May 13, 1969 R. C. ENGELHARDT 3,444,529

CONTROL ADAPTER ASSEMBLY FOR A CHAIN PRINTER

Filed Jan. 3, 1967 Sheet 1 of 3

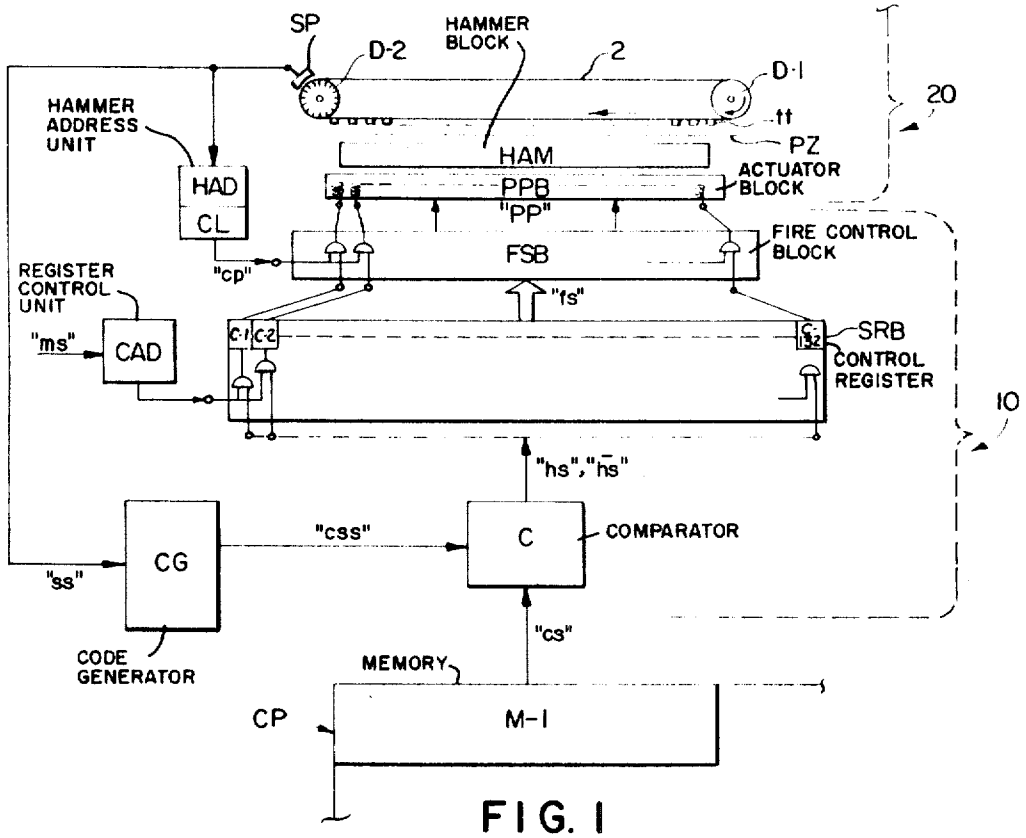


FIG. 1

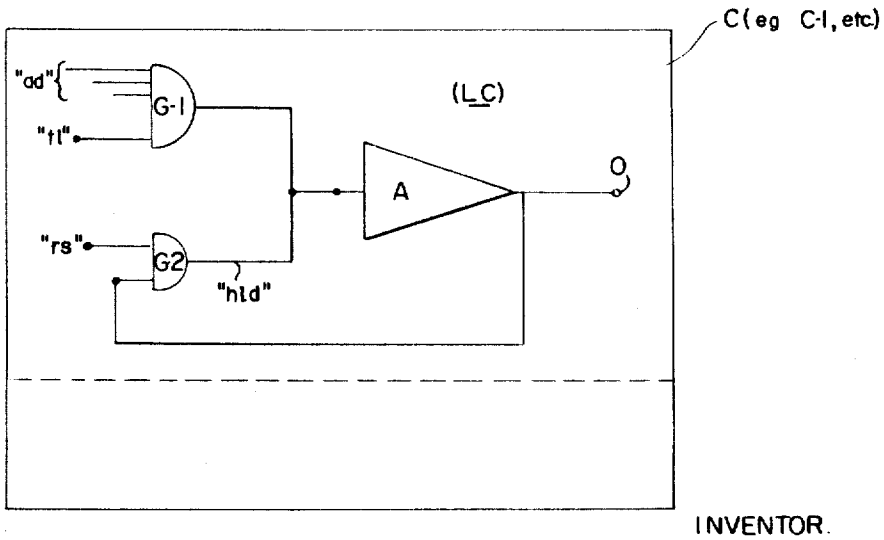


FIG. 2

INVENTOR.
ROBERT C. ENGELHARDT
BY
John J. McCormack
ATTORNEY

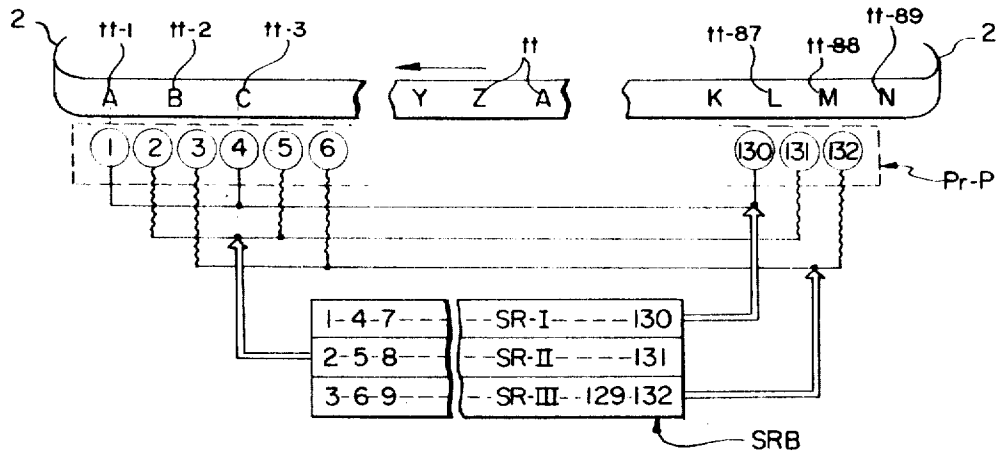


FIG. 3

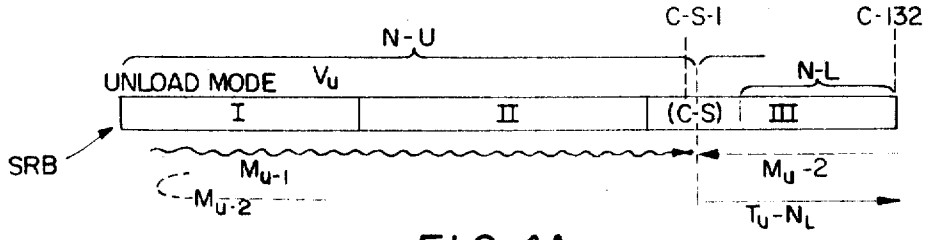


FIG. 4A

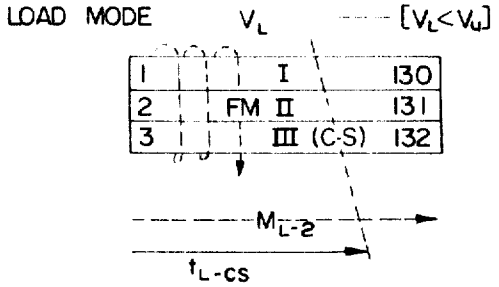


FIG. 4B

INVENTOR.
ROBERT C. ENGELHARDT
BY John J. McCormack
ATTORNEY

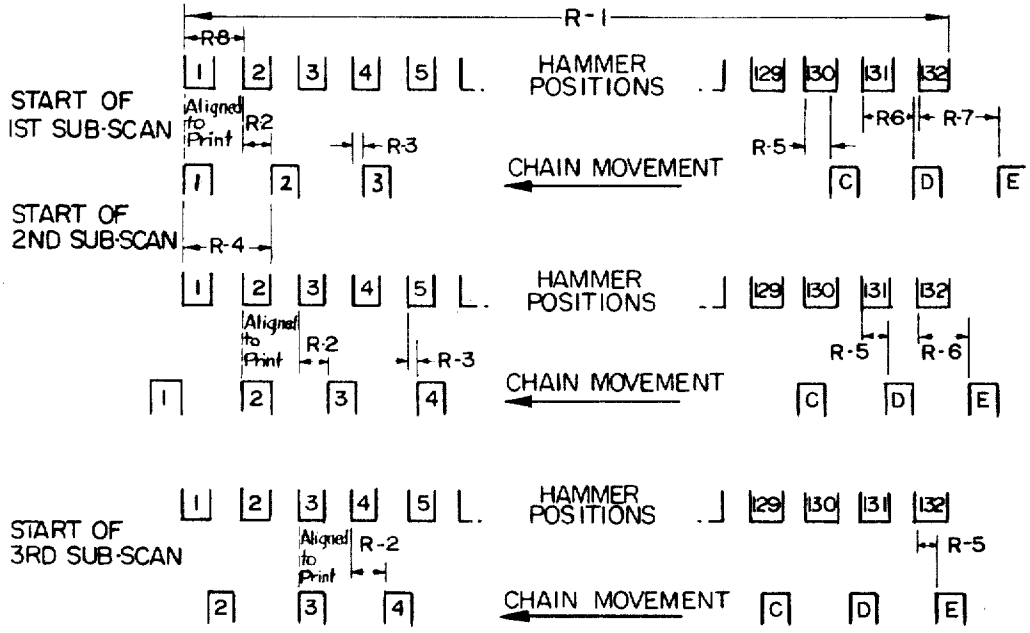


FIG. 5

INVENTOR.
ROBERT C. ENGELHARDT
BY
John J. McCormack
ATTORNEY

United States Patent Office 3,444,529 Patented May 13, 1969

3,444,529

CONTROL ADAPTER ASSEMBLY FOR A CHAIN PRINTER

Robert C. Engelhardt, Waltham, Mass., assignor to Honeywell Inc., Minneapolis, Minn., a corporation of Delaware

Filed Jan. 3, 1967, Ser. No. 606,876
Int. Cl. G11b 13/00
U.S. Cl. 340—172.5 12 Claims

ABSTRACT OF THE DISCLOSURE

A control adapter assembly for controlling a special chain printer having a "leap-frogging," "by-three's" print-slug registration order (for each full print-position-shift of the chain) providing a buffer storage assembly for accessing memory locations serially, i.e., in print-position sequence, and responsively storing hit signals, while accommodating a non-serial, "leap-frogging" buffer unloading operation for actuating print hammers in this "leap-frogging" order. More particularly, this buffer memory uses a resettable latch circuit for each such memory location, each being coupled to control the energization of an associated print hammer, the mode of addressing main memory and filling these locations being controlled so as to accommodate this "leap-frogging" unload mode and also a simple "two-step" reset mode.

Background, features of invention

In deriving printed output from computer and other data-handling systems, it is desirable, in certain instances, to use chain printers of a type well known in the art and understood to be arranged to sweep an array of (type) slugs across a prescribed printing plane, the chain thus scanning all print-positions across a print line (of a data medium) at very high speeds so that each type slug successively passes every print-position, and the hammers associated therewith, to be impacted thereby at selected character coded print times. A familiar such printer chain has five identical sets of 48 different type slugs each, scanning these past 132 print-positions in a print line. Each print line of characters is characteristically stored, in coded form and in printing order, in a prescribed memory bank, such as the core matrix in a computer central processor (CPU) memory. As the chain sweeps past each print-position, character coding means conventionally generate character signals identifying successive character slugs passing that print-position, e.g. a code generator supplying, for each print-position, 48 successive different character codes to be compared with the character code in the CPU memory cell corresponding to that print-position.

It is a peculiarity of this type of chain printer that, for efficiency, etc., the type slugs are slightly staggered to register successively with each of the 132 print-positions, so that when a first slug comes into registry with a first print position, for instance, the following 131 slugs will each successively come into registry with 131 different print-positions at different times (before a second slug next registers with that first print-position). It also happens that, again for efficiency of operation, etc., the order of these successive registrations, and thus of hammer enabling, is not the aforementioned "printing order" (i.e., of the physical arrangement of the print-positions), but rather is a "leap-frogging" one, as detailed below. These organizational sequence differences between the character cells in main memory and the hammer-enabling operation (i.e., "printing-order" vs. "leap-frogging order") is one of the considerations that has heretofore required that the print

commands (hit signals) be loaded into buffer memory at one phase of a print cycle; then unloaded (hammers fired) at a different phase—thus introducing a rather drastic delay in overall printer operation, slowing printout and, absent extensive added memory and/or control means, preventing continuous operation of the chain printer. The present invention confronts these problems, providing an adapter means arranged to allow such continuous printing, yet requiring a very few simple elements without introducing such "expected" disadvantages as requiring extra memory locations, tying-up main memory, or affecting its operational mode or that of the printer itself. Meeting these latter constraints is one of the most surprising features of the invention, as workers in the art can well appreciate. This will become more apparent upon considering the discussion of the prior art below.

Thus it is an object of the invention to meet the aforementioned needs and problems and provide the foregoing and other features and advantages. Another object is to provide a storage adapter arrangement to couple control signals from a memory array ordered in printing sequence to operate an array of print actuators operating in a different, "leap-frogging" sequence. A related object is to provide a storage adapter for buffering such a memory array to a chain printer hammer array operating in "skip-order." Another related object is to provide such an adapter for a printer having a "leap-frogging," "by-N's" operating mode, N being the print-position skip between hammer enablings. Yet another object is to provide such an adapter control unit including a "by-N's" hit register, with N sub-registers. Still another object is to provide such an adapter for such a printer where the skip (N) is 3 print positions. Still another object is to provide such a control/adapter register using resettable latching cells to store "hit" signals. Still another object is to provide such an assembly where such a "skipping" unload sequence for the register can overtake the "non-skipping" loading mode, given conventional control. Still another object is to control such an adapter register so as to begin reloading for the next print cycle during a prescribed portion of a subject unload (printing) cycle. Still another object is to provide such an adapter/control register using conventional latch type circuits. Yet another object is to provide such control wherein said latch circuits may be reset together at a few convenient reset times. Still another object is to provide such control simply using relatively conventional character code generator means, comparator means and hit signal register, together with a simple register control means providing the aforementioned pre-load and reset sequences. Still another object is to provide such a control adapter having each storage location thereof coupled to a respective gate in a fire control block, each gate having its output in turn coupled to the hammer actuator at a respective print-position and also having an input leg thereof coupled to a hammer address block providing the aforementioned "leap-frogging" enabling sequence. Still another object is to provide such control over register operation and actuator operation, sequence, etc., respectively, using control means essentially comprised of simple counter means.

The aforementioned objects and features of advantage are provided in a preferred embodiment of the invention, as described below, generally comprising an adapter register, each memory location therein comprising a resettable latch-circuit coupled to a respective print-position gate in a hammer fire control block of a "skip-firing" type chain printer, i.e., one where hammer firing is enabled in print-position skipping, i.e., "leap-frogging by-three's," sequence for each print-scan (major shift of the chain). Such a fire sequence occurs with each print-scan. These registers are adapted to be loaded from a prescribed print-

line-associated set of memory locations storing character coded signals therein, the output thereof being character code signals compared for each print-scan in "printing order" and resultant "hit signals" being steered into successive print-position-associated latch circuits (cells) in the register. These cells are arranged in three sub-registers, each corresponding to one of the sub-scanning skip sequences of successive hammer enabling, this sub-register allocation and the loading/unloading control thereof being implemented by a register control unit adapted to begin pre-loading the register cells at a prescribed point in the unloading cycle, to reset cell groups in common at a few convenient times, and to generally control the aforementioned operation of the register.

In the drawings, wherein like reference numerals denote like parts:

FIGURE 1 shows, in schematic block diagram form, an embodiment of the control adapter unit according to the invention as coupled to control a particular "skip-firing" type chain printer from a memory bank storing signals for printing a particular line;

FIGURE 2 indicates, in block diagram form, an exemplary one of the "hit" storage circuits comprising the register of the adapter unit indicated in FIGURE 1;

FIGURE 3 illustrates in highly schematic, fragmentary, form the relative positions of a few exemplary (type font) chain slugs with respect to related print-positions, also indicating the relative coupling relationship of these print-positions to the adapter register of FIGURE 1;

FIGURE 4A is an operational-sequence diagram for the register of FIGURES 1–3 indicated in the foregoing figures, correlating the arrangement and grouping of memory cells therein with various unloading sequences; while FIGURE 4B indicates relatively the same thing for related loading sequences; and

FIGURE 5 very schematically indicates the relative alignment of a few exemplary hammer (print) positions and chain slugs at the start of three successive sub-scans, during each of which the chain executes a skipping series of incremental minor shifts.

General organization

FIGURE 1, in a schematic block diagram, shows a print control arrangement **10**, i.e. a control interface or adapter system coupling a known type chain printer (actuate-control) arrangement **20** to a known type print command memory bank CP. Printer arrangement **20** includes a chain **2** (including type elements, or slugs *tt*), a strobe detector SP, a hammer block HAM (132 hammer means, one for each "print-position" along print plane PZ) and an associated bank of hammer-actuating solenoids, etc. in an actuator block PPB. Adapter arrangement **10** includes a bank of solenoid switching means in a fire-control block FSB for controllably energizing "selected" solenoids in PPB, with print-signals PP, responsive to strobe code signals "*cp*" from a hammer address unit HAD, also including a control register SRB, a register control unit CAD, a character code signal generator CG and compare means C. Adapter **10** serves, generally, to couple the "leap frogging" operation of block FSB, etc. with the printing order accessing of memory section M–1, the latter customarily comprising part of a central processor memory CP. This CP will include "print line" sets of memory locations, such as the print-line (coded) character signals understood as stored in memory M–1, these comprising the coded character signals for the given line of print. Coded character signals are stored in M–1 in the aforesaid "printing order," that is the order of disposition of print-positions P*r*–P (**1–132**). Adapter **10** will access and unload M–1 in this natural, "printing order," while also controlling printer unit **20**, to operate in its own different "skip-firing" mode as hit-controlled by the output of M–1.

Printer control **20** is generally organized as follows: A printing area PZ including 132 print positions P*r*–P (FIGURE 3) is disposed in operative relation between chain **2** and an associated array of 132 print-hammers in hammer block HAM. Associated actuating means (solenoids) in solenoid block PPB are controlled by associated fire signals "*pp*" from firing control (or gating) block FSB to thrust an associated hammer in HAM for printing at the selected position. Block FSB is adapted to gate the energization of a selected one of these 132 actuating means under the timing control of hammer (addressing) control HAD, synchronizng FSB with the successive registering of slugs *tt* before different print-positions. HAD includes a "clock" portion CL, adapted to emit clock pulsess "*cp*" at prescribed (11 μs.) intervals, being initiated (and resynchronized) periodically by a strobe pulse from strobe pick-up SP synchronized with the movement of chain **2**. These elements are adapted to characteristically fire the print hammers "somewhat serially" (rather than all together, in parallel, as in a drum printer) and in the aforementioned peculiar "leap frogging, by-three's" (or skipping) firing order. Unit HAD is thus adapted to provide a series of 132 successive solenoid-enabling signals, gated through gating block FSB in the prescribed skipping order for responsive actuation of associated solenoids in energizing solenoid block PPB (when selected by a hit signal coupled from register SRB).

The memory portion of adapter unit **10**, i.e. register SRB, comprises 132 memory locations (cells C–**1**, etc. through C–**132**) in the form of latching circuits for storing "hit-bits" (1 for each print position—see circuit LC in FIGURE 2). Unit **10** also couples cell (addressing) control unit CAD, according to the invention, so as to "steer-in" character-select ("hit," "no-hit") signals (*hs*, $\overline{hs}$ respectively) into cells C–**1** to C–**132**, accessing them serially to be filled in "printing order" from successive associated character locations in memory bank M–**1** for this print line. Signals *hs*, $\overline{hs}$ are coupled through an intermediate compare unit C whose output is controlled by character code generator CG to indicated "hits" (*hs*) or "no-hits" ($\overline{hs}$), as known in the art. According to the invention unit CAD "steers-in" these "hit-bits" at a carefully prescribed cycle time (detailed below) adapted to optimize the unloading of gating block FSB, etc. Unit CG is arranged to generate a code sequence corresponding to the "upcoming" code array (that is, according to the array of upcoming type fonts *tt*, about to register with each of the 132 print-positions at this print-scan). The serial input from C to SRB (signals "*hs*," "$\overline{hs}$"), when so steered into cells C–**1** through C–**132** (corresponding to respective print-positions P*r*–P) will determine whether the hammer at each position will be fired during this print-scan, there being 48 such scans (one for each character on chain **2**).

Chain printer characteristics

The characteristics of the type chain printer **20** that is so usefully controlled according to the invention will now be detailed to illustrate this application thereof, though other analogous "non-printing order" sequencing multi-actuator arrangements may be visualized. As schematically indicated in FIGURE 1, the multi-actuator printer arrangement **20** here comprises the high speed chain printer alluded to above which is, in general, adapted to be controlled by adapter unit **10** in response to character printing signals in exemplary memory bank M–**1**. This chain printer is a familiar such output medium for data processing systems, although other similar types will be recalled by those skilled in the art. Such a printer may be generally understood to comprise a chain assembly **2** driven continuously at high speeds on a pair of sprockets D**1**, D**2**, one of the sprockets (D**2**) being coupled to synchronously emit a strobe output to be read-out by a strobe pick-up SP for thereby providing strobe signals to hammer control unit HAD, and to char-

acter generator CG. The activation of a particular "selected" solenoid means will magnetically thrust an associated hammer means in hammer block HAM through printing zone PZ printingly against a "selected" one of type-slugs *tt* that is confronted at that print-position at that particular time. This printer chain **2** (or character font assembly) will be recognized as comprising a plurality of slugs (2 characters to a slug) fastened to an endless steel tape spanning all print-positions in a print line, thus forming an endless moving type font array, one character high and 240 characters long. Chain assembly **2** is driven continuously in a horizontal plane at a very high speed and a "selected" character-font thereon is impacted by the hammer at the selected print-position, pressing the paper form and intermediate ribbon printingly against the concerned slug, as known in the art. The described alphanumerical high speed printer will be understood as printing 600 lines per minute (including single space time) using a 48-character type array (including numbers and special characters), having five complete such arrays on the chain **2** for a total of 240 characters per chain. This printer operates with 132 print positions, having a horizontal spacing of ten characters per inch, though of course various modifications of this and other features in the aforementioned specifications are inferrable as known in the art.

The printing mode is serial, one character at one print position followed by another (not adjacent) character at another print position, and proceeding in the mentioned "leap-frogging, by three's" or skipping sequence of slug alignment with print-positions (see FIGURE 5 and Table I for illustrated example of sequence). This "leap-frogging by-three's" order of slug registration may be illustrated as follows:

At hammer #**1**, followed by every third hammer (**4, 7** to **130**); then at hammer #**2** followed by every third hammer (**5, 8** to **131**); and then at hammer #**3** followed by every third hammer (**6, 9** to **132**). This sequence is repeated 48 times (for the 48-character array described) to print a single line, insuring that every print-position hammer has been exposed to one complete type array.

Thus, with this alphanumerical 132-position printer, there are 6336 possible print selection times (options to print); however, at most only 132 of these being used to print a line. During each of these times, storage locations are addressed to determine the character in storage corresponding to each hammer position. Of these options to print, one, and only one, is used to select each print position in which a character is to be printed. This use is called "allocating a print time to a print position" and causes the hammer to fire at a controlled time, when the character (slug) to be printed is registered with the selected "print position."

The hammers (and print positions) are spatially related to the type slugs *tt*, as follows (example in FIGURE 3 and FIGURE 5): the distance between the first and the last print position (**1–132**) is called the "line-span." The The distance between adjacent print positions is called the "print-span" (here 0.1 inch). The distance between adjacent type faces along the type chain is called the "type-span." Nominal type span is 50% greater than the print span, but actual type span is about 0.1505 inch. The small difference in length that makes the actual type span greater than the nominal type span exists because the chain travels about one mil in every 11 microseconds, this period being the "print selection time" to be used for storage interrogation and for making a comparison to determine the print time allocation.

The timing arrangement is as follows: The time required for the type to move one-half of a print span is called a "sub-scan." The time required for three sub-scans is called a "print-scan." The first sub-scan starts when a type is aligned with the first print position (FIGURE 5), the second sub-scan starting when a type is aligned with the second print position, and the third sub-scan starting when a type is aligned with the third print position. A print-scan thus comprises a first, second and third sub-scan. During each sub-scan, one-third of the hammers, each in progressing sequence, are optioned, at 11 microsecond intervals, to some character in the type array. At the end of a print scan, each of the hammers will thus have been optioned to one character in the type array. Then, at the end of 48 print scans, all of the hammers will have been optioned to a complete type array (with 48 type characters). Therefore, 48 print-scans are required to print a line. Thus, for the printing rate of **600** lines per minute (type moves at 90 in./sec.) a sub-scan is 555 microseconds, a print-scan is 1655 microseconds and a print-line is about 80 milliseconds. Normally, a clock in the system (CL) is started (resynchronized by SP) at the beginning of each sub-scan and is stopped in each sub-scan after 44 positions are scanned (for 132-position print line) and have had an option to print. This allows the system and the printer chain to be resynchronized for each sub-scan so that any slight difference in the relation of the hammer unit to the type array is corrected with every sub-scan. Thus, it will be appreciated that the physical design of the printing mechanism requires the aforementioned "leap frogging" printing sequence, "by-three's," that is, by every third hammer (every second type character). The system circuits must evidently be designed to accommodate the mechanical operation of the printer.

FIGURE 5 may be understood to be a plot of a theoretical location of type in relation to hammers, remembering that the chain travels about one mil in an 11 microsecond period and also noting the aforementioned relationship of print span to type span. Actual cases may differ slightly, but any actual differences will be reconciled after each sub-scan by the aforementioned resynchronizing of the system to the chain. Thus, FIGURE 5 schematically represents the relation of the chain to the hammers at three specific points in time, namely the beginning of each sub-scan for one entire print-scan. Table I may be used with FIGURE 5 for relating memory location, hammer enabling sequence and chain shifting. Distances R in FIGURE 5 are exemplarily indicated in Table II, below.

TABLE I

	Print option	Cumulative chain movement (in.)	Storage location addressed (in Bank M-1)	Character optioned if present	Hammer position
First Sub-Scan	1st	.000	1	"1"	1
	2nd	.001	4	3	4
	3rd	.002	7	5	7
	44th	.043	130	C	130
Re-Sync		(Add .0075)			
Second Sub-Scan	45th	.0505	2	2	2
	46th	.0515	5	4	5
	47th	.0525	8	6	8
	88th	.0935	131	D	131
Re-Sync		(Add .0075)			
Third Sub-Scan	89th	.101	3	3	3
	90th	.102	6	5	6
	91st	.103	9	7	9
	132nd	.144	132	"E"	132

TABLE II

Segment (R):	Distance (in.)
R–1	13.2
R–2	0.0505
R–3	0.001
R–4	0.1505
R–5	0.043
R–6	0.0935
R–7	0.144
R–8	1.0

To summarize, in order to print one line of information, each hammer must be given the option to print all 48 characters. It takes three sub-scans to have all 132 hammers optioned to print one character, this being called a complete scan. It takes 48 print scans to make one print line and 144 sub-scans. It takes 6336 print options to print a line of information.

Novel adapter unit

As indicated rather generally in FIGURE 1, adapter unit **10** mates the aforedescribed chain printer assembly **20** to the relatively conventional memory bank M–1. The elements and operation of adapter unit **10** may be understood to have the following structural and operational characteristics. Storage means (register) SRB is the focal point of adapter **10** and will be generally understood to have 132 memory cells (C–1 through C–132) filled with 132 hit signals (*hs*, $\overline{hs}$), one for each print position for each of the 48 print-scans. SRB is so filled each time a particular slug *tt* shifts one full print position. It will be recognized, as aforedescribed, that each print-position will have one slug registered therewith every complete print-scan (i.e., every three sub-scans or every 132 incremental chain-shifts) but each slug *tt* will register with some one print position every other sub-scan. Note that each slug registers with a different corresponding print position successively and at a slightly different time, doing so according to the aforedescribed "by-three's" sequence. During each "print-scan," it will be recognized that three sub-scans will successively occur in the "by-three's" sequence, being sequence controlled by HAD through the gates in gate block FSB as described. Thus, in effect, during the time that a slug has shifted one complete print position, adapter memory SRB has been loaded from memory M–1 through comparator C in all 132 storage cells C1–C132, serially; therewhile being also serially unloaded in the "by-three's" order.

How the aforementioned loading and unloading modes of register SRB are effected according to the invention will be described in more detail below. However, it may be generally understood that address control CAD provides control signals to start the loading at the proper time and HAD the unloading at its proper time. Unit HAD also controls the reset time for cells C–1, etc., in SRB, controlling CAD to reset a first portion thereof at a first selected time (when C–S is just unloaded) and then to reset the remainder at second selection time (when C–132 is just unloaded). As explained below, this first portion comprises the cells in SRI, SRII and through C–S in SRIII according to the invention. Thus HAD may preferably include an array of counters to, once started by strobe SP, responsively count successive chain-shift times and indicate the described "by-three's" registration order at the hammer actuator gates in FSB. The unloading of SRB is thus effected in a prescribed controlled manner to apply print (or "hit") signals "*hs*" to respective sequential ones of the 132 gates in block FSB. The loading of SRB is also novel and effectively comprises running the 132 print position locations in memory through comparator C where the coded character signals "*cs*" for each of these print positions are successively compared to associated ones of the "upcoming" array of character code signals for the corresponding 132 slugs *tt* about to register, sequentially, during this upcoming print-scan time. This code pattern is generated by code-generator CG in response to a starting strobe pluse "*ss*" from pick-up SP, as known in the art. Thus, generator CG is adapted to generate 48 different sets of 132 coded character signals, each character set comprising a different sequence of 132 character codes, one for each print-position according to the type character "coming up" there. CAD preferably includes an array of counters which, when initiated (at C–S time by signal "*ms*"), addresses cells C–1, etc., in "printing-order" succession, coupling them to the synchronously controlled operation of compare unit C, generator CG and memory contents M–1 to store "hit" or "no-hit" signals therein for this print-scan (likewise for all 47 other scans).

As indicated in FIGURE 2, each of the 132 memory locations C in register SRB preferably comprises a latched type memory circuit LC. Circuit LC includes an input gate G–1 having a plurality of address legs "*ad*", plus a "hit" leg "*tl*", for application of hit signals from comparator C as aforedescribed. Gate G–1 is coupled to the input of an amplifier A, a known "gate buffer" type amplifier, such as provides sufficient signal gain to derive the needed logical "fan-out." The "hold" function of this circuit is performed at a second gate G–2 to one leg of which the output from amplifier A is fed back to initiate a hold signal "*hld*" until "reset" is signaled at "*rs*." G–2 may be characterized as an "inhibit" gate whereby the prescribed reset signal "*rs*" may be applied to terminate output hold signal "*hld*."

Prior art

A number of alternatives to this control system exist in the art, all more complex, expensive and unwieldy than using merely the disclosed "hit register" SRB. However, such a register will be seen to be faced with a problem of transposing from the aforementioned "printing order" sequence of loading and the aforementioned different skipping unloading sequence (firing order). As stated, hammers are fired in a "by-three's" fire order, that is, the hammers/print positions are fired in the following of print positions (FIGURE 3): First position #**1**, **4**, **7** etc. through **130**; next, position #**2**, **5**, **8** etc. through position **131**; finally, position #**3**, **6**, **9** etc. through the last position **132**.

For example, FIGURE 3 illustrates this "by-three's" firing order (sequence of slug registry across print positions **1** through **132**). By way of illustration and proceeding across chain **2** from right to left, it will be noted that the successive type slugs ("upcoming" slugs *tt*) have been designated as "*tt*–**1**, *tt*–**2**, *tt*–**3**," etc., concluding with *tt*–**87**, *tt*–**88** and *tt*–**89** (registration of these last slugs calling up the 43rd, 88th, and 132nd print-options, respectively). As aforementioned, the peculiar successive-registry disposition of slugs *tt* on claim **2** is responsible for this. Thus, it will be understood that character A comprising slug *tt*–**1** is, now, exactly registered for firing opposite print position (Pr–P) #**1**.

The hammer at Pr–P#**1** will now be fired if, and only if, character A is to be printed at position **1** on the subject print line. All other slugs are out of registry, e.g. slug *tt*–**89** will be seen to register successively with each of the 132 Pr–P in order, there being however, in general, 88 different sub-shifts between each successive registration for any one slug. Thus, a number of scans, corresponding to the number of characters available, will be performed. The next registering slug will be slug *tt*–**3** (character C) about to come into registry with print position #**4**. Continuing, it will be understood that the succeeding slugs (*tt*–**5** next, then *tt*–**7**, etc.) will come up successively at prescribed print-positions (at Pr–P #**7**, then Pr–P #**10** etc.) in the described skipping manner concluding with Pr–P **130** (*tt*–**87** there) in the first sub-scan group, thereafter continuing into the second, "by-three's" sub-scan group of successive registrations (be-

ginning at Pr–P#2, *tt*–2, i.e. B there; then Pr–P#5 concluding with Pr–P#131, slug *tt*–88 there); and lastly, following with the registrations of the third sub-scan group in like manner, (i.e. at Pr–P#3, *tt*–3 there—concluding with Pr–P#132, *tt*–89 there).

Accordingly, it will be seen that each one of these sequential sub-groupings of the 89 fonts *tt* coming up for registry with print positions 1 through 132 will have been registered as "hits" or "no-hits" in SRB for this scan. Further, it will be seen that according to the invention, register SRB has been sub-divided into three sub-registers SRI, SRII, SRIII, the print positions for each sub-register coming up for firing successively within each group and successively by groups (i.e. SRI group first, then SRII etc.). It will appear then that the fire order for print positions 1 through 132 has been divided into three sub-groups, namely positions 1, 4, 7 through 130 (see SRI), positions 2, 5, 8 through 131 (see SRII) and positions 3, 6, 9 through 132 (see SRIII). One advantage to such an arrangement is that the adapter register, SRB comprising SRI etc. can allow individual "hit-cells," C–1, etc. to be readily loaded and to be unloaded in this peculiar "by-three's" sequence. That is, with cells C–1 etc. being in "printing order," control means CAD will, in effect, skip from SRI to SRII to SRIII repeatedly over and over, filling SRB in regular sequence (C–1, C–2, C–3 . . .) but allocating signals to each sub-register SR in skipping sequence (C–1, C–4 etc. in SRI etc.) and filling the sub-registers SR at the same rate. For example, considering first sub-register SRI, one would first load cell C–1 (with a "hit" signal corresponding to print position #1), then cell C–4, then C–7 etc. through cell C–130. At the same time SRII will be filling beginning with C–2, then C–5 etc. through C–131. Furthermore, although the total time for loading the entire register SRB is much less than that for unloading it (800 μs., of 6 μs. per cell, vs. 1700 μs., or 11 μs. per cell plus delay intervals), the skipping "by-three's" organization makes the time for loading a particular sub-register longer than that for unloading it. Hence, for sub-register SRI, for instance, if unloading were to start a few microseconds after loading started, it would soon overtake the loading since successive cells are loaded every 18 μs., but unloaded every 11 μs. This cannot be permitted, of course; hence, loading must be given a prescribed "lead time."

Problem 1

This slow "fill time" for each sub-register gives rise to one major problem solved by the invention. As indicated in FIGURE 4A when sub-registers SRI and SRII have unloaded serially and sub-register III is only partly unloaded in a given print-scan sequence (first unload mode M_{U-1}) loading for the following print-scan sequence is made to begin early (second load made M_{L-2}), so that unloading of SRIII and, thereafter, of SRI (second unload mode M_{U-2} . . . these follow one another with practically no delay since scans are printed continuously) cannot overtake the loading of SRI and prevent continuous printing. Since printing according to the invention is to proceed continuously between scans, one cannot, as in the prior art, wait until one scan is printed before starting to load for the next scan. However, the control system is resynchronized with the chain between sub-scans, i.e. between the unloading of SRI and SRII, SRII and SRIII, SRIII and SRI. Thus, as a feature of the invention a "start cell" (C–S) is selected somewhat along sub-register SRIII such that the time for unloading the rest of SRIII and, thereafter, of SRI is sufficient to allow completing the loading of SRI. That is, C–S must be selected to leave a prescribed number of cells N–L yet to be unloaded (N–U already unloaded) so that the time to unload the remaining cells (N–L; the last in order being C–132) and, then, to unload the cells in SRI (that is N_{U-2}) is sufficient to allow loading all of the cells (M_{L-2}), in SRI (plus any delay intervals).

Problem 2

However, due to the peculiarity of the "latch type" storage means (cf. FIGURE 2) comprising the cells (C–1, etc.) of register SRB, each latching cell (circuit) must be "reset" before it can be reloaded, presenting a rather sticky "reset" problem, whose constraints are relatively opposite those of the above "slow load" problem. That is, it is very messy and expensive to reset the cells individually and not as a group. Thus, although the unloaded cells N–U at the time C–S is selected may all be conveniently reset together, at a first reset time, there is some problem with resetting the rest of the cells in register SRIII, namely cells N–L. Cells N–L should be reset together as a group since it is impractically complicated to reset individually as they are unloaded. To illustrate the problem, note that if cells N–L are unloaded and not immediately reset after unloading the last in this group (C–132) but only after some (unknown) delay which allows the loading process to reach the first few cells in Group N–L, then two problems arise: (1) those cells which the loading process reached could not be properly loaded since they have not first been reset and (2) if these cells had been properly loaded, the information loaded will all be erased by subsequent reset. Thus, it was determined that loading must begin late enough to assure that the unloading cells N–L will be reset together, before loading reaches them and that the "start" cell C–S must therefore be somewhat adjusted in location from the above so as to be far enough along register SRIII so that it is entirely unloaded by the time the filling process starts to load C–S (if not before).

Combining these two constraints, we now have a formula for selection of the range of times (or possible "start cells" in SRIII) for initiating the loading, that is, such times must be:

(1) Late enough (or cell C–S far enough along SRIII) so that SRIII is unloaded completely when the loading process reaches that selected cell C–S; and

(2) Soon enough (or C–S not so far along SRIII) that the unloading of SRI does not overtake the loading thereof, also taking unload-delays into consideration (e.g. delay for resynchronizing, though this is variable) as well as load-delays (such as CP being busy, etc.).

(3) Thus, for simplicity, it appeared convenient to initiate the loading process as soon as possible (especially because of possible delays, such as a busy CP, etc.) and thus, in the preferred case, to select a time (or "start cell" C–S) so located as to just allow SRIII to be unloaded when the loading reached that cell.

Thus, a second reset operation can be timed to conveniently, and safely, reset all remaining cells (N–L) together if it is initiated at the "loading C–S–1" time, locating C–S so that this coincides with "C–132 just unloaded" time. (That is, T_{U-NL} made equal to T_{L-CS} by selected location of C–S.)

It will be apparent to those skilled in the art that the principles of the present invention may be applied to different embodiments from that shown using analogous elements, etc. While in accordance with the provisions of the statutes, there have been illustrated and described the best forms of the invention known, it will be apparent to those skilled in the art that changes may be made in the elements disclosed without departing from the spirit of the invention as set forth in the appended claims and that, in some cases, certain features of the invention may be used to advantage without a corresponding use of other features.

Having now described the invention, what is claimed as new and for which it is desired to secure Letters Patent is:

1. In a data processing output arrangement for marking a record medium along prescribed lines and columns thereof, said arrangement including a reference cycling means; a plurality of marking units arrayed in a prescribed linear order across a prescribed marking plane,

each said unit comprising a marking means adapted to mark a respective column location on said medium and also on associated control means for controlling the energization thereof, the associated control means being adapted for successive enabling of said marking means in a prescribed skipping sequence, different from said linear order and so that every "Nth" marking means is enabled in N enabling sub-scans for each mark scan between each full incremental shift of said cycling means; also said arrangement including a prescribed memory means comprising a plurality of memory banks, each bank comprising a plurality of memory locations, each location having an encoded "mark command" stored therein for enabling a respective one of said marking means during one of said mark scans, said commands being encoded according to the condition of said cycling means during the prescribed mark scan, the combination therewith of an improved adapter control means adapted to interrogate said stored commands in said linear-order to control the energization of said associated marking means, said control means comprising:

code generating means adapted to monitor the cyclical condition of said cycling means and responsively generate a prescribed array of position-coded signals for each said incremental shift, said signals corresponding to said commands at a respective print scan; mark unit address means adapted to sequentially enable said control means responsive to the condition of said code generating means output in said skipping sequence; comparator means connected to receive said code generating means output; storage register means including a plurality of memory cells, each cell having its output coupled to a respective one of said control means for enabling thereof upon correspondence between a respective one of said position-encoded signals, and a respective command in an associated memory location; and register accessing means adapted to sequentially control the comparison of the contents of each memory location with a respective current one of said position coded signals at said comparator in said linear order and responsively inject hit signals in the corresponding one of said cells indicative of said correspondence; said accessing means also being arranged to order and time its operations upon said cells compatible with said skipping marking sequence.

2. The combination as recited in claim **1** wherein said cells comprise resettable storage circuits and wherein said register accessing means includes counter means adapted to successively access and control said cells in N sub-register groups for storage and resetting functions.

3. The combination as recited in claim **2** wherein said register accessing means is adapted to begin loading said cells with a given line set of said hit signals a prescribed lead time prior to conclusion of the prior mark scan and is also adapted to reset said cell circuits in sets at prescribed times so that these sets are related to said N sub-register groups.

4. The combination as recited in claim **3** wherein said marking unit array comprises an array of chain printer hammer means and associated control means; wherein said cycling means comprises a printer chain having a plurality of continually shifting type slugs thereon; and wherein said skipping marking sequence proceeds to unload said register cell circuits, faster, in a single sub-scan, then they can be loaded.

5. The combination as recited in claim **4** wherein said skipping comprises skipping every two hammer positions for enabling every third hammer means and wherein said accessing means accordingly operates said cells in three sub-register groups.

6. The combination as recited in claim **4** wherein said chain printer includes strobe means arranged to signal shifts of said chain and also hammer address means adapted to so skip-sequence the enabling of respective hammer means responsive to said strobe means signals; and wherein said hammer control means each include print hammer actuator means and associated AND gate means for controlling energization of said actuator means, each said gate means being coupled to said hammer address means and a respective one of said register cell circuits for energizing said associated actuator means upon coincident enabling output therefrom; and wherein said code generating means is adapted to generate linear-order code sets synchronous with said stroke means signals and indicative of respective type slugs past then registered with the respective print hammer positions.

7. The combination as recited in claim **6** wherein each of said memory banks comprises a portion of a central processor memory; wherein said hammer addressing means includes a prescribed counter means for enabling said AND gate means in said "by-three's" sequence to thus provide three skipping sub-sequences for each said full position shift of said chain.

8. The combination as recited in claim **7** wherein said register cells each comprise a resettable latch circuit; wherein said register accessing means includes counter means adapted to indicate a point in a subject skipping mark scan sequence for beginning the following load operation and for referencing reset time for said latch circuits.

9. The combination as recited in claim **8** wherein said register accessing means is adapted to begin a following load operation at a prescribed start cell and start time occurring when the time for finishing a subject skipping print scan sequence is approximately equal to the time necessary to load those of said cells in the initial sub-register group.

10. The combination as recited in claim **9** wherein said register accessing means is adapted to reset all said unloaded latch circuits up to said start cell at said start time, thereafter resetting the rest of said cell circuits upon compleion of said subject print scan sequence.

11. In a chain printer control arrangement for controlling a prescribed array of print impact means arrayed across a prescribed print-line segment in a prescribed linear order and arranged to be sequentially enabled in a different "skip every Nth impact means" sequence according to the coded print signals in a prescribed bank of line-printing memory locations corresponding in number and order to said linear order and including a character code generator providing a linear-ordered array of output code signals corresponding to the upcoming type slugs at respective print positions; an adapter arrangement in combination therewith for controlling said impact means responsive to correspondence between said print signals and said code signals, said arrangement comprising:

a storage register comprising a plurality of memory cells, each cell being coupled between said code generator means and a prescribed one of said impact means for enabling thereof with hit signals indicating said signal correspondence; register control means adapted to operate upon said memory cells according to N sub-register groups thereof in accordance with each sub-scan in said skipping impact operation; said control means being adapted to begin a following load operation at a prescribed start cell and start time accurring when the time for finishing a subject skipping print sequence is approximately

equal to the time necessary to load those of said cells in the initial sub-register group.

12. The combination as recited in claim 11 wherein said register cells each comprise a resettable latch memory circuit; wherein said N sequences and sub-register groups comprise three; and wherein said register control means is adapted to reset all of said cell circuits up to said start cell at said start time, thereafter resetting the rest of said cell circuits upon completion of said subject printing sequence.

References Cited

UNITED STATES PATENTS

3,066,601	12/1962	Eded	101—93
3,161,126	12/1964	Saltz	101—93
3,343,131	9/1967	Bloom et al.	340—172.5
3,377,622	4/1968	Burch et al.	340—172.5

PAUL J. HENON, *Primary Examiner.*

RAULFE B. ZACHE, *Assistant Examiner.*