# Memorandum

FOR ALL INTERNAL COMPANY CORRESPONDENCE

DATE: April 20, 1962

COPIES TO:

TO: All Concerned

FROM: W. L. Gordon

SUBJECT: SMALL MACHINE

COMPANY PRIVATE
NOT FOR DISTRIBUTION

The attached is a first stage report on a design study for a low cost data-processing system. It is submitted for comment and criticism on all grounds. It is both a system specification and discussion justifying the design. If it seems lengthy and devoid of detail it is not because the detail is not available, but is due to this writer's feeling that exposition can cover more ground in a report of this nature than by tabular listings of unjustified detail.

Many individuals must share the blame for this concoction which has required the invention of many new techniques. All departments have been extremely cooperative in this endeavor, but in particular without the beautiful development work and interest of a small group of circuit designers, we would still be groping for a low-cost system.

The system design and determination of feasibility is the result of sustained efforts by Richard Little (270), Patrick Nugent (220), and the author with notable assist from William Daly (240) and Leonard Ruthazer (230). The latter must also serve as representative of the aforementioned circuit design group.

The section of this report describing the processor has been prepared by R. Little. In other areas of the report the aforementioned gentlemen can only be credited with the good ideas.

W L Gordon

# SMALL SYSTEM

The following is a rough systems specification for a low-cost processing system of great flexibility and extremely high expansion potential. As will be seen, it is much less a specification for a single system than it is an integrated set of rules for the interconnection of various independent peripheral devices to form a vast number of system configurations. The high degree of modularity is specified intentionally to provide both great system flexibility and low prices for minimal systems.

The underlying system philosophy to be found here is that each subsystem element (e.g., high speed line printer) be given a standard interface with a central source of information and control. This pivot point for all peripheral elements is taken to be a memory system constructed from acoustic glass delay lines.

The central storage unit provides:

1. Distributor action (traffic control) allowing simultaneous operation of all peripheral devices.

2. Information (or buffer) storage for all peripheral devices (including a peripheral processor).

3. A storage depot for exchange of all control information (ie, print control words, busy signals, etc.).

4. Clocking and synchronization for all elements of the system.

The radial system organization is as diagrammed (Figure I below). System enlargement can occur by appending additional spokes to the wheel. An alternative method of allowing the system to grow is shown in Figure 2. This organization is really quite similar to the "off-line" type of operation within an H-800 installation -- the difference being that all peripheral devices are "off-line" to a serial memory rather than a serial tape and the information batch sizes and access times are very much smaller. The general system organization is as "universal" as the H-800, but at far lower cost. Indeed, the H-800 can be appended as a peripheral device.

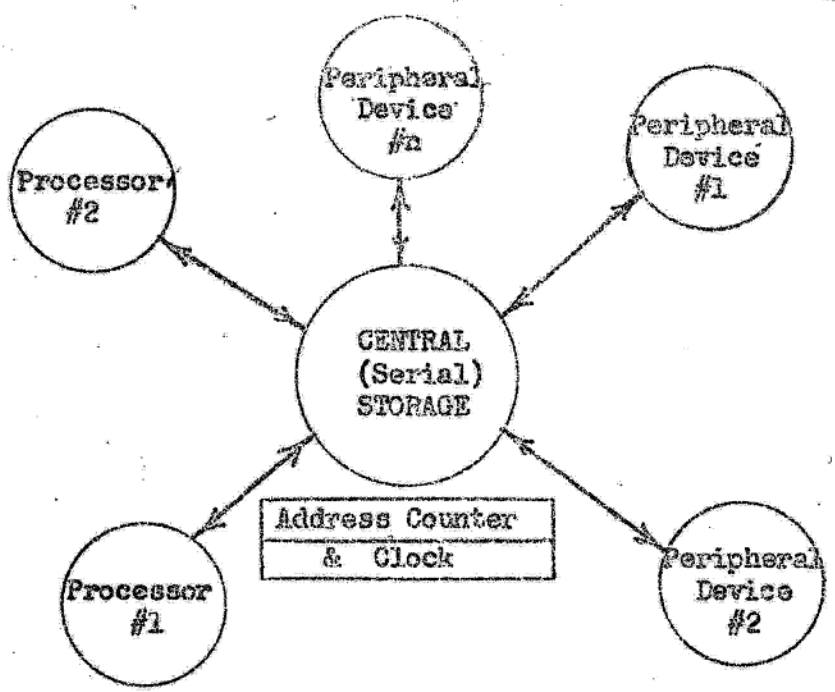


Figure I

BASIC SYSTEM ORGANIZATION

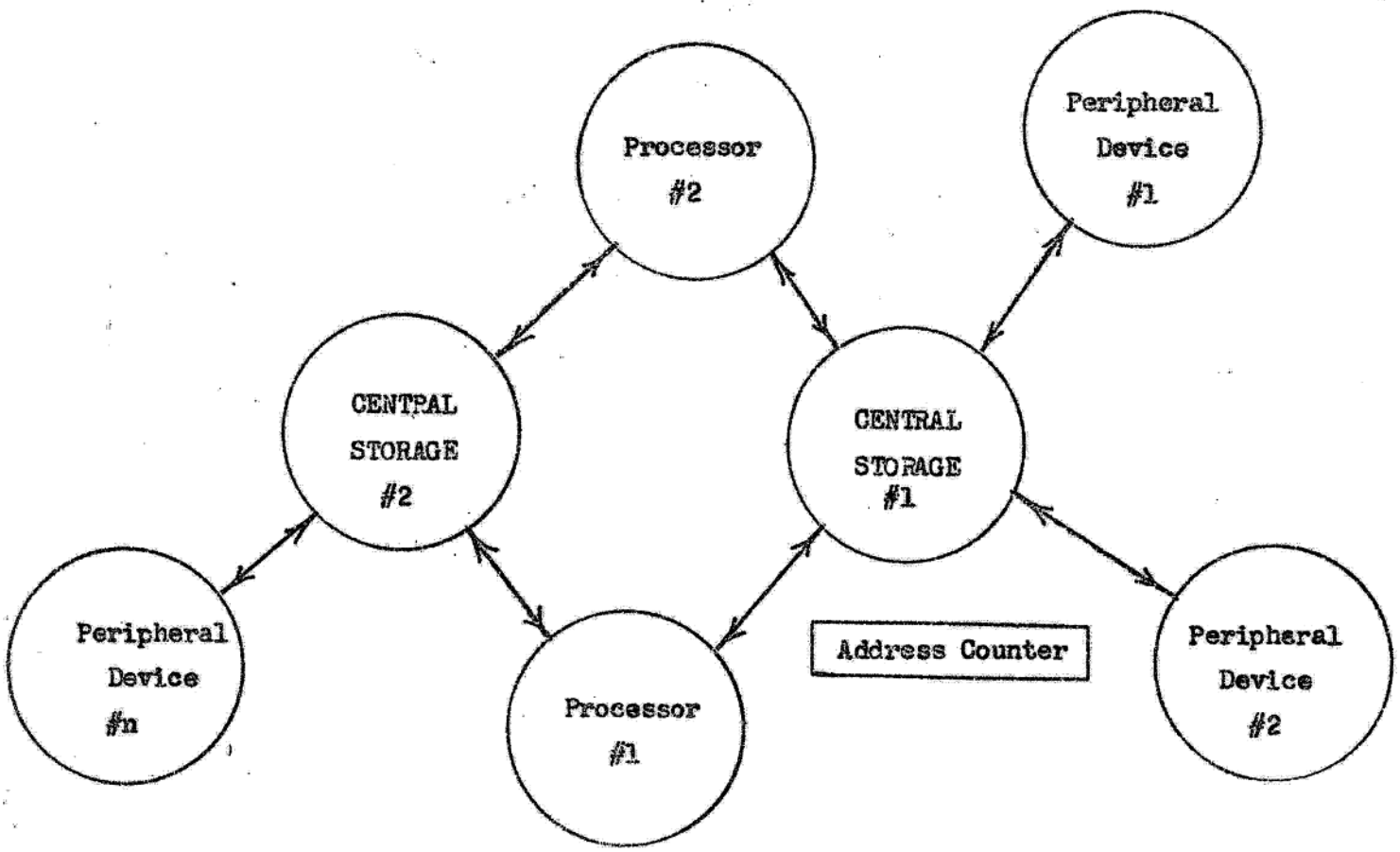


Figure II

EXPANDED SYSTEM

The peripheral devices intended for possible inclusion into any such system configuration can run the entire range of data-processing components. The general criteria for establishing the feasibility of coupling a device to central storage are relatively simple:

(1) The unit of information delivered as a result of peripheral action shall not exceed the capacity of the memory.

(2) It is desirable that the peripheral device can be completely controlled from memory, i.e., be clutched or cycled on demand from a control word in storage.

(3) Barring the above, e.g., in real time input devices such as communication lines, the information delivery rate shall not be too large in the sense that the memory together with processor can process the information faster than it comes in (or out). The clutched device has the lovely aspect of allowing even the slowest processor full control on its average information transfer rate.

(4) In using a serial memory, the access limitation should not impose a storage burden on the peripheral device. The game we are playing with the acoustic glass line is that it serves well as information buffer storage for virtually all peripheral equipment in use today.

The specification of the acoustic glass delay lines (as opposed to other storage devices) is prompted by these major considerations:

1. For the provision of a modest amount of storage the delay line is far less costly than a core counterpart on a dollars per bit basis, or, more importantly in absolute cost.

2. The natural motion of the information storage provides the desired breakdown of time (and storage) into smaller units. This allows complete time sharing of input-output paths to storage without elaborate and expensive traffic control or interrupt hardware and without any waiting lines. Communication is direct to memory.

3. The random access times to storage, although not measured in small numbers of microseconds, but rather in hundreds of microseconds are fully compatible with the requirements of the peripheral devices. This should be compared with the magnetic drum (too slow) and the core memory (too fast - requiring interrupt activity).

4. Modularization of memory is trivial and enlargement of storage can probably be a field modification.

A new system concept employed here is the consistent use of memory as a source of control. By this device we can establish the high degree of modularity that seems to be desirable in a small machine intended for use in many different system environments.

The importance of this notion stems from the fact that the processor is completely alleviated of the burden of interpretation and execution of peripheral orders. This in turn means:

1. Complete time independence of all devices in the system with resulting simultaneity of action;

2. Great simplification of the control areas of the processor with resultant cost savings in the minimal system;

3. The system design need not account for the characteristics of any particular peripheral device except to ascertain that device of its genre can be inexpensively coupled to the system. As they become available, such devices may be appended with no modification to any other portion of the hardware.

4. A varying peripheral complement leaves all other sections of the hardware unchanged. Minimal software for a minimal system could likewise go unmodified.

General Method of Peripheral Control

Each peripheral control unit has associated with it a fixed address in memory wherein it finds control information as deposited by the processor and to which it delivers control information on completion of peripheral action.

For example, suppose the print control unit is to utilize memory character addresses 0101 through 0220 for 120 characters of storage. Utilizing register 0100 as a control exchange point, we establish the following (simplified) rules. Each time address 0100 becomes available to a non-busy printer, it must be examined by the peripheral control unit. If it is 000000, there is no demand for action. If a non-zero configuration appears in this location (as a result of a programmed transfer of a non-zero character to this point) the printer control interprets this as a demand for action - to print and advance paper a specified number of lines - and take appropriate action. When printing from the storage area is complete, the printer control must clear character 0100 to zero, complete its vertical formatting action and start the cycle anew by looking at 0100 when ready.

The program, on the other hand - if printing - interrogates 0100 to see if it is clear. If it is not, it must stall or do other work. As soon as it finds 0100 is clear, it can begin editing into the storage area or transfer an edited line into this area and then demand printer action by appropriately setting up address 0100.

Interlocking is then on a simple programmed basis for the program must test this register before it takes action. However, once it sets up address 0100, it can go about any business it has to do and not look at 0100 until it wants to deliver another line image to paper. Processing is then concurrent with printing.

Errors detected by the peripheral device can deliver suitably coded information to memory describing the error type. This mode of operation (as opposed to a pure demand-release operation) would be employed if the program could take cognizance of the error and do something about it. A condition such as "parity frame error on tape" is of this type, but "no paper in the printer" could merely hold printer-busy on causing the program to stall with the instruction still in memory and ready to be picked up when the error is rectified by the operator.

One character of control information is deemed sufficient to handle all peripheral selection and alternatives as, say, tape drive selection and motion, vertical formatting information for printing, etc.

It may be argued that this method of peripheral control requires that a peripheral device, e.g., printer control, must be able to write in memory as well as read. This is true, but observe that the interface is not complicated by this requirement since one additional wire is required to effect this action and several devices, e.g., tape control and card-reader must be capable of both actions in any event.

Storage:

The disadvantages of delay-line storage are well known but will be briefly discussed here:

1. The worst aspect of this type of storage is its volatility; i.e., a failure within memory, however caused, can completely invalidate the contents of the entire storage unit. Note that general use of even the most protected memory system does not rely on the validity of memory once an error is detected, but calls for full reloading of the processing program from a convenient restart point. In this sense, the volatile memory poses no new procedural problems. Nevertheless, we can expect a delay-line system to be subjected to criticism out of proportion to any true reliability indicators if only because the sanctity of the core memory seems to be as well established as the institution of marriage or, better, the infallibility of IBM.

For at least one manufacturer, this disadvantage has been parlayed into a sales advantage if one is to believe the field criticism of the H-290 that it is not battery operated. For the system at hand, such sub-liminal devices are manifestly impossible because such gadgets as printers, card readers, tape transports use a small chunk of Niagara when operating.

2. Access Limitations - Once information is placed in a delay line, it is irretrievable for the length of time determined by the length of the delay line. Various system gimmicks are employed to offset this access limitation - the notable success story being the drum IBM 650. However, minimum latency coding in the minds of some is as dirty a word as "volatility" and it is important to point out to the prejudiced individual that, in spite of serial storage, we speak here of a device whose basic instruction execution rate including operand access times is of the order of 5,000 to 6,000 instructions per second (already in the range of the D-1000) without any thought to minimum latency. Simple provisions within the processor (access improvement schemes) can raise this figure to about 30,000 operations per second for the programmer wishing to avail himself of this speed potential.

The reason this serial device is not hampered by an access limitation is simply that the maximum access time is in the microsecond range, indeed is 175 microseconds per trip around a delay line. At the postulated 20 Megacycle bit rate this is tantamount to passing over 500 characters of information in each delay line over 5700 times per second. Actual operation time of a serial add of,

say, a 5 digit field will be accomplished in under 2 microseconds, and in this sense the device is access limited.

With no access improvement the system is expected to keep all high speed peripheral devices moving at maximum speed. In part, this is due to the 5,777 instructions executed per second (which figure is competitive with the 1401) but more particularly is due to the system organization which allows full simultaneous operation of all devices - picking up a performance factor of the order of 5 over the (unbuffered) 1401. In other words, the system is not job limited.

Information Representation:

The only representation of information in this system is specified to be 6-bit Alphanumeric utilizing a code fully compatible with the H-800. All numeric information is 6-bit decimal, all computation is decimal, all addressing is decimal and the user should be unaware of the binary number system, octal representation, hexadecimal, etc. Any deviations from this standard (and there is one) are to establish compatibility with H-800 tape records.

The major factors prompting this choice of standard are:

1. The unsophisticated user that may encounter this device should not be plagued with mixed number systems. Extreme simplification of the programmer's problem - in a data processing environment - does occur. Relative ease of programming and debugging in machine language must remain a major consideration. With the painfully restricted storage capabilities of such a small machine, it is extraordinarily unlikely that software will be able to pull the user to a more comfortable linguistic distance from the physical device - except by use of a larger machine (e.g., H-800).

2. Peripheral devices almost uniformly want to talk a character-oriented language and

3. The processing hardware is much more complicated if it must handle mixed number systems. Whatever consistencies of hardware organization occur here are due to the uniform mode of representation of data - or program material.

To match this strictly alphanumeric mode of operation and allow program material to be manipulated by the same instruction set provided for data, all addressing is specified as decimal. The sophisticated reader may immediately suspect one or two system flaws in the establishment of such a rule, namely: (1) inefficient use of storage in that 4 bits rather than 6 are needed to specify a decimal number or (2) that binary addressing is much more economical. Such criticism is indeed true and is countered by the arguments that this type of redundancy is the difference between ease of programming and complications; that without it, software need be more elaborate; and that the system organization, as will be seen, does not truly let all of this storage go to waste. Finally, the command structure need be much less elaborate and costly if data and instruction formats are as comparable as possible.

This general approach is a major characteristic of the 1401. The mere specification of character organization works wonders in the simplicity of both hardware and programming for data processing. Any niceties of programming within the 1401 stem from this feature (and no other) and the device can be criticized from a programmer's point of view on the grounds of complexity of addressing and what appears to be an inconvenience in the use of the (all too invisible) word mark that is important in the processor, but non-existent in any other area of the system. Other variations are possible and are employed in the proposed processor described herein. As will be seen, a variable length field device is specified, but the field size is used to control the information flow.

Buffering:

The importance of simultaneous action of all devices in the system cannot be overemphasized. Of all criticism that may be leveled at the 1401 (or our own H-400 for that matter), the most damaging in terms of system performance is the complete absence of buffering. In the same manner that one may characterize a serial-multiple channel device as hurry-up-and-wait, the parallel-single channel (to storage) device can be spoken of as a wait-and-hurry-up system. In the latter case, the waiting period is not a memory recirculation limitation, but is rather the time required for each peripheral device to effect its action. The unbuffered 1401's (and H-400's) must therefore process at a rapid clip when they can, for per card in or line of print out there is very little time in which to process.

In the case of card reading on the basic 1401 system (similar timing considerations for the H-400), if it is desired to read and process at the full rate of 800 cards per minute, only 10 milliseconds out of 75 milliseconds are available to the processor to get its work done. With card acceleration time thrown in, the processor moves to 31 out of 75 milliseconds to process. Assuming an average of 230 microseconds per instruction executed (or about 4300 instructions per second) in a card reading operation (without options) the program is limited to performing a mere 43 instructions per card. With 31 milliseconds available (at additional rental) the program can do about 123 instructions per card read at full speed - provided it does not encounter a similar lock-out while an output device is active.

If one is printing at 600 lines per minute, a mere 16 milliseconds (no print storage) is available out of every 100 in each print cycle and it becomes impossible to do detail listings at the rated print speed if one is also trying to read a card - for these devices most assuredly bump into each other in time without room for any processor action whatever.

This is not intended to be a critique of the 1401 system (unbuffered) but it is intended to point out that if these timing limitations are removed, the resulting system performance can skyrocket.

Indeed (again) the major justification for this system design is the complete buffering of all peripheral action. The processor to be described will have a worst case performance figure of between 5,000 and 6,000 instructions per second (all hence per card read can execute 450 instructions minimum (as opposed to the 1401's 43 to 123 average).

This potentially means that (1) the proposed system will be able to keep all peripheral devices operating at full rated speed or (2) per item handled more can be done - provided the user purchases the increased storage to house more instructions. It also means that IBM can counter (if pressed) by endowing the 1401 with input buffer storage. Surely some time has been found to lay down the order books to design such an addendum.

Memory is to be constructed of acoustic glass delay lines of standard length 175 microseconds (including all delays in associated circuits) and operated at a bit rate of exactly 20 Megacycles. This configuration of parameters establishes each delay line as storing exactly 3500 bits disposed as 500 alphanumeric characters of 6 bits plus one (odd) parity bit.

Delay lines may be added to the system in modules of 4 (2,000 characters) and a minimum system configuration is taken at 2,000 characters. A tentative layout assumes that the selection mechanism for the lines is associated with the memory system. In adding a second processor, the line selection mechanism must be duplicated.

Each line may be selected by the processor for either reading or writing, but, per module, each peripheral device is constrained to read and write in only one line of the module. No more than 4 peripheral devices are assumed tied to any one delay line; hence, additional peripheral devices require additional storage modules. Since each peripheral device must operate into or out of the memory at distinct points in time, there is no possibility of interference except with the processor which can access any address in storage. With no interlocks, it is up to the program to determine from a control character, that it is permissible to address an information storage area. Performance of the system is unspecified if this interlocking does not take place, but in no case will equipment damage result.

In the presence of a magnetic tape control unit, one module (with peripheral input-output connection) should be reserved for exclusive use by tape and processor. This system specification requires that the tape input area (reading) be confined to 500 characters, all residing within one delay line. The Tape Control is thus unable to switch into different lines and 500 characters is established as a maximum record length for tape. Tape control exchange will invariably come from a fixed address within this fixed delay-line, but reading (or writing) tape may start from any address in this line (c.f. tape control for further restrictions). At the present time, it is an open question as to whether writing may be done from any point in the memory module; at worst it can be considered as an option (in the tape control unit).

Short Lines:

For the user desirous of reducing access time in certain applications, it will be possible to replace a standard delay line (175 usecs. delay) with a "short" delay line of 35 microseconds or exactly 100 characters of storage. Introduction of such a short line will introduce a discontinuity in memory addressing in that five distinct addresses (e.g., 1, 101, 201, 301, 401) will all reference the same information. Provision must be made in the memory address comparison mechanism to guarantee that this occurs. Desirably such

a trade of storage for time should be a field modification. To make this an operator change does not seem quite feasible. Costs for a memory system utilizing short lines are not likely to differ substantially from the costs of the standard line. The attraction is simply that processor performance increases by a factor of five (approximately) when shorter lines are utilized for both program and data storage.

With the above qualification on short lines, the memory is continuously addressed. Since each individual character is addressable and since a field of characters can be of any length, it must be possible to switch across consecutive lines as a field is read out. For example, a ten character field being read out of (or loaded into) location 496 will actually wander across the line boundary and address 500 must be referenced in a different line from address 499.

The limitations on no more than 4 peripheral devices per line are motivated by the hardware considerations of loading of the read out point of any one line as well as by the impracticality of effectively operating many peripheral devices without greatly increased storage for the control program.

Within any one line, address selection for peripheral buffer area is required to be significant in terms of the device being controlled. In particular a card reader is to be associated with addresses 0001 through 0080 and a printer reads out of addresses 0101 through 220. This is simply to establish a simple correspondence between column addresses (card or printer) card memory locations. Control characters located in positions 0000 and 0100 for these two devices respectively. More than one device (and control) of a given type can be appended to the system. Distinct lines should be used for such additions.

With these preliminaries out of the way, we are in a position to discuss general specifications for various units of the system. Note again that all peripheral control devices are optional and electrically independent of one another and that the user pays only for the equipment actually delivered.

Memory Addressing, System Synchronization and Layout:

The system requires full synchronization with a central source, here the 20 Megacycle clock, bit counter (7 bits per character), and character counter (Memory Address Counter) counting decimally from 0 through 499. This counter is truly the heart of the entire system.

Address selection for peripheral devices is in part keyed to simplification of the problem of decoding the decimal Memory Address Counter (MAC). It is anticipated that MAC will be heavily loaded, and as peripheral devices are appended, it must sustain increased loading. Provision must be made for at least 30 loads on each stage of MAC and hardware design should account for the modularity of tapping this central point. Eleven (11) bits constitute the main address register, but inter-stage carry bits and counter-overflow are also to be made available for peripheral decoding. MAC is required to establish a new count well within 350 nanoseconds (character time) to allow selection decisions to be made in time for the next character emerging from the glass delay lines.

The bit counter, moving at 20 Megacycles is likewise expected to be heavily loaded, for all timing relations within a character are derived from it. Exact loading requirements must wait further design effort.

All lines must be synchronous with the clock and with each other. The use of zero temperature coefficient glass is specified unless compatible performance and price can be obtained from other media burdened with a small oven to prevent drift. No cooling facilities other than small fans are tolerable in this system. This specification assumes the presence of an air-conditioned environment for the equipment.

With all communication between components of the system established via the 20 Megacycle interface with memory, hardware layout must assume a portion of each control unit mounted in close proximity to the memory. In general, each device appears to split into a 20 Megacycle area for information flow, a 3 Megacycle area for control on a character basis, and a lower speed area at the interface with a mechanical device. The first two areas will generally be able to effect transfers over a single line at the bit rate indicated. In an effort to reduce cabling costs, it is highly desirable to allocate space for electronics in such a fashion as to allow one small 3 Megacycle line to be cabled from control areas instead of, say, 120 print drive wires.

Printer:

Our ability to print at high speed, currently a rated 50% higher than our major competition is not to be discounted as a feature, particularly since printing appears to be the major job tackled by this type of equipment. Curiously, an examination of the peculiar internal timing relation of the 1401 (notably the 11.2 microsecond memory cycle time) strongly suggests that the original goal of the chain printer was 900 printed lines per minute. This conjecture has been directly substantiated.

However, the high cost of printing is directly related to the number of actuators in the system. It would be pleasant to report that we could print 120 character lines 900 each minute without paying the high price of 120 actuators with associated driving electronics. The best that can be done is to say that at about 600 lines per minute it will be possible to print a more restricted alphabet with one-half the number of driver-actuator combinations. The cost reductions in this area are very large and the appended estimate for printer (+ control) cost can be slashed by a sizable amount if we assume a doubly actuated printer.

The specification here is for a control unit that could be coupled (with minor modifications) to either printer with the reservation that if only one can be specified it should operate at at least 900 lines per minute.

The proposed control unit is designed to take optimum advantage of increased paper moving speeds by revising the means by which control knows a complete line has been printed. At present this is done by guaranteeing exactly one print drum revolution per line of print and requiring all 56 characters to be passed over before paper can be advanced.

The procedure specified here is that of counting the number of blanks in the print image plus the number of character comparisons made of the print image with the moving drum until a total of 120 (a full line) is encountered. The advantage of such a procedure is that paper can be advanced at the earliest possible moment and that careful usage of legal characters and proper placement of infrequently used characters can allow one line of print per drum revolution - upping print speed in most cases to 1200 single spaced lines per minute.

Replacement of the full alphanumeric print drum with a numeric only drum (14 characters repeated 4 times around the drum) will allow one line of print each 13 milliseconds. A paper feed time (single line) of 12 milliseconds would then allow 2400 lines per minute (numeric only) - or twice the IBM 1403 speed with numeric chain. This could be accomplished with no modification of control. The assumption made here is that a character code wheel is replaced with the drum. If possible, the drum should become an operator replacement. No program modification would be required to get numeric output to this rate.

The area of greatest concern in making such a specification is that of paper feed and clutching. The buffered processor (and tape input) could deliver predigested information at this rate.

In communicating with the serial delay line, when a character demand is made from the print drum, comparisons with print storage can be started within 175 usecs. and must be completed within 42 microseconds. Print hammer storage in the form of a 3 Megacycle shift register is adequate to select hammers for firing. The fact that a character demand (occurring once each 840 microseconds) can be accommodated within 217 microseconds enables a great simplification in the storage-preamplifier driver circuitry associated with each hammer. This occurs simply because the bits firing hammers on a given character cycle can be held for about 600 microseconds - eliminating the need for time delays in the print circuitry, allowing the print driver to be treated as a storage flip-flop directly.

Costs in this area are of considerable interest, for with each circuit duplicated 120 times, printer electronics easily become the most expensive item in the system (with printer). This problem is significant enough to warrant consideration of the double-duty actuators in an effort to reduce the multiplier 120 to a less expensive 60. The considerations outlined above can reduce printer electronics costs (uninstalled and untested) by as much as $1,200 per manufactured unit.

The interface of printer control with the memory requires two 20 Megacycle lines (read and write) and one 3 Megacycle line controlling writing. In addition the print control unit must obtain from the memory address counter enough information to control its scanning cycles. Comparison outputs occur at (approximately) a 3 Megacycle rate. It will be desirable to mount all print hammer storage (shift register) in the proximity of the print hammers, rather than handle 120 wires on the floor.

Manual Controls other than those required to get the mechanical device moving are restricted to Start and Stop Print, Advance to Head of Form and Print one Line (forced print cycle - for printer adjustment while processing).

Card Reader and Control:

Without question an end-fed card reader is easier and cheaper to control than a broadside-fed reader. However, no end fed reader provides punching facility and few applications are envisaged in which punching (at least of program material) does not occur in the presence of card input. Holley Carburetor has described work on an end-fed (static) punch at 600 cards per minute, but this would probably not be available (if at all) for some time.

The system described here can operate with any type of reader or punch provided the appropriate control device goes with it. In view of the remarks above, all estimating has been done assuming the IBM 1402 Reader-Punch unit as the terminal. The 1402 does have the pleasant system feature of stacker selection.

The principal costs associated with this type of reader lie in the need to interrogate 80 brushes - and because they are brushes - an additional 80 must be read for error detection.

Program control is established via address ØØØ and character addresses 1 through 80 are reserved for the card image. Delivery to address zero of a 1,2,4 instructs the card reader control to read another card and deliver the previously read card to the selected stacker 1, 2, or 3. The card image area will remain intact during the acceleration period of the card. No interlock whatsoever exists for this time and it is provided for the programmer who wishes to use this area of memory for working storage during the read cycle. On receipt of the 9 row of the card, the reader control delivers to the image area partial information about the card and destroys the information previously found there. Having passed the R row of the card, the 6-bit image (corrected for Honeywell code) is complete and the card reader control delivers a code to register Ø indicating completion of the card read. In case of error detected, a special code is delivered to zero and the program can take appropriate action (stacker selection). A card read character can be set during deceleration time - allowing the image area to be used as a working area for 31 milliseconds per card.

A second card read station (available on the 1402) is utilized for card read error detection by a hole count accumulation module 128. No illegal punch detection is specified nor is there any binary card option. All 64 possible 6-bit codes are obtainable from the card and can be punched.

Manual controls on the card reader must allow the card read cycle to occur independently of program demand. Bootstraping procedures for program loading (from cards - if available) are facilitated by forcing a card read with processor off. A manually executed transfer of control to the card read area will then complete bootstrap and program loading.

Card Punch:

System (and cost) considerations here are related to printer control. All 64 six-bit codes must be capable of being punched, however.

Magnetic Tape:

The addition of a tape control unit will allow communication to and from magnetic tape to occur simultaneously with all other activity. The tape control is, however, forced to read into one and only one memory line thereby limiting record size on tape to under 500 characters.

Communication with the tape control unit (if present) is via a fixed address in memory. Two characters of control storage are required; the first character selecting one of 5 tape mechanisms and determining mode of operation of the addressed device. Control is established for Forward Reading, Writing, Backspacing, and Rewind. Only one device may be selected by any one tape control unit at any one time. Thus, if simultaneous reading and writing are to occur, two (or more) tape control units must appear in the system. The second control character selects the hundreds digit from which (or into which) reading (writing) is to occur. Writing from memory continues until a sequence of stereotyped end-record characters (e.g., # # # #) is encountered. Reading from tape into memory continues until end-record is detected or address 500, whichever is first.

Although all information in storage is on a 6-bit plus parity basis, communication with H-800 compatible tape requires the tape control unit to assemble 8 information bits per frame and write an independent frame parity. On reading, frame parity is checked, but parity is recomputed for delivery to memory on a 6-bit basis. A tape frame error will result in the delivery of an appropriate control character to storage on completion of the read.

General operation of a tape read involves the use of an additional delay line to assemble a piece of each tape record. That this is needed is apparent for the frame rate from magnetic tape is such that a 175 usec. memory cannot be accessed per frame and additional storage must be provided to effect gear shifting. A single delay line of length 17.5 usecs. (50 characters) provides sufficient storage to couple to both a main memory line and a standard density H-800 tape.

Operation is as follows: Each frame delivered from tape is judiciously shifted into this intermediate store. At a nominal rate of 21 usec. between frames, sufficient time is available to file each frame into the intermediate delay line. Two frames of storage (8 bits each) are required at the tape interface, one assembling the tape frame and synchronized by the tape clock with transfers occurring into the secondary frame. Information is shifted out of the secondary frame at a 20 Megacycle rate into intermediate storage. Since within the tolerance of the 80h tape transport, two frames may be as little as 15 usec. apart, the situation seemingly arrives where access to a 17.5 usec. store may not be made. Even in the worst case, however, an early frame must be followed by a late frame and sufficient time is available to file from secondard frame storage, reload from primary frame storage (assembly) and continue.

When the intermediate buffer is exactly half full, the control unit references main memory to discharge this information. It must therefore be able to access a 175 usec. loop before its remaining half will be filled. With a nominal frame time of 21 usec., the remaining 25 characters will be filled by $\left(\frac{6}{8} \cdot 25\right) = 18$ 8-bit frames in no less than 18.21 = 378 microseconds, ample time to access the main line.

A major responsibility of the tape control is handling the H-800 tape format, affectionately known as kangaroo. The additional cost burden imposed by the need to unscramble the tape resides in the control of shifting out of secondary frame storage. As mentioned earlier, the 8-bit frame yields 1 1/3 characters in memory

and a consequent disparity between frame parity and memory parity. The result is that all orthotronic control information is lost to the processor. The possibility of prefacing the buffer with 2 H-800 words of storage and control to enable ortho-correction within the tape control has not been fully explored.

In order to account for certain properties of the H-803 control unit, namely its inability to handle frames other than 6 at a time, a programming restriction fixing record size as a multiple of 8 characters is imposed. The control unit will compute and append longitudinal parity and in general will do whatever is necessary to prepare a tape that will read without introducing control errors at an H-800 (or H-400).

Suggested alternative means of tape buffering which should be explored in detail prior to actual design include one system in which frames are dumped into a delay line store as above, but removed as early as possible. This allows the use of a shorter line, thereby making high-density tape input possible, but also increases control costs by requiring full, instead of partial, address storage and a more elaborate comparison mechanism. A third method is to fractionate the buffer into many shorter lines holding one (or very few) characters and constructed from coax delay line elements. This eliminates the access problem, but is potentially more difficult to control.

Writing tape from memory is essentially a reverse procedure to the above, but in any one control unit cannot be simultaneous with reading (or backspacing) on any drive controlled by the same unit. Simultaneity of reading and writing can be done only by adding another full control unit. In this case it is recommended that each control unit have its own line. For any system configuration tape reading or writing will be coincident with processing. Indeed, under program control, processing can be performed on information as it arrives.

One further comment having to do with the relationship of the processor to tape. This system is heavily character oriented and can easily process alphanumeric information. Conversion from binary or BCD to alphanumeric, although possible, is not a pleasant job within either the processor or the program. As a result, the programming and storage burden will grow with the amount of non-alphabetic information to be handled.

Communications Systems and Miscellaneous Peripheral Devices:

Almost all considerations have been directed toward problems of card readers, punches, line printers, magnetic tape control, and processors. As indicated earlier, there is no reason to believe that there is any difficulty in establishing suitable control of other devices.

Several are of immediate potential interest and will be briefly discussed.

1. Communications Control via telephone imposes trivial access demands on the processor. With no buffering required other than to establish synchrony we can handle characters at over a 5 kilocycle rate -- far in excess of the

capabilities of the telephone line. The interface with a Dataphone subset can be relatively simple, and unlike the 1401-1009 combination the full processing power of the computer is available during transmission each way.

In this fashion a minimal processor and memory seems reasonable at a remote terminal, controlling that equipment (and no more) required to process locally and/or talk on the telephone.

At a local terminus a more elaborate version (storage increase) can handle a substantial amount of traffic and indeed can couple on an off the shelf basis to several Dataphone subsets. The local repository of telephone information can be whatever is desired - tape, cards, print or larger computer. We would then not be bound to a fixed, say, tape to tape system but could offer a variety of systems. Unlike the 1401 we would not be bound to the card reader (and could get program material in from a cheaper source).

An interface with the H-800 (or H-400) would consist of little more than a word sized buffer. Indeed it almost exists in the current RTCU design.

A direct memory to memory connection between two small systems can be easily established with one character of information storage and one address stored plus comparisons and control. A cheaper alternative would force both systems to be completely synchronized. This is essentially the expanded system with two processors operating off of one clock (Figure 2).

Other devices such as optical scanners, Paper Tape readers, etc. having a character delivery rate under 5KC will have no difficulty in operating into or out of memory. The general pattern for each such device is that it will require a frame of serial or serializable storage, a partial address counter, logic for detection of a control address of interest, one bit to decide if a demand has been made, additional storage to discriminate between possible actions such as stacker selection, a bit to control writing a release indication, moderate control for interrelating these small pieces and finally, the special circuits and logic indigenous to the device being controlled.

Hardware & Maintenance:

To anyone that has progressed this far it should be clear that the system proposed is modular without limit. Its actual realization would desirably be the same - and could be if all devices occupied zero volume.

Many conflicting conditions are present in the physical realization, a few of which are that the peripheral devices are bulky and rather noisy (acoustically and electrically); that 20 megacycle and 3 megacycle lines can not be long; that equipment then wants to huddle in small floor space; that equipment requires access for repair and adjustment; cables are expensive and should not be stepped on; false floors are expensive; maintenance and operator controls are minimal.

Preliminary hardware considerations indicate that the processor and memory can be housed in approximately the volume occuppied by one H-800 bay of equipment or one 19 inch rack. This volumetric estimate is merely for reference purposes and does not necessarily reflect the appearance of the unit. From many points of view it could be desirable for the hardware layout to follow the general form of the system design - with a plan view of an installation reflecting the figures on Page 2.

The presence of 20 megacycle signals will no doubt require new packaging and mounting hardware, with modularity raising an additional complication and tending to force as much as possible of peripheral control into the terminal unit itself.

Power must be supplied as cheaply as possible, with no power central of H-800 or F-400 nature at all tolerable. Neither the user nor ourselves must be burdened with high installation costs.

If this device is to be sold as an independent or remote satellite unit, the problem of maintenance becomes extraordinarily important. Although there is obviously no reason to believe that 20 megacycles dynamic logic will be easy to debug, we do have the factor in our favor that the total number of active components in the system is low. This can only result (with good basic design) in an increased mean time between failure for the electronics. The electromechanical equipment cannot be wrapped in cellophane and will probably form the largest part of the maintenance problem. No firm solution is proposed at this time except to say that the design must be very carefully executed with maintenance in mind at all times.