B. T. Crutcher

D. E. Sederholm

The IBM SELECTRIC Composer

Letter Keyboard

Abstract: This paper describes the design of the letter keyboard for the IBM SELECTRIC Composer. The design requirements were to provide an information transfer mechanism not only for character selection using the SELECTRIC Typewriter printing system, but also for proportional escapement selection and print impact velocity control. The keyboard was required to be exceptionally reliable, to have the positive SELECTRIC Typewriter touch characteristics, and to be consistent with the keyboard layout standards that are familiar to the operator or typist.

The keyboard that has been developed has a completely serial operating sequence. Each event in a sequence is positively latched as it occurs, and remains latched until its function is completed. This insures reliable information transfer without compromising speed or input-signal storage capability.

Introduction

For a typewriter, and for the SELECTRIC Composer, the keyboard is defined as the mechanical means by which an operator transfers information to the machine. Operator *control* can be applied in other ways in certain cases, but operator *information* (what is to be printed and the order in which characters and spaces are to appear) must always pass, at some time, through the keyboard, even if it is temporarily stored.

In a typewriter the primary purpose of the keyboard is to transmit the operator's character selection to the printing mechanism for immediate output. In the Composer, however, the keyboard must perform two other control functions—proportional escapement selection and print-impact velocity selection. These are not directly influenced by the operator. The escapement mechanism[1] and the velocity control mechanism[2] are set by the operator to correspond to the type font being used, but the keyboard itself is not affected by typefont changes. Instead the number of escapement units and the velocity level for each character (and in the case of escapement, each space as well) are pre-coded inside the keyboard, and the latter, in effect, "reads" the operator's character selection, "interprets" it, and itself selects the velocity level and escapement length in units that will be produced. It is thus required to deliver three classes of information to the operating mechanisms but receives only one at its input.

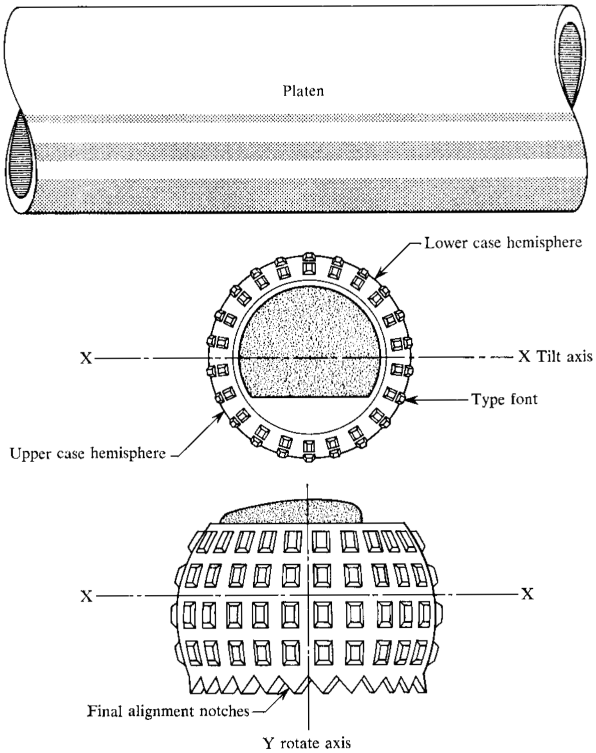
With minor exceptions, the arrangement of typewriter keyboard characters has not changed since 1868. This standardization has led to a "touch" system for the operator that produces a rapid input rate and permits different machines to be used interchangeably. The Composer keyboard retains that arrangement, and changes to accommodate the new functions have been restricted primarily to the internal portion of the mechanism.

The design objectives, therefore, were to develop a keyboard that would respond to a one-mode input signal by transmitting three channels of information to the output mechanisms. High reliability was required for all three channels, the familiar external format was to be maintained, and the touch characteristics (mechanical feedback to the operator, discussed further in later sections) were to be equivalent to or better than those of a high-quality typewriter.

Design requirements

• *General*

In order to insure high reliability, the keyboard is required to produce a uniquely defined output from each impulse above a certain minimum. Inputs below this minimum, or trip threshold, cannot be accepted nor be permitted to interfere with other signals.



(a)

Upper case

Tilt position code	Pay						Pull				
	R-5	R-5-1	R-5-2	R-5-1-2	R-5-2-2A	R-0 (Home)	R-1	R-2	R-1-2	R-2-2A	R-1-2-2A
T-0	M * 6	M † 6	M (4	M) 4	M ! 4	H S 6	H $ 6	M + 6	M / 4	L ‘ 3	L ’ 3
T-1	H % 8	H & 8	H B 7	H T 7	H A 8	M I 4	H L 7	H C 7	H ½ 8	L — 8	H ¾ 8
T-2	H K 8	H W 9	H D 8	H O 8	H R 8	H N 8	H H 8	H M 9	H U 8	H V 8	H X 8
T-3	M J 5	L : 4	H Q 8	H G 8	H F 7	H E 7	H Y 8	H P 6	H Z 7	H ¼ 8	H @ 8

Lower case

Tilt position code	Pay						Pull				
	R-5	R-5-1	R-5-2	R-5-1-2	R-5-2-2A	R-0 (Home)	R-1	R-2	R-1-2	R-2-2A	R-1-2-2A
T-0	H 8 6	H 2 6	H 9 6	H 0 6	M 1 6	M s 4	H 4 6	H 3 6	H 6 6	L , 3	L . 3
T-1	H 5 6	H 7 6	H b 6	M t 4	M a 5	M i 3	M l 3	M c 5	M [5	L - 3	M ? 5
T-2	M k 6	H w 8	H d 6	M o 6	M r 4	M n 6	M h 6	H m 9	M u 6	M v 5	M x 6
T-3	M j 3	L ; 3	M q 6	H g 5	M f 4	M e 5	M y 6	M p 6	M z 5	M] 6	M = 6

Rotate position code

(b)

Figure 1 (a) Sketch of type element; (b) character location matrix.

The reliability requirement adds other constraints; if two or more inputs can be entered completely simultaneously, all should be rejected, and if sequential inputs can occur at a rate faster than the output rate (speed of the printing mechanism), there must be some provision for temporary storage in the correct sequence. Finally no signal, once accepted, can be allowed to escape (in the conventional sense of "getting away") until accepted by the printing mechanism; this implies a means for latching each mechanical event along the information path, with release permitted only after the last device requiring the signal has used the information.

Timing is another factor. The delay of the signal in traversing the keyboard must be small enough to escape the operator's notice; longer delays would affect typing rhythm and lead to both errors and slower output.

The last general consideration is the dynamic path of all moving parts within the keyboard. It was recognized that multiple, variable signals would cause considerable variation in loading conditions for certain parts, and that stability conditions and displacement paths would vary depending on load. It was therefore necessary to consider more than the usual "strength vs. lightness" analysis. Each moving part has been analyzed under all conditions at every position in the operating path, and not just at the extremes or "worst cases," in order to guard against errors and failures due to rebound, idling, erratic information rate, etc.

• *Print (character) selection*

The printing mechanism and character selection scheme are very similar to those of the IBM SELECTRIC Typewriter. The Composer design was required only to employ the same general approach; this will be described briefly in the following paragraphs. It must be remembered, however, that in the Composer this function is only one of the three controlled by the keyboard, and the six-"bit" information channel that controls the printing element had to operate compatibly with two other channels.

The SELECTRIC Composer print element is a spherical typehead containing four rows of characters with eleven columns per row in each hemisphere as shown in Fig. 1. Character column selection is accomplished by rotating the sphere about its vertical axis to select one of the eleven columns in a hemisphere. As shown in Fig. 1b, rotation is in either direction about the "home" (or zero) position. One hemisphere contains the lower case font; the other the upper case. The familiar shift operation at the keyboard simply rotates the typehead one-half revolution prior to character selection. Row selection is done by tilting the sphere about its transverse axis (parallel to the paper), but tilting is done in one direction only with the uppermost row as "home."

As mentioned above, the channel that controls tilting and rotation requires six "bits" of information, exclusive of the shift operation (which is a one-key, separate function and not part of the three-channel information scheme). Four bits are used to select one of eleven "rotate" positions—one bit, if selected, causes a one-column increment of positive (clockwise as viewed from above) rotation. Two other bits provide two increments each, also positive, and the fourth bit provides a five-column negative increment. Of the sixteen possible four-bit combinations, eleven are used as shown in Fig. 1b.

The two additional bits are used for row selection. The first bit causes one unit of tilt motion; the second causes two. Combining these with the home (no signal) position creates four combinations that can be used to select one of four rows.

As a keyboard design specification, each of the six bits required a pull motion of 0.058 in., to be accomplished within seven milliseconds after the start of a print cycle. (This requirement is similar to that of the IBM SELECTRIC Typewriter since the same selection mechanism is used.)

• *Escapement selection*

Each character used in the Composer is assigned a proportional width value, or escapement, of from three to nine units.[1] (The width per unit is also variable, permitting the use of multiple character sets, but this is not a keyboard function and is discussed instead in Ref. 1.) The escapement requirement was reflected in the keyboard as a need for six information "bits": The channel was to contain a separate signal path for each escapement value from three through eight units; a nine-unit escapement would be indicated by the absence of signal in all six paths. Thus, in this information channel, there had to be six available "bits" but only one (or none) would be active during a given character selection.

There was also a need to change the escapement value assigned to certain characters whenever the shift from lower to upper case was active (e.g., the upper case L requires seven units of escapement while the lower case l requires three). For characters affected in this way, some provision was necessary to change the escapement "bit" assignment in response to the action of the shift mechanism.

The output specifications were a 0.058 in. pull motion for each selected bit in the channel to occur within seven milliseconds, as before.

• *Velocity selection*

In order to achieve the desired print quality, it was decided (among other things) to allow the print element to strike the paper at more than one impact velocity. The character set was divided according to character size into three groups, and each group was assigned a different velocity.[2] For each character selection, therefore, the keyboard had to provide a signal that would correspond to the selected character's group assignment, and a third information channel was required. Only two "bits" were needed, however, one for each of two velocity levels. The absence of signal would select a third level.

Again because of size variations between upper and lower case, the shift function has an effect on velocity of the same kind as that on escapement, and again the requirement for shift-activated bit reassignment was imposed. Output specifications were also the same.

• *Special requirements: variable spacebar and no-print*

As in any typing system, the Composer spacebar was required to cause a machine escapement but suppress the printing action. Unlike a typewriter, however, the Composer needed a space escapement that would be variable and operator selectable. This was not so much because of character proportionality (although this did play a part), but because some method of line expansion was needed for justification.[3]

There is another function, called the no-print operation, that was also a part of the keyboard design. In effect it is similar to the spacebar function, in that escapement is variable and printing is suppressed. In this case, however, operator selection is indirect: Occasionally (as in "centering," error correction or certain cases of justification), it is desirable to cycle the machine as if a character were being typed but without actually printing the character. If the spacebar were used, the operator would need to know the proper escapement width for the selected character; in view of the large number of typefonts and the number of characters per font this would be impractical. If, however, a "no-print" key function could be designed to use the spacebar suppression signal but the character-key escapement signal, this difficulty could be avoided. The design specification, as finally established, was for a no-print key that would operate in conjunction with a character key and do only one task: print suppression using the spacebar's suppression signal path. With this key depressed, a character key input would cause the pre-coded escapement, and its character-print signal would be transmitted but negated.

A third special function is opposite in effect—printing is allowed and escapement is suppressed. This operation, called "dead key," was included to permit the printing of built-up characters by using two print cycles to print from two different places on the typehead but in the same position on the paper (for example, an accented character). Although in part a keyboard function, this is discussed elsewhere[4,5] and will not be covered here.

Choice of design approach

The design requirements above imply a unique set of characteristics. Some members of the set may be found in other keyboards and all of them, probably, exist individually or in partial combination. But the set as a whole required either a new design or, perhaps, extensive modification of some existing keyboard.

One major decision to be made was the choice between parallel and serial approaches. The parameters involved were speed, which probably favors a parallel approach, reliability, which favors serial operation, and complexity, which does not certainly favor either.

A parallel operation in a keyboard implies that two or more events* must occur simultaneously. This can increase

* The term "event" is used here to describe a single element of the sequence required to transmit a signal from the keyboard (input) to the mechanism (output) whose response is required.

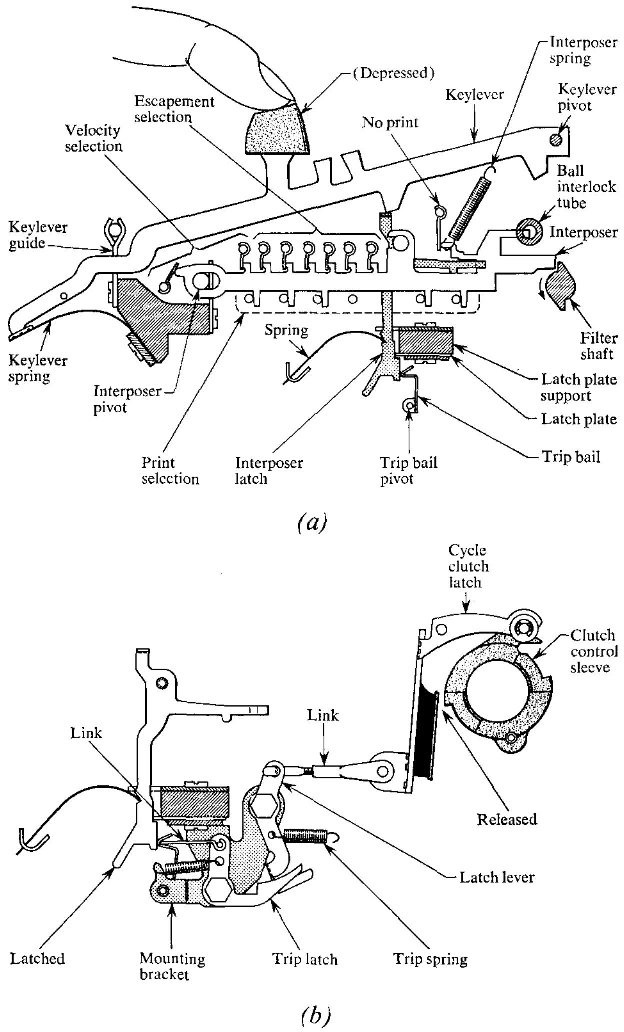


Figure 2 Section view of (a) Composer keyboard and (b) interposer latch assembly.

over-all speed, but if an accidental time delay exists between two events (we are choosing the simplest example), the initial event may return to its starting condition before the information can be used by the printing (or other final) device, which is waiting for *all* events to occur. If, on the other hand, the initial event does *not* return to its starting condition soon enough, erroneous information may be transmitted (as, for example, when the late-terminating event overlaps a later event or another cycle). Failure results in either case.

In a serial keyboard, events are always in an ordered sequence. Each event of a cycle is mechanically latched in place, and the latching action triggers the next event. No event is unlatched until the last device has used the required information. This approach is intrinsically more reliable, but at some investment of operating time and possibly greater complexity.

The decision was to use a serial keyboard. The deciding factors were that, first, reliability was a cardinal requirement because of the quality and accuracy implied by the term "composition," and second, the SELECTRIC Typewriter keyboard was a proven mechanism that already satisfied the character selection criteria and was extremely reliable. It would of course have to be modified to achieve wholly serial characteristics and accommodate the new Composer functions, but it was concluded that this would be possible. The interposers* in the SELECTRIC Typewriter keyboard contained six code stations but could be redesigned to contain the necessary fifteen (six for character selection, six for escapement, two for velocity selection and one for no-print). The serial requirement could be met by the addition of a latch that would be activated by the keylever, trip the power mechanism to begin a print cycle, latch the interposer until power-induced motion has occurred, and then release the interposer and allow it to transmit its signal. This would guarantee retention of the signal information until the cycle has advanced sufficiently to use it; it would also insure that, at the proper time, the interposer would be restored to its initial condition. Careful design would assure the necessary operating speed.

Based on this approach, the serial keyboard was designed and tested. The outcome, i.e., the present Composer keyboard, will be described below, after which the analytical work will be reviewed with emphasis on reliability and the impact of the new Composer functions.

Mechanical description

A section view of the keyboard is shown in Fig. 2. The sequence of events begins with the depression of a keylever. After a small (0.010 inch) clearance is taken up between the keylever projection and the interposer latch, the keylever displacement is transferred to the interposer by the vertical movement of the latch. The input information is then held in place by the latch, which locks against a latch plate to secure the vertical displacement of the interposer. The counterclockwise rotation of the interposer latch, which produces the locking action, causes clockwise rotation of the trip bail, thereby releasing the trip latch lever (Fig. 2b) and tripping the cycle clutch. The cycle clutch latch releases the cycle clutch sleeve, causing the cycle clutch of the printing device to engage and produce rotation of the two-fluted keyboard drive shaft—the filter shaft. The counterclockwise rotation of the filter shaft produces horizontal translation of the interposer-latch assembly and the resulting output information from the keyboard is transferred to the

* The interposer is part of the mechanical linkage between a keylever and the output mechanisms. There is one interposer for each keyboard character; each contains projections whose position corresponds to the information code for the associated character. Each projection represents a "bit" of the code and, through its motion, the interposer divides the single signal received from the keylever into its component parts and sends those parts along separate paths. Although mechanical it is, in effect, the input logic device.

printing device. The cycle clutch latch is restored by a cam.

The areas labeled "escapement selection" and "velocity selection" are a section view of a group of rods, called bails, which span the keyboard and have coded projections extending from them. The horizontal translation of the interposer produces a rotation of these bails. The connection from each bail to the velocity and escapement mechanisms is by pull cables.

The area labeled "print selection" is a section view of another group of bails. These, however, do not rotate about their centers. The pivot axis is located 0.500 inch above the bails, and the motion of each bail section is approximately linear translation. Proper coding is achieved by adding or removing the projections on the interposer. A rod link forms the connection to the print selection mechanism.

• *Timing*

Figure 3 is a timing diagram that illustrates the beginning of a typical print cycle. The input time required to depress a keylever and interposer will vary, depending on the typing speed of the operator. The clutch "pick-up" time will vary from 3 to 12 milliseconds, depending on the relationship of the drive pawl to a pulley tooth when the clutch is released. The velocity selection, character selection, and escapement signals all occur within 7 milliseconds from the starting time. The clutch trip bail will rotate about its pivot and release the cycle clutch latch in 0.005 seconds. The trip bail is designed for low mass-moment of inertia while retaining maximum torsional stiffness.

• *Shift operations*

The escapement selection bails and the velocity selection bails are displaced axially by the shift mechanism to change the code for the upper case. The code bails are spring-urged in one direction and driven by the shift mechanism in the other direction. Each bail contains code tabs for each condition—upper and lower case; the removal of the tab from alignment with an interposer projection removes the code of that "bit" from the associated character's channel.

• *Interlock*

The ball interlock tube (see Fig. 2) is a device to eliminate the simultaneous entrance of two or more input signals. The device consists of a tube containing one ball more than there are interposers. The balls are adjusted for clearance to permit the entrance of only one interposer at a time. A horizontal displacement of 0.040 inch removes the first interposer and permits the entrance of a second.

• *Spacebar*

Figure 4 shows the variable spacebar selection system and its relationship to the escapement selection system. The spacebar keylever is made to operate an interposer, similar to a letter interposer. This will provide the same key de-

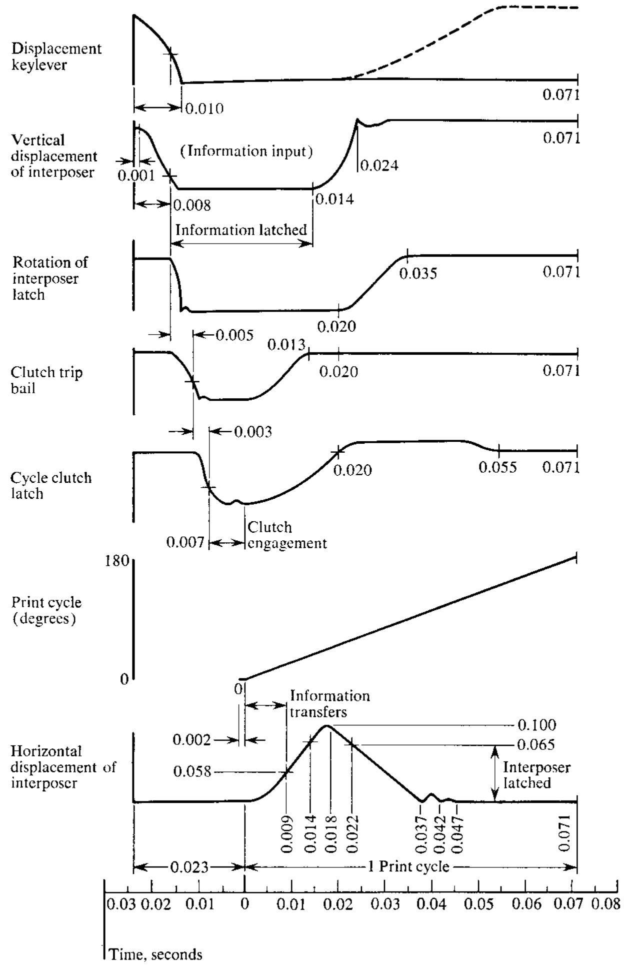


Figure 3 Keyboard timing diagram.

pression interlock feature as a letter-key. It provides the necessary motion for an escapement selection signal and its character selection signal is coded to select home position on the print element. An additional signal taken from the interposer is used to provide the no-print function (see Fig. 2 for location of the no-print bail). The machine will perform all the normal functions of a print cycle except that the print element does not strike the paper. The no-print signal moves the print cam follower to a low rise cam which suppresses the print impact. The escapement signal is generated by the spacebar interposer and this motion is transferred to the code bails through a drive key and pull links. The escapement value is made variable by adjusting the drive key to engage the pull links one at a time. Value selection is made at the dial through a pinion and rack that slide the drive key laterally back and forth. The interposer has essentially the same flight path as a character interposer and

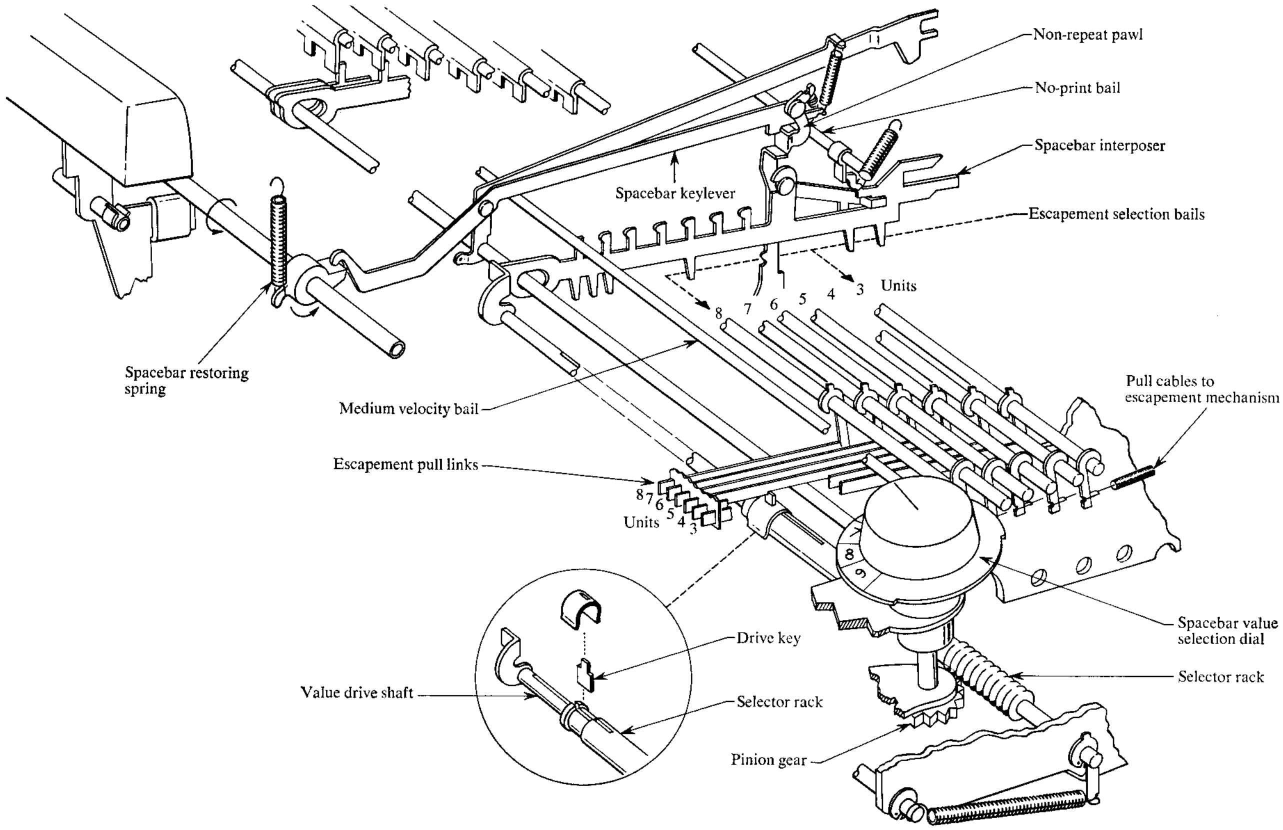


Figure 4 Selection mechanism for variable spacebar.

a timing relationship similar to that shown in Fig. 3. The keylever has a movable pawl which is used to trip a single cycle, and a fixed projection is used to hold the interposer depressed so it will recycle the machine. The spacebar can then be held in repeat mode and at any escapement value.

Analytical design

Although the chosen approach was apparently feasible and not difficult to visualize, there were obviously a number of conditions imposed by the new elements that would have to be analyzed before the design could be called successful or reliable. The exact relationship, for example, of the latch and tripping mechanism to the cycle clutch would need to be known. The addition of new bails for escapement and velocity selection, no-print, etc., could radically change the loading conditions of the interposer and affect its flight path; at the very least the number of variations in loading would considerably increase. The latching action and the geometry of the latch itself had to be so designed that there would be neither stalling nor clashing as the latch slides beneath the latch plate (and later out again). The interposer assembly would have to be reversible; that is, it would have to be able to return to its home position from a point short of latching if its cycle is begun but blocked from completion by the interlock. And finally, the touch characteristic (the feedback to the operator of the loads developed as the cycle completes or as the interlock is encountered) had to be consistent with the original specification to assure good operator efficiency.

The five examples just listed are developed in detail below. The reader is asked to refer to Fig. 2 for illustration of mechanical elements.

- *Clutch release and interposer unlatching*

The free body diagrams for the release of the cycle clutch are shown in Figure 5. Force F_1 (Fig. 5) is applied from the clutch mechanism to the cycle clutch latch (part 1). F_2 is the force required from the trip spring to release the clutch. Force F_3 (Fig. 5b) is the spring return force for the latch bellcrank (part 4). F_4 is the applied force from the latch lever (part 3) to the latch bellcrank (part 4). F_5 is the release force for the entire mechanism. The parameter requirements for unlatching are found as follows:

From Fig. 5a, Σ Force = equivalent mass $\times$ acceleration,

$$\Sigma F = k_2(S_0 - S) - \mu F_1\left(\frac{c}{b}\right),$$

$$M_{\text{eq.}} \times \text{acceleration} = \left(M_2 + \frac{I_1}{b^2} + \frac{I_3}{a^2}\right)\frac{d^2S}{dt^2},$$

where k_2 is the spring rate of the spring producing force F_2, S_0 and S are the initial and final displacements of the top of the latch lever, M_2 is the mass of part 2, μ is the coefficient of friction between the clutch sleeve and the cycle clutch latch, I_1 is the moment of inertia of part 1, and I_3 is the moment of inertia of part 3.

The solution of the equation is:

$$S = \frac{k_2S_0 - \mu(c/b)F_1}{k_2} \times \left(1 - \cos\left[k_2\Big/\left(M_2 + \frac{I_1}{b^2} + \frac{I_3}{b^2}\right)\right]^{\frac{1}{2}} t\right).$$

k_2S_0 is found knowing the desired S (0.150 in.) and t (0.003 sec.).

$$F_4 = F_2\frac{a}{d}.$$

The required unlatching force F_5 is evaluated from the relationship (Fig. 5b):

Σ Force = equivalent mass $\times$ acceleration

$$\left(F_5 - F_3\frac{f}{g} - \mu F_2\frac{ah}{dg}\right) = \left(M_5 + \frac{I_4}{g^2} + \frac{I_6}{e^2}\right)\frac{d^2x}{dt^2},$$

where x is the displacement of the trip bail link (part 5). Applying the boundary conditions when $t_0 = 0$, $dx/dt = 0$, $x_0 = 0$ gives the solution:

$$x = \frac{[F_5 - F_3(f/g) - \mu F_2(ah/dg)]}{2[M_5 + (I_4/g^2) + (I_6/e^2)]}(t^2).$$

By knowing the desired x (0.040 in.) and t (0.005 sec.), the unlatching force can be found. The value F_5 is assumed to be a constant since the bowed interposer latch spring supplies this force. This assumption was verified experimentally.

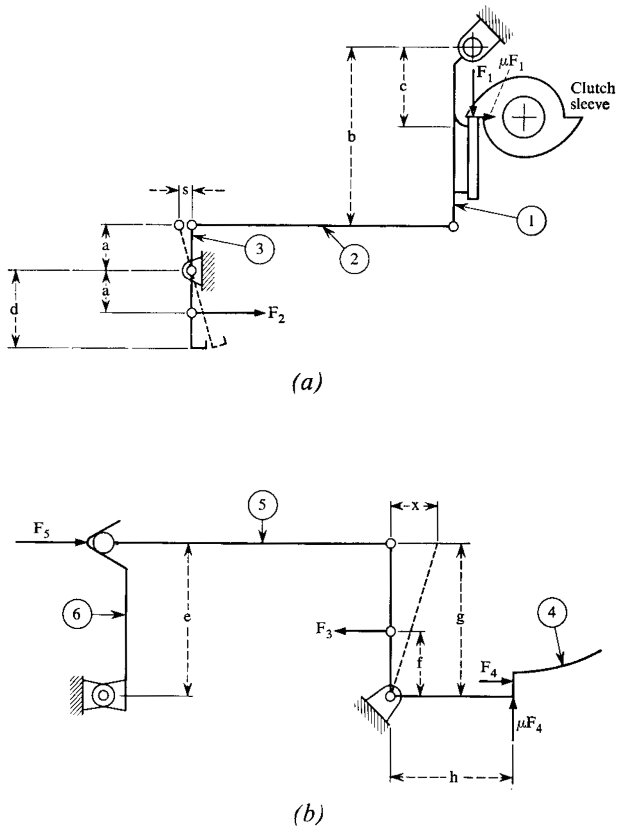


Figure 5 Free body diagrams of clutch release mechanism: (a) cycle clutch latch; (b) trip bail and latch.

- *Interposer flight path*

The letter interposers must impart sufficient horizontal motion to the selector bails to produce print selection, velocity selection, and escapement signals at the respective mechanisms. Also, they must provide a critical segment of time in each print cycle to lock the keyboard against any other letter-key input. These events must occur very early in the print cycle to allow time for subsequent preconditioning of the selection mechanisms prior to actual printing. To accomplish these functions, each interposer must follow a prescribed flight path and stay within the given limits. An examination of the keyboard geometry and six critical areas on the interposer will demonstrate the forces involved and how they interact to produce this flight path.

The interposer horizontal motion is produced by rotary action of the filter shaft flute (Fig. 2); this motion must then be transferred to any combination of six print selection bails; any one of six escapement selection bails; and to either one of two velocity bails or to a no-print bail. This horizontal drive motion depends on the amount of engagement of the interposer tip with the filter shaft and the amount of friction load at point of contact. The friction load will vary, depending on the number of bails to be moved in making a selection. The combined ranges of vertical and horizontal interposer spring loads must be carefully selected and be coupled with a wide range of bail-load forces to produce an acceptable flight path. For example, heavy bail load will cause a high contact load and the interposer tip will "nose dive" to the downstop, then bounce away. If there is no bail load, the contact load will be reduced to the sum of horizontal spring components and the

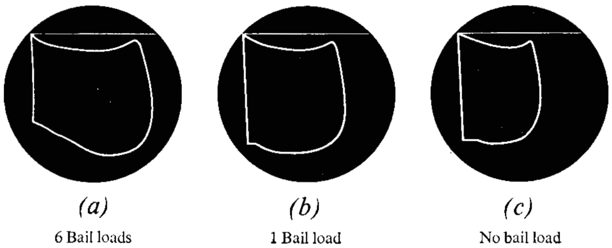
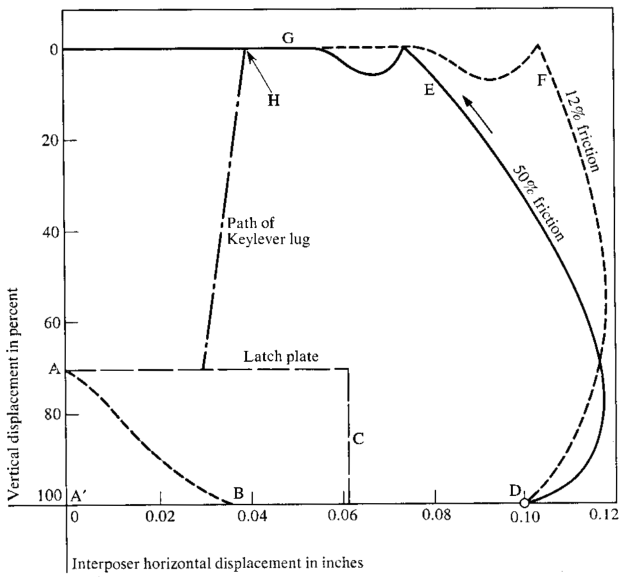


Figure 6 Oscilloscope traces of interposer flight path (a) with 6 bail loads; (b) with one bail load; (c) with no bail load.

Figure 7 Line sketch of interposer flight path.



stronger vertical spring components will tend to strip the interposer up and away from the drive shaft too soon. To prevent a malselection due to insufficient horizontal bail motion, the interposer must be held engaged to the drive shaft long enough to disengage the latch pawl from the latch plate. The elevation of the latch plate can be adjusted to provide this motion. (This was later demonstrated by testing as necessary.)

Figure 6 shows oscilloscope illustrations of the flight path under three different conditions. Figure 6a shows the path of an interposer that is influenced by six letter-bail loads. (The velocity and escapement bail loads are extremely light; they have no appreciable effect on the flight path and will not be discussed further.) The part of the trace at the extreme left is the path taken by the interposer as it follows the vertical motion of the keylever. The trip point, at the lower left corner, is the point at which the filter shaft engages and begins to drive the interposer horizontally. In this case, horizontal motion begins before the limit of vertical travel (the keylever downstop) is reached. The interposer latch is clear of the latch plate before it reaches the limit of horizontal travel, but the filter shaft remains engaged and horizontal displacement continues until this influence ends (lower right corner).

Figure 6b shows the interposer flight path with the single (no-print) bail load on the interposer projection adjacent to the filter shaft as shown in Figure 4. The horizontal drive travel starts after the interposer is vertically driven to the release point. The interposer is then driven to its downstop by the drive shaft before it enters free flight.

Figure 6c shows the flight path with no bail load. In this case, the horizontal drive starts at keylever bottom. The contact friction load is balanced by the vertical spring loads and the resulting motion is horizontal (Compare the total horizontal drive shown in Fig. 6a for six bails with that of Fig. 6c for none.)

A schematic of the dynamic path is shown in Fig. 7. The vertical travel of the interposer from 0 to the trip point A, the same as in Fig. 6, is from the vertical motion of the keylever.

The keylever may continue the vertical displacement of the interposer to point A′ before horizontal displacement from the filter shaft. If the interposer is located at point A, the frictional load between the interposer and filter shaft produces path AB. The vertical coupling between the keylever and the interposer is disengaged after 0.04 in. of horizontal displacement of the interposer. The interposer is driven horizontally by the filter shaft from A to D. The interposer at point C is free of the latch plate and leaves the filter shaft at point D with an initial horizontal velocity of seven inches per second. The horizontal components of the interposer spring and the interposer latch spring govern the horizontal displacement at this point, while the vertical components of these two springs govern the path of the vertical restore mode. This is illustrated in the free body diagram (Fig. 8), where:

k_1 = horizontal spring rate of interposer spring,
k_2 = vertical spring rate of interposer spring,
c_1 = horizontal value of latch spring force,
c_2 = vertical value of latch spring force,
F_f = assumed friction, % of applied spring force, and
V_0 = initial horizontal velocity of the interposer and interposer latch.

The result is the plot shown in Fig. 7. Path F assumes a frictional force of 12% of the applied spring force, while path E assumes a frictional force of 50% of the applied spring force. The force of friction is a result of the dynamic load between the interposer code lugs and the output bails.

The return path G of the interposer to its origin should intersect the path of the keylever lug (point H) when the interposer is moving horizontally. This prevents further vertical motion of the interposer until the keylever is reset to the starting position.

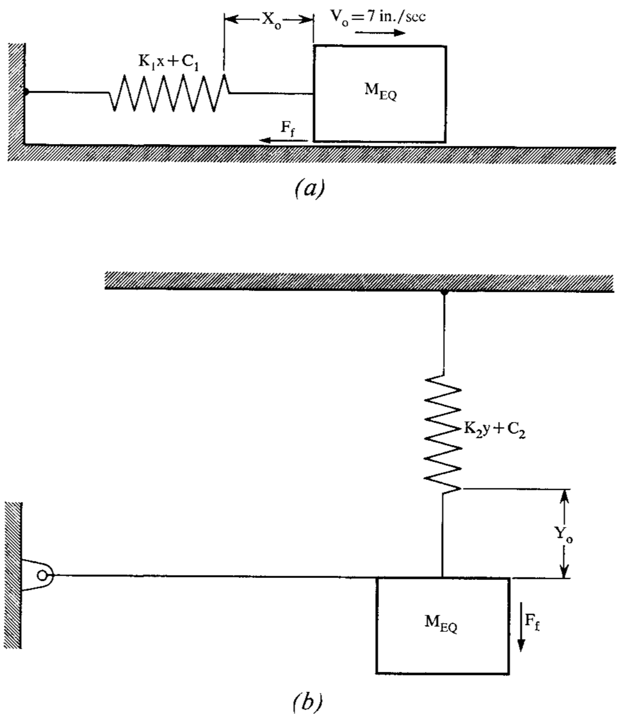


Figure 8 Free body diagrams of interposer: (a) horizontal; (b) vertical.

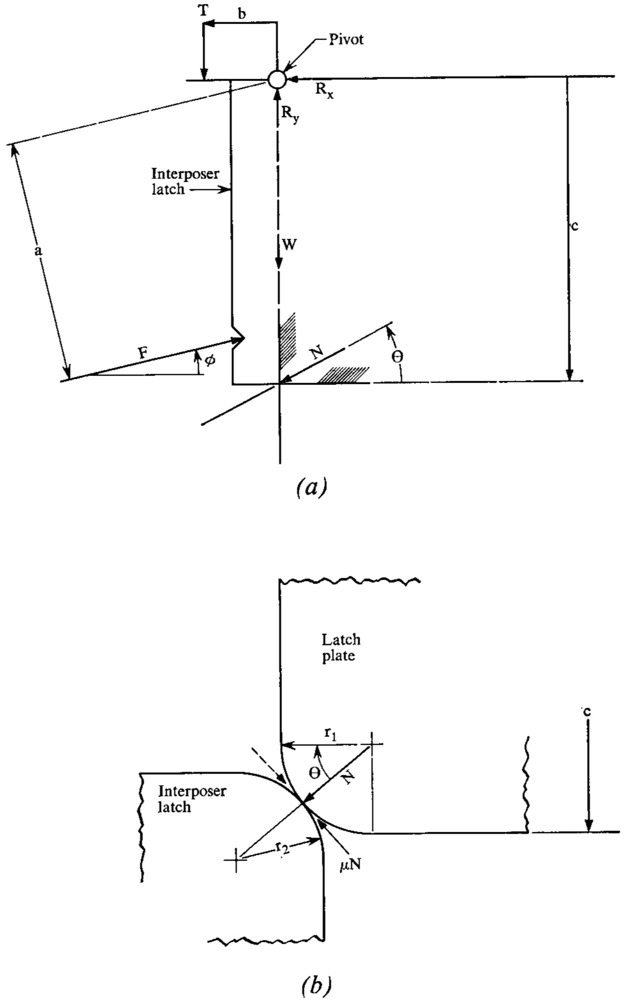


Figure 9 Illustration of interposer stall condition: (a) free body diagram of interposer latch; (b) mating surfaces of latch and latch plate.

- *Latch pawl geometry*

The interposer latch may experience a stall condition (illustrated in Fig. 9). A stall condition is defined as a total of vertically applied forces that is insufficient to lift the assembly, in conjunction with a total of horizontally applied forces that is insufficient to force the interposer latch surface under the latch plate. The static balance of forces illustrates this condition as:

$$\Sigma F_y = R_y - T - W + F \sin \phi - \mu N \cos \theta ,$$

$$\Sigma F_x = R_x - F \cos \phi - \mu N \sin \theta ,$$

$$\Sigma M_0 = Fa + Tb - Nc (\cos \theta + \mu \sin \theta) ,$$

where

R_y = vertical reaction at pivot,
R_x = horizontal reaction at pivot,
T = applied load from keylever,
F = applied load from interposer latch spring,
W = weight of interposer latch,
ϕ = angle of action from horizontal of force F,
θ = angle of reaction of force N,
N = reaction force on latch plate,
$a,b,c,$ = linear distance,
r_1,r_2 = curvature radii of the contacting surfaces, and
μ = coefficient of friction at the point of contact.

The angle θ can assume any value from 0° to 90°, depending on the point of contact of the two parts under consideration (see Fig. 9b). The point of contact varies the angle of the reaction force N. An undefined condition may therefore result.

To reduce the stall condition the corner radii were made to approach zero. The reason for this is that the angle θ changes rapidly with a minute motion of the interposer latch; to eliminate the stall zone the radii must approach zero, so that angle θ becomes undefined.

- *Interposer reversibility*

Reversibility is defined as the ability of the interposer assembly to return to the home position in the event of partial depression of the keylever. Reversibility is important since

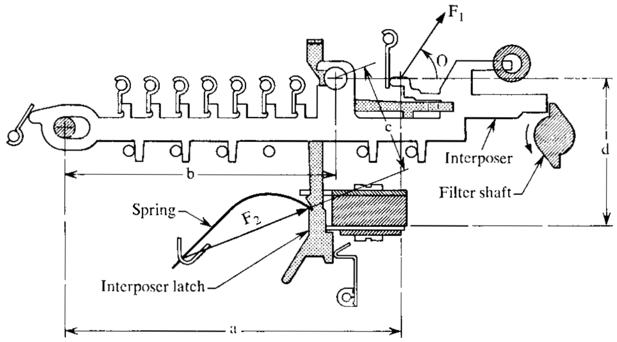


Figure 10 Reversibility diagram.

two interposers may overlap the filter shaft during the driving portion of the cycle. If this overlap is prevented by an interlocking device, the irreversibility of an interposer will lock the keyboard. With reference to Fig. 10, the reversibility is described mathematically as:

$$R = \frac{(F_1\ a \cos \theta)}{b} - \mu F_2(c/d)\ ,$$

where

F_1, F_2 = spring forces,
a,b,c,d = distance, and
μ = coefficient of friction.

The reversibility R must be positive to assure the return to home of the interposer assembly. The force F_2 is determined from the boundary conditions of the interposer flight path; F_1 is determined from the touch characteristics.

By assuming μ, a, c, and d to be constants, increased reversibility can be achieved by decreasing the distance b; however, the lower limit is the amount of vertical displacement of the interposer latch that can be allowed without critical adjustment of the latch plate.

• *Keyboard touch*

The efficiency and accuracy of typing are largely attributed to the keybutton load which is felt as the keylever is depressed. The buildup of button load to a cutoff point is also a contributing factor. An acceptable touch pattern has existed for some time in the IBM SELECTRIC Typewriter keyboard, and this has set the precedent for design criteria in the Composer keyboard:

Figure 11 consists of two superimposed traces, representing two successive key strokes. One trace is the key stroke with the normal operational loads; the second shows the additional load increase as the interposer encounters the ball interlock. The diagram represents the load increase of a second-row keybutton as the keylever is depressed to the keylever stop point. The load "touch pattern" develops in steps as three spring loads are encountered in succession. The first load is the initial tension of the keylever spring; the second load is the vertical component of the latch spring; and the third load is the interposer return spring. Reduction in keybutton load is due to the release of input information to the machine. The final load occurs when the keylever contacts the keylever stop. The ball interlock load is a result of displacement of the interlocking balls as the interposer enters the ball interlock tube. The extra load will vary in size, depending on how many balls must be pushed aside before the interposer can enter.

• *Validation*

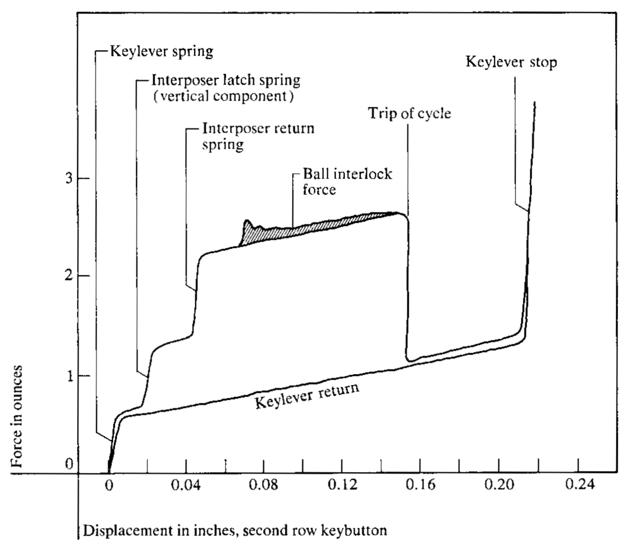
The considerations above, and others of similar nature, were applied to a prototype keyboard and verified experimentally. It was found that the design approach, properly executed, would indeed produce a keyboard having the desired reliability and speed and incorporating the necessary new functions. Timing of most of the major components, e.g., interposer, cycle clutch, trip bail and latch, and keylever, was verified by engineering instrumentation as conforming to the original timing objectives (Fig. 3). Reliability and life testing indicated acceptable failure characteristics, wear conditions, and in-use degradation.

Summary

The objective of the Composer keyboard—to provide information to the printing device to print correct characters, to print with the required level of velocity, and to escape the required displacement after printing—was achieved.

The design was accomplished by modifying the IBM SELECTRIC Typewriter keyboard and rearranging the location of the output coding. Reliability was increased by mechanically latching the information interposer in place

Figure 11 Keylever touch characteristic.



to prevent erroneous output signals from the keyboard. The functions of escapement selection and velocity selection were accommodated by increasing the number of output code signals transmitted by the interposer. The standard external format was maintained with only minor exceptions (and these exceptions, such as the no-print key, were additions only; standard keys were not changed.*) The Composer keyboard requires no increase in space over that needed for the IBM SELECTRIC Typewriter keyboard, except that the pull links of the escapement selection extend somewhat beyond the internal side plate.

Acknowledgments

The Composer keyboard is a composite of several keyboard design efforts contributed by many designers and engineers; however the authors are particularly grateful to R. D. Dodge for his experience and leadership in guiding the development of the serial keyboard, the major portion of which has been used in the IBM SELECTRIC Composer.

* Interlingual compatibility requirements did create external keyboard changes for certain specific cases. These occur, however, only with changes in the language for which a specific machine is to be used. A discussion of this subject appears in Ref. 4.

References

1. B. W. Miles and C. C. Wilson, "IBM SELECTRIC Composer Escapement Mechanism," *IBM Journal* **12,** 48 (this issue).
2. G. T. Slaughter, "IBM SELECTRIC Composer Mechanical Print Quality," *IBM Journal* **12,** 34 (this issue).
3. J. S. Morgan and J. R. Norwood, "IBM SELECTRIC Composer Justification Mechanism," *IBM Journal* **12,** 68 (this issue).
4. H. Pijlman, "IBM SELECTRIC Composer Type Font Compatibility," *IBM Journal* **12,** 26 (this issue).
5. G. E. Siemer, "IBM SELECTRIC Composer Dead Key and Dead Key Disconnect Functions," *IBM Journal* **12,** 32 (this issue).

Received April 29, 1967.