The Bell System Technical Journal
Vol. 52, No. 6, July–August, 1973
Printed in U.S.A.

Switching Networks of Planar Shifting Arrays

By R. S. KRUPP and L. A. TOMKO

(Manuscript received June 2, 1972)

An array of shift registers that may operate in two orthogonal directions can be called a planar shifting array. This article shows how two basic building blocks, fashioned from planar shifting arrays, may be interconnected to form a time-division switching network of arbitrary size. The characteristics of magnetic bubble and charge-coupled devices are compatible with the concept of planar arrays, and it is in these emerging technologies that switching networks of planar shifting arrays may become practical.

I. INTRODUCTION

Switching machines in the Bell System have grown in both number and capacity to meet the growing traffic demand. Early machines consisted of a small amount of distributed logic embodied in electromechanical devices but, as technology has permitted, the machines have evolved into largely solid state systems with central processor control. The environment in which switching machines must operate has also changed from a relatively small collection of analog voice grade circuits to an overwhelming number of circuits of various bandwidths with an increasing proportion of digital facilities. It is the purpose of this paper to look at a possible future realization of one portion of a switching machine that might have advantages in meeting future requirements in a largely digital environment.

We may consider that a switching machine consists of three major subdivisions. One is a switching network, which makes cross connections for each call. A controller, used to direct the operation of the network, is another. Finally, some interface is needed between the network, the controller, and external circuits. The subject of this paper is a switching network, one that may reduce the complexity of the tasks of the other two subdivisions of a switching machine as well as have advantages of its own in conjunction with some emerging technologies.

Most switching networks in the past have been "space-division" networks; that is, a spatially distinct path is assigned to each of the many simultaneous calls that might pass through the network. The present paper, however, refers to a "time-division" network, in which interleaved samples of several simultaneous calls may share parts of the same spatial path. A potential savings of equipment is implied by this time-sharing process, and its merits have already resulted in plans for a large-scale digital electronic switching machine with some time sharing in the switching network (the No. 4 ESS, now under development in Bell Laboratories).[1]

The possibility of constructing a somewhat different time-division digital switching network from two basic building blocks is explored in this paper. The two blocks, or subsystems, may be realized in the form of planar shifting arrays which are basically shift registers that can perform shifting operations in two orthogonal directions. These capabilities seem to be consistent with those of the emerging technologies of magnetic bubble and charge-coupled devices,[2,3] which do shift data on a plane. Although these types of devices may be particularly well suited for use as planar shifting arrays, the concepts presented below are not restricted to implementation by any particular type of device.

The first building block, a time-slot interchanger, is the only actual switching element in the system. The other type of block, a mass serial-to-parallel converter, performs a time-space mapping and thereby acts as the interconnection links between successive stages of time-slot interchangers. Networks of arbitrary size and blocking probability can be fashioned from these two building blocks.

In the sections that follow, the basic interconnections of time-slot interchangers and mass serial-to-parallel converters necessary to perform multistage switching functions are explained, and some logical details of the two building blocks are presented. Although the network is only a part of a total switching machine, the concepts presented here may simplify the interface with digital external circuits and may help reduce the burden on the central control processor.

II. NETWORK ARCHITECTURE

In this section, multistage switching network structures are described that use pure time-division techniques only. Such a network may be diagrammed using two types of functional blocks, as in Figure 1.

Each block labelled TSI denotes one time-slot interchanger—a familiar subsystem in the time-division switching art. The attached notation $N \times M$ indicates that the TSI rearranges words from an

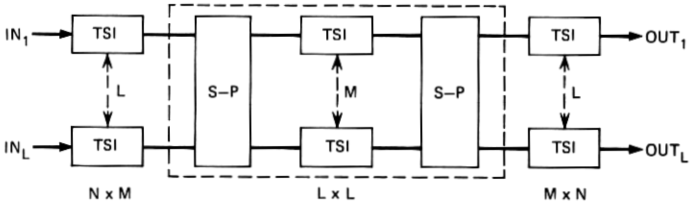

Fig. 1—Three-stage switching network.

N-slot input frame into an M-slot output frame. The TSI is the only actual switching element in the network. It corresponds to an $N \times M$ array of crosspoints in an analogous space-division network.

Each block labelled S–P denotes a mass serial-to-parallel converter. The S–P does not perform switching functions, but corresponds to the links between crosspoint stages in a multistage space-division network. Its basic function is that of interchanging the space and time coordinates of each word it handles. As inputs to one S–P, the figure shows L time-division multiplexed lines, each carrying M time slots per frame. The output is M lines with L slots per frame. An input word in slot m of line ℓ will always be routed by the S–P to slot ℓ of output line m, where $1 \leqq m \leqq M$ and $1 \leqq \ell \leqq L$. Thus, the mth words of all L input frames are combined to form a single output frame on line m, and conversely the input frame on line ℓ gets distributed into the ℓth slots of all the M output frames.

A three-stage switching network is diagrammed in Figure 1 using the two functional blocks. To illustrate its operation, suppose that a word in slot n of input line ℓ must be sent to time slot ν on output line λ, where $1 \leqq n,\nu \leqq N$ and $1 \leqq \ell,\lambda \leqq L$. First an intermediate time slot m is assigned for some $1 \leqq m \leqq M$, which permits the middle stage to complete a connection. Thereafter:

(*i*) The ℓth input TSI switches word n to slot m.
(*ii*) The first S–P places this word in slot ℓ on line m.
(*iii*) The mth intermediate TSI switches word ℓ to slot λ.
(*iv*) The second S–P places the word in slot m on line λ.
(*v*) The λth output TSI switches word m to slot ν, and the task is finished.

The portion of the figure surrounded by a dashed line performs the same function as the time-shared space-division switch (TSSDS) stages of the No. 4 ESS.[1] That is, all words that enter the first S–P in time slot m will exit the second S–P in the same slot. In their passage through

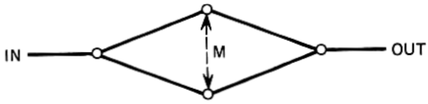

Fig. 2—Spiderweb graph for Figure 1.

the S–P's and the *m*th intermediate TSI, however, these words may be spatially rearranged to different output lines. A direct realization of the same TSSDS function using magnetic bubbles has been suggested by P. I. Bonyhard.[4]

Figure 2 is the probability linear graph[5] (also called spiderweb or Lee graph) of all paths for one call in Figure 1. It is the standard Lee graph for a three-stage network and yields the usual blocking formulas. In particular, if $M \geqq N - 1$ and we assume independent occupancy p of all input and output time slots, then a modification of C. Y. Lee's argument[5] yields the mismatch blocking probability first given by M. Karnaugh[6] in 1954:

$$P_B = (p^2)^{M-N+1}[1 - (1 - p)^2]^{2N-M-2}(N - 1)!^2/M!(2N - M - 2)!. \quad (1)$$

On the other hand, if $M \leqq N - 1$ and we assume all intermediate time slots m have independent occupancy p, a more appropriate estimate is:

$$P_B = [1 - (1 - p)^2]^M. \quad (2)$$

When $M \geqq 2N - 1$ in Figure 1, P_B vanishes and the network is nonblocking.[7] Figure 3 illustrates such a case for $M = 2N$. A part of the time-space interchange is accomplished, not by the S–P, but by splitting each input and output TSI into two separate TSI's. Now each TSI in the network handles input and output frames of equal size. This might prove convenient for certain applications or implementations. Note that Figure 3 may also be interpreted as a pair of three-stage networks whose inputs and outputs are tied in parallel. A third, inactive, network could be placed in parallel as well, and held in reserve against failure of one of the first two.

There is an unavoidable signal delay of at least one frame for each stage of the TSI's. This could contribute to the cost of echo suppression on long toll circuits; however, the duration of the frame itself may be shortened within the switch to mitigate this effect.

The three-stage network of Figure 1 has a capacity of $C = LN$ terminations. Such networks may be nested to achieve a higher switching capacity. A five-stage example having capacity $C = JLN$ is dia-

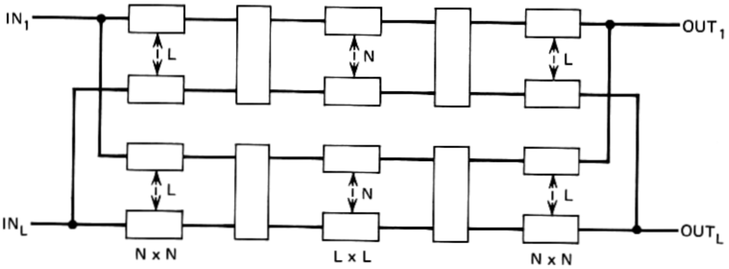

Fig. 3—Nonblocking three-stage network.

grammed in Figure 4. Shown are M three-stage TSSDS sections within dashed lines, as well as one large overall TSSDS which rearranges JL words spatially in each of M time slots. Blocking probabilities for the five-stage nested network are obtained by arguments similar to those for eqs. (1) and (2), using the spiderweb graph in Figure 4.

A different kind of five-stage network organization is shown in Figure 5. No section of this network performs the TSSDS function. In fact, a word entering its first S–P in time slot m may exit its last S–P in any time slot μ for $1 \leqq m,\mu \leqq M$, depending upon the path it follows through the network. The number M^2 of possible pairs (m, μ) yields the same number of possible paths for routing a message, as the spiderweb graph in Figure 5 shows. At full occupancy, this graph reduces to

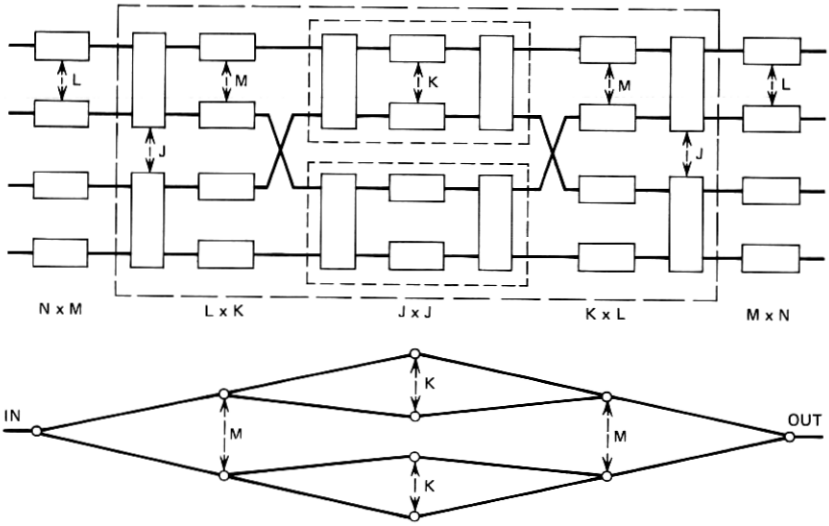

Fig. 4—Nested five-stage network.

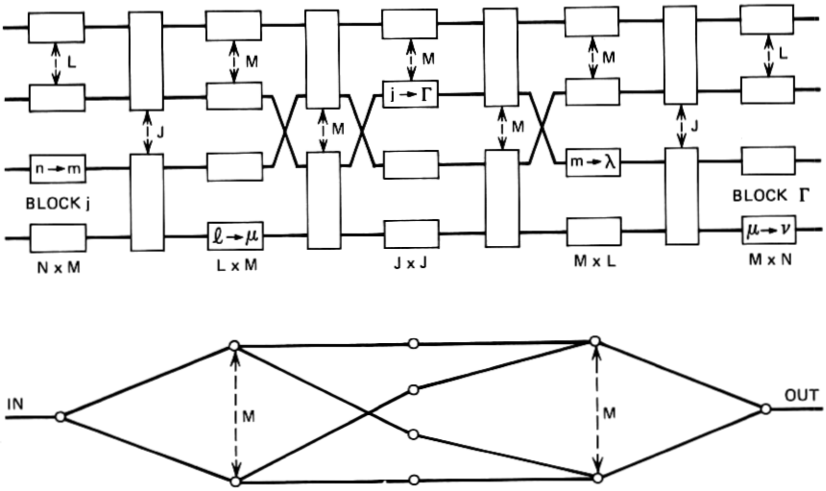

Fig. 5—Cross-connected five-stage network.

Figure 2, but with $M' = (M - N + 1)^2$ center-stage switches, so that (2) applies with link occupancies $p = N(L - 1)/M^2$ to yield the following blocking probability expression:

$$P_B = [1 - (1 - p)^2]^{(M-N+1)^2}. \tag{3}$$

Thus, parameters $L = N = 48$ and $M = 60$ yield $P_B < 10^{-10}$ at any occupancy level; essentially, this is a nonblocking network. The same parameters in a nested arrangement produce a much higher degree of blocking, $P_B > 0.027$ at full occupancy, based on (1) for the case $p = 1$.

To illustrate operation of the network in Figure 5, suppose that a word in time slot n on line ℓ of input block j must be transmitted to slot ν on line λ of output block Γ for $1 \leqq n,\nu \leqq N$ and $1 \leqq \ell,\lambda \leqq L$ with $1 \leqq j,\Gamma \leqq J$. First, intermediate time slots m and μ must be chosen for some $1 \leqq m,\mu \leqq M$ which permits completion of the connection. Then TSI's in the five stages perform the following switching operations consecutively to complete the task, as indicated on Figure 5:

(*i*) Switch word n into slot m (Stage 1).
(*ii*) Switch word ℓ into slot μ (Stage 2).
(*iii*) Switch word j into slot Γ (Stage 3).
(*iv*) Switch word m into slot λ (Stage 4).
(*v*) Switch word μ into slot ν (Stage 5).

The networks of Figures 4 and 5 each have a center section consisting of the third-stage TSI's and the two rows of S–P's to which they attach. A total of J independent input and output blocks of LN terminations each connect to the center. Each pair of blocks consists of an input S–P, an output S–P, and their attached TSI's in the first, second, fourth, and fifth stages. This structure immediately suggests an appropriate strategy for growing the network capacity in modules of LN terminations.

III. NETWORK PATH SEARCH

Section II mentioned the need to choose one or two intermediate time slots m before setting up a message path through the network. This amounts to specifying which of the many possible paths in the spiderweb graph is currently free and will be used. The decision might be made by a central control processor after consulting its memory records of the current status of the network. In this section, an alternate procedure is described by which some simple operations within the network itself can identify suitable paths for routing a new call. A possible advantage is reduced demands upon processor time and memory for call processing.

To provide necessary information about network status, a "busy-bit" is used. This is a bit in each word which travels through the network with that word and serves to indicate whether or not the word is part of a call currently in progress. The busy-bit may occur in every word of a message or less frequently, perhaps every tenth or hundredth frame. For concreteness, assume the busy-bit is "one" for a call in progress and "zero" otherwise.

To determine occupancy of a time slot on some line in the network, we merely consult the busy-bit in that slot. For example, in routing a word from input line ℓ of Figure 1 to output line λ, an intermediate time slot m was chosen that was vacant both at the output of the ℓth first-stage TSI and at the input of the λth third-stage TSI. Such m may be found by comparing the two corresponding streams of busy-bits. This is illustrated by Figure 6a for the case $\ell = \lambda = 1$. Output from the first input TSI and input to the first output TSI are sent to a NOR gate. When simultaneous zeroes are found in the busy-bit positions of words m, an output pulse is produced by the gate identifying an available path through the network by its timing.

Other means of sampling and matching busy-bits are possible. For instance, the stream of busy-bits from the ℓth input TSI will exit the first S–P simultaneously in word time ℓ, while the busy-bits entering the second S–P at word time λ are those for the λth output TSI. Hence, timed sampling pulses to the two S–P's can select the two de-

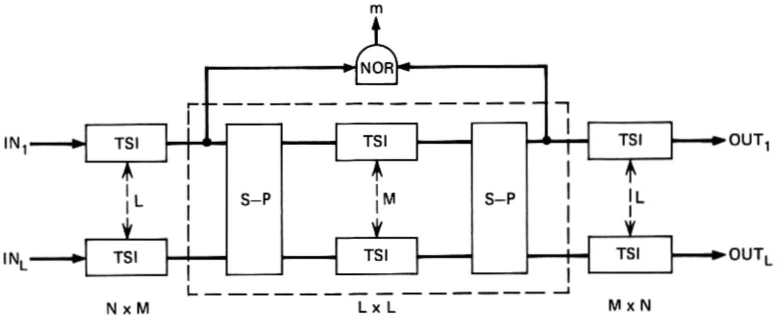

Fig. 6a—Path search in three-stage network.

sired strings of busy-bits rather than use spatial selection of an input and output TSI. Such an option might serve to reduce the amount of control and signal wiring associated with the network.

Once an intermediate time slot m is chosen, a central processor could issue the necessary orders to the first-, second-, and third-stage TSI's to establish the path. But again some additional logical apparatus in the network could perform the same task. The significance of this observation is that the dependence of the network upon external intervention may be reduced. Indeed, for each new message, the principal external control required would be a specification of which input and output terminations must be connected. Such a structure might lend itself, for example, to applications in which all processor functions are performed by a remote computer that sends its instructions to the network over a data link.

The path search procedure in the nested five-stage network of Figure 4 would consist of two nested three-stage path searches. That is, the busy-bit streams of the input and output TSI's are compared, as above, to determine which one m of the M intermediate three-stage networks will carry the message. Then the mth three-stage network is searched for some intermediate time slot $1 \leqq k \leqq K$ which will complete the connection.

A different search procedure is required for the cross-connected network of Figure 5, since all M^2 paths in the spiderweb graph must be tested. This involves sorting the M busy-bit streams leaving the second-stage TSI's of the input block j by using them as inputs to an S–P with M output lines. The S–P outputs are then compared by a set of OR gates to the M busy-bit streams entering the fourth-stage TSI's of the output block Γ. A "zero" at word time m from gate μ identifies a possible path in Figure 6b. Note that, if time slot m from the first-stage

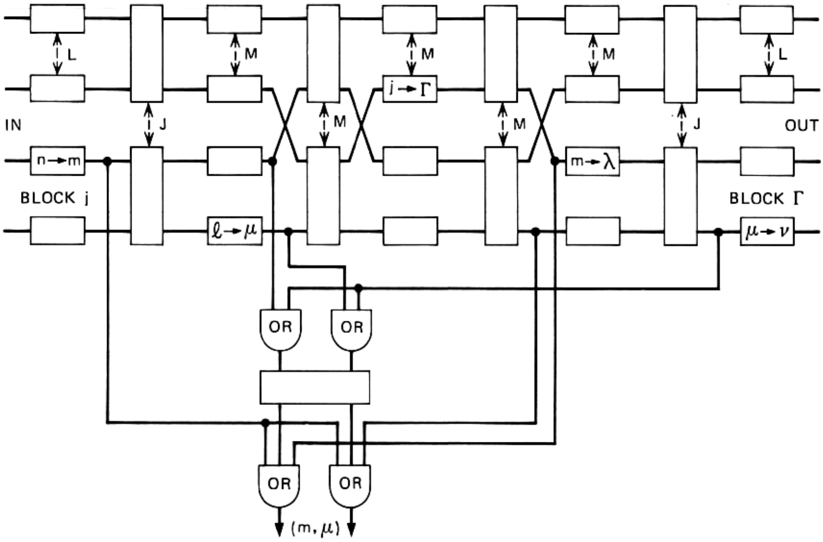

Fig. 6b—Path search in five-stage network.

TSI or time slot μ into the fifth-stage TSI is busy, then path (m, μ) cannot be used. To eliminate such possibilities, all the OR gates should be pulsed during each busy first-stage slot m, and all gates μ corresponding to busy fifth-stage slots should be pulsed for the entire frame, as is accomplished in Figure 6b by feeding parallel "ones" to the S–P. This last path search requires only one pass instead of two nested steps.

IV. PLANAR SHIFTING ARRAYS

To understand the role of planar array devices in the realization of the functions described above, it is helpful to study the building blocks at the logical level, with a notation suggestive of the two-dimensional nature of the devices. The device technologies mentioned in Section I differ considerably in their physical principles. Nevertheless, certain common features of their operation may be abstracted to aid in discussing their use for switching applications. To this end, a "planar shifting array" (PSA) notation will be introduced, exemplified by Figure 7. Circles ○ represent fixed word-storage locations in one-dimensional or two-dimensional shift registers. Diamonds ◇ stand for shifting apparatus, which can move the contents of each circle to the next in line, under control of clock pulses A and B. Gates $\boxed{n}$ allow transfer of individual words between the adjacent circles under control of switch-

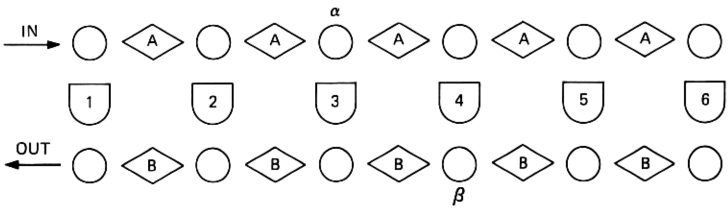

Fig. 7—Switch elements for time-slot interchanger.

ing signals n. The distinction between ◇ and ⊔ is one of function rather than physical structure. In this section, a word, rather than a bit, is the basic quantum of information. It is not necessary to specify whether this word is in a serial or parallel digital representation or some other form.

A basic scheme for performing switching functions in a time-slot interchanger appears in Figure 7. Gates 1 to 6 allow transfer of input words from a frame stored in the upper shift register into output slots in a frame stored in the lower shift register. The input and output frames move, relative to one another, in the two adjacent shift registers, and the switching strategy is: For each input word α, wait until its destined output slot β lies directly underneath and then transfer through the appropriate gate.

It is important to note a particular difficulty. If both the input and output frames are shifted simultaneously, then word α in the diagram will pass slot β halfway between gates 3 and 4, so that it is not possible to transfer. This we will call the "half-word" problem. At least three kinds of solution can be suggested:

(*i*) Alternate clock pulses A and B, so that only one frame shifts at a time.

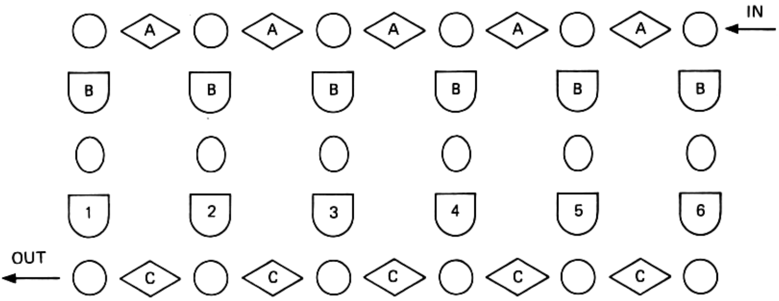

Fig. 8—Switch elements for time-slot interchanger with buffer.

(*ii*) Omit one set of clock pulses, say A, so that one frame is held in stationary buffers.

(*iii*) Let each circle contain only a half-word, so that two consecutive locations hold an entire word. Then it is never necessary to transfer between first and second half-word locations.

A fourth solution, based on the technique of bubble expansion, has been found by P. I. Bonyhard.[4]

The first solution above may be illustrated by Figure 7. The basic switching cycle would consist of the following four steps:

(*i*) Diamonds ◇A move the input words right one slot.
(*ii*) Gates ▢n may transfer input words to output slots.
(*iii*) Diamonds ◇B move the output words left one slot.
(*iv*) Gates ▢n may transfer input words to output slots.

For an M-word frame, switching is completed in M cycles. Each shift register operates M times, and the M gates each have as many as $2M$ chances to operate in every frame.

The second solution to the half-word problem is illustrated by Figure 8. The diamonds ◇A read in an entire input frame, which is stored in buffers ○ by simultaneous operation of all gates ▢B at the end of the input frame. Then the following two-step switching cycle:

(*i*) diamonds ◇C move the output words left one slot,
(*ii*) gates ▢n may transfer words from buffers to output,

is repeated $2M$ times, and the M gates each have M chances to operate in an M-word frame. This scheme has the additional advantage that the input and output clocks need not be synchronous. Each solution to the half-word problem has advantages with respect to particular hardware.

Figure 9 uses the PSA notation to illustrate a simple scheme for implementing the S–P function. The S–P shown has $L = 4$ input lines, each carrying six-word frames, and $M = 6$ output lines, each with four time slots per frame. The diamonds ◇A operate to load an entire frame into the S–P from each input line. Then the diamonds ◇B operate to unload an entire frame from the S–P onto each output line. Loading and unloading may proceed at twice the external basic word rate so that the S–P empties in time to receive the next frame, or else an alternate S–P may handle the next frame while the first one is unloading. Other schemes permit the S–P to load and unload simultaneously. While the S–P is drawn as a distinct unit in preceding figures,

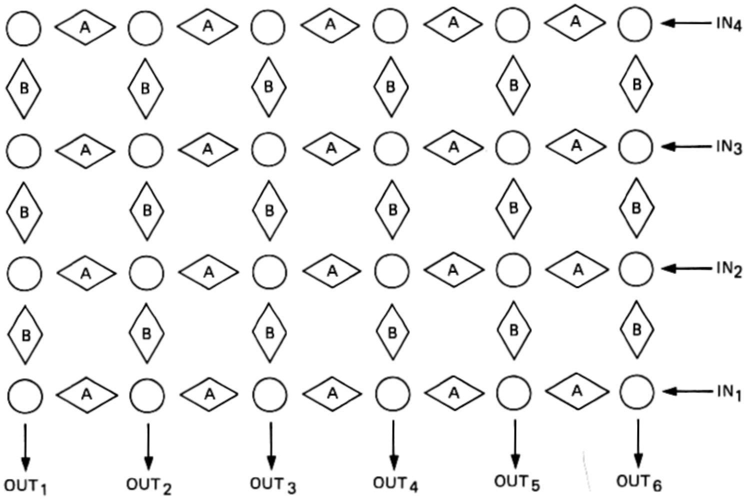

Fig. 9—Mass serial-to-parallel converter.

only its function need be distinct. The S–P as a device may be broken up and its parts integrated onto the same chips as the various TSI's which it serves. Schemes to obviate the S–P completely using specific device properties have been proposed by W. F. Chow and P. I. Bonyhard.[4]

V. MEMORY STRUCTURE

In the switching schemes described above, each gate n should operate at the same word times in each consecutive frame; it is natural to place it under the control of a recirculating local memory. This could consist of a shift register fabricated from the same materials as the switching elements and preferably integrated onto the same chips. Such an arrangement is diagrammed in Figure 10, using the PSA notation. The switching elements appear at the right. The rest of the diagram is the local memory array to control these elements. Each shift register in the control memory moves its contents one word to the right under the action of clock pulse *C* whenever a pulse *A* or *B* occurs. A word in shift register *n* is read when it reaches the box $\boxed{n}$, which is considered to generate a pulse *n* that operates gates n . For example, the gating operations might be realized through repulsion between control bubbles and message bubbles. Gates n at the left start new words down the shift registers to accomplish recirculation of the local memory.

The size of the local memory array depends on the number of words Q in each register and the number R of such registers, which is the same as the number of switching gates. Although the exact size depends on the particular solution chosen for the half-word problem presented above, we can conclude in each case that the local memory in a TSI grows quadratically in frame size (as QR), while the number of switching elements grows only linearly (as R). For moderate-size frames (say, 20 words or more), each TSI becomes a large recirculating memory plane with a small quantity of logic elements at the edges. A word in the control memory corresponds most closely to a crosspoint in a space-division switch, since the number of crosspoints also grows as the square of the number of circuits, and each crosspoint stores just one bit of information.

VI. CONTROL INSERTION AND ERASURE

The switching operations of a TSI are fixed from frame to frame on a short time scale (say, thousands of frames); thus, the contents of its local memory array are fixed. On some longer time scale, though, it is necessary to be able to change portions of local memory; for instance, in response to external signals emanating from a central control processor.

Specifically, one requires the capability to address single word positions in the local memory in order to close or open a "crosspoint" by writing or erasing a bit. In the case of erasure, it would be sufficient to simultaneously erase all memory locations affecting a given input word, or alternatively a given output slot. This is due to the following two properties of the TSI:

(A) Each input word is switched into one output slot at most.
(B) Each output slot receives at most one input word.

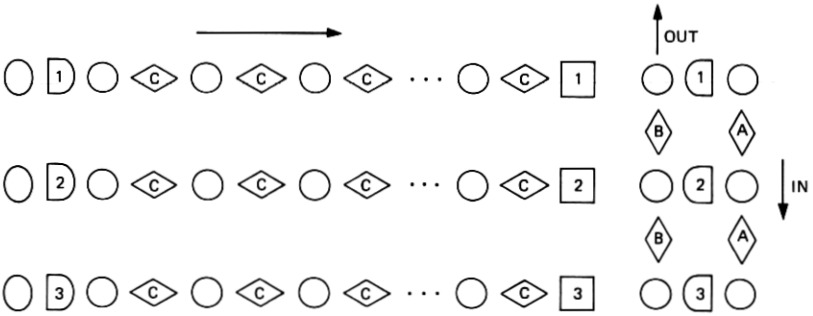

Fig. 10—Recirculating memory array for TSI gates.

Hence, one can provide for erasure of a control word without knowing exactly where it is in the local memory by erasing a whole class of words.

The fact that local memory recirculates will greatly simplify the problem of addressing specific locations, since this allows access to any location from the edge of the array. Indeed, one efficient and convenient scheme would be to build an analog of the switching elements and operate it "backwards" to insert control bits into the local memory. Insertion could be directed by two external control pulses whose timing would specify an input word and the output slot for which it is intended. Such an approach is illustrated in Figure 11, in which switching elements are deleted to concentrate attention on the memory plane. Control insertion elements appear at the right-hand side. A bit is inserted in order to control gate n at word time m as follows:

(*i*) At the start of a frame, D gates a "one" into the vertical shift register, where it propagates downward.

(*ii*) After n of the A clock pulses, gates I operate to place the "one" in buffer ○ at the right of the nth recirculating shift register in local memory.

(*iii*) In the next frame, gates J operate with the mth clock pulse C, to place the "one" in word m of memory line n. This is indicated formally by reading the "one" at n , but the actual details of bit injection will depend on the type of hardware.

It is clear now that any word location in the local memory may be addressed employing a pair of time-coded pulses I and J. The scheme shown would be particularly appropriate to the case of buffered input, since n then becomes the number of the input word to be switched.

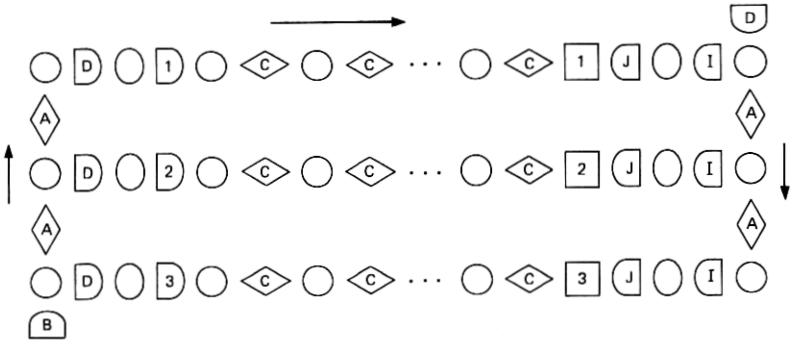

Fig. 11—Addressing and erasure schemes for the TSI local memory.

Indeed, the nth shift register contains only those locations that can affect input word n. By property (A) above, at most one location on this memory line has nonzero contents. It is sufficient now to erase the entire shift register, by interrupting recirculation, for example, before a new control word is inserted.

The left side of Figure 11 shows a simple scheme for erasure in the case of buffered input, which also guarantees that at most one nonzero word is resident in each memory line. Once each frame, the buffers ○ at the left are loaded with "ones." Gates $\boxed{n}$ then start these "ones" down the various shift registers at the appropriate times to accomplish recirculation. Clearly, at most one nonzero word can circulate. To erase memory line n, the "one" is simply deleted in its recirculation buffer for a single frame.

A single time-coded pulse will suffice to specify erasure of any word in the local memory above, if the buffers are loaded serially. Just such an arrangement appears at the left side. Gate $\boxed{B}$ enters "ones" in the vertical shift register at each input word time, and these are loaded into the buffers by gates $\boxed{D}$ at the start of each frame. By deleting the B pulse at the nth word time, a "zero" is sent to the buffer for the nth memory line, erasing it and opening any "crosspoint" which might affect the nth input word. The deletion of pulse B could coincide with pulse I, which routes a new control word to memory line n. Then the pair of external signals I and J would accomplish erasure together with address insertion. Examples at device level of this procedure have been given by P. I. Bonyhard and W. F. Chow.[4]

VII. BUSY–BITS FOR PATH TAKE–DOWN

The preceding section considered two schemes for erasure in the local memory:

(*i*) A direct instruction from the central control processor.
(*ii*) Automatic erasure when a new control word is inserted.

A third scheme is to provide for automatic erasure when the message that is being switched terminates. This might be accomplished through use of the busy-bits introduced previously. Recirculation of a word in local memory would be made contingent on the presence of a "one" in the busy-bit position of the particular input word which that control word switches. When the message terminates, the path along which it was routed through the switch is taken down simply by sending through one word with a "zero" busy-bit. This function is analogous to that of the sleeve-lead in electromechanical switches and might be considered an "electronic sleeve-lead" application of busy-bits.

This strategy of making recirculation in the local memory contingent on busy-bits can be easily implemented in Figure 11. It is merely necessary to read the busy-bit stream for the input frame into the left-hand shift register and load it into the recirculation buffers once each frame. Now each busy-bit will prime the recirculation of the memory register associated with the word of that busy-bit.

VIII. CONCLUSION

The possibility of constructing a time-division switching network using two building blocks of planar arrays was discussed in this paper. The compatibility of planar arrays with the emerging technologies of magnetic bubble, charge-coupled, and bucket-brigade devices[2,3] may lead to application of these concepts in the construction of relatively inexpensive large-capacity switching machines.

The need for external connections to a network composed of these building blocks can be minimized by including path search and path maintenance functions with the blocks. As a result, a relatively small amount of information must be exchanged between the network and the supervisory processor, and some of the processing burden on the controller is shared by the network itself.

Some examples of compatible network architecture have been given, although no specific design is proposed. The process of switching is accomplished by time-slot interchangers within the network rather than a space-division crosspoint array, but the task performed by some of the described subnetworks, when viewed from their external ports, is the same as a time-shared space-division switch.

ACKNOWLEDGMENTS

We wish to thank the reviewers and referee for their comments and suggestions. We have received invaluable advice on the properties and capabilities of bucket-brigade and charge-coupled devices from W. F. Chow, C. H. Sequin, G. E. Smith, and M. F. Tompsett, and on magnetic bubble devices from P. I. Bonyhard, who also suggested improvements in the network design.

REFERENCES

1. Vaughan, H. E., "An Introduction to No. 4 ESS," paper presented at the International Switching Symposium, Massachusetts Institute of Technology, Cambridge, Mass., June 6, 1972.

2. Bobeck, A. H., Fischer, R. F., Perneski, A. J., Remeika, J. P., and Van Uitert, L. G., "Application of Orthoferrites to Domain Wall Devices," IEEE Trans. Magnetics, *Mag-5*, No. 3 (September 1969), pp. 544–553.
3. Sangster, F. L. J., and Teer, K., "Bucket Brigade Electronics—New Possibilities for Delay, Time Axis Conversion and Scanning," IEEE J. Solid State Circuits, *SC-4*, No. 3 (June 1969).
4. Chow, W. F., and Bonyhard, P. I., unpublished work (1971) and private conversations.
5. Lee, C. Y., "Analysis of Switching Networks," B.S.T.J., *34*, No. 6 (November 1955), pp. 1287–1315.
6. Karnaugh, M., unpublished work, August 1954.
7. Clos, Charles, "A Study of Non-Blocking Switching Networks," B.S.T.J., *32*, No. 2 (March 1953), pp. 406–424.