, sequential mode of hammer actuation, and the need therefor, will be considered with respect to one particular belt printer of the type embodied herein wherein adjacent type characters on the carrier having a spacing of 1.5 times the spacing between adjacent hammers as in the aforementioned Babler application. As such, every second type character may be brought into registry with every third print hammer (and column) of an array thereof at any one time. Inasmuch as the type characters are moving at a constant rate of speed past the array of hammers, it is readily seen that there are continuously changing groups of type characters and print hammers brought into alignment during each print cycle. Thus, it can be readily seen that when there are a large number of characters to be printed along a given line, each print cycle may involve the printing of from 1 to the maximum number of type characters and hammers that can be aligned at any one time during a print cycle. It thus follows that the characters and hammers involved in one print cycle normally would have no particular relationship with the type characters and hammers involved in any other of the print cycles required to complete the printing of a given line.

As described herein, the logic process of successively identifying each group of characters that is sequentially

brought into alignment with an associated group of hammers is referred to as a sub-scan, with three sub-scans constituting a scan period. A scan period is required to advance the type carrier by a distance equal to the spacing between two adjacent type characters. With the spacing between hammers (or columns) being equal to only two-thirds the spacing between type characters (for a 1.5 pitch relationship therebetween), it is seen that in order to print any given character of a font in any one of the column positions of the printer, the number of sub-scans must equal 3 × N, where N is the number of character in the font.

It also logically follows that the shortest possible sub-scan print period for any new type character-column (hammer) alignment must necessarily encompass the time required for the carrier to be displaced one half the distance between hammers (or one-third the distance between type characters). In one illustrative embodiment, this displacement of the type carrier requires 840 microseconds.

The logic control circuitry employed to effect high-speed impact printing "on-the-fly" heretorfore has generally comprised a single storage memory having a plurality of storage locations corresponding in number to the total number of print columns of the printer. Associated with the memory typically are means for identifying the characters in the sequence in which they appear on the type carrier, means for actuating the hammers as the proper characters sequentially register therewith, means for timing the various control and print functions, and means for initiating and terminating the various electrical and mechanical operations involved in connection with the operation of the printer.

With respect to the storage memory, it has generally comprised a magnetic core arranged in a particular core plane matrix having, for example, a core plane for each bit of a chosen binary code, with each plane comprising plural cores arranged in a row and column configuration. Such a core plane matrix, for example, might comprise 160 cores arranged in 16 rows of 10 cores each so as to accommodate a corresponding number of print hammers.

Disadvantageously, such core memories require rather complex and expensive (from a fabrication standpoint) X-Y read-write driver circuitry, including X-Y windings, drivers, rings and switches. Such magnetic core matrices also often impose undesirable speed-power restrictions on the associated circuitry, as well as limitations on the size and layout of the composite control circuitry.

More recently, advancements in solid state technology have resulted in very versatile and multi-faceted integrated circuitry referred to as medium or large scale integration (MSI or LSI). Particularly in applications where very high switching speeds have not been required, the active devices in such circuitry have increasingly been of the metal-oxide-semiconductor-field-effect type, hereinafter generally referred to as MOSFETs. Such devices have proven to be very effective, reliable, and readily amenable to high volume, low cost manufacture, even in rather complex, high density MSI or LSI circuitry. To that end, MOSFETs are gaining wide acceptance for use in the fabrication of diverse logic circuitry, including gates, inverters, flip-flops, counters, shift registers, and the like.

With respect to printer logic control circuitry, however, all of the required circuit logic cannot at present, at least, be economically fabricated on a single semiconductor chip. This has thus necessitated a number of different logic circuit chips heretofore, with rather complex and extensive interfacing being required therebetween in order to assemble a composite printer hammer logic control circuit. This, of course, does not lead to low cost, high yield composite circuits that would be possible if every circuit chip could be of identical construction, be capable of independently processing print data for only an assigned subgroup of print columns, and be interconnected in a simple, modular fashion so as to constitute the complete logic circuitry for an 80 column printer, for example.

Such a modular form of circuit construction would also have the advantage that it would be conducive to accommodating printers with a smaller or larger number of print columns, typically ranging from 40 to 132 columns, for example, by simply adding or subtracting one or more identical circuit modules from the composite logic circuitry. Unfortunately, with most hammer logic control circuits employed heretofore, the maximum number of print columns that can normally be controlled is fixed by unalterable circuit parameters.

Accordingly, there exists a definite need for a modular constructed hammer logic control circuit, particularly wherein the number of print columns chosen for any given application need not be divisable by any given number of identical circuit modules, each of which may be designed normally to accommodate and control the processing of print data for a given predetermined number of print columns. To that end, what is needed is circuit modules designed to be compatible with simple, switch-option circuitry that may be employed outside of the last of a series of modular circuit chips, for example, so as to accommodate and control the processing of print data for any number of print columns less than the predetermined number. In that way the need for any expensive, specially designed circuit chips would be obviated.

A concomitant problem affecting both circuit speed requirements and total line printing speed in endless carrier, on-the-fly printers heretofore is related to the need to first store a complete line of data before printing commences. This has generally been true whether synchronous or asynchronous storage memories were employed, and whether the hammer-type character spacings were the same or different.

Such total print line loading of characters, of course, substantially reduces the line printing speed of the apparatus. Such a mode of operation can also impose other serious constraints on the printer when, for example, input data is being received at a much faster rate than the printer is capable of printing it. An auxiliary storage memory could be employed, of course, but this is a particularly costly adjunct to the basic printer when the incoming data may be only sporadically received at a rate faster than the printer can handle it.

It is thus seen that there is a need for hammer logic control circuitry for on-the-fly "belt" printers that can fully exploit the many advantages of MOSFET circuitry, without requiring a cost-performance trade-off in the logic circuitry in order to attain the high speed printing rates desired.

SUMMARY OF THE INVENTION

It, therefore, is an object of the present invention to provide new and improved type character selection and print logic control circuitry for use in on-the-fly high speed printer apparatus.

It is a further object of the present invention to utilize the virtues of MOSFET integrated circuitry in on-the-fly printer apparatus in a manner that allows reliable, simplified, low cost fabrication thereof, through the use of selective, logic controlled, dual line character storage read out of the information to be printed.

In accordance with the principles of the present invention, these and other objects are accomplished through the utilization of a plurality of intercoupled, but independently operated data processing circuit modules. Each module includes a free-running, logic controlled dual line storage memory capable of selectively directing intput encoded character data into and out of either of two discrete storage areas thereof. Both areas are associated with a specific and common subgroup of hammers (and print columns), but each storage area is associated with a different print line. Such dual line storage areas in each memory advantageously allows incoming data, often received at varying rates, to be temporarily stored for subsequent printing without having to resort to a batch process technique requiring an auxiliary main storage memory, for example.

The utilization of multiple data processing circuit modules, with each associated with a different subgroup of print hammers (and columns), advantageously allows the switching speeds of the various active integrated circuit elements embodied in each processing circuit to be lower, for a given line printing speed, than would be possible with a single storage memory having sufficient storage for all of the character data to be printed along a given line. Considered more specifically, with each circuit module controlling only a fractional number of the total print hammers employed in a given printer, the number of storage positions in the memory thereof are likewise reduced. Accordingly, the scanning (or read out) rate of the circuit can be reduced, as the limited number of encoded characters stored may be processed and printed independently of, but concurrently with the input data received in and processed by the other circuit modules.

In contrast, with a typical single shift register type of memory employed heretofore, if there are 80 columns of possible data to be stored for subsequent printing along a given line, for example, then that data must normally be shifted through 80 elements of the shift register memory every time a different type character is brought into alignment with a different column and associated hammer, until all possible alignment combinations therebetween have been compared. It thus becomes readily apparent that as the number of storage positions required in the memory increase, the circuit logic switching speed of not only the memory, but of the other associated processing circuitry must also increase for a given line printing speed.

The modular nature of the logic circuitry also has the advantage that various length print lines can easily be accommodated by simply adding or subtracting one or more of the identical data processing circuit modules as required for a given application. Moreover, in accordance with the principles of the present invention, the number of print columns chosen need not be divisable by the standard number of columns assigned to each circuit module by design. Rather, each circuit module is designed so as to be compatible with outside strap-option circuitry so as to accommodate less than the standard number of print columns. Thus, there is no need for the last circuit module of a series, for example, to be specially designed to accommodate a non-standard number of print columns.

As all of the circuit modules may be formed on discrete chips of identical integrated circuit design, chip size may be chosen to minimize costs of both manufacture and composite circuit assembly, while simultaneously maximizing circuit chip yield and performance.

It is thus seen that the sub-divided, modular construction of the hammer logic control circuitry embodied herein advantageously allows individual data processing functions to be performed at a slower rate, for a given line printing speed. This, in turn, makes it possible to exploit the desired features and characteristics of inexpensively designed MOSFET integrated circuitry, for example, without any adverse effect on printer performance. Stated another way, the present modular logic circuitry does not necessitate the use of bi-polar active devices, or customized MOSFET devices, fabricated through special and expensive processing techniques, in order to achieve the switching speed characteristics otherwise required in prior hammer logic control circuits for a given line printing rate.

The hammer logic control circuitry embodied in the present invention, as will be discussed in greater detail hereinbelow, also exhibits a number of other features and advantages relating not only to control of the printer mechanism per se, but to the control of other important operating functions, such as error character detection, conditional processing time out, data overloads, even and odd parity checks, and test mode operations. The control circuitry is also designed to receive Standard Serial Interface (SSI) information, with the modular data processing circuits packaged on circuit cards for ease of assembly and maintenance. The outputs from the circuit cards also advantageously drive the hammer magnets directly, which simplifies the hammer actuation circuits and the wiring thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a partially broken away perspective view of a high-speed impact printer, with some parts being omitted for the purpose of illustration, together with a simplified block diagram of the associated hammer logic control circuitry for use therewith, which embodies features in accordancc with the principles of the present invention;

FIG. 2 is a simplified schematic circuit diagram generally depicting the modular nature of the composite hammer logic control circuitry;

FIG. 3 is a simplified block diagram of the major associated circuits embodied in one data processing circuit module of the composite hammer logic control circuitry;

FIGS. 4A and 4B, in block diagram form, illustrate in greater detail the nature and functions of, and the inter-

connected relationship between, all of the major circuits embodied in one processing circuit module;

FIG. 5 is a more detailed block and schematic circuit diagram primarily of the logic controlled dual storage memory embodied in each of the data processing circuit modules in accordance with the principles of the present invention;

FIG. 6 is a block and schematic circuit diagram illustrating in greater detail the association of the character position and print column counters with respect to the load control circuitry disclosed only generally in block diagram form in FIG. 4B;

FIG. 7 is a more detailed block and schematic circuit diagram primarily of the character generating portion of one of the circuit modules of the composite logic control circuitry;

FIG. 8 is a block diagram of the hammer driver circuitry, including clock and reset inputs associated therewith;

FIGS. 9 and 10 are detailed schematic circuit diagrams of interrelated logic circuitry for controlling the selective storage and read out of data from the dual storage areas of one of the circuit module memories, and for effecting the erasing of erroneous data from the memory, respectively;

FIG. 11 is a symbolic representation of two typical sub-scan alignment patterns at one point in time between certain print columns and type characters, having a 1.5 pitch relationship therebetween, for the first and last of seven interconnected sub-circuit processing modules;

FIG. 12 is a timing chart depicting the time duration and sequence of three sub-scans, for each of three successive print periods, relative to the time delayed hammer firings initiated in response to each associated sub-scan;

FIG. 13 is a symbolic representation of the functional relationships, time-wise, between the slot positions in the position and column counters relative to the slot positions in the dual storage areas of one free-running storage memory; and

FIG. 14 is a timing chart illustrating certain printer control and data processing control signals generated during a single line printing cycle.

DETAILED DESCRIPTION OF THE INVENTION
GENERAL DESCRIPTION OF LOGIC CONTROLLED PRINTER APPARATUS

The logic control circuitry embodied herein is particularly adapted for use in high speed, on-the-fly, impact printers of the type depicted in FIG. 1. As disclosed therein, the printer, identified generally by the reference numeral 15, is of the class that utilizes an endless type carrier 16, entrained about a pair of spaced and aligned sprockets or pulleys 17*a* and 17*b*, as described in the aforementioned Huntoon-Kearney patent. The pulleys are journalled by any suitable means on the frame (not shown) of the impact printer. The carrier 16, which may comprise a chain or a toothed timing belt, is adapted to transport a plurality of type pallets 18 (only several shown in FIG. 1) in an essentially orbital path, which is oriented so as to define upper and lower line printing courses in the areas generally defined by the numerals 22 and 23, respectively.

Each of the type pallets 18 has a front face portion having a type character die (not shown) secured thereto or otherwise formed as an integral part thereof. The pallets 18 are mounted on the carrier 16 and uniformly spaced and oriented transversely therealong by means of integral shank portions 18*a* that extend through suitably formed channels (not shown) formed in the carrier. As will presently be seen, in accordance with the present printer embodiment, there are normally at least twice as many type pallets-characters on the carrier than in any chosen font. This means that a given sequence of characters is repeated more than once along the entire length of the carrier. If desired, of course, several fonts of different lengths could also be employed for a special printing application. For further details as to one preferred embodiment of the carrier 16 and type pallets 18, including the mounting and driving structure associated therewith, reference is made to the aforementioned Huntoon-Kearney patent.

In the illustrative embodiment of the printer depicted in FIG. 1, carrier pulleys 17*a* and 17*b* constitute idler and drive pulleys respectively. Driving torque is continuously transmitted to the pulley 17*b* through a shaft 24 which is coupled to a prime mover, herein shown as a motor 26, through a drive train designated generally by a reference numeral 27 in FIG. 1. The drive train includes a worm gear 28, secured to a shaft 30 of the motor 26, a gear 34, operably enmeshed with worm gear 28, and secured to one end of an elongated drive shaft 35, a worm gear 36, secured to the other end of the drive shaft 35, and a gear 38, operably enmeshed with worm gear 36. Gear 38 is secured to the same shaft 24 as the pulley 17*b*.

A clutch 58, which may be of conventional design, is mounted in juxtaposition with gear 38 so as to releaseably couple a split shaft 60 to the coextensively aligned shaft 24. Selective actuation of the clutch 58 is effected through control signals generated by the logic control circuitry embodied herein, as will be discussed in greater detail hereinafter. Whenever the clutch 58 is operated it also supplies torque through a drive train not shown to incrementally advance a web 41, or paper on which printing is to take place in the direction shown by arrow 42, one line at a time.

For further details as to the drive trains for both the type carrier 16 and the web 41, reference is again made to the Huntoon-Kearney patent.

In the illustrative impact printer embodiment depicted in FIG. 1, the character printing mechanism further comprises both an array of spoke-like impellers 84 (only several shown in FIG. 1), and a plurality of respectively associated impactors, or print hammers, designated generally by the reference numeral 85 (only several shown). There is one impeller and associated hammer for each possible printing position or column across the width of the web 41, with the spaced impellers being concentrically secured to the longitudinally extending drive shaft 35. Each impeller 84 has a plurality of radially extending, uniformly spaced spoke-like impeller teeth or elements 84*a*, all of which are oriented in a common plane perpendicular to the axis of the shaft 35. In a preferred embodiment of the printer, the impellers 84 are preferably arranged in a manner as disclosed in the aforementioned copending Babler application.

Each hammer 85 is mounted in a channel (not shown) forming part of the printer frame so as to be selectively driven along a rectilinear path, perpendicular to the web 41, between a normal, untensioned, or non-

printing position, and a printing position whereat an enlarged forward head portion **85***a* of each hammer is propelled against the back side of the web **22**, as described in a second copending application of Egon S. Babler, Ser. No. 268,237, filed July 3, 1972. In order to facilitate printing at any given time, it is apparent that selective hammers must be axially aligned respectively with different type pallets **18**, the latter being mounted in an array on the carrier **16**, as described hereinabove. As such, the continuous movement of the pallet-type character assemblies along a path extending across the width of, and closely adjacent to, the front side of the web **41**, makes it possible for each pallet **18** (with a type character die on the front face thereof), or groups thereof, to be successfully brought into momentary axial alignment with the hammers **85**.

The agency through which each impeller **84** drives an aligned hammer **85** against the back side of the web **22** comprises an interponent **92** (only several shown in FIG. **1**), preferably arranged in accordance with a third copending Babler application, Ser. No. 268,238, filed July 3, 1972, now U.S. Patent 3,822,641, herein incorporated by reference. Each of the interponents essentially comprises an elongated upright finger-like member, operably associated with one particular impeller **84** and hammer **85**. The interponents **92** are disposed in a lateral array, extending parallel to and respectively aligned with the adjacent free rearward ends **85***b* of the hammers **85**. Briefly described herein, each interponent can be positioned in: (1) a first or vertically raised upper position, with its upper free end portion disposed in the path of movement of a then immediately adjacent impeller spokelike element **84***a* (for transmitting force from the aligned impeller **84** to an aligned hammer **85**), in consequence of which printing occurs; or (2) a second or vertically lowered, nonprinting position, in which the upper free end portion of the interponent **92** is displaced from the path of an impeller spoke **84***a*.

The manner in which the hammers **85** are mechanically driven (as distinguished from electronically controlled) against the web **41** forms no part of the present invention. For details relating to the cooperating mechanical relationship between the impellers **84**, hammers **85** and interponents **92**, reference is made to the three copending Babler applications previously cited.

An inked ribbon **95**, as depicted in only fragmentary form in FIG. **1**, is continuously driven in one direction or the other between, and in alignment with, the array of type pallets **18** and the web **41**. One preferred embodiment of an improved ribbon spool driving, reversing and tensioning mechanism for use with a printer of the type depicted in FIG. **1** is disclosed in a copending application of A. F. Riley, Ser. No. 345,407, filed Mar. 27, 1973, also assigned to the assignee of the present invention, and incorporated herein by reference.

For the purposes of understanding the logic control circuitry of the present invention, and its significance in an on-the-fly, endless carrier type of impact printer, it will suffice to simply state at this point that the printer **15**, as thus far described, constitutes a line-at-a-time printer, i.e., a plurality of characters are normally sequentially printed across the width dimension of the web **41** during each of a plurality of print cycles that occur between successive index advancements of the web **41**. The printing of actual images on the back or rear side of the web **41**, as viewed in FIG. **1**, is accomplished, of course, by the impellers **84** generating, and transferring through the respectively associated and selectively actuated interponents **92**, sufficient force against the respectively associated hammers **85** to cause the latter to be driven against the back side of the web **41**. The discrete hammer-propelled areas of the web **41** are then, in turn, driven against correspondingly aligned areas of the aligned inked ribbon **95**, and then driven against the particular type characters on the pallets **18** aligned therewith with sufficient force to effect controlled impact printing of character images on the front side of the web **41**.

The logic control circuitry required to selectively and sequentially actuate the interponent-hammer combinations through the energization of respectively associated electromagnets **126** will now be described in greater detail. Mechanically, each electromagnet **126**, when energized, serves to pivot a corresponding armature **127** such that it elevates a selected interponent **92** into the path of an oncoming impeller tooth **84***a*. This propels the associated hammer against the backside of the web **41** and thereby effects the printing of a character. The armatures **127** are preferably arranged in accordance with a fourth copending Babler application, Serial No. 292,003, filed September 18, 1972, now issued as U.S. Patent 3,805,695, herein incorporated by reference. The electromagnets are also preferably constructed, arranged, and energized through physical circuit connections in the manner described in the commonly assigned copending application of James F. Kearney, Ser. No. 290,192, filed Sept. 18, 1972, now U.S. Patent 3,785,283, and herein incorporated by reference.

As simply represented in block diagram form in FIG. **1**, a unique and improved composite hammer logic control circuit, identified generally by the reference numeral **130**, is employed for sequentially energizing the hammer associated electromagnets **126** at the proper times, with respect to desired type character registration, in order to effect the printing of stored characters along each successive print line sequentially, but ultimately positioned in the serial order in which they were initially received. It is to be understood, of course, that the hammer logic control circuitry **130** described and claimed herein produces type character-associated control signals, relative to selected column positions, that may be employed to actuate any type of print-inducing means, whether of the solenoid or magnet-driven hammer type, or of the non-hammer type, wherein selectively pulsed magnetic fields, for example, are utilized to inductively impact magnetic type characters mounted on a nonmagnetic carrier, or disk, against a print medium.

CARRIER CONTROLLED TIMING CIRCUIT FOR COMPOSITE LOGIC CIRCUITRY

As also depicted in FIG. **1**, timing pulses for the logic circuitry are derived directly from the actual speed of the type carrier **16**. More specifically, through the utilization of a code disk **131**, with appropriately sensed markings **131***a* thereon, and a suitable conventional transducer **132**, shown only symbolically, the basic timing signals, designated herein as a TYPE CARRIER ADVANCE (TCA) and a TYPE CARRIER ADVANCE THREE (TCA-3) are generated to synchronize movement of the type carrier **16** relative to the firing times of selected ones of the array of hammers **85**. Precise synchronization is very important, of course, if

accurate type character-hammer registration is to be established as required for high quality printing.

Each TCA pulse senses when a new type carrier-print column alignment has occured, and conditions the logic circuitry to start a new process cycle. Three TCA pulses occur while the type carrier moves the distance between two adjacent type pallets **18** (0.150 inch in one embodiment). Stated another way, a single TCA pulse occurs each time a type pallet has moved a third of the distance between adjacent type pallets, which corresponds to each possible printing instant as described in the Babler application Ser. No. 268,236.

Each type carrier generated TCA pulse is thus employed to provide an indication to the hammer logic circuitry **130** (FIG. **1**) that there is a new alignment of font type characters and print columns of the printer. Each TCA-3 pulse is normally generated during every third TCA pulse time slot and when it is missing from its customary third slot position, this provides an indication to the logic circuit that the first type character of a given font on the carrier is aligned with the first, or some other chosen print column position of the printer. A TYPE CHARACTER RESET (TCR) pulse is then generated and indicates that the first type character of a font on the carrier **16** (represented by a **0**, for example), aligns with the first (or some other reference) column position of the printer.

GENERAL DESCRIPTION OF I/O CONTROLLER AND OF LOGIC CONTROL CIRCUITRY

With particular reference now to FIG. **2**, it is seen that an I/O controller **134** essentially functions as a buffer between a conventional Standard Serial Interface (SSI) and the composite hammer logic control circuit, which actually comprises a plurality of essentially identical and uniquely interconnected data processing circuit modules **135**. In conjunction with one illustrative **80** column printer, the logic control circuitry, in accordance with one illustrative embodiment, employs seven of such circuit modules **135** (only the first **135***a* and last **135***g* being shown in FIG. **2**), with each of the first six performing the necessary logic and data processing functions to effect the desired printing along twelve respectively associated columns of an eighty column printer. This, of course, results in the seventh sub-circuit module **135***g* being associated with only the last eight columns constituting a print line. It is to be understood that each circuit module **135** can be readily constructed to process the necessary print data and control the firing of either a larger or a smaller number of print hammers, and that there is no limit on the number of circuit modules that may be utilized in the interconnected manner depicted in FIG. **2**.

As also seen in FIG. **2**, each of the circuit modules **135** has a number of control leads connected to the input/output (I/O) controller **134**, which may be of conventional design. The purpose of the controller is to receive incoming encoded data to be printed, such as from the aforementioned conventional Standard Serial Interface (SSI), and to transform that data into the desired binary encoded form, including the necessary control information to operate the printer mechanism.

Considered more specifically, the input data to be processed for printing, when of a common format known as the American Standard Code for Information Interchange (ASCII), for example, wherein the binary numbers **1** through **32** constitute special commands for non-printable characters, and the numbers **33** through **126** constituted printable character data, is changed into a format of weighted binary order so as to be compatible with the logic circuitry embodied herein.

This data is received by and transmitted from the I/O controller **134**, as depicted in FIG. **2**, over a four wire (double twisted pairs) input/output connection designated generally by the reference numeral **141**, from a conventional parallel interface **142**, which serves no part of the present invention and, thus, is shown only symbolically. The data transmitted over the two twisted pairs of leads **141** is accomplished, for example, by switching the operating signal currents, between voltage levels of 0 and −5 volts DC, supplied from a voltage source (not shown). A conventional differential receiver (not shown) in the I/O controller **134**, may be used to detect the resulting voltage shifts.

In one preferred embodiment, all print data (i.e., ASCII print data and message control characters), as well as other necessary control information for the printer, is sent to the logic control circuitry **130** in the form of an 18-bit word, with such data being transmitted serially at 56 k baud. Only control information is sent from the printer to the I/O controller **134**.

As depicted in FIG. **2**, the most significant non-printable command signals, identified by their respectively associated leads, comprise the TYPE CARRIER ADVANCE (TCA) sync lead **163**, TYPE CARRIER RESET (TCR) sync lead **165**, END OF LINE (EOL) lead **168**, LINE FEED lead **169**, REQUEST NEXT CHARACTER (RNC) lead **172**, INITIALIZE lead **174**, and EMPTY lead **176**. As will be described in greater detail in connection with FIG. **10**, the EOL lead **168** actually includes what essentially constitutes a common OR-buss on which EOL signals may be sent from the I/O controller **134** or from circuit module **135***g* to all other circuit modules **135**, under specified circumstances. Similarly, the LINE FEED lead **169** also includes what essentially constitutes an AND-buss, in that all of the circuit modules must send a LINE FEED signal (−**1**) to the I/O controller before the latter, after a specified time duration, forces the bus to a NO LINE FEED state (−**0**). The circuit modules **135** then detect this latter state and release the LINE FEED lead **169**. The significance of all of the control signals will be described in greater detail hereinafter.

The printable character data, as distinguished from the non-printable command signals, is transmitted over Serial Data lead **181**, and generally comprises encoded alphanumeric characters, punctuation marks, etc.

As is also readily seen in FIG. **2**, all of the input-output leads of the controller **134**, with the exception of the TCR lead **165**, are directly connected to each of the logic circuit modules **135***a–g*. The TCR signal is successively passed from one sub-circuit module **135** to the next after some internal logic functions are carried out. Considered only briefly at this point, before each circuit module **135** is ready to process print data and to selectively actuate any of the proper hammers **85** associated therewith, after it has been INITIALIZED, it must have received a TCR signal and passed it through its own circuit module to the next one.

A given circuit module **135** accepts coded input data to be printed only if the immediately preceding circuit module has the proper 12 column storage area full, and if its own proper 12 column memory storage area is not full. This, of course, does not mean that there has to be a printing character stored in every slot position of a

given storage area, as there may be one or more columns where no data is to be printed (as represented by SPACE characters), or where a character has already been printed and the memory slot therefore is empty. In such cases, the circuitry simply recognizes this fact and acts as if the memory were full.

A PREV CKT FULL STATUS signal is shown only symbolically in FIG. **2** by leads **197***a* and *g* within the outline of the first and seventh circuit modules **135***a* and *g*. The OWN CKT FULL STATUS signal is similarly only shown symbolically by leads **199***a* and g in the same circuit modules. The logic circuits for these signals and their significance will be described in greater detail in connection with a description of the composite hammer logic circuitry depicted in FIGS. **3–10.**

A signal lead **201** connects each successive circuit module to the next, and actually comprises an extension of the TCR lead **165** depicted in FIG. **2.** As the lead **201** actually interconnects successive modules **135,** it is shown symbolically in FIG. **2** as having both an output portion **201''** and an input portion **201'**. In actual operation, two different encoded signals are multiplexed on the intercoupling lead **201,** one signal comprises OWN CKT FULL STATUS information with respect to a particular preceding adjacent one of the circuit modules **135***b–g* (excluding **135***a*), and the other signal comprises TYPE CARRIER RESET (TCR) information. The TYPE CARRIER RESET (TCR) information is shifted into and out of each of the seven circuit modules **135***a–g* in succession.

Considered more specifically, the TCR signal transmitted over the lead **201** is actually shifted through an eight element shift register **261** (not shown in FIG. **2,** but described below in connection with FIGS. **3** and **4**), at a rate corresponding to the periodic alignment of type characters with print column positions (or hammers) of the printer. As mentioned hereinabove, the generation of the LINE FEED (lead **169**) and EOL (lead **168**) signals between the I/O controller and circuit modules controls the processing of each new line of binary encoded information, and the ultimate printing of the characters represented thereby.

Two additional inputs to each circuit module **135,** which are identified by leads **203***a* and **203***g* in FIG. **2,** provide information as to the number of characters in the largest character font on the type carrier. The significance of these inputs will become more apparent hereinbelow.

As also generally depicted in FIG. **2,** each circuit module **135** produces a maximum of **12** possible output print signals for respectively driving directly, and selectively, **12** hammers **85** (FIG. **1**), or any other type of print-inducing means, out of the total array thereof, which total is equal to the number of print columns of the printer. These signals are respectively sent over **12** output leads designated generally by the reference numerals **205***a* and **205***g* in FIG. **2.**

A plurality of printer signal control leads, forming a part of a cable **210** in FIG. **2,** also interconnect the I/O controller **134** with the printer. Such printer control leads transmit the necessary control signals to effect, or provide information of, by way of example, the following printer conditions: forms versus roll paper, single versus double line feed, low paper, paper out, local line feed versus form feed, end of forms, line feed magnet energization, and motor on/off. These signals are normally not binary encoded and may be generated as simple voltage level changes in a conventional manner and, thus, will not be considered in further detail herein. The previously mentioned TCA and TCA-3 signals, produced for carrier hammer synchronization, are also transmitted between the I/O controller **134** and the transducer **132,** associated with the printer carrier **16,** over separate leads in the group forming the cable **210** in FIG. **2.**

Before considering the basic logic circuitry in greater detail, it is believed beneficial to describe at this point the nature of the font(s) carried by the carrier **16,** as well as the manner in which they are positioned on the type pallets **18** in two illustrative impact printers. In an 80 column printer, for example, the type carrier has 192 type pallets mounted within a corresponding number of slots spaced equidistant therealong, whereas in a 132 column printer, there are 288 type pallets mounted within a corresponding number of slots.

Every font of type characters on the carrier **16** must begin with a SPACE character and have the subsequent characters positioned in increasing binary order (according to the ASCII code, for example). When two or more fonts are employed, each font may be terminated on any character, as long as every slot of the carrier **16** has a type pallet mounted therein. As a SPACE is not a printing character (although it is assigned a type pallet slot), the type pallet for a predetermined ERROR symbol is inserted in that slot. This ERROR pallet also identifies the particular font set, if two are employed, by means of two peculiar alpha characters, for example, formed on the face of the type pallet in question.

When two or more fonts are employed, they may be of equal or different lengths, with each being repeated as often as is required in order to completely fill the slots in the type carrier **16.** As each of the character fonts, as previously mentioned, must begin with a SPACE, all of the characters of the shorter font are necessarily contained in the longer font. The 64 (including SPACE) and 96 (including SPACE and DELETE) character fonts are most commonly used, as three fonts of the former and two of the latter will evenly fill the **192** pallet slots in the carrier of an 80 column printer. To fit a 64 character font on the carrier for a 132 column printer of the type embodied herein, four complete fonts plus the first **32** characters of a fifth font would be required to fill the 288 slots in the carrier with pallet-characters.

OVERVIEW OF INDIVIDUAL CIRCUITS FORMING ONE COMPLETE CIRCUIT MODULE

Attention is now directed to FIG. **3,** which discloses in simplified form the basic hammer logic circuitry as embodied in just one of the essentially identical data processing circuit modules **135***a–g* depicted only symbolically in FIG. **2.** A more detailed description of the circuitry and of the various functions thereof will be described in connection with a discussion of FIGS. **4–10** hereinbelow.

Starting at the point of data entry in FIG. **3,** the print data initially is typically received in preferably converted ASCII data format from the output of the I/O controller **134** in serial format, and then supplied through a shift register **220** to a dual storage free-running memory **221.** As previously mentioned, the memory is effectively sub-divided into two distinct **12**

element storage areas, designated stores 1 and 2 herein, by logic control circuitry (to be described in greater detail hereinbelow) associated with the input and output thereof. Such dual storage areas advantageously allow data previously stored in one storage area for one print line to be read out and further processed (so as to effect the printing of characters represented thereby), while data for the next print line is being received in the other storage area.

The binary encoded character data read-out of each memory store together with the character data generated in a type character font generator **230**, are compared in an 8-bit comparator **232**. The binary encoded font characters are generated in successive groupings (of eight character-numbers each), with each group starting with the next higher order character as they appear on the carrier **16**. The character font generator **230** is driven at two-third the frequency *f* of the storage memory **221**, and with a TCA/$_3$ ENABLE control signal applied to an input thereof, only every third output encoded character from the memory is effectively compared with every other encoded type character generated by the font generator **230**. This selective sequence of signal comparisons thus satisfies the 1.5 pitch relationship between the type characters and hammer-print columns, as best illustrated in FIG. **11**. In that figure, it is readily seen that only every other type character on the carrier **16** registers with every third print column at any one instant.

The output of the comparator **232** comprises serialized, binary encoded data for selectively energizing the drive magnets **126** for the print hammers in sequential groupings, after such data has been time-delayed and selectively routed through a hammer data store and drive control circuit **235**, described in greater detail hereinafter.

A type character position counter **241** and a print column counter **243** are employed, together with a 4-bit comparator **249**, to gate the input binary encoded data into a given one of the stores of the memory **221** on an assigned basis. More specifically, the position counter **241**, as in the case of the memory **221**, is free-running at the frequency f, which in one illustrative embodiment is 56 KHZ. As such, the position counter **241** successively counts through the 12 positions or elements associated with one of the two stores of the memory **221**, before being recycled to count through the 12 positions or elements of the other store. The column counter **243**, on the other hand, successively counts the number of encoded characters that are actually loaded into each respective store of the memory.

The outputs of the position counter **241** and the column counter **243** are then compared in the 4-bit comparator **249**, with each equal comparison output signal therefrom controlling the selective loading of encoded characters into the proper slots of each store of the memory **221** on an assigned basis. Such selective loading is accomplished, in part, by a load, line feed and process control circuit **251**, which supplies LOAD signals over leads **252** and **253** to the memory **221** and column counter **243**, respectively. The nature and functions of the control circuit **251** will be described in greater detail hereinbelow.

A phase counter **255** and a 2-bit comparator **257**, in conjunction with the composite control circuit **251**, are employed to generate the one-third *f* and the two-third *f* control frequencies. These clock frequencies are necessary for ascertaining which one of the three sub-scans or phase sequences at any given time should be processing binary encoded data to be printed, and for synchronizing and controlling the data storing, read-out and comparing operations involved during each of the sub-scans.

An 8-bit shift register **261** is employed to keep track of each received TYPE CHARACTER RESET (TCR) pulse, which indicates a beginning of character font, and to reset the type character font generator **230** so that the latter will start a new count sequence for stored signal comparisons in the comparator **232** at such times. This is accomplished by feeding each TCR pulse in succession from the first processing circuit module **135***a* (FIG. 2) to the second, and from the second to the third, etc., over the interconnecting lead **201**, depicted in FIG. **2**.

FIGS. **4A** and **4B** disclose the various sub-circuits comprising one circuit module in somewhat greater detail than depicted in FIG. **3**. In particular, it is seen in FIG. **4A** that the phase counter **255** drives a clock **262** which gates the firing of a hammer control circuit **235***a*, which forms part of the circuit **235** depicted in FIG. **3**. A 4-bit shift register **235***b* and **12** magnetic drive circuits **235***c* completes the simplified hammer driver circuit **235** of FIG. **3**.

With reference to the storage memory **221**, the output thereof, as depicted in FIG. **4A**, is compared with the type character data supplied from the process character counter **230***a* in the 8-bit comparator **232**. The counter **230***a*, together with a parallel connected master character counter **230***b*, comprises the simplified type character font generator **230** depicted in FIG. **3**. The process counter **230***a*, when enabled, generates eight successive binary encoded output signals, with alternate ones being representative of the four font characters that selectively align with the particular four (out of twelve) print columns associated therewith during each sub-scan period.

Any equal signal comparison that is subsequently made in the 8-bit comparator **232**, as depicted in FIG. **4A**, produces an output binary encoded character signal that is fed to the 4-bit shift register **235***b*. Circuits **235***a–c* together direct the print signals successively applied thereto to the particular ones (up to four) of the 12 print hammer magnets **86** (FIG. **1**) having the proper type characters respectively in register therewith during any particular one of the aforementioned sub-scan periods (each being defined by a TCA pulse, with adjacent pulses having a time duration of 840 us therebetween). Further details as to how the print hammer signals are stored, stretched in time duration, and routed will be discussed hereinafter in connection with FIG. **8**.

FIG. **4B** depicts in somewhat greater detail the nature of the load, line feed and process control circuit **251** of FIG. **3**. To that end, the latter circuit is shown in FIG. **4B** as being subdivided into a load control circuit **251***a*, a process control circuit **251***b* and a line feed control circuit **251***c*. The nature and function of the load control circuit **251***a* will be considered in greater detail in connection with FIG. **6**, whereas circuits **251***b* and **251***c* will be considered in greater detail in connection with a description of the circuits depicted in FIGS. **9** and **10**, respectively, hereinbelow.

DETAILED DESCRIPTION OF ONE CIRCUIT MODULE RELATIVE TO COMPOSITE LOGIC CONTROL CIRCUITRY

In considering certain of the various block diagram circuits depicted in FIGS. **3**, 4A and B now in greater detail, from both a circuit construction and a functional standpoint, reference will be made to FIGS. **4–10** in particular. In the interest of clarity, like reference numerals will be used to identify corresponding circuit elemets in all of the drawings whenever applicable. It should also be understood that the detailed descriptions hereinbelow of the various circuit elements and associated circuitry will be made with reference to a typical rather than specific one of the circuit modules **135***a–g*, unless otherwise indicated.

The code converted output from the shift register **220,** (FIG. 4A), is fed over a lead **263** to the input of the dual storage memory **221** which, in one preferred embodiment, comprises an 8 × 24 bit dynamic shift register of the type depicted in greater detail in FIG. **5.** The data in this dynamic memory is recirculated at a frequency *f* and is accessed, so as to allow input data to be stored serially therein, but then sub-divided, positionwise, into two distinct storage areas, generally referred to hereinafter as STORES **1** and **2.** Each storage area in accordance with the principles of the present invention is capable of storing character data for 12 columns of the associated printer for a given one of two adjacent lines to be printed. As the storage memory is free-running, selective storage in and read-out of data from the two discrete stores thereof is made possible by input and output logic control circuitry described in greater detail hereinbelow.

As best seen in FIG. 4A, the dyamic, free-running storage memory **221** is not only supplied with a (56 KHZ) clock signal, having the frequency designated *f* herein, but with a load signal. The manner in which the load signal is generated and its significance will become more readily apparent after a description of some of the logic processing and frequency generating circuits.

With the memory **221** being subdivided into dual (12 slot) storage areas, it becomes readily apparent that after every 24 advancements of encoded character data therethrough (e. g., at a rate of 17.85 microseconds per bit, based on the clock rate of 56 KHZ), each successive loaded character will reappear (i.e., be recirculated, unless erased after printing), at the output of the memory. Thus, as best seen in the timing diagram depicted in FIG. **12,** the actual data processing time for each sub-scan period only encompasses approximately 214 us out of the total of 840 us alloted for each sub-scan period. The non-scan portion of each sub-scan period is required primarily for electromechanical functions of the printer, for energization of the print hammer magnets, and for resetting some of the logic circuitry associated with mechanical functions of the printer. Such functions will be described in greater detail hereinbelow.

During the data processing portion of each sub-scan period, any encoded characters stored in the proper third of the **12** slots of one store are actually compared with the four particular type characters — pallets that are then momentarily respectively aligned with four (out of **12**) of the print columns (and hammers) of the printer. As previously mentioned, for each TCA pulse-initiated sub-scan period, the carrier **16** advances the type pallets **18** one half the distance (0.050 inch in one illustrative embodiment) between print columns, with three such sub-scans coinciding with the spacing between adjacent type pallets.

It thus becomes readily apparent that for the 1.5 pitch relationship between type pallets and print columns that exists in the illustrative printer embodiment, that three sub-scan periods are required in order for every print column (and hammer) to become aligned with a type character on the carrier **16.** This can best be seen from an examination of the type characterprint column registration sequences symbolically depicted in FIG. **11.**

Advantageously, by utilizing a plurality of subdivided dual storage memories **221,** each forming a part of a discrete circuit module **135,** incoming encoded data to be printed may be received, and printing commenced in the print columns associated with that memory, independently of all of the other memories. This advantageously allows the process logic circuitry scanning rate to be considerably less than would be the case with only one large memory.

In addition, the modular construction of the logic control circuitry, by utilizing independent circuit modules **135** interconnected and controlled in the unique manner depicted in FIG. **2,** advantageously allows the processing of print data concurrently (as well as selectively) therein. It thus follows that processing may start in any particular circuit module as soon as a data character has been stored therein, without having to wait for all of the data characters to be stored either with respect to a given memory **221,** or with respect to all of the memories, as required for a full print line.

Also as previously mentioned, the utilization of dual memory stores advantageously allows the incoming data to be received and processed over an appreciable variable rate of transmission, as one line may be temporarily stored, if necessary, until the previously stored line of encoded character data has been printed.

With reference now to the character position counter **241,** previously identified in FIG. **3,** but best seen in FIGS. **4**B and **6,** it also is free-running with the storage memory **221** (at *f*). This counter is preferably of the type wherein a modulo three counter drives a modulo four counter so as to produce repetitive 12 count outputs. Each stage of the counter **241** (excluding a final stage) is employed to identify a distinct slot or position in one of the two stores of the memory **221.** The counter **241** is adapted to count down rather than up primarily because of the direction in which the type characters are transported by the carrier **16** past the array of print columns (and hammers), and because of the manner in which the encoded type characters are loaded and recirculated in the dual storage memory **221.** This will become more readily apparent in a more detailed description of the composite logic circuitry hereinafter.

The position counter **241,** as depicted in FIG. **6** not only produces repetitive 12 count sequences alternately for each store of the memory **221,** but is wired so that the last stage thereof, such as a flip-flop **241'**, toggles over after every 12th count generated by the counter. This provides an indication over leads **264** and **265** of, and conditions other related circuits to be described hereinbelow with respect to, which of the two

memory stores are to be scanned, loaded, compared and otherwise utilized at any point in time.

The column counter **243**, initially identified in FIG. **3** and depicted more specifically in FIGS. **4B** and **6**, is preferably constructed as a modulo 12 counter in the same fashion as the character position counter **241**, and is advanced by a LOAD CHAR signal once for every character that is actually loaded in the memory **221**, up to a maximum of 12 characters for any one line printing cycle. By comparing the outputs of the character position counter **241** and the column counter **243** in the 4-bit comparator **249** (briefly identified in FIG. **3** and disclosed more specifically with associated circuitry in FIGS. **4B** and **6**), the next empty slot for storage in the memory **221** is determined.

This is accomplished in the following manner. With particular reference to FIG. **6**, in response to each equal signal comparison detected in comparator **249**, an output signal is produced, applied to, and partially enables an AND-gate **266** which conditionally provides an output signal designated LOAD CHAR On a lead **268**. This signal is applied to and controls the loading of characters into the particular memory **221** by periodically conditionally enabling one of each pair of a plurality of input AND-gates **270***a*—*a* to **270***h*—*h* (FIG. **5**), each pair respectively being associated with a different one of the eight vertical planes of the memory. The LOAD CHAR signal is also applied as a clock pulse input to the column counter **243**, as seen in both FIGS. **4B** and **6**.

In addition, the LOAD CHAR signal, on lead **268**, is applied to an OR-gate **272** (FIG. **5**), which forms part of a reset circuit for the (13 bit) shift register **220** (also depicted in FIGS. **3** and **4A**). Considered more specifically, after each binary encoded character has been loaded into the associated memory **221**, there is no further need to store the character in register **220**. When characters are not to be accepted by this particular circuit module **135**, the negative of an ACCEPT CHAR signal on a lead **275**, will also cause the shift register **220** to be reset. Since the LOAD CHAR signal is only one clock pulse wide, each such signal also only advances the column counter **243** (as best seen in FIGS. **3** and **4B**) up by one count and, as previously mentioned, only allows the loading of one encoded character in one particular store of the memory **221** at a time. The manner in which the ACCEPT CHAR signal is derived at the output of an AND-gate **277** in FIG. **6** will be described in greater detail hereinbelow.

It is also seen in FIG. **6** that an actuable latch **278**, driven by the output of the column counter **243**, through an OR-gate **279**, is set to an OWN STATUS FULL state each time the column counter counts to 12. This indicates that a given store of the associated memory **221** is full. Latch **278** may also be set to an OWN STATUS FULL state by means of a second input to the OR-gate **279**, connected through a lead **280** to one output of a 4-bit comparator **281**. Thus, regardless of how the latch **278** is set, it is capable of producing an OWN STATUS FULL signal, but only when: (1) data characters have either been loaded into all **12** slot positions of the proper one of the dual stores of each memory **221** (as indicated by the output of the column counter **243**), or (2) when an equal signal comparison has been detected in, and an output is produced by, the 4-bit comparator **281** of the last circuit module. The latter output indicates that the last stage memory **221** (*g*) as been loaded with a data character to be printed in the highest column of the printer.

Considering the significance of the 4-bit comparator **281** (FIG. **6**) used in only the last circuit module **135***g* in greater detail, it is needed because the highest print column assigned for processing therein need not be the same as for the other circuit modules. For example, in an 80 column printer of the type illustrated herein, the sub-dividing of the print columns into groups of 12 results in the last group not being assigned twelve print columns, but rather, only eight. The last circuit module **135***g* in the present case therefor obviously has some unused print column processing capacity. This is most clearly indicated by the type character-print column alignment associated with the 7th circuit module **135***g* depicted in FIG. **11**, where it is seen that print data for only print columns 73 to 80 is actually being processed therein. As such, it is readily appreciated that there must be logic circuitry for ascertaining when the output of the column counter **243** in circuit module **135***g* corresponds numerically with the highest column of the printer.

To that end, and as seen in FIG. **6**, the output of the column counter **243** is continuously compared with a HIGH COL. COMP signal, generated by the I/O controller **134**, in the 4-bit comparator **281**. The latter signal is representative of the highest column number of the printer, and is transmitted on the SERIAL DATA lead **181** (FIG. **2**) as part of each encoded data character, which typically comprises eight ASCII coded data bits and four high column data bits. Thus, each sensed column count comparison in the last circuit module **135***g* is indicative of the fact that the associated memory **221** has stored a character therein for the highest print column available in the printer. As such, it is appreciated that the 4-bit comparator **281** (FIG. **6**) is only used in the last circuit module **135***g*.

A second output of the 4-bit comparator **281** in the last circuit module **135***g* not only transmits an EOL signal back to the I/O controller **134**, but to every other circuit module (**135***a*–*f*), over its lead portion **168'**, which is connected to a common bus represented as part of lead **168** in FIG. **2**. An EOL signal appearing on this lead **168** indicates to all of the circuit modules **135** that all data for the present line has been received and that any data characters subsequently appearing on the SERIAL DATA lead **181** are to be printed on the next line. Generally, the I/O controller **134** sends an EOL signal to all of the circuit modules on the lead (or bus) **168** in response to a new line command from the SSI Interface. As previously noted, the last circuit module **135***g* may also output on the EOL lead **168** whenever data has been stored in the memory slot associated with the highest print column of the printer. It should also be noted that it is because of the two-way communications that takes place over the EOL leads **168, 168'** in the last circuit module **135***g* (FIG. **2**), that bidirectional arrows are shown relative to the input of only that module.

From the foregoing, it is seen that only the last circuit module **135***g* utilizes the HIGH COL COMP signal to initiate an EOL signal at the output of the comparator **281**, and then only when the latter is enabled by AND-gate **282** (FIG. **6**). One input to the AND-gate comprises the previously described LOAD CHAR signal, with the other input comprising a LAST CKT signal. The latter signal is actually generated in the illustrative

embodiment by a simple strap option connection which, as shown, may ground (or otherwise change the signal level on) a LAST CKT lead **284***g* depicted in FIG. **2.** It should be understood, of course, that a LAST CKT condition may be established in any one of a number of conventional ways, to indicate to the last circuit module (whether **135g** or some other one) that it is to perform the high column comparison.

As also depicted in FIG. **6,** the ACCEPT CHAR signal on lead **275** is produced at the output of the AND-gate **277** whenever an OWN STATUS FULL signal is not present, which results in one enabling input thereto by reason of an inverter **287,** and a PREVIOUS CKT FULL STATUS signal on a lead **288** being active. These operating conditions indicate that a particular circuit module **135** is in an operating state wherein it can accept and subsequently store data characters applied thereto on the SERIAL DATA lead **181.**

Attention will now be directed as to how signals transmitted on the ACCEPT CHAR lead **275,** and on the previously identified RNC bus **172,** control data character transfer between the I/O controller **134** and the circuit modules **135***a–g.* As seen in FIG. **2,** all of the circuit modules **135***a–g* output on the RNC bus **172,** which essentially functions as an OR-bus. More specifically, and with particular reference to FIG. **5,** only one particular circuit module **135** at any given point in time has an active ACCEPT CHAR signal appearing on the associated lead **275.**

This signal will output onto the RNC bus **172** and, thereby, request the I/O controller **134** to send a data character to be printed back to the circuit module **135** in question on the SERIAL DATA lead **181.** When such a data character is received by the circuit module, a CHAR AVAIL signal is produced at the output of the shift register **220.**

The RNC bus **172** is released by each circuit module **135** in response to the receipt of a CHAR AVAIL signal on a lead **289** (FIG. **5**), which signal in passing through an inverter **291** disables an AND-gate **292.** The subsequent loading of a data character in a given store of each circuit module memory **221** then resets the shift register **220** by reason of a LOAD CHAR signal being applied to the input of the previously identified OR-gate **272.** This results in a CHAR AVAIL signal level change over lead **289** so as to enable the AND-gate **292** and results in a REQUEST NEW CHARACTER (RNC) signal being sent over the similarly designated bus **172.**

As will be described in greater detail hereinbelow, when two EOL command signals have been received without either one having been executed in conjunction with a new line feed command by the printer, a PRINTER BUSY signal is supplied over a lead **293** to the input of an inverter **294,** which signal turns the RNC bus **172** off, indicating that no more characters are requested from the I/O controller **134.** The PRINTER BUSY signal is generated by circuitry depicted in FIG. **9** which will be described in detail hereinbelow.

Considering at this point the PREV CKT FULL STATUS signal for each circuit module **135** in greater detail, it is generated by the flip-flop **295** (FIG. **7**), which is selectively enabled by a pair of AND-gates **296** and **298** respectively connected to the J and K inputs thereof. One input to the AND-gate **298** is connected through an inverter **299** to the incoming data lead **201′.**

The other input of each AND-gate in question is connected through an inverter **301** and leads **302** and **303** to every 3rd TCA pulse output of an AND-gate **304.** The latter gate has two inputs connected to the two lower order bit outputs of the modulo 3 phase counter **255,** with a third input being connected to the TCA pulse input to the logic circuitry.

Before considering the circuit function of the flip-flop **295** in greater detail, it might prove beneficial to review briefly the significance of the input and output lead portions **201′** and **201″,** respectively, which are associated with each of the circuit modules **135** as depicted in FIGS. **2** and **7.** The incoming lead portion **201′** is actually a continuation of the output lead **201″** of the immediately preceding circuit module **135.** As previously mentioned, there are two multiplexed signals transmitted on each lead portion **201,** one being the PREV CKT FULL STATUS signal and the other being TYPE CARRIER RESET (TCR) information. At all times other than during every third TCA pulse, a PREV CKT FULL STATUS signal (or the absence thereof) is detected on the lead **201′.**

Thus, each flip-flop **295** of each circuit module **135** effectively follows the signal state of the associated lead portion **201′,** thereby sampling and storing information relative to the full status of the previous circuit module **135.** Upon receipt of every third TCA pulse, the flip-flop **295** is disabled and, thus, prevented from following the signal state of lead portion **201′.** As a result, the shift register **261** associated with the flip-flop **295** is advanced one slot, causing the output thereof to be fed to the input of the next succeeding shift register over the continuous lead portions **201″, 201′.**

BRIEF DESCRIPTION OF MEMORY LOAD CONTROL CIRCUIT

With reference again to FIG. **6,** between the position counter output stage **241′** and the AND-gate **266,** there is a memory load control circuit defined within the dash-lined box **251***a.* This circuit forms a part of the load, line feed and process control circuit **251** in FIG. **3,** and is disclosed by the similarly numbered box **251***a* in FIG. **4B.** With particular reference to FIG. **6,** circuit **251***a,* in addition to the previously described AND-gates **266, 277,** and inverter **287,** further includes a pair of AND-gates **307, 308** each having an input connected to a different one of the previously identified leads **264** and **265.** The other ends of these leads are respectively connected to opposite sides of the last-stage (e.g., flip-flop) logic element **241′** of the position counter **241.** It will be recalled that it is the toggled position of logic element **241′** after every 12th count that determines, in part, whether STORE **1** or **2** of each memory **221** has its output available for either the storing or processing of data. The other two inputs to the AND-gates **307, 308,** respectively designated STORE **1** and **2,** are connected to logic control circuitry of the type depicted in FIG. **9,** described in greater detail hereinbelow.

It will suffice to simply state at this point that a STORE **1** or **2** signal indicates that data is to be stored in either correspondingly numbered STORE of the associated memory **221.** The presence of either STORE signal results in an output from an OR-gate **309** being applied to an input of the AND-gate **266.** When an ACCEPT CHAR signal is present on lead **275** (indicating

that the associated circuit module **135** is the one to accept data characters), and when a CHAR AVAIL signal is present on lead **293** (indicating that a character has been received on the SERIAL DATA lead **181**), and when an output COMPARE signal is also provided by the 4-bit comparator **249** (indicating that the proper slot in the proper store of the memory **221** into which the next data character is to be loaded is available at that memory output), a previously described LOAD CHAR signal is sent over lead **268** to gate the input AND-gates **270***a*—*a* to **270***h*—*h* depicted in FIG. **5**.

BRIEF DESCRIPTION OF MEMORY PROCESS CONTROL CIRCUIT

Consideration will now be directed in greater detail to the process control circuit **251***b* depicted in only block diagram form in FIG. 4B. That circuit is shown in greater detail in two parts, namely, within the dash-lined boxes designated **251***b'* and **251***b''* in FIGS. **7** and **6**, respectively. With particular reference first to FIG. **7**, it is seen that every TCA(SYNC) signal received on the lead **163** (from the I/O controller **134**, FIG. **2**) is an indication that a new type carrier/print column alignment has occurred, as described hereinbabove, and that a new process cycle is to be initiated. The process cycle cannot, however, be started immediately since synchronism with the free-running position counter **241** and the memory **221** of each circuit module **135** must first be achieved. This is accomplished by circuit portion **251***b'* (FIG. **7**) in the following manner. Upon receipt of each TCA pulse on lead **163**, a flip-flop or latch **401** acquires an operating state indicative of that fact. With the latch **401** enabled, the output thereof, in turn, enables a latch **402**, and as it is clocked by the **12** count output of the position counter **241** (FIG. **6**) over a lead **403**, the latch **402** produces a synchronized output PROCESS ENABLE signal on a lead **404** (see FIG. **14** timing diagram).

Referring now to the processing circuit portion depicted in FIG. **6**, and defined within the dash-lined box designated **251***b''*, it is seen that the PROCESS ENABLE signal generated at the output of the latch **402** in FIG. **7** is applied over the lead **404** as a common gated input enable to two AND-gates **406** and **407**. The PROCESS **1** and **2** input signals are correlated with the STORE **1** and **2** input signals applied to AND-gates **307** and **308** described hereinabove, in that the former signals respectively indicate whether the data stored in STORE **1** or **2** of the associated memory **221** is to be processed. By gating the AND-gates **406, 407** with the high order bit output of the position counter **241**, applied through lead **265**, control is provided over whether STORE **1** or **2** of the memory **221** is available for access (either storing or processing). An output from either AND-gate **406** or **407** enables an OR-gate **408**, the output of which produces a SYNC PROCESS signal on a lead **409**. The latter signal provides lead **412**, **12** count interval at the proper time, during which the data available at the output of the memory **221** is to be processed. At the conclusion of this process interval (sub-scan cycle), the circuit portion **251***b'* (FIG. **7**), which is still generating the PROCESS ENABLE signal on lead **404**, is reset by means of the SYNC PROCESS signal applied to the latch **401** through an AND-gate **411**.

The manner in which the COMPARE ENABLE signal is generated and used will now be discussed in greater detail. As depicted in FIG. **6**, that signal appearing on a lead **412**. is generated at the output of an AND-gate **413** which forms a part of the previously identified PROCESS CONTROL circuit portion **251***b''* . Gate **413** has three inputs applied thereto, namely, a COMPARE signal from the 2-bit comparator **257**, an ANY STORED signal transmitted on a lead **415** from a multi-input OR-gate **417** (FIG. **5**), connected to the output of the associated memory **221**, and the previously identified SYNC PROCESS signal derived at the output of the OR-gate **408**.

As described hereinabove, the SYNC PROCESS signal on lead **409** provides one **12** count interval for every TCA pulse (i.e., a timed interval for every new type character/print column alignment) at the time the character data that is to be processed is accessible at the output of the memory **221**. The ANY STORED signal on lead **415** (FIGS. **5** and **6**) is produced, and applied as an input to the AND-gate **413**, only when a character is actually stored in the memory **221**.

In connection with the function of the 2-bit comparator **257**, it will be recalled that the MOD **3** phase counter **255** (FIG. **7**) is clocked once for each TCA pulse in a modulo **3** fashion and, thus, determines for any one sub-scan which four of the **12** slots in the associated memory **221** (each slot associated with a particular print column) are indicative of four particular print column-type character alignments, and are to be reviewed in the process cycle initiated by each TCA pulse.

Concomitantly, the position counter **241**, whose first two elements also count in a modulo **3** fashion (at the same free-running rate f at which data in the memory **221** is circulated), counts in an order such that each of its **12** counts (excluding the last stage **241'**) is associated with one of the **12** slots in the memory **221**. Comparison of the outputs of these two last-mentioned counters in the 2-bit comparator **257** (FIG. **6**) then generates the proper **4** out of **12** signals required to gate the AND-gate **413** (FIG. **6**) and, thereby, partially effect the generation of the COMPARE ENABLE signal, which is employed to control the proper phasing of the character data read out of each associated memory **221**.

The actual effect of the COMPARE ENABLE signal is that it only allows a comparison of character data that is not only read out of the proper one of the two distinct free-running memory storage areas, but also only during the times that the particular **4** out of **12** memory slots thereof correspond with the four type character-print column alignments for that particular process print cycle.

As also depicted in FIG. **6**, the COMPARE signal generated at the output of the **2**-bit comparator **257** is applied to one input of each of two AND-gates **421** and **423**, with the input to the latter being inverted by an inverter **425**. By utilizing the SYNC PROCESS signal as a gating input, the one-third *f* and two-third *f* clock frequencies are generated at the outputs of the AND-gates **421** and **423**, respectively.

From the foregoing, it is readily seen that upon receipt of each TCA pulse, the logic control circuitry is conditioned to go through a sub-scan print cycle. In such a process, there must be generated serial, binary encoded numbers for each of the eight characters carried by the type pallets that at any given time are in juxtaposed relationship with the **12** print columns associ-

ated with each circuit module **135** (depicted only in outline form in FIG. **2**).

MODE OF OPERATION OF TYPE CHARACTER FONT GENERATOR

The type character font generator **230** depicted in simplified form in FIG. **3**, is shown in greater detail in FIGS. **4A** and **7** as comprising an 8-bit process character counter **230***a* and an 8-bit master character counter **230***b*. Attention will now be directed to the manner in which these counters relate to the shift register **261**, and to a shift register **450** in performing the character generating function, with particular reference to FIG. **7**.

The master character counter **230***b* always contains the binary encoded number representative of the character/type pallet that is at any given time associated with the high order print column (12th) that is assigned to a particular circuit module **135**. Thus, when the TCR signal, as described above, is shifted into the high order A slot position (8th) of the shift register **261**, as depicted in FIG. **7**, the master counter **230**b is reset to a O state. This state of the counter **230***b* thus indicates that the first or beginning of font character on the carrier **16** has been moved into a position which is aligned with the 12th hammer **85** controlled by that circuit module.

On receipt of every TCA pulse applied to the phase counter **255** (as best seen in FIG. **7**), that pulse in the form of a binary encoded character is shifted into the process counter **230***a* (while also remaining in the master counter **230***b*) in preparation for a sub-scan print cycle. On every 3rd TCA pulse, the pulse also in the form of a binary encoded character is also counted up 1, and then shifted back up to the master counter **230***b* to reflect the next binary encoded character that has now been moved into the high order position. In this connection, it should be noted that in the illustrative embodiment, the type carrier **16** moves the type characters in each font from left-to-right, while the binary encoded numbers representative thereof are generated in increasing binary order from right-to-left.

In order to provide selective character signal comparisons in the comparator **232**, (FIG. **5**) of only the four type characters (every other one out of eight) and the four print columns (every third out of 12) that are in register during any one subscan period (see FIG. **11**), the process counter **230***a* clocks the output data therefrom at two-thirds the frequency of the associated free-running memory **221**. The output of the comparator **232** is gated by the input COMPARE ENABLE signal applied to an AND-gate **419** (FIG. **5**) in producing print hammer data signals which are transmitted over lead **488** to the HAMMER DATA STORE AND CONTROL circuitry **235** depicted generally in FIG. **3**, and disclosed in greater detail in FIGS. **4A** and **8**.

As a result, the process counter **230***a*, clocked at two-thirds *f*, counts upwardly eight counts, each being respectively representative of the eight consecutive type characters then positioned adjacent the twelve associated print columns of the printer, but with only four type characters being in actual alignment with print columns at any one time.

As also depicted in FIG. **7**, a logic circuit is associated with the output of each shift register **261**, and includes two parallel paths, one comprising an inverter **436** and an AND-gate **438**, and the other comprising an AND-gate **443**. The output of both AND-gates are coupled through an OR-gate **444** to the output lead **201''**, previously identified in connection with FIG. **2**. Either an OWN STATUS FULL signal together with an inverted 3rd TCA pulse must be applied to the input of the AND-gate **438**, or a 3rd TCA pulse — and the output of the shift register **261** must be applied to the input of the AND-gate **443** in order to transmit data over the lead portion **201''** to the next processing circuit module **135**. The OWN STATUS FULL input to the inverter **436** is derived from the output of the previously described latch **278** depicted in FIG. **6**.

With particular reference now to FIGS. **4A** and **7**, an 8-bit shift register **450** is associated with the 8-bit shift register **261**, and together provide conditional enabling signals to an 8-bit AND-gate compare circuit designated generally by the reference numeral **455** (shown only in block diagram form in FIG. **4A**). These three logic circuits are employed to reset the process counter **230***a* every time a **0** character (or some other beginning of font character), for example, is present in one or more of the eight type characters that are associated with a particular circuit module **135** during any particular subscan period.

Considering the significance of the shift register **450** in greater detail, at the beginning of a process cycle, a TCA marker bit is applied as an input to that register as depicted in FIG. **7**. At that time, the first slot position of the shift register **450** coicides with the high order or 8th slot position of the shift register **261**. The marker bit is thereafter shifted through the shift register **450** (right-to-left, i.e., in the same direction as the process cycle proceeds) at a frequency of two-thirds f which, in effect, allows successive slot position comparisons to be made between the shift registers **261** and **450**.

If a beginning of font character (such as **0** in one particular illustrative embodiment) appears at any particular slot position in the shift register **261**, in advancing from right-to-left through the eight positions thereof, and at a time when it coincides position-wise with the continuously shifted marker bit in register **450**, the common dual outputs from such coincident storage positions will enable the associated one of the plurality of AND-gates in a logic array **455**. This, is turn, will produce a Reset signal that is transmitted through a multi-input OR-gate **457** and a lead **459** to, and will reset, the process counter **230***a*. The binary encoded output of the counter **230***a* then corresponds to the first or start of font character on the carrier **16**. The count up of counter **230***a* then proceeds at a two-third *f* rate for the remainder of the sub-scan (process cycle), generating the proper binary encoded characters corresponding to the type characters that successively and sequentially are brought into alignment with the print columns and hammers that are assigned to the associated circuit module **135**.

With each successive shift register **261** shifting only TYPE CARRIER RESET (TCR) information through the respectively associated circuit modules **135** at the same rate, and in the same direction, as the type carrier **16** moves across the front of the printer, there advantageously is no need to shift the 8-bit binary encoded character representations through the circuit modules **135***a–g*. If the latter approach were undertaken, eight shift registers analagous to shift register **261** would be required per circuit module, for each of the eight character bits, and would also require either 14 additional input/outputs to accommodate such data, or would require a complex scheme of multiplexing. Inasmuch as the circuit modules **135** are advantageously implemented as an LSI package, the additional pinouts

would require a subdivision of the circuit logic into a larger number of circuit modules (each driving less than 12 columns), while the multiplexing scheme would entail a substantial increase in logic complexity. As such, either solution would add substantially to the cost of the composite hammer logic control circuitry.

STRAP-OPTION RECOVERY CIRCUIT FOR PURGING MEMORY OF INVALID CHARACTERS

There are two different recovery circuits and associated modes of operation embodied in the present hammer logic control circuitry for purging the memory **221** of any invalid or erroneous character data. Such data could be generated and stored, for example, as a result of spurious noise transmitted in a given message, or generated locally. Special recovery techniques are required since erroneous data characters may typically never compare with a generated binary representation of a type character and would, therefore, never be removed from a given memory **221.**

One circuit, depicted in FIG. **5,** is adapted for use with standard length character fonts, and incorporates two selectable strap options in a manner that allows erroneous data to be compared for printing, erased from the memory and an ERROR symbol printed therefor, in a manner very similar to that described hereinabove for valid data, and advantageously within the normal sub-scan process cycle. An auxiliary invalid character recovery circuit depicted in FIG. **10,** and described in detail hereinbelow, is adapted for use with non-standard character font lengths. That circuit necessarily employs a pre-determined print cycle time delay before the memory is purged of any erroneous data.

Considering at this point only the strap option controlled recovery circuit associated with each memory **221,** as depicted in FIG. **5,** it is seen that within the dash-lined box **470** is a dual strap option circuit comprising a plurality of AND/OR-gates **471** through **476,** a zero count detector **477,** and a three-input AND-gate **479.**

Considering circuit **470** more specifically, the AND-gate **471** is connected to the 6th and 7th parallel-by-bit outputs of the associated memory **221** so as to be enabled and provide a binary encoded output whenever stored data characters register at those designated memory outputs. The OR-gate **472,** by having one input connected to the output of the AND-gate **471,** and the other input connected to the 8th bit output of the memory **221,** provides an output whenever 6th and 7th bit outputs are detected, or when an 8th bit output is similarly detected. Considered another way, an output from the OR-gate **472** is produced whenever a binary representation of a data character equal to or greater in numerical serial order than **96** is detected at the output of the memory **221** in the illustrative embodiment. In a similar manner, the OR-gate **474** provides an output whenever a binary representation of a data character greater in numerical serial order than **64** is detected at the output of the associated memory **221.**

It is thus seen that when either the AND-gate **473** or **475** is enabled, which one depending upon the strap option chosen, the resulting output from either gate will enable the OR-gate **476.** It will be noted that the third input to the OR-gate **476** is a Recover signal applied thereto over a lead **481** that originates at the output of the other erroneous character detection circuit mentioned hereinabove, and depicted in FIG. **10.** As will presently be seen, if the latter auxilliary circuit is not employed, the RECOVER input to the OR-gate **476** is not required. The output of a multi-input AND-gate **483** is also connected to an input of the AND-gate **476,** and is used to effect the printing of an ERROR symbol for special characters, as described further hereinbelow.

Referring again to the OR-gate **476,** any **1** input applied thereto will result in a **1** output which provides a conditional input signal to the previously identified three-input AND-gate **479.** The center input to the latter gate is connected to the zero count detector **477.** This detector effectively interrogates the type carrier character encoded data appearing on an eight wire cable **484** connected to the output of the process counter counter **230***a* (FIG. **7**), and determines when the latter is set at a **0** count. The third input to the AND-gate **479** comes from the ANY STORED output of the previously identified multi-input OR-gate **417** (FIG. **5**). This latter gate as previously mentioned is connected to all eight parallel-by-bit outputs of the storage memory **221,** and produces an output whenever any data is stored in any of the **12** slots of a given store of the memory being interrogated.

Thus, the AND-gate **479** is only enabled by three coincident input signals supplied thereto. When enabled, the resultant output from the AND-gate **479** is applied to an OR-gate **486.** Either that input, or an input thereto from the 8-bit comparator **232** will, of course, provide an output therefrom on a lead **487** which can partially enable the previously identified inhibiting AND-gate **419.**

Thus, upon the AND-gate **419** having both a COMPARE ENABLE signal and an ANY STORED signal (indicating a stored character other than a SPACE) applied thereto, then, even in the absence of an output signal from the 8-bit comparator **232** (required for the printing of valid characters), an output ERROR signal for actuating the appropriate print hammer will be produced on a lead **488,** provided a FALSE COMP signal is produced at the output of the AND-gate **479.** The output on lead **488** also serves as a COMPARE signal which is sent over a lead **489** to control circuitry depicted in FIG. **10,** for reasons discussed in greater detail later.

The ERROR symbol is printed in the following manner: It will be recalled that the **0** count, by definition, is the first character of each character font on the carrier **16.** Thus, by placing the ERROR symbol on the type pallet **18** occupying that particular slot position along the carrier **16** (which actually constitutes an otherwise non-printing SPACE slot), the ERROR symbol will be printed in the same manner as any other valid type character whenever a FALSE COMPARE signal (from the AND-gate **479**) is simultaneously applied with the COMPARE ENABLE and ANY STORED signals as inputs to the AND-gate **419.**

The ERROR symbol is prevented from being printed when the SPACE character is stored in the normal message text because in being a binary **0** character, it cannot result in an ANY STORED signal being generated for it. Accordingly, in that instance, the AND-gate **419** is disabled and, hence, inhibits any print hammer out-

put signal representative of a SPACE character from being generated on lead **488.**

However, when a print signal does appear on lead **488,** regardless how generated, it is employed to effect the erasure of the stored data in the memory **221** that gave rise to it, upon such character being printed. Such operation is effected by enabling a NOR-gate **491** by either a COMPARE signal applied thereto over a lead **493,** or the previously identified INITIALIZE signal applied thereto over the lead **174** (see FIG. **2**). With the NOR-gate **491** enabled, an ERASE signal is sent over a lead **496** to the common inputs of the array of AND-gates **270***a*−*a* to **270***h*−*h*. The INITIALIZE signal applied to the NOR-gate **491** allows the associated memory **221** to be erased from an initial POWER ON RESET condition.

From the foregoing it is seen that the strap option circuit depicted within the dash-lined box **470** in FIG. **5** provides an effective way to rapidly detect the presence of any invalid data characters, or characters having a binary number higher than the chosen strap option font length, and to effect the erasing thereof from the memory **221** with no time loss being incurred in the process cycle. As mentioned hereinabove, an optional recovery circuit for purging the memory **221** of erroneous character data, by generating a RECOVER signal on the input lead **481** applied to the OR-gate **476** will be described in detail hereinbelow in reference to FIG. **10.**

DESCRIPTION OF ERROR SYMBOL PRINTING FOR SPECIAL CHARACTERS

The capability is provided in each of the circuit modules **135** of the illustrative logic control circuit to allow the I/O controller **134** to access the ERROR symbol included in each font of type characters, and to effect the printing thereof at any selected print column in place of any type character. For example, the I/O controller **134** can make a vertical parity check and choose to have incorrect vertical parity characters printed as the ERROR symbol.

Considered more specifically, and in reference to FIG. **5,** the previously defined multi-input AND-gate **483** is connected to all eight parallel-by-bit outputs of each memory **221,** and thus detects an all bit **1** condition, and provides a **1** output whenever the binary representation of character **255,** for example, is detected at the output of the memory. This output from the AND-gate **483** is transmitted over a lead **498** to the input of the OR-gate **476.** As described hereinabove in connection with purging each memory **221** of invalid characters, any **1** input to the OR-gate **476** will result in the ERROR symbol being printed for the associated binary representation(s) of any given type character(s) which would otherwise be printed.

HAMMER MAGNET DRIVE CIRCUITRY

Attention will now be directed to the active utilization of the PRINT HAMMER DATA derived at the output of the inhibiting AND-gate **419** on lead **488** (FIG. **5**). As best seen in FIG. **8,** that data comprises all of the serial information required to actuate the proper print hammers **85.** Considered more specifically, the serial encoded data on lead **488** is applied to the 4-bit shift register **235***b*, previously identified in connection with FIG. **4A.** As there can never be more than four print hammers in registry with different type pallet-characters in any one sub-scan or process cycle (for a 1.5 pitch relationship therebetween), there will never be more than four discrete hammer actuation signals supplied to the shift register **235***b* at any one time. This data is advanced to the latter shift register at the one-third *f* clock frequency which, of course, is the rate at which the type character data is advanced through the inhibiting AND-gate **419** (FIG. **5**), and controlled by the COMPARE ENABLE signal discussed hereinabove.

The information successively stored in the shift register **235***b* is thereafter shifted out of that register, upon receipt of each successive TCA pulse, and applied selectively to the particular one of three 4-bit latches designated **503, 504,** and **505** in FIG. **8.** Which one of the three latches receives information at any given point in time is controlled by three precisely phase clock inputs derived from clock source **262** (FIG. **4A**). These clock pulses are phase-controlled by the MOD **3** phase counter CTR **255** (FIG. **7**) which, of course, determines which third of the print hammers (or columns) are aligned with type characters on the carrier **16** for that particular sub-scan process cycle.

Three additional 4-bit latches **509, 510,** and **511** are respectively parallel coupled to the latches **503, 504,** and **505,** with the former latches effectively functioning as pulse stretchers. More specifically, through the utilization of the latches **509–511,** which are respectively and sequentially actuated by appropriately timed RESET pulses numbered **1–3,** generated by a conventional multi-phase clock source **513,** the hammer magnets **126** (FIG. **1**) may be actually energized for the time encompassed by four, rather than by three TCA pulse periods. This expanded time period is normally desired, if not required, for line printing speeds of the order of three or more lines per second. The pulse stretching effect is clearly illustrated in the sub-scan and print cycle timing diagrams depicted in FIG. **12.**

DESCRIPTION OF TIMING AND CHARACTER SLOT POSITION DIAGRAMS

In order to better understand the operating relationship of the dual storage memory **221,** relative to the position counter **241** and column counter **243** in each circuit module **135,** attention will now be directed to the pictorial diagrams depicted in FIG. **13,** and to the timing waveforms depicted in FIG. **14.** Starting with the bottom pictorial representation of STORES **1** and **2** of the memory **221** in FIG. **13,** it is seen that the slots or elements of each STORE are numbered in increasing order from left-to-right so that they will correspond spatially with the print column positions of the printer. It should also be noted that the slot numbers associated with each STORE are depicted with respect to one instant in time and, in particular, during a reset period, which, by definition, is the time when any data character to be printed in the **12**th column of a given line is accessible at the memory store output. The particular storage slot that is available at the output of the memory will change, of course, at the rate of the clock frequency (*f*) 56 KHZ, or once for every 17.85 microseconds in the illustrative embodiment.

Looking at the position counter slot position diagram at the top of FIG. **13,** it is important to note that rather than showing a picture thereof at one instant in time (as the memory **221** is illustrated), time is measured on the horizontal axis, increasing from left-to-right. As such,

that diagram represents each successive count, starting with the 12th, generated by the position counter **241** at the frequency *f*. Arranged in this manner, it is seen that during time slot **12** of each position count (reset position) of the position counter **243,** storage slot **12** of a given STORE is accessible at the memory store output. Similarly, each succeeding position count is associated with a corresponding storage slot upon becoming accessible at the memory store output. In contrast to the position counter **241,** the slot positions of the column counter **243** are numbered in the same order as the slots in the two STORES of each memory **221** for reasons which will become apparent by several illustrative examples.

First, let it be assumed that the data character A is to be loaded in slot **2** of STORE **1** of the memory **221,** as pictorially depicted in FIG. **13.** This implies that the column counter **243** is now at count **2.** As previously described, the loading of characters in each memory **221** is effected by comparing the outputs of the position counter **241** and the column counter **243** in the 4-bit comparator **249,** as best seen in FIGS. **4B** and **6.** If an equal signal comparison is effected in the comparator **249,** the particular data character to be printed in the then identifiable print column is loaded in the correspondingly numbered slot, namely, slot **2** of STORE **1** of the memory **221.** In order to accomplish this, the free-running position counter **241** must count downward until a count **2** output is generated therefrom, which will coincide in time with the count **2** output from the column counter.

At that point, the data character to be printed in column **2** is loaded into slot **2** of STORE **1** of the memory, because the latter is operated in synchronism with the position counter **241.** The same load pulse or signal employed to effect the storing of a data character in a given memory **221** also effects the counting upward of the associated column counter **243** from count **2** to **3.** As will be described hereinbelow, the logic circuits of FIGS. **9** and **10,** which generate the necessary PROCESS **1** and **2** and STORE **1** and **2** control signals, control the timing of the LOAD, SYNC PROCESS and COMPARE ENABLE signals so as to insure that each data character received for printing is stored in and read out from the proper store of each memory **221** at the proper times.

With the position and column counters **241** and **243** respectively operated in the manner just described, a very beneficial result is realized. Specifically, as soon as a data character is received in the STORE (**1** or **2**) being processed, printing may commence in the appropriate one of the **12** print columns associated with that memory as soon as the loaded character is read out and compared, even though all of the data characters to be stored in that memory have not yet been received.

One somewhat complicating effect of such sequential printing is that in a typical operating case, there will often by a number of empty slots in the accessed STORE of a given memory, because characters previously stored therein have already been printed, which results in those stored characters being erased. If the printing rate and the rate at which the data is received are closely matched, it is readily possible, and in fact often happens, that about half the slots in a given store would be empty to receive new data characters at most times during a given print cycle. It thus becomes apparent that with printing taking place in a sequential rather than serial manner, successive data characters cannot always be stored in the next available empty slot position of a given store of the memory.

By way of further illustrating this operating condition, assume that the first five data characters received in a given message are to be printed in the first five print columns and, thus, stored in the first five slots of memory STORE **1,** for example, and that before the sixth data character is received for storing the third data character is printed, and thereafter immediately erased from the memory in the manner described hereinabove. Under these conditions, it would be possible without the particular use of the position and column counters **241** and **243** (associated with each circuit module **135**), to have the sixth data character inadvertently loaded in the then empty third slot of the associated STORE, as that slot next presented itself in a free-running manner.

It thus becomes imperative that the position counter **241** continuously identify the rapidly changing output-input memory slot positions as they become available in a synchronous manner, and that the column counter **243** constantly keep track of each data character loaded in a given STORE, up to the maximum number of **12,** and relative to a particular time in the illustrative printer embodiment. Only in this manner can the outputs of the position and column counters be compared and effectively utilized to correlate the slot positions in the memory with the print columns in the illustrative printer embodiment.

Reference is now made to the particular timing waveforms depicted in FIG. **14.** These waveforms illustrate how the circuit depicted within the dash-lined box **251***b*′ (described hereinabove) of FIG. **7** provides synchronous timing intervals in response to successive asynchronous TCA pulses received from the printer mechanism (transducer **132,** FIG. **1**) so as to precisely define the duration of each process cycle. The position counter C_{+12} output waveform (*a*) is simply representative of each successive high order output count (12th) of a given position counter **241** (see FIG. **6**), and is the signal that is used to achieve synchronism. As described hereinabove, the output of the flip-flop **401** (FIG. **7**) is enabled, producing the output waveform (*b*), whenever a TCA pulse occurs and, thereby, sets the associated flip-flop **402.** The necessary PROCESS ENABLE signal (waveform *c*) to effect a synchronous process cycle, however, is not produced on lead **404** until the flip-flop **402** receives a C_{+12} clock pulse (waveform *a*), as evidenced by a comparison of the waveforms *a–c* in FIG. **14.**

The dotted line in the timing waveform (*d*) in FIG. **14** simply indicates that the SYNC PROCESS signal may occur simultaneously with the PROCESS ENABLE signal, or be delayed up to 12 counts of the position counter **241** in order to commence a print cycle in a synchronous manner. Whether a delay is necessary or not is dependent upon when the initial type carrier advance (TCA) pulse was generated relative to the accessibility at the memory output of the particular STORE (**1** or **2**) then being processed.

DETAILED DESCRIPTION OF STORE 1, 2 (AND PROCESS 1, 2) SIGNAL GENERATING CIRCUIT

Attention will now be directed to the manner in which the necessary STORE and PROCESS signals are generated for use in the circuitry previously described

and depicted in FIG. 6. It will be recalled that the STORE 1 and 2 signals are employed to produce the LOAD CHAR signals for each associated memory 221, and the PROCESS 1 and 2 signals are employed to effect the processing of stored data at the proper times and out of the proper one of the two memory storage areas. With particular reference to FIG. 9, it is seen that a flip-flop 525, together with an associated array of output gates 530–534 produce the STORE 1 and 2 output signals for use in loading the associated dual storage memory 221 (best seen in FIG. 5). Considering first the case when there is no PRINTER BUSY signal appearing on the previously identified lead 293, it is seen that the gating array at the output of the flip-flop 525 is functionally transparent since AND-gate 531 is enabled, AND-gate 532 is disabled, and thus the N (normal) output of the flip-flop 525 is transmitted to the output of the OR-gate 533.

Considered more specifically, whatever logic states appear at the N/I outputs of the flip-flop 525, these outputs will also appear at the outputs of the OR-gate 533 and inverter 534, respectively. The condition when the PRINTER BUSY signal on lead 293 is present is considered hereinbelow in connection with a description of a data stacking feature incorporated in the illustrative composite logic control circuit.

Flip-flop 525 switches state upon receipt of each END OF LINE (EOL) signal applied to the J and K inputs thereof through an AND-gate 537. If the flip-flop 525 is initially in a 1 output state, for example, this indicates that the data is presently being stored in STORE 1, whereas if the flip-flop is in a 0 output state, data is being stored in STORE 2. A free-running clock input f controls the switching rate of the flip-flop 525, as well as all the other flip-flops in FIG. 9.

The primary function of a process signal generating flip-flop 541 is to insure that the stored data is processed for printing in the same order as the data is initially received in the memory 221. These are several possible storing and processing conditions that make the proper time-related operating sequences of the flip-flop 541 imperative.

For example, on an intial POWER ON RESET condition, which is a point in time when no data is stored in the memory 221, there would be a NO EOL 1 and a NO EOL 2 signal applied to the associated inputs of an AND-gate 543. The presence of both of these signals is indicative of the fact that neither memory STORE 1 or 2 is full, and that both storage and processing of data characters should be taking place out of the same STORE initially. The manner in which the NO EOL 1 and 2 signals are generated will be described shortly.

Referring now to the output of the AND-gate 543, it is seen that it is applied to one input of an OR-gate 546, with the output thereof applied to an input of each of two AND-gates 548 and 549. Thus any 1 input applied to the OR-gate 546 will result in a 1 output causing both AND-gates 548 and 549 to be conditionally enabled. The other input to each of these AND-gates has an associated STORE 1 or STORE 2 signal applied thereto. The flip-flop 541 will thus follow or track the STORE 1 and/or STORE 2 output state of the flip-flop 525 by a one single pulse delay, and will thus cause the loading and processing of data to occur out of the same STORE whenever a 1 signal is applied to either input of the OR-gate 546.

Consider now the case where, after a POWER ON RESET condition, loading and processing is occurring out of STORE 1 and an EOL signal is subsequently applied to the J and K inputs of the flip-flop 525 through AND-gate 537, which is enabled as long as a printer busy signal is not present. This causes the flip-flop 525 to switch from a STORE 1 to a STORE 2 operating state, which results in a STORE 2 signal appearing at the output of the inverter 534. Receipt of an EOL command signal on lead 168 (from the I/O controller 134), of course, indicates that the line of character data that was being stored is to be terminated, and that subsequent data received is to be printed on the following print line.

This same EOL signal also causes a flip-flop 554 to be set by means of an enabled AND-gate 556, with an associated AND-gate 557 remaining disabled. As a result, an active EOL 1 signal is produced and the previous active NO EOL 1 signal is changed to a O level at the respective outputs of the flip-flop 554. The OR-gate 546 therefore looses its 1 input, since the AND-gate 543 has lost its NO EOL 1 (= 1) signal. Thus, the flip-flop 541 remains frozen in the PROCESS 1 output state. This, of course, is the desired result which allows the storage of character data to commence in STORE 2 while the processing of data is still taking place in STORE 1.

However, before data can be loaded in STORE 2 of a given memory 221, the associated column counter 243 must be reset, which is accomplished in the following manner. Both inputs to an OR-gate 559 are initially at a O signal level, indicating that neither an EOL 1 nor 2 signal is at that time applied thereto, because an EOL signal has not yet been received on lead 168. The latter signal is required in order to conditionally enable the AND-gates 556 and 557 through lead 560 and, thereby, effect a change in the output state or either flip-flop 554 or a flip-flop 561. As such, an AND-gate 562 is initially conditionally enabled through an inverter 563, so that upon receipt of the next EOL pulse, which is applied to the other input of the AND-gate 562, through a lead 564, a COL CTR RESET signal will be produced at the output of an OR-gate 565, and sent over the previously identified lead 286 to the counter 243 (see FIG. 6). Two AND-gates 566, 567, a flip-flop 568 and an inverter 569 are employed in conjunction with the aforementioned data stacking feature which will be described shortly.

Referring again to the generated STORE 1 and 2 and PROCESS 1 and 2 signals, when the first EOL signal is received on lead 168, an EOL 1 signal is generated and the PROCESS 1 signal remains active so as to conditionally enable an AND-gate 571. Processing of data in STORE 1 of the memory then continues until it is empty, as indicated by an EMPTY 1 signal supplied on a lead 573 to an input of the AND-gate 571. Only at that time is a line feed (L. F. 1) command signal generated at the N output of a flip-flop 575, as called for by the previously received EOL signal.

The flip-flop 575, of course, is set in response to the AND-gate 571 being enabled, with the resultant L. F. 1 output signal being sent over a lead 576 to an input of an OR-gate 577. The resulting output from the OR-gate 577 causes the associated circuit module 135 to "release" the line feed lead 169. As descirbed hereinabove, lead 169 essentially functions as an AND-bus, in that all of the circuit modules 135*a–g* must "release"

that bus before it changes to the line feed state. Once a LINE FEED signal appears on the lead **169**, the I/O controller **134** detects this signal level state and causes the printer to perform a mechanical line feed.

It is also seen in FIG. **9** that the presence of a LINE FEED signal on line **169** enables a flip-flop **579**, which forms part of a LINE FEED COMPLETED signal generating circuit **580**. The resulting N output of the flip-flop **579** is applied to an input of an AND-gate **581**, with the other input thereto being a conditionally disabling LINE FEED signal produced by an inverter **583**.

When the I/O controller **134** has completed a line feed of data, it will force the LINE FEED signal appearing on the bus **169** to a O (NO LINE FEED) signal level, such as by a simple switching voltage level change (not shown). The output of the inverter **583** is then at a **1** signal level. The flip-flop **579** remains in the **1** state for one clock cycle, and is then reset by the O level of the LINE FEED bus **169**. As such, the AND-gate **581** is momentarily enabled, and provides a one clock period (17.85 microseconds in the illustrative embodiment) L. F COMPLETED output pulse. This pulse, transmitted selectively over lead portions **584***a*, *b* and *c*, resets both the L. F. **1** generating flip-flop **575** and the EOL **1** generating flip-flop **554**, and enables the OR-gate **546**, the output of which switches the flip-flop **541** to a PROCESS **2** output state so as to follow or track the then STORE **2** output state of the flip-flop **525**. Accordingly, both the loading and processing of character data will then take place relative to the same STORE (**2**) of the associated memory **221**.

It is also seen that the L. F. COMPLETED pulse, by having reset rhe EOL **1** flip-flop **554** (with the EOL **2** flip-flop **561** having not yet been set), will also result in both inputs to the AND-gate **543** being at a **1** signal level, thereby, causing a **1** signal to be applied to an input of the OR-gate **546**. This, of course, results in the process flip-flop **541** remaining in a PROCESS **2** output state, even after the removal of a L. F. COMPLETED pulse to the OR-gate **546**, until the next EOL is received. Until that time, the loading of character data into STORE **2** and the processing of such data out of STORE **2** will continue in the same manner as previously described with respect to the loading and processing of data in STORE **1**.

When the next succeeding EOL signal is received on lead **168** from the I/O controller **134**, the sequence of events described above for STORE **1** will now occur again for STORE **2**. More specifically, the second EOL signal will cause the EOL **2** generating flip-flop **561** to be set by the previously identified and enabled AND-gate **557**. The EOL **2** output of the flip-flop **561** will, in turn, switch a flip-flop **588** to its L. F. **2** output state, through an enabled AND-gate **589**, upon an EMPTY **2** and a PROCESS **2** signal being applied thereto. This, of course, will result in the L. F. **2** signal being sent over a lead **590** to the other input of the aforementioned OR-gate **577**, which again causes the associated circuit module **135** to "release" the line feed lead **169**, and subsequently effect a new LINE FEED COMPLETED signal at the output of the AND-gate **581**. As such, the loading and processing of character data will subsequently once again take place relative to the same common STORE (**1**) of the associated memory **221**, in accordance with an operating logic sequence identical to that described above after the AND-gate **537** was enabled.

DATA STACKING FEATURE

In the above examples describing how data may be loaded in and processed out of different stores of each memory **221**, the processing of a given line of print data was always completed before the EOL signal for the next print line was received. More specifically, only one circuit module-generated EOL **1** or EOL **2** signal was stored and generated by the associated one of flip-flops **554** and **561** at any one time and, consequently, a PRINTER BUSY signal was not produced on the previously identified lead **293** (FIG. **5**), as a result of an AND-gate **591** being enabled by both EOL **1** and **2** input signals applied thereto over leads **592** and **593** respectively.

Consideration will now be directed, therefore, to the case where both EOL **1** and **2** command signals are stored and generated, and a data stacking mode of operation is effected.

Starting with a POWER ON RESET condition, let it be assumed that the loading and processing of character data both occur initially relative to STORE **1** of a given memory **221**. When the first EOL signal is then received, under the operating conditions described hereinabove, the EOL **1** generating flip-flop **554** is set, the associated column counter **243** is reset and the received character data will begin to be loaded in STORE **2** of the memory. The L. F. **1** flip-flop **575** is not yet set, however, because the processing of character data in STORE **1** has not been completed (i.e., an EMPTY **1** signal has not yet been received on lead **573** to enable the AND-gate **571** and, thereby, effect the generation of an L. F. **1** signal).

Since the flip-flop **554** is now latched to the EOL **1** output state, that output is applied over leads **592** and **592***a* as an input to the OR-gate **559** which, in turn, results in an EOL STORED output signal that is passed through the inverter **563** so as to produce a disabling signal at an input of the AND-gate **562**. The previously identified AND-gate **566** is, however, conditionally enabled by the presence of the EOL STORED signal being applied to one of the three inputs thereof.

In response to a second EOL signal at this time, the EOL **2** flip-flop **561** is also set as a result of the enabled AND-gate **557**, and both inputs to the AND-gate **591** then comprise EOL **1** and **2** **1** signals. As such, a PRINTER BUSY signal is generated on the lead **293**. The printer is considered "busy" at this time because two full print lines of character data have been stored with processing still continuing out of STORE **1**, hence, no more character data can be accepted.

The RNC bus **172** (FIGS. **2**, **5**) is then turned OFF by means of the PRINTER BUSY signal appearing on lead **293** being applied to the inverter **294** (FIG. **5**), the output of which is applied as an input to the AND-gate **292**, previously described and shown in the latter figure. This latter input disables the AND-gate **292** and causes a "stop" condition in the transfer of data over serial data lead **181** between the I/O controller **134** and the circuit modules **135***a–g*. This is a desirable result so that no character data is lost while processing in STORE **1** continues.

Consider now the case where the printer and, therefore, the I/O controller **134**, are connected to a data

source that is sending data at some fixed rate, rather than on a character-by-character basis, as controlled by the RNC bus 172 depicted in FIGS. 2 and 5. This a very common situation for either mobile or fixed teleprinters, where the receiving printer does not control data transfer on a character-by-character basis, but rather, is a slave to the incoming data rate. Since the data source cannot know exactly how long it takes each individual printer to print each given line, (since that is the function of the position of the type carrier at that instant in time), the problem of not over-running the printer has typically been resolved in one of two ways: One technique is to limit the data source to some maximum line per second rate, which is determined by the maximum time it can take the printer to print any possible print line. The other technique is to utilize a buffer store, larger than the maximum message length, as an adjunct to the printer.

In either of the above cases, if the data source violates the set requirements by sending character data faster than the printer can print it, data is lost. When looking at the printed copy subsequent to that event, of course, it is very difficult to tell why the data was lost, and often impossible to ascertain the context of the lost data.

Typically, character data is received faster than it can be printed when short lines are sent, that is, the print line has been terminated by an END OF LINE character(s) from the data source before many of the possible print positions (or columns) are filled. This arises, of course, since it can take the printer as long to print a short line as a long line, whereas it takes the data source much less time to send short lines of data compared to long lines filling every possible print column.

From the foregoing, the importance of and definite need for a data stacking feature in a printer becomes very apparent. Both the mode of operation involved in and the unique logic circuitry embodied in the present invention for implementing such a feature will now be described in detail.

Consider first the situation referred to hereinabove where the flip-flop **554** (FIG. 9) is initially latched to its EOL 2 output state, and where a second EOL signal has just been received, which results in the flip-flop **561** also being latched to its EOL 2 output state so as to produce a PRINTER BUSY signal on the lead **293.** As will be recalled, this signal results in the RNC bus **172** being turned OFF by the disabling of the AND-gate **292** (FIG. 5).

Thereafter, any additional print characters received by the I/O controller **134** will continue to be sent to the circuit modules **135** on the serial data lead **181.** These data characters, actually assignable to a separate line, will not be lost, but rather, be stored in STORE 2 of the memory, and subsequently printed in the next print line (such as line 2) up to the point where either that line has been completely filled with print characters, or the processing in STORE 1 has been completed. In the latter case, subsequent print characters for the following line (such as line 3) are then loaded in STORE 1 and subsequently printed in line 3. Line format is thus sacrificed before print characters are lost whenever the maximum printing rate of the printer is exceeded. This can be very useful from not only the system trouble-shooting point of view, but also can be advantageous even in the normal operation of the printer so as to avoid having to provide an expensive, auxilliary data buffer for use in connection with the occasional printing of short lines.

With reference again to the receipt of the second EOL signal, it does not produce a COL CTR RESET signal on lead **286** (FIG. 9) since, as described above, the EOL 1 signal previously generated disables the AND-gate **562** by reason of the input applied thereto through the inverter **563.** The second EOL signal does, however, set the flip-flop **568** to the 1 output state, thereby providing a stored indication that a second EOL signal has, in fact, been received, and also provides an operating condition that allows the column counter **243** to be reset at a later point in time. Accordingly, until the column counter **243** has been reset, data characters will be received for loading in the memory **221** by the proper circuit module **135** (assuming that line 2 for example, has not yet been filled with print characters) as if no EOL 2 signal had been received. This is evident from the load control circuit **251***a* depicted in FIG. **6,** wherein it is seen that only the OWN STATUS FULL and the PREV CKT STATUS FULL signals determine whether a given data character is accepted for loading in the memory or not.

Although the second EOL signal in question appearing on lead **168** (FIG. 9) at this point in time has toggled STORE flip-flop **525** from a STORE 2 to a STORE 1 output state, the PRINTER BUSY signal then appearing on lead **293** would cause that output to be inverted to a STORE 2 output signal. This is achieved by the inverter **530** preventing the N (normal) output of the flip-flop **525** from enabling the AND-gate **531,** while the I (inverted) output of the flip-flop is allowed to enable the AND-gate **532** and the OR-gate **533.** This is, of course, the desired result so that any data characters received at this time are still loaded into STORE 2, as if no EOL 2 signal had been received.

This "stacking" of charactr data in STORE 2 (e.g., for line 2) is terminated by either print line 2 being filled with print characters, or by the processing of character data in line 1 (STORE 1) being completed.

With respect to the first case, and with reference to FIG. **6,** it is seen that when a HIGH COL COMP signal is applied to the 4-bit comparator **281,** which indicates that a character has just been loaded into the high order print column of a particular memory store, the resulting output signal on lead **280** sets the flip-flop or latch **278** to its OWN STATUS FULL output state through the enabled OR-gate **279.** Any additional data characters sent on the serial data lead **181** are then lost since the ACCEPT CHAR signal has been removed from lead **275.**

The fact that an EOL signal is also sent on the bus **168** by the last circuit module **135***g* is irrelevant, since this signal is prevented from passing through the AND-gate **537** (Fig. 9) by the disabling output of the inverter **594.** As a result, the EOL signal is prevented from actuating the flip-flop **525.**

The more typical sequence of events occurs when processing of data in line **1,** for example, and the associated line feed, are both completed before STORE **2** (for the next line) is filled. In this condition, the resulting L. F. COMPLETED pulse will perform the following functions. All three inputs to the AND-gate **566** are then at a 1 signal level for the one clock cycle (17.85 microseconds in the illustrative embodiment) duration

of the L. F. COMPLETED pulse, resulting in a COL CTR RESET signal appearing on lead **286.** This signal, in addition to resetting the associated column counter **243,** also resets the flip-flop **568.**

As will be recalled, the OR-gate **546** is also enabled by the L. F. COMPLETED pulse, which causes the PROCESS flip-flop **541** to change to a PROCESS 2 output state, since the gate array **530–534** still effects the generation of a STORE 2 output signal. However, the L. F. COMPLETED signal, by resetting the EOL 1 and L. F. 1 flip-flops **544** and **575,** respectively, results in a loss of the PRINTER BUSY signal on lead **293.** This results in the gates **530–534** again becoming logically transparent, and as the flip-flop **525** has already been set to a STORE 1 output state (by the second EOL signal), a STORE 1 output signal will be produced. A new data character received on the serial data lead **181** from the I/O controller **134** will thereafter correctly be stored in the first empty slot of STORE 1 (e.g., line 3) of the associated memory **221.**

In briefly considering the purpose of the aforementioned AND-gate **567** and inverter **569** also depicted in FIG. **9,** it will be recalled that when the first EOL signal is received, the AND-gate **562** is still enabled and a COL CTR RESET signal is generated. This EOL signal will thus attempt to set the flip-flop **568,** but the COL CTR RESET signal will prevent this from happening by the action of the inverter **569** and, thus, will result in the flip-flop **568** being reset by a signal being applied to the K (clear) input thereof. This is, of course, the correct logical function desired since the flip-flop **568** is to store only EOL command signals that have not yet caused a COL CTR RESET signal to be established.

INITIALIZE AND EMPTY CONTROL SIGNALS

Referring now in greater detail to the aforementioned POWER ON RESET (POR) signal, which will be present on the INITIALIZE lead **174** (FIG. **2**) it is employed to initialize all of the logic in each of the process circuit modules **135.** As such, that signal is employed, for example, to effect the purging of the storage memory **221** of any invalid data after power has been applied to the module, and to automatically reset the various data processing shift registers and counters of each circuit module **135.** In addition, the INITIALIZE signal affords protection against a loss of synchronism between the mechanical and electrical logic portions of the printer. More specifically, any time at the end of a message, for example, when the drive motor for the type carrier **16** would normally be shut off, the process circuit logic looses synchronism with the type carrier generated TCR and TCR pulses.

As a result, there must be some way of regaining the loss in the electrical-mechanical synchronism of the printer when the motor is subsequently turned on and brought up to proper speed. This is accomplished in the present printer by having the I/O controller **134** momentarily activate (e.g., by a simple voltage level change) the INITIALIZE lead **174** (FIGS. **2** and **5**) any time a new message is received. This signal condition effectively resets all of the registers and counters of the data processing circuit modules **135***a–g*. Thereafter, the I/O controller **134** generates a control signal over a lead not specifically shown, but included in the multilead cable **210** depicted in FIG. **2,** to turn on the motor **26** (FIG. **1**) which drives the endless belt type carrier **16.** Two other leads (shown at the output of the transducer **132** in FIG. **1**) included in the multilead cable **210** provide the TCA and TCR signal information to the I/O controller **134** from the printer mechanism. The remaining leads in the cable **210** are associated with other mechanical control functions of the printer and are not of interest with respect to understanding the logic control circuitry disclosed herein.

Thus, it is seen that the I/O controller **134** performs the following "handshaking" function between the parallel interface **142** and the logic control leads connected to the data processing circuit modules **135,** as depicted in FIG. **2.** When a message turn-on command is received by the controller **134,** all of the logic elements of the process circuit modules **135***a–g* are reset, the carrier drive motor **26** is then turned on, TCA pulses from the printer sensor **132** are regenerated by the I/O controller **134** and passed to the circuit modules **135***a–g* over the lead **163** (see FIG. **2**), and when the motor is brought up to speed (as ascertained by the controller in a conventional manner), the controller passes the first TCR signal to module **135***a*. Thereafter, the I/O controller will continue transmitting the TCR pulses over the lead **165** to the first circuit module **135***a*, as depicted in FIG. **2,** and subsequently in a multiplexed and time-delayed manner over the interconnecting leads **201** to each of the remaining circuit modules **135** in succession.

Each of the seven circuit modules **135** in the illustrative embodiment is not ready to print any data characters supplied thereto until a TCR pulse has been received, and the beginning of font marker bit has been shifted into the eighth element of the associated shift register **261** (best seen in FIG. **7**). Only at that point in time is the first circuit module **135***a*, for example, prepared to process, compare, and commence the printing of encoded data characters. The I/O controller **134** detects readiness of a given circuit module **135** to receive print data over the lead **181** (FIG. **2**), by the circuit module in question activating the previously identified REQUEST NEXT CHARACTER (RNC) lead **172** in response to the AND-gate **292** (FIG. **5**) being enabled. As also previously described, printing may then take place in a non-serial fashion, depending upon the order of alignment between the successive type characters on the carrier **16** and the data characters received for printing. Advantageously, printing in the first **12** print columns associated with the circuit module **135***a* (as well as in other random **12** column groups associated with other respective circuit modules) in the illustrative printer embodiment may commence, and actually may be completed, before all of the print data to be printed in a given line has been received and/or processed in the other circuit modules **135.**

The EMPTY signal lead **176** is actually an AND-bus on which all of the circuit modules **135** provide an output. The bus becomes active when all of the circuit modules **135***a–g* have no data or EOR command signals stored. When a message turnoff command is received by the controller, this signal must go to the EMPTY state, indicating that all printing and processing has been completed, before the motor is turned off by the I/O controller **134.**

DETAILED DESCRIPTION OF EMPTY MEMORY STORE SENSING CIRCUIT

In order to ascertain when all of the characters within a given store of the memory **221** have been printed, the

circuitry depicted on the left side of FIG. **10** is employed. Since parallel access to all of the bit positions in the memory is, of course, not provided, a logic circuit-controlled technique is desired to serially monitor the outputs of the memory **221.** This circuitry is designed to generate EMPTY and NOT EMPTY storage signals for both STORE 1 and 2 conditions, such that after an EOL signal has been received, a LINE FEED signal can be generated in response to the EMPTY state of the memory.

Considered more specifically, and with particular reference to FIG. **10,** at the end of a given **12** count period of the position counter **241,** the previously described C_{+12} signal is produced at the output thereof (best seen in FIG. **6**), applied over a lead **621** to the K input of a flip-flop **625,** and resets the latter. A new **12** count period then begins, which means that the other of the two STORES of the associated memory **221** is then accessible at the output thereof.

For purposes of discussion, let it be assumed that STORE 1 is then being accessed. If any data is stored in STORE 1 in the first **11** slots thereof, the flip-flop **625** will be set by an enabled AND-gate **627,** which will then have both an ANY STORED signal and an inactive C_{+12} signal applied thereto, the latter being inverted to a 1 signal level after passing through an inverter **629.**

The ANY STORED signal is received on lead **415** from the previously identified OR-gate **417** of FIG. **5,** and constitutes a sampling of the 8-bit output of the associated memory **221.** This sampling simply provides a free-running indication of the fact that there is one or more bits of character data stored in the specific STORE-defined group of memory slots then being accessed at the output of the memory. The resulting 1 output from the flip-flop **625** is applied to an OR-gate **631** which, in turn, provides an active ANY STORED LATCH signal that is applied as one input to both an AND-gate **637** and an AND-gate **638.** After this 12 count period in question, and regardless whether any data had been stored in STORE 1 prior thereto or not, the next C_{+12} signal from the position counter **241** is again received on lead **621** and resets the flip-flop **625.** If an ANY STORED signal is received during this C_{+12} time interval, which, of course, is still associated with a STORE 1 non-empty condition, the other input to the OR-gate **631** assures that an active ANY STORED LATCH signal is provided independently of the state of flip-flop **625** The flip-flop **625** is reset in any case since the presence of the second C_{+12} signal results in a O input signal being applied to the AND-gate **627** through the inverter **629,** so as to assure that the J input of the flip-flop **625** remains at a O signal level, while the K input is at a 1 signal leve.

During the second C_{+12} signal interval, the presence or absence of an ANY STORED LATCH signal thus respectively indicates whether the previously interrogated half of the associated memory **221** (STORE 1 in this example), has any character data bits stored therein, or is empty at that time.

A Q_N and a Q_I signal are respectively applied as inputs to two AND-gates **641** and **642.** These signals, as seen in FIG. **6,** constitute the normal and inverse outputs of the last stage **241'** of the free-running position counter **241,** and respectively indicate, when at a 1 signal level, whether STORE 1 or 2 of the dual storage memory **221** is then accessible at the memory output. When the AND-gates **641** and **642** are further gated with the 12th count signal C_{+12}, the particular one of these gates, when enabled, provides an output pulse to interrogate or sample the state of the ANY STORED LATCH signal and, thereby, ascertain whether the memory store that has been interrogated during the previous 12 count interval is empty.

This is accomplished in the following manner. The AND-gate **637,** and a complementary AND-gate **643** are conditionally enabled by the AND-gate **641,** whenever both a Q_N and the second C_{+12} signals are applied to the inputs thereof. A flip-flop **645** is then set to a 1 (NOT EMPTY) output state whenever the ANY STORED LATCH signal is active during the proper sampled pulse period, and is set to a O (EMPTY 1) output state by means of an inverter **647** whenever the ANY STORED LATCH signal is not present. The aforementioned AND-gates **638** and **642,** and AND-gate **648,** and an associated flip-flop **649** similarly ascertain the storage status of STORE 2 of the memory.

The flip-flops **645** and **649** are thus seen to constantly monitor the dual storage areas of the memory **221** and, thereby, selectively and respectively provide EMPTY (1 or 2) and (NOT EMPTY 1 or 2) output signals indicative of the storage status of the associated STORES (1 or 2) as they alternately become accessible at the output of the memory. The EMPTY 1 and EMPTY 2 signals, of course, are employed as conditionally enabling inputs to the respective AND-gates **571** and **589** depicted in FIG. **9.**

An AND-gate **651,** when enabled, provides an active 1 output signal whenever data in STORE 1 of the memory **221** is being processed, and when that store is not yet empty, whereas an AND-gate **652** performs the same function relative to STORE 2.

An OR-gate **655** has an input connected to the output of each of the AND-gates **651** and **652** and, thus, provides an output PROCESS IN PROGRESS signal on a lead **658** whenever a particular one of the two STORES that is being processed contains character data that is to be processed and printed. The significance of this latter signal appearing on lead **658** will now be considered in connection with an auxiliary circuit portion of FIG. **10.**

OPTIONAL RECOVERY CIRCUIT FOR PURGING MEMORY OF INVALID CHARACTERS

The circuitry depicted within the dash-lined box **670** in FIG. **10** essentially comprises an auxiliary invalid character recovery circuit for purging the associated memory **221.** This circuit functions essentially as an adjunct to the invalid character strap option recovery circuit **470** described hereinabove in connection with FIG. **5.** It will be recalled that the strap option circuit is employed with common font lengths of **64** or 96 characters, for example, and effects the purging of the memory of any erroneous characters with no time-out delays being required, and with no restrictions being placed on the maximum number of different length character fonts that may be employed.

The recovery circuit depicted within the dash-lined box **670** in FIG. **10,** however, is designed for use with non-conventional font lengths, and purges the memory **221** of any erroneous characters only after a predetermined time-out period has elapsed. In such cases, neither the CHAR MAX **64** or **96** leads depicted in FIG.

5 are activated and, thus, the associated strap-option circuit within the dash-lined box **470** is effectively disabled.

As briefly pointed out hereinabove, in situations where the strap-option recovery circuit is not effective because of an uncommon font length, the presence of any erroneous characters in a given memory, which could be created somewhere in the control circuitry due to electrical noise, for example, could not be printed, as there would be no data comparisons possible with any of the generated encoded signals representative of the type characters in a given font on the carrier **16.** It is thus important to have special logic circuitry for detecting the presence of invalid characters, and for purging them from the storage memory after a predetermined time has elapsed, otherwise the printer would simply continue operating in an attempt to print the invalid characters. The circuit **670** in FIG. **10** performs this function in a manner to be described in detail after a brief initial explanation of some of the salient factors involved.

If it is assumed for one particular operating condition that all of the character fonts on the type carrier **16** are identical, it can be seen that the maximum time required to print any type character in any print column is the time required for one character font to completely pass a given fixed reference point on the printer. During this time, every type character has aligned itself with every print column position since the type carrier moves continuously, and each of the type pallet-supporting slots in it are associated with a different type character of a given font.

Consider now an alternative operating condition wherein a maximum of two different length character fonts are allowed (remembering that no such limitation applies if the fonts are of the more common 64 and 96 strap-option character lengths). In this latter case, a time-out for two TYPE CARRIER RESET (TCR) pulses is sufficient to assure that every type character has aligned itself with every print column. This time-out period is performed in the present logic control circuit **670** by a modulo (MOD) **4** CTR **671** being held out of its reset state.

More specifically, after an initial POWER ON RESET condition occurs, i.e., when no character data is stored in either STORE of a given memory, the PROCESS IN PROGRESS signal on lead **658** is then at a **0** signal level which, in turn, results in a **1** input signal being applied to an OR-gate **672** through an inverter **674.** As such, the counter **671** is held in its then existing reset state.

When the first data character to be printed has been stored, the PROCESS IN PROGRESS signal becomes active (i.e., changes to a **1** level), and results in a **0** signal being applied to the OR-gate **672.** As described hereinabove, the PROCESS IN PROGRESS signal provides an indication that character data remains in a specific STORE of the dual storage memory **221,** and that such data is presently being processed. At the same time that the latter signal is received, a non-active **0** level COMPARE signal is applied to an input of an AND-gate **676** and, thus, results in a **0** output signal from that AND-gate being applied at the other input of the OR-gate **672.** It will be recalled that the COMPARE signal is derived from the output of the inhibiting AND-gate **419** depicted in FIG. **5,** and is always at a **0** signal level when no valid data character comparisons are taking place. With both inputs to the OR-gate **672** being at a **0** signal level, the resulting output therefrom allows the counter **671** to begin its time-out.

It should be appreciated, however, that the counter **671** cannot simply time-out for two TCR pulse intervals starting from the time that the first data character is loaded into a particular STORE of the associated memory **221.** This follows from the fact that if data characters are received at some very slow rate, a **12**th character, for example, may not be loaded into a given STORE of the memory in time to be compared with the type characters of the two full (and different) fonts while making a single pass past the print columns of the printer in the illustrative example presently being considered.

This problem is resolved in accordance with the principles of the present invention by resetting the counter **671** whenever a data character comparison has occurred, and that character is about to be printed, as indicated by the presence of a COMPARE signal applied to the input of the AND-gate **676.** This **1** signal input, together with another **1** (**0** inverted) signal input supplied thorough a lead **677** and an inverter **678,** results in the output of the OR-gate **672** resetting the counter **671.** The input to the AND-gate **676** through lead **677** is at a **1** signal level since two complete TCR pulses have not yet been received, which would otherwise produce a pair of output signals from the counter **671** to enable an AND-gate **679.**

It is thus seen that the AND-gate **679** is employed to provide an output whenever: (1) two complete type character fonts have passed every column position of the printer, (2) data is still stored in the memory to be printed, and (3) no valid comparison has occurred between any binary representation of a character stored in the memory and any of the type characters on the type carrier **16.**

An AND-gate **681** has one input connected to the output of the AND-gate **679,** and provides a RECOVER output signal on the previously identified lead **481** (FIG. **5**) whenever the SYNC PROCESS signal (also from FIG. **5**) is at a **1** signal level. This assures, from a timing standpoint, that the purging of any erroneous data from the memory **221** will take place in the proper STORE thereof.

As also described hereinabove with reference to FIG. **5,** the generation of the RECOVER signal will cause the ERROR symbol to be printed for any character(s) that remain in a given store of the memory being interrogated. While the COMPARE signal appearing on lead **489** becomes active as the ERROR symbol (or symbols) are being printed, the otherwise resetting of the counter **671** is inhibited by the now active output of the AND-gate **679,** which provides a **0** level (disable) signal to the input of the AND-gate **676** through the inverter **678.** Purging the memory of erroneous characters, by printing the error symbol(s), thus can continue until the particular STORE of the memory being interrogated is completely empty, as indicated by the PROCESS IN PROGRESS signal changing to a **0** signal level. The counter **671** is then reset.

The choice of limiting the type character **16** to a maximum of two different length character fonts (each of which may be repeated as often as is required to completely fill the type carrier slots is, of course, what determines how long must be the time-out of the counter **671.** Considered another way, the larger the number of different length character fonts allowed, the

longer will be the required time-out and, therefore, the less efficient (time-wise) will be the optional recovery circuit of FIG. 10.

In summary, a very unique and versatile composite print hammer logic control circuit has been described and claimed herein. Through the utilization of a plurality of unique, data processing circuit modules, each having its own free-running, logic controlled dual storage memory, not only may sequential printing on a given line be effected as the print data is being received (as distinguised from after a whole line of data has been received), but the printing of character data previously stored for one line may occur simultaneously with the storing of data for the next adjacent line, with character stacking in a given line also being possible on an emergency basis. Such control circuitry insures reliable high-speed on-the-fly printing, regardless of variations in the input data rate. Further, by using both built-in strap-option and auxiliary control circuits, the memories are routinely purged of any erroneous characters stored therein.

In addition, through the utilization of modular construction, wherein each circuit module is assignable to and performs the necessary data processing for a specified sub-group of print hammers and columns, inexpensive MOSFET integrated circuits with relatively slow switching speeds may be employed without any sacrifice in printer performance. Finally, through the utilization of pulse stretching, print hammer magnet energization time may also be extended over a much longer period than the print cycle scanning time. As such, logic circuit control functions need not be made dependent on the mechanical and/or electromechanical hammer actuation time periods involved in a typical impact printer.

It is obvious, of course, that various modifications may be made to the present embodiment and alternatives provided without departing from the spirit and scope of the invention.

What is claimed is:

1. A print control logic circuit for use in high speed on-the-fly printers, for selectively actuating one or more out of a plurality of character print-inducing members whenever selective type characters forming a part of a continuously moving font of type characters are momentarily brought into respectively aligned and juxtaposed relationship with said print-inducing members so as to effect sequential printing of type character images along each successive line on a medium, but with the character images ultimately being positioned in the serial order in which the input data characters to be printed were received, said logic control circuit comprising:

sub-dividable memory means for storing input binary encoded data representative of the characters to be printed at particular positions along a given print line, said memory means having at least two logic controlled and operationally distinct storage areas, and including means for selectively storing input character data in and reading said data out of said storage areas in a manner that allows incoming data intended for printing on one line to be stored in one of said distinct storage areas while any previously received data is still being read out of the other storage area, processed and printed in the preceding line,

means for generating type character font data in multiple phase sequences during each print cycle, said data being representative of and identifying the particular ones of said moving font type characters that are respectively brought into registry with associated ones of said character print-inducing members at any particular point in time during each phase sequence, and

means for processing any stored character data for printing after having been read out of said memory means, by comparing said stored data with said generated type character data, and for generating a distinct print signal, in response to each valid equal signal comparison, for use in actuating at the proper time the particular one of an associated character print-inducing member that will effect the printing of a character image representative of the compared type character at the proper position along a given print line.

2. A print control logic circuit in accordance with claim 1 wherein said memory means includes a free-running memory, and phase controlled type character position and print column counter means for logically controlling access to and sub-dividing said memory into distinct dual storage areas.

3. A print control logic circuit in accordance with claim 1 wherein said circuit is sub-divided into a plurality of essentially identical, but interconnected data processing circuit modules, each circuit module including separate ones of said memory means, character generating means and comparing means, and with each circuit module being assigned to process print data for a pre-determined number of print columns forming only a sub-group out of the total number of print columns defining the length of a given print line, with each circuit module thereby storing, processing and effecting the printing of input data characters as received in only the print columns assigned thereto.

4. A print control logic circuit in accordance with claim 2 wherein said memory means further includes memory recovery circuit means for detecting the presence of and effecting both the erasing of any invalid data characters stored in said memory during the printing of each print line, and the printing of error symbols therefore.

5. A print control logic circuit in accordance with claim 2 wherein said memory means further includes:

presettable strap-option circuit means coupled to said associated memory for detecting the presence of and purging said memory of any invalid data characters stored therein whenever such characters represent an encoded binary number higher than a preset number representative of the highest order character in a chosen font of type characters employed in a given application.

6. A print control logic circuit in accordance with claim 5 further comprising:

an auxiliary invalid character recovery circuit associated with said memory means and being capable of detecting any invalid data characters stored in said memory means, and for optionally erasing said invalid characters independently of said strap-option circuit means, regardless of the length of a given character font employed in conjunction therewith, after a predetermined time delay that is chosen to be greater than the time re-

quired for the number of different type character fonts employed to successively pass a given reference point of the printer, and for thereafter causing each equal character-signal comparison representative of an invalid data character to effect the actuation of the proper print-inducing member at the proper time when an error symbol character included in a given font of type characters is momentarily brought into register therewith and, thereby, effect the printing of an error symbol.

7. A print control logic circuit in accordance with claim **3** wherein said memory means of each of said circuit modules further includes means for selectively generating first and second storage signals and first and second processing signals for controlling the loading of encoded data characters in and the readout thereof from the first and second storage areas of said memory means in a selective manner.

8. A print control logic circuit in accordance with claim **7** wherein the generated storage and processing signals are associated with the same storage areas of the memory means unless input data characters are received for storage after one storage area of said memory means has been filled, but the processing and printing of that data has not been completed.

9. A print control logic circuit in accordance with claim **3** wherein each of said memory means includes a free-running storage memory and phase-controlled type character position and print column counter means for logically controlling selective access to and sub-dividing said memory into dual storage areas, and wherein each of said circuit modules further comprises:

memory recovery circuit means for detecting the presence of and effecting the erasing of any invalid data characters stored in said associated memory during the printing of each print line, and for effecting the printing of error symbols therefor.

10. A print control logic circuit in accordance with claim **9** wherein each of said circuit modules further comprises:

circuit means for generating line stacking control signals for over-riding the normal operation of said associated memory so as to allow character data received at a rate faster than the printer can print the data stored in a first storage area of said memory, for a given line, to be temporarily stored in the second of said storage areas for not only the adjacent print line, but for any succeeding line thereafter until said second storage area has been filled, subject to any data for any line after the adjacent line, having not been previously transferred to said first storage area in the normal manner upon the processing and printing of the previously stored character data therein having been completed.

11. A print control logic circuit in accordance with claim **9** wherein said memory recovery circuit means in each of said circuit modules comprises presettable strap-option circuit means coupled to said associated memory for detecting the presence of and purging said memory of any invalid data characters stored therein whenever such characters represent an encoded binary number higher than a preset number representative of the highest order character in a chosen font of type characters employed in a given application.

12. A print control logic circuit in accordance with claim **11** wherein each of said circuit modules further comprises:

an auxiliary invalid character recovery circuit capable of detecting any invalid data characters stored in said associated memory, and for erasing said invalid characters independently of said strap-option circuit means, regardless of the length of a given character font employed in conjunction therewith, after a predetermined time-delay that is chosen to be greater than the time required for the number of different type character fonts employed to successively pass a given reference point on the printer, and for thereafter causing each equal character signal comparision representative of an invalid data character to effect the actuation of the proper print-inducing member at the proper time when an error symbol character included in a given font of type characters is momentarily brought into register therewith, thereby effecting the printing of an error symbol.

13. A print control logic circuit in accordance with claim **9** wherein each of said processing circuit modules further includes:

pulse-stretching means associated with the comparing and print signal generating means so as to allow said print signals to be respectively and selectively generated for a longer time period than is encompassed by the predetermined number of sub-scan periods defining a given print cycle.

14. A print control logic circuit in accordance with claim **9** wherein each of said processing circuit modules further includes:

means for genrating timing pulses in response to and dependent on the rate of speed at which each type character font moves past the print columns, to operate the circuitry in each circuit module in a manner which maintains a continuous synchronous relationship between the font type character signals generated and the read out of stored data characters from the associated memory during each sub-scan period, said timing pulses being transmitted in succession to the serially connected circuit modules.

15. A print control logic circuit in accordance with claim **14** wherein each of said circuit modules further includes:

means for generating an own circuit full status signal whenever a given storage area of said memory has been filled with data characters to be printed, and

shift register means, responsive to said timing pulses, for effecting the multiplexing of said timing pulses with said own full status signals, and for transmitting said pulses and signals to the next succeeding circuit module, but in a manner contingent upon all of the data characters to be stored, processed and printed by each preceding circuit module having been received and, as a result thereof, having generated its own circuit full status signal and then multiplexed that signal with a previously received timing pulse and transmitted both to the next succeeding circuit module which similarly responds thereto, in serial fashion.

16. A print control logic circuit in accordance with claim **14** wherein each of said processing circuit modules further includes:

means responsive to externally controlled signal stimulus for allowing each circuit module to receive, store and process character print data for any selected number of assigned print columns less than the maximum number of print columns that may be assigned to each circuit module by circuit design.

17. A print control logic circuit in accordance with claim **12** wherein each of said circuit modules further includes:

pulse-stretching means associated with the comparing and print signal generating means so as to allow said print signals to be respectively and selectively generated for a longer time period than is encompassed by the predetermined number of sub-scan periods defining a given print cycle, and

means for generating timing pulses in response to and dependent on the rate of speed of each character font in moving past the print columns, to operate the circuitry in each circuit module in a manner that maintains a continuous synchronous relationship between the font type character signals generated and the read out of stored data characters from the associated memory during each sub-scan period, said timing pulses being transmitted in succession to the serially connected circuit modules.

18. A print control logic circuit in accordance with claim **17** wherein each of said circuit modules further includes:

means for generating an own circuit full status signal whenever a given storage area of said memory has been filled with data characters to be printed,

shift register means, responsive to said timing pulses, for effecting the multiplexing of said timing pulses with said own full status signals, and for transmitting said pulses and signals to the next succeeding circuit module, but in a manner contingent upon all of the data characters to be stored, processed and printed by each preceding circuit module having been received and, as a result thereof, having generated its own circuit full status signal and then multiplexed that signal with a previously received timing pulse and transmitted both to the next succeeding circuit module which similarly responds thereto, in serial fashion, and means responsive to external signal stimulus for allowing each circuit module to receive, store, and process character print data for any selected number of assigned print columns less than the maximum number of print columns that may be assigned to each circuit module by circuit design.

19. A print control logic circuit for use in on-the-fly impact printers, for selectively actuating one or more out of a plurality of character print-inducing members whenever selective type characters forming a part of a continuously moving font of type characters are momentarily brought into respectively aligned and juxtaposed relationship with said print-inducing members so as to effect sequential printing of type character images along each successive line on a medium, but with the character images ultimately being positioned in the serial order in which the input data characters to be printed were received, said logic control circuit comprising:

a plurality of essentially identical, but interconnected data processing circuit modules, each one being assigned to process print data for a pre-determined number of print columns forming only a sub-group out of the total number of print columns defining the length of a given print line, with each circuit module thereby storing, processing and effecting the printing of input data characters as received in only the print columns assigned thereto, each of said circuit modules including:

sub-dividable memory means for storing input binary encoded data representative of the characters to be printed at particular positions along a given print line, said memory means including a free-running storage memory having two logic controlled storage areas, and further including phase controlled logic circuit means for selectively storing input character data in and reading said data out of said storage areas in a manner that allows incoming data intended for printing on one line to be stored in one of said distinct storage areas while any previously received data is still being read out of the other storage area, processed and printed in the preceding line, said phase controlled circuit means including a phase counter, a type character position counter and a print column counter to control the selective loading of data characters in and their read out from the two storage areas of said memory,

means for generating type character data in multiple phase sequences during each print cycle, said data being representative of and identifying the particular ones of said moving font type characters that are respectively brought into registry with associated ones of said character print-inducing members at any particular point in time during each phase sequence, and

means for processing said stored data for printing after it has been selectively read out of said storage areas of said memory, by comparing it with said generated type character data, and for generating a distinct print signal, in response to each valid equal signal comparison, for use in actuating at the proper time the particular one of the associated character print-inducing members that will effect the printing of a character image representative of the compared type character at the proper position along a given print line.

20. A print control logic circuit in accordance with claim **18** wherein each of said circuit modules further includes:

means for generating timing pulses in response to and dependent on the rate of speed at which each type character font moves past the print columns to operate the circuitry in each circuit module in a manner that maintains a continuous synchronous relationship between the font character signals generated and the read out of stored data characters from the associated memory during each sub-scan period, said timing pulses being transmitted in succession to the serially connected circuit modules,

memory recovery circuit means for detecting the presence of and effecting the erasing of any invalid data characters stored in said associated memory during the printing of each print line, and for effecting the printing of error symbols therefor,

means for selectively generating first and second storage signals and first and second processing sig-

nals, for controlling the loading of encoded data characters in and the read out thereof from the first and second storage areas of said memory in a selective manner, and

pulse-stretching means associated with the comparing and print signal generating means so as to allow said print signals to be respectively and selectively generated for a longer time period than is encompassed by the predetermined number of sub-scan periods defining a given print cycle.

21. A print control logic circuit in accordance with claim **20** wherein each of said memory recovery circuit means comprises presettable strap-option circuit means coupled to said associated memory for detecting the presence of and purging said memory of any invalid data characters stored therein whenever such characters represent an encoded binary number higher than a preset number representative of the highest order character in a chosen font of type characters employed in a given application, and wherein each of said circuit modules further comprises:

circuit means for generating line stacking control signals for over-riding the normal operation of said associated memory so as to allow character data received at a rate faster than the printer can print the data stored in a first storage area of said memory, for a given print line, to be temporarily stored in the second of said storage areas for not only the adjacent print line, but for any succeeding line thereafter until said second storage area has been filled, subject to any data for any line after the adjacent line, having not been previously transferred to said first storage area in the normal manner upon the processing and printing of the previously stored character data therein having been completed;

an auxiliary invalid character recovery circuit capable of detecting any invalid data characters stored in said associated memory, and for optionally erasing said invalid characters independently of said strap-option circuit means, regardless of the length of a given character font employed in conjunction therewith, after a predetermined time-delay that is chosen to be greater than the time required for the number of different type character fonts employed to successively pass a given reference point on th printer, and for thereafter causing each equal character signal comparison representative of an invalid data character proper time when an error symbol character included in a given font of type characters is momentarily brought into register therewith and, thereby, effect the printing of an error symbol, and

strap-option circuit means for allowing each circuit module to receive, store and process character print data for any selected number of assigned print columns less than the maximum number of print columns that may be assigned to each circuit module by circuit design.

22. In a printer mechanism including a moving type carrier having at least one font of characters positioned therealong, a plurality of actuable print-inducing members aligned with the carrier, and an input-output controller for interfacing the incoming encoded print data with the printer, a print control logic circuit for use in generating character print signals to selectively actuate one or more of the print-inducing members out of a plurality of such members when selected ones thereof are periodically and sequentially momentarily aligned with selected and respective ones of said type characters inducing members in accordance with the order in which input data characters are received by the control circuit for subsequent printing, said circuit comprising:

sub-divided memory means for storing input binary encoded data representative of the characters to be printed at particular positions along a given print line, said memory means having at least two logic controlled and operationally distinct storage areas, and including means for selectively storing encoded character data in and for reading said data out of said storage areas of said memory means in a manner that allows incoming data intended for printing on one line to be stored in one of said distinct storage areas, while any previously received data is still being read out of the other storage area, processed and printed in the preceding line,

means for generating type character font data in a predetermined number of phase sequences during each print cycle, said data being representative of and identifying the particular ones of said font type characters on said continuously moving carrier that are respectively brought into registry with associated ones of said character print-inducing members at any particular point in time, during each phase sequence, the number of said phase sequences being dependent on the relationship between the spacing of adjacent type characters on the carrier and the spacing of print-inducing members, with the spacing of the former being wider and with the number of phase sequences being chosen so as to allow every type character to be brought into alignment with every print-inducing member during a given print cycle, and

means for processing any stored character data for printing after it has been read out of said memory means, by comparing said stored data with said generated type character data, and for generating a distinct print signal, in response to each valid equal signal comparison, for use in actuating at the proper time the particular one of the character print-inducing members that will effect the printing of the desired character image representative of the compared type character at the proper position along a given print line in accordance with the order in which the input data characters were received for printing.

23. In a printer mechanism in accordance with claim **22,** said print control logic circuit being sub-divided into a plurality of essentially identical, but interconnected data processing circuit modules, each circuit module including separate ones of said memory means, character generating means and comparing means, and with each circuit module being assigned to process print data for a pre-determined number of print columns forming only a sub-group out of the total number of print columns defining the length of a given print line, with each circuit module thereby storing, processing and effecting the printing of input data characters as received in only the print columns assigned thereto.

24. In a printer mechanism in accordance with claim **23,** said memory means of each of said circuit modules including a free-running memory, phase controlled type character position means and print column counter means for logically controlling access to and

sub-dividing said memory into distinct dual storage areas, and memory recovery circuit means for detecting the presence of and effecting the erasing of any invalid data characters stored in said memory during the printing of each print line.

25. In a printer mechanism in accordance with claim **24,** each of said recovery means of said control circuit further including:

presettable strap-option circuit means copuled to said associated memory for detecting the presence of and purging said memory of any invalid data characters stored therein whenever such characters represent an encoded binary number higher than a preset number representative of the highest order character in a chosen font of type characters employed in the printer mechanism, and for thereafter causing each equal character signal comparison representative of an invalid data character to effect the actuation of the proper print-inducing member, corresponding in position to the position in which the invalid data character is stored in the memory, at the proper time when an error symbol type character included in a given font of type characters carried on the carrier is momentarily brought into registry therewith, thereby effecting the printing of an error symbol.

26. In a printer mechanism in accordance with claim **25,** each of said processing circuit modules of said control circuit further including:

an auxiliary invalid character recovery circuit capable of detecting any invalid data characters stored in said associated memory, and for optionally erasing said invalid characters independently of said strap-option circuit means, regardless of the length of a given character font employed in said printer mechanism, after a predetermined time-delay that is chosen to be greater than the time required for the number of different type character fonts employed to successively pass a given reference point on the printer, and for thereafter causing each equal signal comparison representative of such an invalid data character to effect the actuation of the proper print-inducing member, corresponding in position to the particular position in which the invalid data character was initially stored in said memory, at the proper time when an error symbol character included in a given font of type characters carried on the carrier is momentarily brought into register therewith, thereby effecting the printing of an error symbol, and

pulse-stretching means associated with the comparing and print signal generating means so as to allow said print signals to be respectively and selectively generated for a longer time period than is encompassed by the predetermined number of sub-scan periods defining a given print cycle.

27. In a printer mechanism including a moving type carrier having at least one font of type characters positioned therealong, a plurality of actuable print-inducing members aligned with the carrier, and an input-output controller for interfacing the incoming encoded print data with the printer mechanism, a print control logic circuit for use in generating print signals to selectively actuate one or more of the print-inducing members out of a plurality of such members when selected ones thereof are periodically and sequentially momentarily aligned with selected and respective ones of said type characters in accordance with the order in which input data characters are received by the control circuit for subsequent printing, said circuit comprising:

a plurality of essentially identical, but interconnected data processing circuit modules, each one being assigned to process print data for a pre-determined number of print columns forming only a sub-group out of the total number of print columns defining the length of a given print line, with each circuit module thereby storing, processing and effecting the printing of input data characters as received in only the print columns assigned thereto, each of said circuit modules including:

sub-dividable memory means for storing input binary encoded data representative of the characters to be printed at particular positions along a given print line, said memory means including a free-running storage memory having two logic controlled storage areas, and further including phase controlled logic circuit means for selectively loading input character data in and reading said data out of said storage areas in a manner that allows incoming data intended for printing on one line to be stored in one of said distinct storage areas while any previously received data is still being read out of the other storage area, processed and printed in the preceding line, said phase controlled circuit means including a phase counter, a type character position counter and a print column counter to control the selective loading of data characters in and their read out from the two storage areas of said memory,

means for generating type character data in multiple phase sequences during each print cycle, said data being representative of and identifying the particular ones of said moving font type characters that are respectively brought into registry with associated ones of said character print-inducing members at any particular point in time during each phase sequence, and

means for processing any stored character data for printing after it has been read out of said free-running memory, by comparing said stored data with said generated type character data, and for generating a distinct print signal in response to each valid signal comparison, for use in actuating at the proper time the particular one of the character print-inducing members that will effect the printing of the desired character image representative of the compared type character at the proper position along a given print line.

28. In a printer mechanism in accordance with claim **27,** each of said processing circuit modules of said control circuit further including:

means for generating timing pulses in response to and dependent on the rate of speed at which each type character font moves past the print columns, to operate the circuitry in each circuit module in a manner that maintains a continuous synchronous relationship between the font type character signals generated and the read out of stored data characters from the associated memory during each sub-scan period, said timing pulses being transmitted in succession to the serially connected circuit modules,

memory recovery circuit means coupled to said associated memory for detecting the presence of and effecting both the erasing of any invalid data characters stored in said memory during the printing of each print line, and the printing of error symbols therefor, and wherein each of said memory means further includes:

means for selectively generating first and second storage signals and first and second processing signals, for controlling the loading of encoded data characters in and the read out thereof from the first and second storage areas of said memory in a selective manner.

29. In a printer mechanism in accordance with claim **28,** each of said memory recovery circuit means further comprising presettable strap-option circuit means coupled to said associated memory for detecting the presence of and purging said memory of any invalid data characters stored therein whenever such characters represent an encoded binary number higher than a preset number of representative of the highest order character in a chosen font of type characters employed in a given application, and wherein each of said circuit modules further comprises:

pulse-stretching means associated with the comparing and print signal generating means so as to allow said print signals to be respectively and selectively generated for a longer time period than is encompassed by the predetermined number of sub-scan periods defining a given print cycle.

30. In a printer mechanism in accordance with claim **28,** each of said circuit modules further including:

an auxiliary invalid character recovery circuit capable of detecting any invalid data characters stored in said associated memory, and for optionally erasing said invalid characters independently of said strap-option circuit means, regardless of the length of a given character font employed in conjunction therewith, after a predetermined time-delay that is chosen to be greater than the time required for the number of different type character fonts employed to successively pass a given reference point on the printer mechansim, and for thereafter causing each equal character signal comparison representative of an invalid data character to effect the actuation of the proper print-inducing member at the proper time when an error symbol character included in a given font of type characters is momentarily brought into register therewith, thereby effecting the printing of an error symbol, and

operable circuit means within each circuit module, responsive to an externally applied signal, for allowing each circuit module to receive, store and process character print data for any selected number of assigned print columns less than the maximum number of print columns that may be assigned to each circuit module by circuit design.

* * * * *