Control Unit System

By THOMAS E. BROWNE, THOMAS M. QUINN, WING N. TOY and JOHN E. YATES

(Manuscript received March 7, 1969)

The control unit system consists of the combination of elements which direct and control the operation and functions of the system. It includes the stores and processing units as well as the elements of the man-machine interface. This paper describes the organization of these units, the processor order structure, and some rationale as to why certain design choices were made.

I. INTRODUCTION

The goal of this paper is to provide a general view of the organization and structure of the No. 2 ESS control unit. An attempt is made to emphasize the background and philosophy behind some of the design choices and their relation to the broad requirements of an economical, flexible, and reliable stored program control system.

1.1 *Definition of Control Unit System*

The role of the control unit is to direct the processing of calls and maintenance in the system. It is comprised of the central processing unit, the stores, and the interface to the peripheral units. Two control units and the maintenance and administration center are referred to as the No. 2 ESS control complex. The various subunits which form a control unit include the program control, the input-output control, the call store, and the program store.

The program control is the central processing unit which executes the instructions read from the program store. The input-output control performs the function of interfacing with the peripheral units by providing circuits such as a central pulse distributor, address bus drivers, and scan answer bus receivers. In addition, the input-output control autonomously performs certain processing functions associated with high rate, time critical, peripheral communication. The call store is

the variable memory used to hold information of a transient nature and also that which must be easily modified. The program store is a read-only memory which contains the operational programs as well as parameter and translation data. Physically, the program control, input-output control, and up to 16,384 words of call store are contained in one double bay frame called the central processor frame. Up to four 16,384-word modules are contained in a program store frame (65,536 words), four of these frames may be addressed by the system. A companion article in this issue has photographs of these frames.[1]

1.2 *Design Objectives and Their Influence*

The control unit was designed to control a small, low cost, electronic central office. Much of the economy was realized by the program and call store word sizes which, in turn, determine the size of various registers and gating paths in the processor. In addition, the use of a "micro" type of instruction repertoire allows the multiple use of logic elements for many of the instructions and the efficient use of the program store.

The command structure is somewhat tailored to the progress mark approach of call processing with efficient means of handling information contained in call registers or records.[2] However, the majority of instructions are general purpose, simple, and compact. Hardware check circuits are judiciously used in areas that are difficult if not impossible to check by program and where immediate detection is important. A high degree of internal access is provided for fault detection and diagnostic purposes. Much of this access is common to call processing as well as maintenance.

Some of these economies and the rather serial manner in which processing is performed sacrifice real-time capacity. A fairly modest investment in input-output wired logic minimizes the sacrifice by relieving the program of performing some of the highly repetitive simple functions and quite significantly avoiding the burden of a more frequent input-output interrupt with its associated overhead. This design resulted in a small enough quantity of hardware that an entire control unit forms one switchable system block, thus avoiding the expense and complexity of providing communication paths and control for switching subunits within the control unit.

II. GENERAL ORGANIZATION

Figure 1 shows the No. 2 ESS control complex which consists of two control units and the maintenance and administration center. The

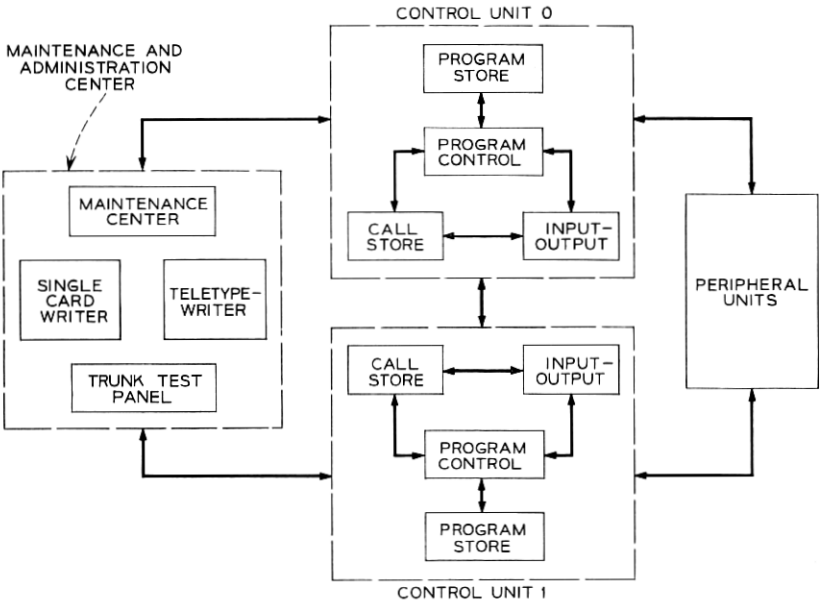

Fig. 1 — No. 2 ESS control complex.

control unit is duplicated and normally running in synchronism with its mate. One control unit activates the peripheral equipment and is called the ACTIVE or more descriptively, the "on-line" control unit. The control units are matched at the maintenance center. The call store is shared by both the program control and input-output control within a control unit. Each control unit has access to only one peripheral communication link but each link provides access to the duplicated controllers of the peripheral unit frames.[3]

The subunits are defined in Fig. 1 and are described in detail later in this article. The bus system is considered part of the control unit and a fault in the active bus will require switching of control units. Communication between control units is for maintenance purposes as in the communication between the control units and maintenance center.

The savings involved by avoiding the interswitching of subunits within the control unit has already been mentioned. Additional savings result from the compact nature of the design in that simple direct wire (dc) communication is possible among all units in the central processor frame, between the pair of control units, and between the control units and the maintenance center. The need for

transformer-coupled (ac) drivers, receivers, and receiving registers is avoided in these instances. However, the nature of the program store signals, the store's physical size, and the growth needs, dictate the use of transformer-coupled paths for the address and read out signals between the program control and program store. This is not a severe economic penalty since there are only about 20 pairs involved in each direction. For the most part, the connections between major units are handled via plug-in cables. This connectorization of blocks leads to advantages in testing, installation, and growth and is covered in detail in a separate article in this issue.[1]

III. LOGIC CIRCUITS AND STORES

3.1 *Building Block Logic Circuits*

The logic circuits of the control unit are fabricated using the transistor-resistor NOR logic gate of Fig. 2 as a basic building block. Constructed with high-speed silicon transistors and tantalum-nitride thin film resistors as described in a companion article in this issue,[1] this gate, with a typical propagation delay of 35 nanoseconds, competes favorably with other available discrete component logic forms from an economic point of view. To provide building blocks with medium and high fanout capability, and a binary counter-shift register function, variations on the basic gate have been developed. These are shown in Fig. 3.

The fan-out of the basic gate is three and the fan-in is two. By connecting the collectors of a number of transistors to a common load resistor, a gate with greater fan-in (maximum of 40) can be formed, although not without suffering some loss of speed. The margin of this gate against false turn-on is determined primarily by the difference between the saturated collector-emitter voltage and the conduction threshold of the base-emitter junction of the transistor. In the worst case, this is only about 200 mV, and this imposes a constraint

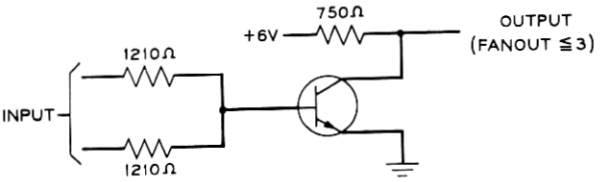

Fig. 2 — Transistor-resistor logic gate.

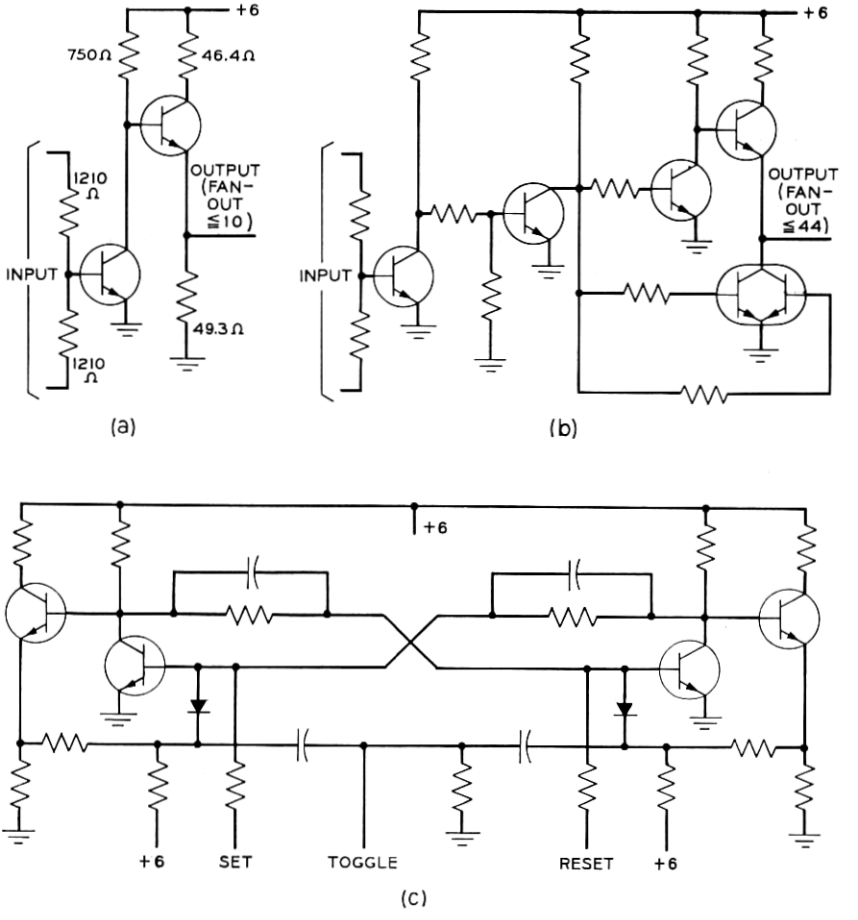

Fig. 3 — (a) Intermediate fan-out gate, (b) high fan-out gate, and (c) binary counter-shift register.

on the backplane wiring of the machine. The wiring is also constrained by the need to avoid excessive negative noise pulses and ringing on signal leads since these may cause "on" transistors to turn off falsely. In some cases, "catching diodes" and terminating resistors have been connected on the far end of long signal leads to prevent negative undershoots from turning off gates. In the development of the No. 2 ESS, these various constraints have been evaluated and wiring rules formulated to assure reliable performance of the circuits over the range of operating voltages and environmental conditions.

3.2 *Call Store Circuit*

The call store circuit provides the temporary information storage required in the control unit. It is a random-access memory of 8192 words of 16 bits each and operates with a destructive read-write cycle time of 6.0 microseconds.

3.2.1 *Memory Device*

The basic memory device in the call store circuit is a 1-inch square ferrite sheet approximately 30 mils thick, having a 16-by-16 array of 25 mil diameter holes located on 50 mil centers. [4,5] This is the same device that is used in the No. 1 ESS call store circuit.[6] Two hundred fifty-six sheets are wired together to form a 4-wire coincident current memory module of 4096 sixteen-bit words. Two wires are used for access current (X and Y), a third (Z) for inhibit current, and the fourth is used for sensing the readout signal.

The wiring pattern used in this construction is designed to maximize the readout signal-to-noise ratio by minimizing the coupling of access circuit transients into the readout circuit and by canceling the delta noise along the readout signal wire. Signal-to-noise ratio is further controlled by time-staggering the application of drive currents in reading the memory. The sheet material characteristics are such as to allow a 1.90 microsecond destructive read access time and total cycle time of 6.0 microseconds with coordinate access currents of 280 mA and 0.4 microsecond rise times.

3.2.2 *Call Store Operation*

Figure 4 is a block diagram of the call store circuit. Address and input data signals are brought to the store by dc connections to the call store address and call store input registers in the program control circuit. Output data from the store is delivered by dc connections to either the general register in the program control circuit or the digit-data output register in the input-output circuit depending on which of two kinds of read command is being executed. Because of these dc connections, no registers are needed in the call store itself, and none are shown in Fig. 4.

The X and Y access windings are the crosspoints of bilateral voltage selection matrices in each of which one horizontal and one vertical coordinate is selected by the address information presented to the store at the time of a read or write command. This selection is made by the address decoders and bilateral switches under control of the

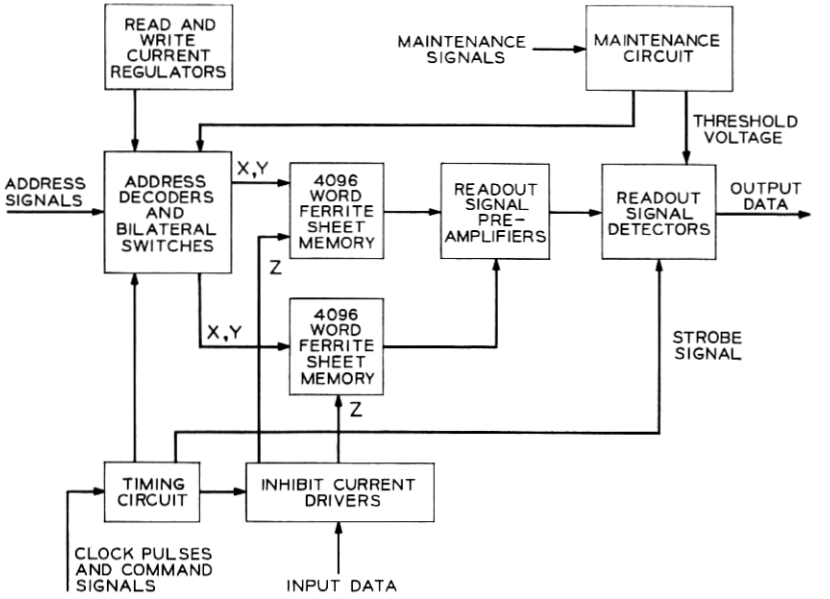

Fig. 4 — Call store block diagram.

timing circuit shown in Fig. 4. Access current flowing through the switches and selection matrix is controlled in both amplitude and rise and fall times by the read and write current regulators. Output signals from the ferrite sheet memories are amplified and discriminated during a specific time interval under control of a strobe signal generated by the timing circuit. This strobe signal is inhibited by the program control on "call store write" commands when the information being read is to be destroyed.

Following the read operation, the access currents reverse polarity to accomplish writing. This will switch the ferrite material at the selected holes to the "one" state except at those holes where inhibit current is applied. This is done on the inhibit windings of those bits which are to be zero, thereby cancelling the effect of half of the access current. The access currents persist long enough to switch all of the material surrounding the selected holes to the "one" state in the absence of inhibit current. Following the write operation, a post-write disturb current is applied in the read direction on the inhibit windings in order to adjust the magnetic state of the ferrite material to achieve minimum delta noise.

3.2.3 *Maintenance*

In Fig. 4, a "maintenance circuit" is shown. This circuit monitors the operation of the access circuitry to insure that on each store cycle the access currents flow in the correct direction at the correct time and that exactly two of the bilateral switches on both the X and the Y selection matrix operates. On any one cycle, if all of these conditions are not met, a "call store error" signal is sent to the program control and maintenance center circuits, indicating a fault in the access circuits of the store. In order to insure the integrity of these checking circuits, maintenance signals are brought into the store which logically simulate the access failures that should be detected by the check circuits.

Other maintenance signals which cause the store to write all 1's or all 0's at a particular address can be activated by the maintenance programs to facilitate diagnosis of faults in either the store or the circuits to which it connects. Finally, there is a third set of maintenance signals by which the threshold voltage of the sense amplifiers may be varied, either manually or by program, to evaluate the operating margins of the store.

3.2.4 *Growth*

In order to achieve minimum cost in the smaller office sizes, the call store has been designed to provide only 4096 words by eliminating one of the memory modules. Furthermore, the physical design of the call store and the program control has included connectorization to allow for growth of up to four 8192-word stores, any of which may be partially equipped so that the call store may be any size from 4096 to 32,768 words in 4096-word increments. The smallest office is expected to use 8192 words.

3.3 *Program Store*

The program store circuit provides memory for semipermanent information in the No. 2 ESS control unit. This information is the program and translation data used by the system. It is called semipermanent because it is fixed and independent of the calls being processed by the system at any given time. However, as the services provided by the system are changed, or as lines and trunks are changed, the program and translation data must change. The store is a random-access read-only memory of 65,536 words of 22 bits each and operates with a 2.5 microsecond access time and a 6.0 microsecond

cycle time. Each half of the control unit may be equipped with up to four stores (in the largest office), and each store may be partially equipped to provide as few as 16,384 words.

3.3.1 *Memory Device*

The program store uses the permanent magnet twistor memory for the basic storage device, in a fashion similar to the way it is used in the No. 1 ESS program store.[7] Information is stored in the state of small magnets affixed to aluminum cards, and the state of the magnets is sensed by the twistor wire. A detailed description of the actual read-out process has been published in a number of earlier papers. The present store differs from No. 1 ESS in its organization, maintenance features, and capability for growth. These aspects of the store are described in the following paragraphs. As in the No. 1 ESS store, information is changed by withdrawing the magnet cards from the twistor module and changing the state of the magnets by means of a special writing mechanism. The operational aspects of this feature are described in some detail in Section VI.

3.3.2 *Program Store Operation*

Figure 5 is a block diagram of the store. Because of its physical size and the cable lengths that would be encountered when four stores are provided in an office, the store is connected to the rest of the control unit by way of an ac bus system. Thus, within the store, there is an 18-bit address register which is cleared and filled by the signals on the address bus. All stores receive the address information and a "read" command signal from the bus and the store identified by the two most significant bits of the address responds on the answer bus. A store contains up to four connectorized memory units, each of which contains a twistor memory module and its associated address decoding and high current access circuitry. Each unit stores 16,384 22-bit words in the form of 4096 88-bit words. Readout signals from corresponding bits of the four memory units are paralleled through a resistive impedance matching network and amplified in 88 readout amplifiers.

On the basis of two of the address bits, a one-out-of-four selection is made on the 88-bit word to produce a 22-bit word which is transmitted to the program control on the answer bus. Since, on the basis of the address, only one memory unit is active at any time, the access current regulation, timing control, readout circuits, and answer cable drivers are common to the four memory units.

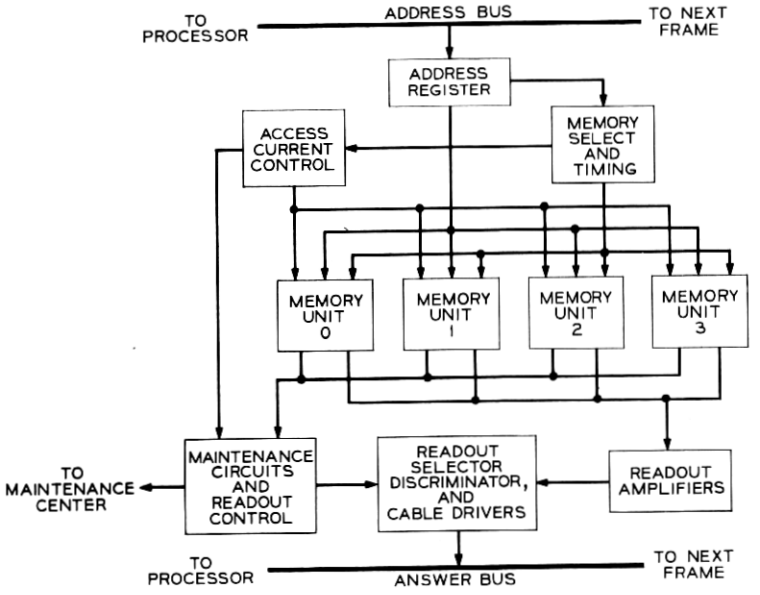

Fig. 5 — No. 2 ESS program store.

3.3.3 *Maintenance*

In order to provide rapid detection of access circuit failures, the program store includes checking circuits to verify the correct operation of the store on every read cycle. These circuits verify the correct flow of module access currents and the correct selection of a unique current path through the selected module. In the event that these conditions are not met in a given read cycle, a "program store error" signal is sent to the program control and maintenance center circuits and certain bits of the readout information sent on the answer bus are forced to be "ones," depending on which of the check circuits detected the failure conditon. This provides for fast resolution of a failure to specific circuit packages by the diagnostic program. All that is required is that the program read a word which contains all "zeros." As in the call store, there are dc control signals brought to the program store from the maintenance center to simulate the various kinds of failures that the check circuits are designed to detect. These signals are used to verify the integrity of the check circuits and may also be used in resolving certain kinds of failures.

There is also a set of maintenance signals which provide for varying

the sense amplifier threshold voltage either manually or by program to obtain a measure of the operating margin of the store. There are two discrete amounts by which the threshold can be varied by the program from its nominal position. These are called "inner margins" (closest to nominal) and "outer margins." A store which works properly with the threshold voltage at the "outer margins" is presumed to be in good health and to contain well written magnet cards.

A store which fails to work with outer margins applied but does work with the inner margins applied is presumed to contain either an incipient hardware fault or poorly written cards. If replacing the cards with a well written set does not enable the store to operate with outer margins applied, then the existence of an incipient failure may be concluded. If a store fails to work when "inner margins" are applied, then a hard fault is presumed to exist in the store. Thus, by virtue of the access checking circuits and the program controlled maintenance signals, hardware failures and magnet card writing errors can be quickly and accurately detected and diagnosed.

3.3.4 *Growth*

To provide orderly simple growth procedures, as well as to minimize installation and repair time, the physical design of the program store uses connectors at both the frame and unit level. This means that a frame can be added to an office by simply bolting it to the floor, connecting power, and connecting it to the adjacent frame by plug-in connectors. Furthermore, a memory unit can be added to an existing frame (either as growth or replacement of a failed unit) by simply bolting it in place in the frame and plugging umbilical cables from the unit into matching connectors in the frame. As a result, a given office can have from 16,384 to 262,144 words of storage in 16,384-word increments. The estimated time to add an increment to an existing frame or to add a frame is less than one hour.

IV. ORDER STRUCTURE

The No. 2 ESS order structure was designed with emphasis on data processing rather than arithmetic operations. The basic 22-bit instruction words, shown in Fig. 6, are divided into two types. The type 1 contains one instruction with a 5-bit operation code, 16-bit address, and a transfer check bit. This check bit is used for detecting illegal transfers and has proved to be quite useful in indicating hardware faults as well as program faults resulting from bugs or data mutilation. The instructions belonging to the type 1 class are rather few.

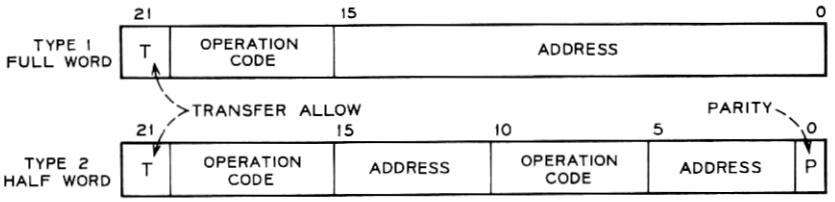

Fig. 6 — No. 2 ESS instruction formats.

They are used primarily for absolute program transfers and for supplying constants and masks for various functions.

The type 2 word consists of two 10-bit instructions, each with a 5-bit operation code and a 5-bit address. As for the remaining two bits, one is assigned as a transfer check bit and the other as a parity check bit. The 5-bit address is used to denote a value or a modifier. For example, a value associated with a rotate instruction specifies the amount of rotation. A modifier associated with the gating operation specifies the source and destination register combination. In many cases, the combination of the two 5-bit fields can be more appropriately considered as a 10-bit operation code rather than the division just outlined. In addition, the 10-bit operation can be assigned as a full-word instruction and the second 10-bit field is available for option bits, modifiers, addresses, or other pertinent data. The majority of instructions fall into the simple type 2 category and provide for efficient program storage use.

4.1 *Clock and Cycle Timing*

The operation of logic circuitry within the central processor is generally synchronous, whereby the clock circuit provides all the basic timing signals. These signals define a machine cycle and phases within that cycle. The duration of a machine cycle is primarily determined by the rate at which the instruction can be fetched from the program store. With the type 2 order format of two instructions per program word, a machine cycle is set equal to one-half of the program store access time, or 3 microseconds. The number of clock phases is chosen to permit many of the short word instructions to be implemented in one cycle, thereby making it possible to process two short word instructions while the next word is being fetched from the program store. This maximizes the average data processing rate of the system.

A machine cycle consists of eight 750-nanosecond clock phases with

each pulse overlapping the preceding and succeeding pulses by one-half their width (375 nanoseconds). This arrangement provides an adequate number of intervals to fully realize the data processing steps of the system in response to a diversity of instructions. At the same time, it is consistent with the speed of the transistor-resistor logic gate. The basic timing signal is provided by a 2-⅔ MHz crystal-controlled oscillator with the output from the on-line oscillator driving the clock in the off-line processor. In this way, synchronism between the processors is maintained.

The clock circuit provides timing signals throughout the entire system. These signals coordinate the various activities as dictated by the program sequence. Improper timing not only causes errors in the system, but causes them in such a manner that they are extremely difficult to diagnose. For this reason, error detection circuits have been incorporated into the clock. Furthermore, accessibility is provided to allow ease of program check of the clock circuitry for fault detection and diagnosis.

4.2 *Order Decoding*

The central processor decodes and executes only one instruction at any given time. Most of the orders are relatively simple, that is, setting and clearing various registers and bits, gating data from one register to another, or testing various registers and bits. The decoding of some simple instructions leads directly to the specified actions without any additional logic circuitry. In the more complex instructions, outputs from the order decoder are combined logically with the appropriate system conditions and clock signals, thus producing the necessary controls required for implementing a given order.

Each order, regardless of its format or complexity, has a 5-bit operation code and is decoded into one lead active out of 32. For most full-word and for some half-word orders, the address portion of the instructions (16-bit or 5-bit) is used directly as an address or a constant. These instructions are concerned mainly with providing direct addresses for program transfers and call store access, loading registers with constants, or providing data for certain logical operations. For the remaining half-word orders, the address and the operation code form a 10-bit binary combination that specifies a unique instruction. These instructions are decoded by two stages of one-out-of-32 decoders (Fig. 7). Eight outputs in the main decoder are dedicated to enabling one of eight auxiliary decoders. Each one decodes the 5-bit address to one lead active out-of-32 basis when it is

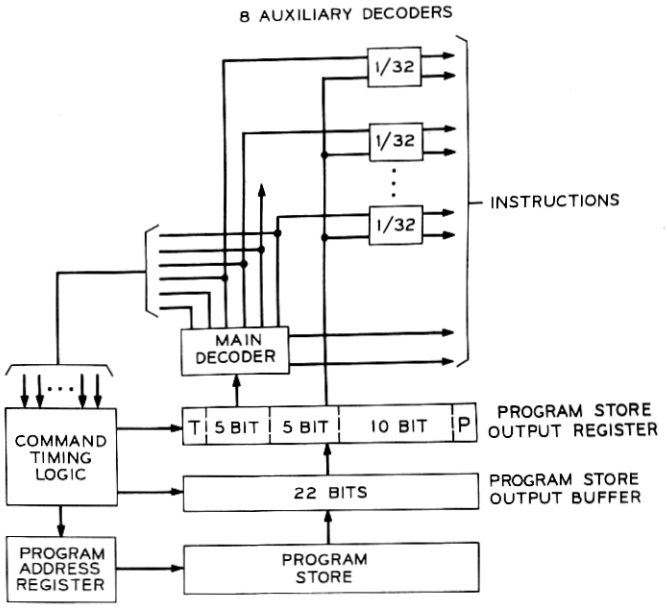

Fig. 7 — Instruction order decoding arrangement.

selected by the output from the main decoder. This arrangement accommodates numerous short and simple instructions.

The decoder outputs extend throughout the system to implement the prescribed actions for each instruction. Outputs for a subset of orders extend to the maintenance center and to the duplicated central processor to permit the on-line machine to communicate with the off-line processor for maintenance purposes. The main decoder outputs also combine logically in the command timing logic to control the flow of instructions from the program store to the decoders.

4.3 *General Description of Order Flow*

The flow of program store information is normally in sequential order with the program address incremented by one each time the store is read. At times, it is necessary to branch from the regular sequences and transfer to a new one. The address will then be changed completely corresponding to a starting point for a new sequence.

The 22-bit output from the program store is first temporarily stored in a flip-flop register. It acts as a buffer for storing the information until the preceding instruction has been fully processed. The

information is then gated to the program store output register for decoding and executing. Immediately thereafter, the program address counter is advanced by 1 and the program store is instructed to read the next sequential word.

The buffering action of the program store output buffer permits the next program word to be read while the instruction in the program store output register is being executed, thus reducing the overall sequencing time. If a transfer should be executed, the revised address, rather than the next sequential location, will be used in addressing the store. However, if the next location has been read, and a transfer should be executed, the contents of the program store output buffer will be discarded. The store is read at the new location.

In many cases, concurrent operation of executing the present instruction and reading the next program word requires no additional time because of transfers. If the word contains two short instructions, the decoder acts on the left one first. After the first instruction has been executed, the second instruction is then gated to the left half of the program store output register for decoding and execution. Upon completion, the new information (the next program word) in the program store output buffer is gated to the program store output register. The instructions are thus stepped along one at a time to the decoder circuit.

The program store can be read once every 6 microseconds. Consequently, most of the half-word instructions are implemented in 3 microseconds to allow the program store to operate at its maximum rate. The full-word instructions are allotted a minimum of 6 microseconds since the next word is not available for that length of time.

4.4 *Data Transfer and Register Functions*

The central processor, as shown in Fig. 8, contains a large number of flip-flop registers in addition to the data processing logic. In general, the content of any one register can be gated to any other register in the system. The transfer of information is done by means of the common bus designated as the program gating bus. Most of the registers are 16 bits wide, with two sets of gates (input and output) associated with each. The corresponding bits from the output gates are ORed to form the common bus outputs, fanning out to all parts of the system.

Normally, all the input and all the output gates are disabled by their respective control signals. In transferring information from one

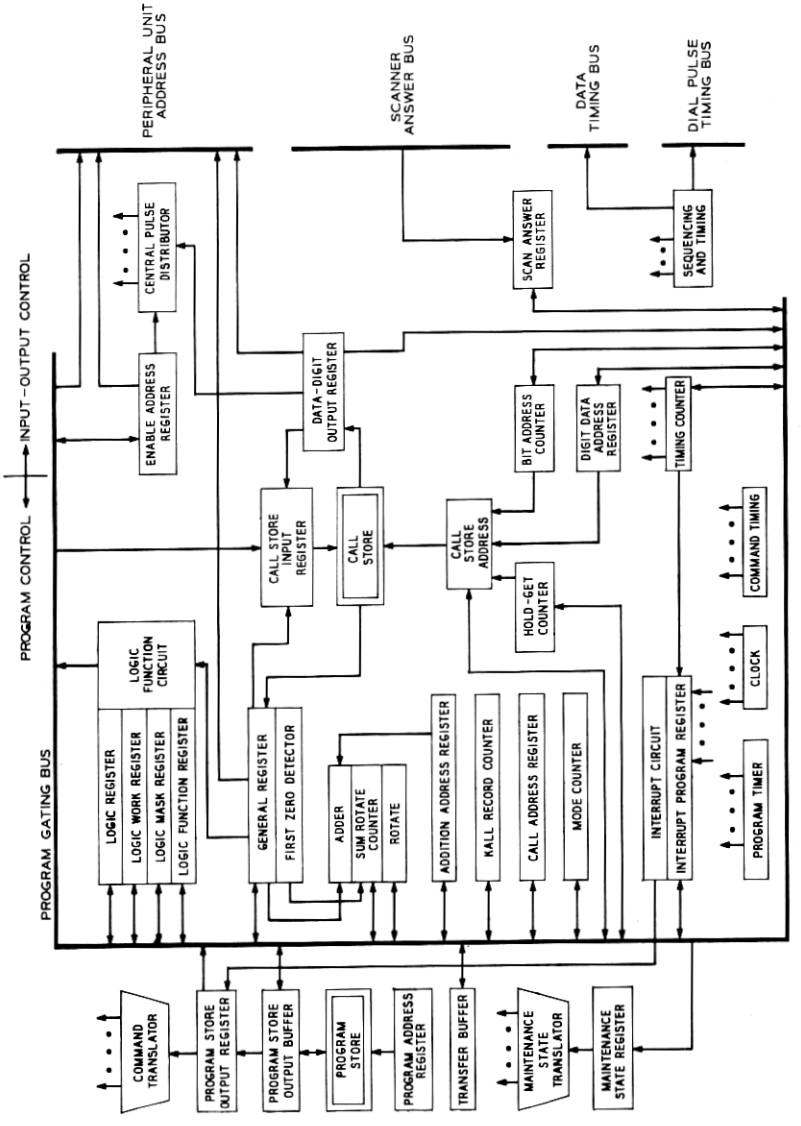

Fig. 8 — No. 2 ESS call processor block diagram.

register to another, the output gates from the first are enabled to the gating bus and the input gates to the second are enabled from the bus. The use of a common bus is an excellent method of transmitting data signals within the processor, keeping the number of gates and the number of long leads to a minimum. The bus outputs extend to the input-output circuit, the duplicated control unit, and the maintenance center to allow data transfers via the bus among the circuits under programmed control.

There are several registers associated directly with the operation of the call store. As Section V elaborates, this store is time-shared between the processor and the input-output circuits. The outputs from the store are directed either to the general register in the processor or to the digit-data output register in the input-output depending upon which circuit is accessing the store. In addressing the addition address register or the call address register may be gated to the call store address register for use as an address in reading or writing. The data in the call store input register, normally gated from the general register, is the store input.

The registers mentioned thus far are related mainly to the operation of the call store and the program store. Other functional flip-flop registers are used either with associated logic circuits for data manipulation by the program or are dedicated to various system functions. The data modifications involve such functions as addition, rotation, identification, and marking the least significant zero in a register, and various logic functions of two variables. Some of the specific system functions are concerned with interrupt facilities, subroutine facilities, program timing, and maintenance features.

Adding to the numerous flip-flop registers in the system, is the call store, which is segmented into many registers. Much of the call processing program is concerned with call store registers. A group of eight words is assigned as a register or record for each call in progress. Several instructions are provided for quick access to these records.

4.5 *Transfer and Decision-Making Orders*

Call processing programs tend to be highly decision-oriented. In order to facilitate decision making, a multiplicity of test instructions are provided to check various data words and individual bits of certain data words. Test instructions indicate their result by the state of the condition flip-flop which is later utilized by a conditional transfer instruction. If the specified test condition is satisfied, the condition flip-flop is set to the "1" state; otherwise, it is reset to the "0"

state. Many system functions also affect the state of the condition flip-flop. For example, in adding two binary numbers, the end carry is stored in this flip-flop. Most of the decision making is based the state of the condition flip-flop.

Many conditional transfers involve short jumps, hence coding economy can be realized by the use of a 5-bit address. When a transfer is made, the low order five bits of the current address are modified. This range, however, can be extended by preloading the transfer buffer with the next five higher order bits. In this case, the five bits from the instruction and the five bits from the transfer buffer form the low ten bits of the program store address when a transfer is made. In addition to short transfers, the entire range of the 18-bit program store address is accessible by the long, or full, transfer instruction. The low 16 bits are provided by the instruction while the remaining two bits come from the transfer buffer. Indirect transfers, that is, transfers to an address contained in a system register, are also provided.

4.6 *Data Manipulation Orders*

Data manipulation orders allow the operation on data, contained in system registers, in a particular manner to implement the call processing functions. Logical rather than arithmetic needs are paramount. However, a binary ADD order is included in the instruction repertoire. This allows other arithmetical operations to be implemented rather easily by a software combination of other logical operations.

A very flexible and useful set of orders for data manipulation is the Boolean function of two variables. The general register and the logic register are the two variables which combine logically to form the four minterms ($\bar{G}_n\bar{L}_n$, $\bar{G}_nL_n$, $G_n\bar{L}_n$, G_nL_n) bit by bit. The subscript n represents a unique bit of the general register (G), or logic register (L). The 4-bit logic function register (a, b, c, d) specifies the coefficients of the minterms to give the 16 possible combinations. The Boolean function of two variables represented by:

$$f(G_n , L_n) = a\bar{G}_n\bar{L}_n + b\bar{G}_nL_n + cG_n\bar{L}_n + dG_nL_n$$

generates many useful logical functions. For example, if $a = d = 1$ and $b = c = 0$, the function ($\bar{G}_n\bar{L}_n + G_nL_n$) is used to match or compare two binary numbers. The results are placed in the logic work register and, optionally, also placed in the general register. The logic mask register controls or selects the bits in which the logic function is to be performed.

Most of the data in the call store are continuously undergoing changes. In many instances, the data read out are logically operated on with the result written back into the store. This usually involves a logical operation between two memory cycles. Since the operation of read-change-write occurs rather frequently in the system program, a considerable savings in program words and real-time can be realized by combining all three steps. This is possible within the framework of one memory cycle with no additional timing penalty. More importantly, it eliminates the possibility of having an interrupt program modifying the same word in the midst of read-write-change operation. Therefore, the store interwrite problem is avoided. Figure 9 shows that the data return to the call store input for writing may come directly from the general register or from the logic output via the gating bus or a combination of the two. The combination is a form of insertion masking which permits the bits specified by the logic mask register to be returned unchanged directly from the general register to the call store input register. The other bits are modified according to the content of the logic register and the setting of the logic function register.

Other useful data manipulation functions include rotation of a register's contents and a particular function of detecting and identifying the rightmost "0" in a register. The latter function's implementation by several conventional instructions is rather awkward and time consuming. Therefore, a special order, find low zero test, is provided to facilitate operations such as the selection of idle trunks. The status of each trunk is indicated by a binary bit, a "0" corresponds to the idle condition and a "1" to the busy condition. A word of 16 bits represents a group of 16 trunks with the bit position denoting the trunk number. In the find low zero test instruction, the logic searches for the first zero in the general register, starting with the least significant bit, and successively checks through the higher order bits in sequence until a zero is found. When a zero is detected, the logic automatically ignores the remainder of the bits, sets the selected "0" to "1," records the bit position, and marks the condition flip-flop to "1" to indicate finding an idle trunk.

4.7 *Interrupt Facilities*

A multilevel interrupt mechanism is provided in the central processor to permit a signal to enter at any time and seize control of the system. The interrupt circuit responds to an interrupt request by injecting or replacing the next program word with a program in-

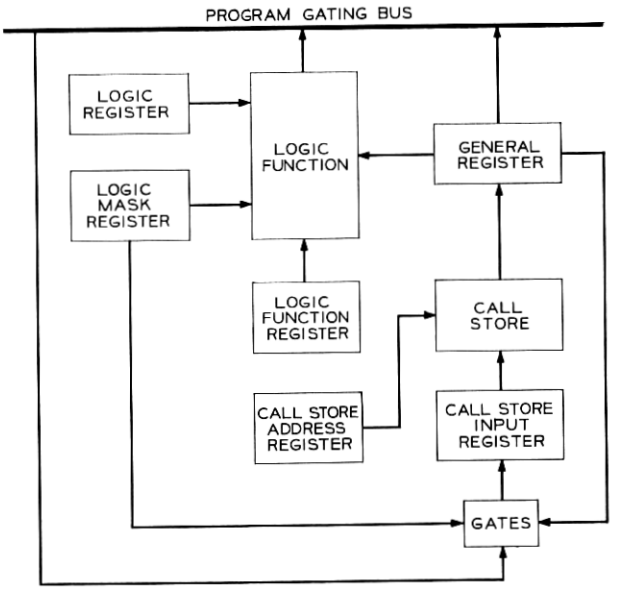

Fig. 9 — Call store operation.

terrupt begin order. The order stores the program address where the program was interrupted, and then the program transfers to the entry point of the appropriate interrupt program. The first chore in the execution of the interrupt program is to store the state of the central processor in memory. At the conclusion of the required work function, the central processor is restored to its initial state. The interrupt program then ends with a program interrupt end order which transfers back to the previous interrupted program.

Presently, interrupts are used for two functions. One is to handle input-output data processing, that is, digit receiving, digit sending, scanning, data sending, and peripheral communication functions. These tasks are scheduled regularly in real time at 25 (or a multiple of 25) millisecond intervals.

The other function is to gain immediate access to one of several maintenance programs. The interrupt program register and logic are arranged in a hierarchy of eight levels to allow the running of the program with preference to the most urgent or highest level present at any time. Only three of the eight interrupt levels have been assigned; two for maintenance and one for input-output. The remaining five are for future growth and other applications of the No. 2 ESS control complex.

4.8 *Subroutine Facilities*

Many of the No. 2 ESS programs consist largely of a series of subroutine calls. This not only has the advantage of saving program store words, but also makes debugging simpler since a subroutine need only be debugged once. A special push-down address arrangement has been implemented and storage allocated in the call store to facilitate entry, nesting, return, and releasing of major system registers for use by subroutines. This arrangement provides 32 groups of eight words within the call store. Each time a new subroutine level is entered by means of a transfer and save address order, the push-down level or the hold-get counter is decremented by one. In execution of the transfer and save address order, the return address is stored within the eight-word block. The remaining words are available for the major system registers. Special HOLD and GET instructions are provided to facilitate storage and retrieval of data between a selected register and the call store. When the subroutine is returned via the transfer to saved address order, the push-down level (hold-get counter) is incremented by one.

4.9 *Orders for Repetitive Functions*

Certain tasks are very repetitive and can be accomplished by a sequence of operations which is repeated a predetermined number of times forming a loop. Each time the program is repeated, different data are encountered. The transfer on index instruction in the No. 2 ESS is used primarily for program loops. An 8-bit transfer on index counter (kall record counter) is used to keep track of the number of loops. In executing the transfer on index instruction, when the kall record counter is not equal to 1, the kall record counter is decremented by 1 and a transfer is made to the beginning of the loop to repeat the given set of operations.

Many of the repetitive functions can be implemented rather easily with program loops using the transfer on index instruction. However, there are some repetitive functions that must be performed very frequently by the system. Although the program to be repeated usually contains only a few instructions (including transfer on index), the program's high usage consumes a considerable amount of system time. A real-time saving can be realized by combining these instructions into a single highly efficient macroinstruction. One order thus performs the actions of several and is repeated as many times as is specified by the contents of the kall record counter unless the data are

being tested by the program and satisfies the exit conditions prior to that. The real-time saving is obtained chiefly by: (*i*) not having to fetch a series of instructions from the program store, (*ii*) not having to perform a transfer at each loop back, and (*iii*) concurrent operation of a set of instructions.

There are several repetitive or macroinstructions used in the No. 2 ESS. Among these are: (*i*) an instruction for performing logic on the contents of a number of words contained in call store, (*ii*) an instruction that scans ferrods and compares the results with call store status words, (*iii*) an instruction for sending serial data to peripheral units, and (*iv*) an instruction that scans call store records and performs indirect transfers to programs that process the records. The last instruction is the ADV (advance) command and is described in the call processing article in this issue.[2] The ferrod scanning instruction as well as the other peripheral unit orders are described in Section 5.4.

4.10 *Maintenance Facilities*

In the duplicated system under normal conditions both the on-line and off-line processors are run in synchronism while handling the input-output data and solving their problems independently. Only the outputs from the on-line processor are permitted to activate the external equipment, such as setting up network connections, outpulsing, and data sending. In this synchronous mode, both call stores contain identical information and both processors are in step, executing the same sequence of instructions. As a detection means, the information being written into the call stores is matched by circuitry in the maintenance center. In addition, a number of error checking circuits are strategically located throughout the system for the detection and diagnosis of faults. They provide:

(*i*) Parity checks of words read from the program store (odd parity for the program words, and even parity for the data words),

(*ii*) Transfer checks to ensure a legal program entry,

(*iii*) Double output checks for more than one output active from the instruction decoders,

(*iv*) Sequence checks of the instruction timing logic,

(*v*) Subroutine level checks for exceeding the number that can be nested,

(*vi*) System clock checks for the proper sequence and rate of the clock output signals,

(*vii*) Simultaneous access checks of the call store by the processor and the input-output control,

(*viii*) Multiple access checks in the call store and the program store,

(*ix*) Timing checks in the input-output control for checking dial pulse and data timing buses, and

(*x*) Input-output error checks involving proper access to and response from peripheral units.

Check circuits are generally provided to cope with faults which are too difficult or too time consuming to detect strictly by program. An example is the double output detector which checks for more than one output active from the instruction decoders. Within the central processor one and only one instruction is decoded and executed at any given time. Faults causing no output at all are relatively simple to detect.

If, however, the instruction decoder produces two simultaneous outputs, an additional extraneous instruction will be performed. This can seriously affect the overall operation of the system and produce obscure errors. Therefore, the double output check circuit is incorporated as part of the processor for quick detection and diagnosis. Except for input-output errors, when a check circuit in the on-line system indicates an error condition exists, the faulty processor is immediately taken off line. Input-output error signals are handled by special "working mode" programs which determine the proper actions to be taken.

As previously indicated, a method of trouble detection available when the control units are running in synchronism is the matching of call store information by a match circuit located in the maintenance center. A mismatch causes a maintenance interrupt program to run detection tests on the on-line control unit. Upon failure, the control unit is immediately switched off-line and stopped while the other control unit is given an initialized start. In some cases, the faulty processor may be incapable of making any decision. Under this condition, the hardware emergency timer will time out and initiate the switching and initiatization functions.

In order to analyze the defective circuit, the on-line processor has direct access to the off-line machine by means of a set of external orders. Simplicity of implementation rather than coding efficiency has been the goal in the design of these orders since the total number of external orders used in the system program is relatively small.

Consequently, each external order is a full word and takes two cycles to execute. The order format consists of a 5-bit operational field and two 5-bit addresses designated as the infield and the outfield. The operation code and the infield form a 10-bit binary combination which specifies the unique operation within the processor. Similarly, the operation code and the outfield specify the unique operation for the other processor. By a proper combination of the functions being performed in each of the two processors, many different operations are possible for communication and information transfer between the two processors and for direct control of the off-line processor. Some of the special orders are exceptionally useful in implementing the maintenance routines of the more difficult logic circuits, such as orders that permit the on-line machine to perform a step-by-step check of the off-line machine, or orders that start and stop the off-line clock to check timing leads.

After the fault has been diagnosed and repaired, the off-line processor must be restored to its normal working mode. In order to do this, the call store must first be updated to contain the same data as the on-line call store. If the complete transfer of the entire store is done by one continuous program, excessive time will be consumed, thereby causing errors in call processing. As a result, the data transfer from the on-line to off-line call store is interleaved with the normal program, moving a small block of words at a time until the entire store is updated. However, the data in the on-line call store is transient and the transferred portion of the store may undergo a continuous change. To ensure that the updated portion of the off-line store remains current, its address and store input data are "slaved" by hardware to the on-line store. When any read or write operation is performed, the same address and input data are used to alter the data in the off-line store. The updating of the call store is thus accomplished by the combination of software and hardware.

4.11 *Means of Checking Error Detection Circuits*

Error detection circuits are checked periodically to ensure their usefulness. When they fail in the error mode, the failures are self-detecting and are recognized immediately. On the other hand, if the failures prevent the circuitry from giving any error indication, faults normally detected will be ignored. Consequently, a periodic program check of error detection circuitry is essential to guarantee the reliable operation of these circuits. As an aid for testing the check circuits and for testing maintenance equipment that is not easily tested by

normal program functions, a maintenance state register and an associated decoder are provided for simulating faults and test conditions. The outputs from the maintenance state decoder are used, for example, to check the instruction double output detector or to enable a programmed margin check of the call store.

4.12 *Emergency Switching of Control Unit*

In any duplicated system, the mechanism for switching controls must be highly reliable. The emergency timer and associated circuitry provide the very vital function of switching the bad control unit off-line and the good control unit on-line. During the process, proper steps are taken to ensure a smooth transition in the transfer of controls.

The timer provides a further protection against faults or errors that cause programs to loop indefinitely or to bypass certain major program functions. The main call processing program is cyclic and returns to the starting point upon completion of all the tasks required by the call processing. The time required to complete one scan varies from scan to scan depending upon the amount of work required of the program. The basic timeout interval is 320 milliseconds. If the program takes longer or does not reset the timer, an equipment or program error has occurred. The timer and its associated logic then produce an output signal to switch controls to the other processor.

There are two timers, one in each processor. The off-line timer is used as "backup" and will take over the task of switching if the on-line timer does not work. Under most error conditions, immediate switching is performed to minimize any detrimental effect on call processing.

V. INPUT-OUTPUT FACILITIES

In the No. 2 ESS, all inputs and outputs between the peripheral units and the central processing unit (program control) are handled by the input-output control. Figure 1 shows the control complex block diagram. The input-output control performs the interface function both by acting as an extension of the program control in providing and controlling the special purpose circuitry required to communicate with the peripheral units under control of program instructions and also providing wired logic for autonomously performing a portion of the digit receiving, digit sending, line scanning, and data sending tasks. By assigning the repetitive high rate functions which must be performed

on a tight schedule to the wired logic, program real-time is conserved by not having to perform these functions as well as not bearing the burden of an input-output interrupt and its associated overhead more frequently than every 25 milliseconds. The division of work between the input-output control and the input-output interrupt program was based on the criterion that only the simple, highly repetitive functions should be performed autonomously. To retain their flexibility, functions are directed by information placed by program in the common call store. In addition, economic justification requires the delegation to wired logic only those functions which return a substantial real-time saving for logic cost since the wired logic not only adds to the cost of the hardware for that function but also adds to the maintenance costs. This cost in borne in maintenance access as well as detection and diagnostic programs. The maintenance costs are often high because of the special purpose nature of the functions that must be tested and the difficulty in controlling autonomous circuitry.

5.1 *The Peripheral Interface*

A characteristic which frequently determines the general usefulness of a control system is its interface with the outside world (referred to here as the periphery). No. 1 ESS has developed an inexpensive yet attractive form of communication utilizing ac bus circuits over simple twisted pairs.[8] The bus frequency response, transmission characteristics, and noise immunity, satisfactorily meet the requirements and goals of the No. 2 ESS. Consequently, the natural conclusion was to design an "ESS 1-compatible" peripheral interface.

For economy and simplicity, the No. 2 ESS peripheral interface provides a limited bus structure with fanout in only two directions and access from only a single control unit. In addition to the buses which are used for parallel data transmission, a pulse distributor allows for private pair communication with particular receivers. Again, simplifications in the pulse distributor organization has resulted in further system economy. Figure 10 shows the peripheral connections.

5.1.1 *Peripheral Buses*

The peripheral interface consists of the peripheral unit address bus, the scan answer bus, and the central pulse distributor. Both of the buses are groups of balanced twisted pairs which are transformer-coupled to and from the communicating circuits. The peripheral unit address bus is the output bus (from the input-output control) while the scan answer bus is the input bus. The bus character-

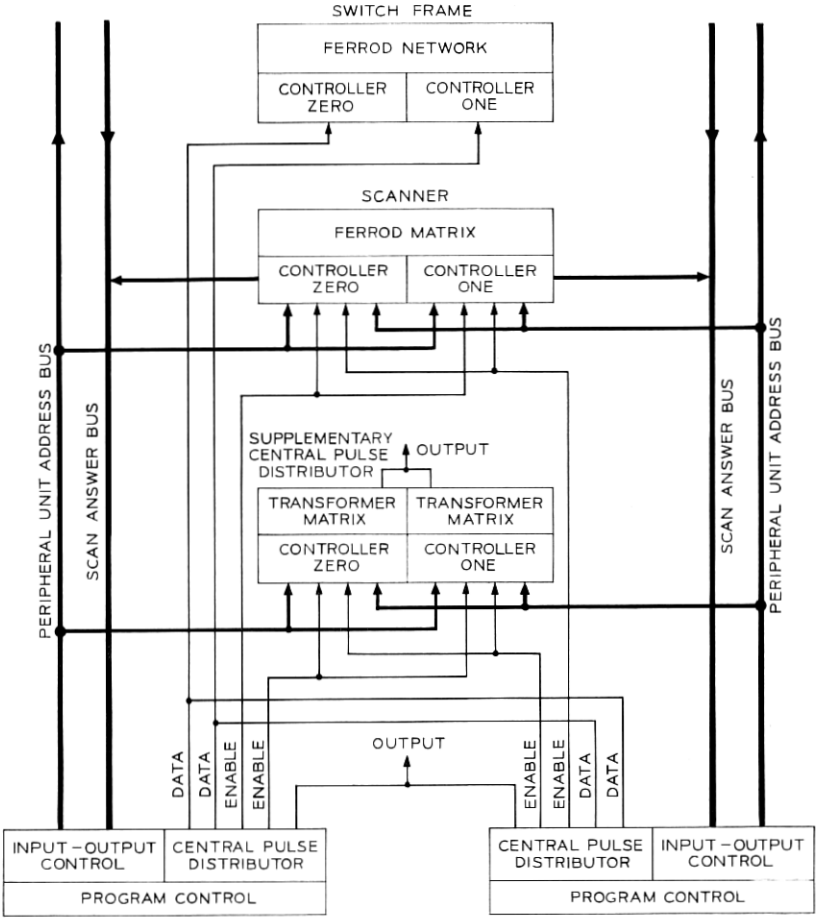

Fig. 10 — Peripheral connections.

istics are identical to those of No. 1 ESS; however, the cable drivers and receivers interface with transistor-resistor logic. Special input-output orders produce particular data formats on the peripheral unit address bus as well as general purpose binary information depending on the specific use.

5.1.2 *Central Pulse Distributor*

The local central pulse distributor is an integral part of the central processor frame and is directly controlled by input-output registers and control signals.

A signal from the central pulse distributor can be used for:

(*i*) Selecting and enabling a peripheral unit to receive information from the output bus (peripheral unit address bus),
(*ii*) Communicating with a shift register device via a stream of positive and negative pulses, and
(*iii*) Setting or resetting a flip-flop in the periphery.

There are a number of special purpose instructions which activate the peripheral interface in various modes which will be described in the next few pages.

The central pulse distributor is a three-stage transformer matrix through which a shaped pulse is steered to one-out-of-512 output pairs. The output is transformer-coupled and may be of either polarity depending on the selection address. The electrical characteristics are similar to those of No. 1 ESS,[9] while the organization and control is specific to No. 2 ESS. Selection of the output is determined by the enable address register for program orders and autonomous line scanning while the digit-data output register performs selection for autonomous digit scanning. Remotely located supplementary central pulse distributors allow growth beyond 512 outputs for high runner items such as trunk circuits.

The general use of the enable address register allows the high 10 bits to select the one-out-of-512 outputs and the polarity. The particular order which is used to simply transmit pulses through the central pulse distributor, as in uses (*ii*) and (*iii*) of Section 5.1, uses the low six bits of the enable address register to determine whether the local or a supplementary central pulse distributor is to produce the output pulse. A supplementary distributor contains a fully duplicated set of control logic and output transformers which provide multiple access to a common output pair, as shown in Fig. 10. Selection of a particular supplementary central pulse distributor and its control logic is determined by the low six bits of the enable address register. If these bits are all zero, the local central pulse distributor is selected. If not, five of the bits pick one of the supplementary distributors and the remaining bit selects the control logic. The system is thus capable of selecting up to 31 supplementary central pulse distributors.

An output pulse from the local central pulse distributor enables the selected supplementary distributor to receive the desired output code from the peripheral unit address bus which contains the upper ten bits of the enable address encoded into $\frac{1}{8}$, $\frac{1}{8}$, $\frac{1}{16}$. This coding allows

direct control of the supplementary distributor matrix as well as error checking to be performed at the supplementary distributor. Since the supplementary central pulse distributor requires use of the peripheral unit address bus to be enabled, it cannot in turn be used for enabling other circuits, but is used to provide independent signaling.

5.2 *Autonomous Work Cycle*

The input-output control performs certain tasks associated with digit receiving and sending, data sending, and line scanning. Since the input-output control requires use of the common call store and peripheral access, it interleaves its work with the program use of this equipment in order to avoid interfering with program execution. Figure 11 contrasts the call store interaction mode with the program interruption mode. Input-output work performed by "cycle stealing" not only saves the program from doing that work but also avoids the substantial overhead required to save and restore the state of the machine at the time of the interrupt. The call store usage priority structure

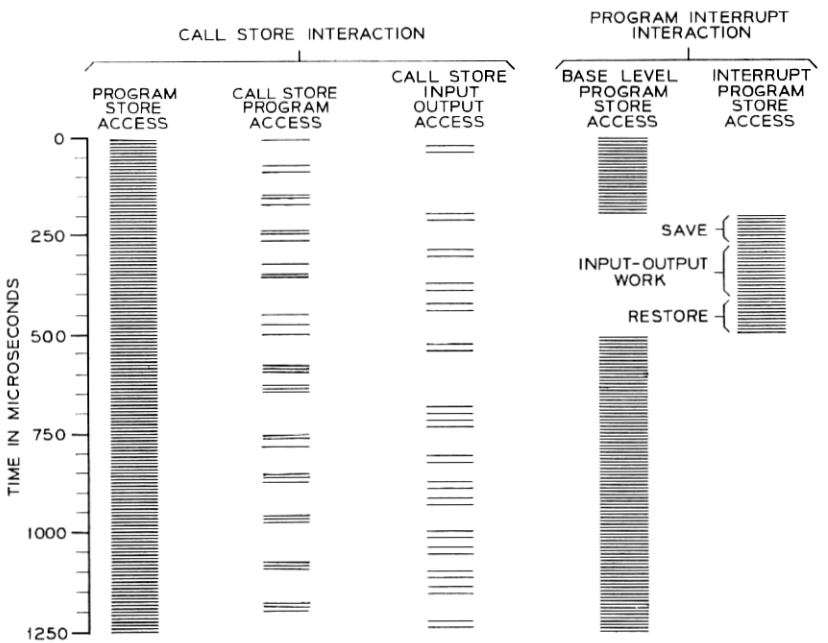

Fig. 11 — No. 2 ESS shared storage.

allows the program first choice, the digit scan portion of the input-output control second choice, and the line scan portion of the input-output control third choice. The input-output control has a mechanism for upping its priority in situations when the program usage has denied the input-output control sufficient access to keep up with its work schedule.

The input-output control is driven by a wired logic schedule which causes the performance of the various input-output tasks at the proper repetition rates. The lowest rate function determines the major repetition rate which, in this case, is dial pulse sending (10 pulses per second), requiring a 100 millisecond rate. A 10 millisecond rate is provided for scanning digit receivers. Division by eight yields the 1.25 millisecond interval which is the basic input-output work cycle used to control work slippage as described in the following paragraphs. The source of timing is the timing counter which is incremented by the system clock. The timing counter also generates the 25 millisecond input-output signal to interrupt the program so that the cooperative functions performed by the input-output interrupt program will be in step with the wired logic functions. The autonomous input-output control requires access to the peripheral unit address bus and central pulse distributor in order to interrogate scanners, and it requires access to the call store for control information, to record inputs, and to get outputs.

The primary function of the autonomous input-output control is digit receiving. Digits are stored in an eight word area of the call store (16 bits per word) called an originating register as shown in Fig. 12. There are a maximum of 128 originating registers, but the program can adjust the number to be used. The input-output control requires 12 microseconds to process an originating register, accessing two call store words and a ferrod scanner during that interval. The originating registers are sequentially scanned every 10 milliseconds with the input-output and program controls competing for use of the call store, peripheral unit buses, and central pulse distributor. The program control normally has priority in the use of this equipment in order to maintain efficient use of program time.

The 10 millisecond scan interval is divided into eight intervals of 1.25 milliseconds, each of which should handle 16 originating registers. This distributes the work load and minimizes the variations in scan rate. The work assigned to each interval is performed as soon as possible in the interval but may be delayed varying amounts because the program control is using the call store or peripheral unit access.

If the input-output control cannot complete the work assigned to a given 1.25 millisecond interval, it will raise the priority of its request for call store and peripheral unit access so that it can become active at the conclusion of the program control command that is denying the input-output control access. The input-output control remains in this force mode until work for that interval has been completed. In this way, the scan interval for originating registers is maintained at 10 milliseconds plus or minus 1.25 milliseconds.

Except when in the forcing mode as just described, once the input-output control determines that the program control instruction is noninterfering, it seizes the store and periphery and has control for 12 microseconds. The program control must now wait until the end of the 12 microseconds if it requires access to the call store or peripheral units. Since instructions may be multiples of 3 microseconds, the program control may have to wait 3, 6, or 9 microseconds. The input-output usage within a 1.25 millisecond interval is only 16 percent of the time. Computer simulations have shown that for a random program control usage of the call store or peripheral units at one-third of the time, the real-time loss resulting from input-output blockage is approximately 2 percent.

The 1.25 millisecond interval provides a convenient data bit rate (800 bits per second) for the data sending function. How data sending is performed and how it fits into the work cycle will become evident as more detail is presented.

5.3 *Input-Output Autonomous Functions and Their Relation to Programs*

5.3.1 *Digit Receiving and Sending*

As mentioned previously, digit receiving is the primary autonomous function of the input-output control but only those functions which require attention every 10 millisecond are performed by this logic. The remaining functions are performed by the input-output 25 millisecond interrupt program and the base level call processing programs. An originating register contains storage for up to 16 digits. Any originating register may be used in combination with a receiver or sender of many different types. Both receiving and sending of digits may be performed at the same time with the same originating register for nearly all combinations of receiver and sender types.

The input-output control reads the first word of the originating register to obtain the scanner address of the receiver ferrods and sends an order to interrogate the scanner. It then reads the second

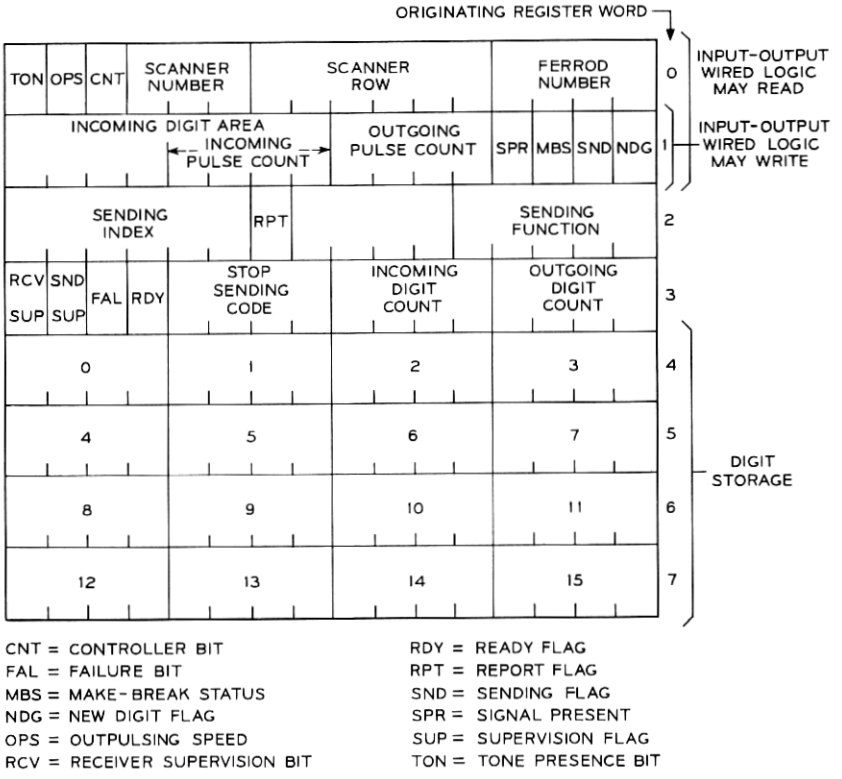

Fig. 12 — Originating register layout.

word of the register and combines this with the scanner reply and some information saved from the first word to determine the modifications to be made to the second word. The modified second word is then written back into the originating register and the input-output control goes on to the next originating register. These actions require two read-write cycles in the call store and a scanner interrogation, all performed in a 12 microsecond input-output cycle. The same cycle time is required for data sending since two 16-bit words must be read from the call store in order to get one bit for each of 32 possible data channels. The data cycle and 16 originating register cycles are allocated to a 1.25 millisecond interval. The combinational logic that modifies the second originating register word and the input-output program interaction combine to perform the various types of digit receiving and sending.

The input-output control is designed mainly to handle dial pulse, multifrequency, or *Touch-Tone*® telephone receivers. However, provision has been made for revertive and panel call indicator as well. In each of these cases, the cooperative action of both wired logic and program make up the whole job of receiving and sending digits. Rather than attempt to cover the details of each type of signaling, the following descriptions of dial pulse receiving and sending illustrate the type of interaction.

5.3.1.1 *Dial Pulse Receiving.* When an originating register is selected by call processing programs for dial pulse receiving, the first word is loaded with a number indicating the scanner, scanner word, and ferrod number in that word where the status of the pulsing relay can be observed. The remainder of the originating register is all zero. As the wired logic scans the originating register and digit receiver, every 10 milliseconds the state of the relay is compared with its pervious state stored in the make-break status bit (see Fig. 12). If there is a change from break to make, the wired logic increments the incoming pulse count. If there is *any* change in the state of the relay, the make-break status is updated and the new digit flag bit is *reset*. Every 125 milliseconds, the input-output 25 interrupt program checks the new digit bit and then sets it to one. If it does not get reset in a wired logic scan for a make-break status change over the next 125 milliseconds, the input-output 25 program will see the flag and assume pulsing is complete for the digit. The program checks that the party is still off-hook and inserts the digit in the proper position in the originating register, incrementing the count of digits received. Call processing programs check the originating register periodically to act upon digits as they are accumulated and to time between digits to insure that the interdigital timeout interval is not exceeded. Translation for office code, area code, and routing are performed by call processing programs.

Dial pulse receiving requires that the input-output 25 program check for a new digit flag at 125 millisecond intervals. Other types of receiving or sending require more frequent attention, so the input-output program looks for new digit flags at 50 millisecond intervals. This means that if a new digit flag is found in an originating register that is receiving dial pulses and the interval is not yet 125 milliseconds, the flag must be ignored. Dial pulses for the digit may still be arriving even though no change has occurred for 50 milliseconds.

The dial tone must be turned off by the program when the first dial pulse or tone is received after the receiver is connected. Since the input-output program is normally not flagged until a new digit is complete, some dif-

ferent means is required for the logic to detect when to flag the program to turn off the dial tone. Since digit sending cannot be in progress at this time, one of the values (zero) of the outpulse count is used to indicate dial tone is still on. The input-output control recognizes a nonzero incoming digit area, and sets a flag (sending flag bit SND) for the input-output program only if the outpulse count is zero. When the input-output program finds this flag on the next 50 millisecond pass, it will turn off the dial tone and set the outpulse count to its maximum value (decimal 15) to prevent the flag from being set again for this purpose. Both the outpulse count and the sending flag bit are normally used for sending functions as described in a later section. The outpulse count of 15 remains until sending is required, or the originating register is cleared, or until a second dial tone must be given.

5.3.1.2 *Dial Pulse Sending.* A dial pulse sender is enabled and disabled for pulsing by a private twisted pair from the central pulse distributor. Make-break timing for the dial pulses are supplied over common twisted pairs (buses) from the input-output control. Timing is provided for both 10 and 20 pulse-per-second sending.

When an originating register is to be used for dial pulse sending, call processing programs will place the sender circuit identity in the third originating register word, and indicate the type of signaling (dial pulse, tone, and so on) in coded form. It is assumed that the digits to be outpulsed were received in the same originating register or have been placed there by call processing programs. The digit number where sending is to start (outgoing digit count) and stop (stop sending code) are also entered in the originating register and the outpulsing speed bit is marked to indicate outpulse speed prior to sending. The last initializing action of call processing programs is to set the outpulse count to a value from 1 to 14, depending on the delay desired before sending starts. Normally, the input-output control will decrement the outpulse count at the outpulsing rate indicated (outpulsing speed) unless the outpulse count is 0, 1, or 15.

The outpulse count normally contains the count 15 when not in use for sending and the count 0 is used to indicate that dial tone has not yet been turned off for receiving. The count of one indicates that the number placed in the outpulse count by the program was decremented the proper number of times and that the input-output program should be flagged. The input-output control sets the sending flag bit as a flag to the program anytime the count of one is detected in

the outpulse count. The input-output program will see the sending flag bit on a 50 millisecond pass and set a constant in the outpulse count for interdigital timing, or obtain the next digit to be sent from the originating register and place the digit, plus one, in the outpulse count. This program also enables or disables pulsing for the dial pulse sender as required. The outgoing digit count is incremented by program for each digit. Sending proceeds in this way until the outgoing digit count is equal to either the stop sending code or the incoming digit count. This latter condition prevents sending from overtaking receiving in the case of overlapped operation.

5.3.2 *Data Sending*

Provision has been made for up to thirty-two 800 bit per second serial data channels, each of which could be associated with the control of a remote switching unit or other general data receiving facility. Data sending requires a message bit from the call store for each channel every 1.25 milliseconds. Corresponding message bits for 16 channels are read from the same word in call store by the input-output control each data bit interval. Messages of up to 64 bits are assembled vertically in the call store by the program with one bit in each of 64 words in preparation for sending. Data senders, located in separate frames from the control unit, receive message bits over private transformer-coupled twisted pair leads. Data receivers would also be located in separate frames from the control unit. Information received is stored in shift registers which are interrogated by a program using a ferrod scanner.

This method avoids the need for buffering outgoing data in shift registers. Incoming data are buffered in shift registers because they cannot be sychronized easily with the timing in the input-output control. In general, the quantity of incoming data is small so that the register size and program interrogation rate are also small.

The outgoing data bits are buffered at the transmitter and the 1.25 millisecond data timing signal generated in the input-output control is used to gate from the buffer to the transmitter itself. When the input-output control has sent the last bit of each message, it will set a bit in the input-output control register that stops data sending. New messages can then be stored in this area by the program. In order to start sending, the program sets the bit address counter in the input-output control for the address of the first message bits and resets the inhibit in the control register.

5.3.3 *Line Origination Scanning*

The autonomous input-output control performs the detection of line service requests or originations by sequencing through the ferrod scan points associated with customer lines and searching for an off-hook state. Once an off-hook state is encountered, the wired logic will stop sequencing and set a flag for the program. The identity of the scanner row can be determined by the contents of the enable address register. A program which is executed in the input-output 25 interrupt will perform a directed scan of the row, 50 milliseconds after storing the row identity in the call store and restarting the autonomous line scan function. Thus, some hit protection is provided.

The directed scan will result in the scan point number of the off-hook line being passed onto the main program for processing which includes the disconnection of the line ferrod. Thus, only unserved off-hooks are seen by the autonomous wired logic. Unless stopped by an off-hook signal, the wired logic continues to scan until the end of a scan block is reached at which time the program must provide a new starting address. The scan block can consist of a maximum of 512 consecutive addresses or 8192 lines. There can be a number of blocks and the blocks need not be maximum as long as they are continuous to the end of block boundary.

The line scanning function uses the central pulse distributor and peripheral unit buses in order to interrogate the ferrod scanners. The call store is not used since last look is not required to detect the simple off-hook state. The autonomous digit scanning of originating registers competes with line scanning for peripheral unit access but each has its own register access to the central pulse distributor and peripheral unit address bus. However, the program instructions use the same access register (enable address) as used by line scanning and, therefore, will interrupt the line scanning function (saving and restoring its state) when program access to the peripheral units is required. This access is mainly in the input-output 25 interrupt program so that the line scanner need only be stopped at the beginning of the input-output 25 program and restored at the end. The common and exclusive access provides simplicity and economy as opposed to a cycle-stealing mode which would provide less interference to the line scanning function. The time remaining outside of the input-output 25 interrupt is more than sufficient to allow the wired logic to scan the maximum number of lines at an adequate rate.

5.4 *Program Controlled Input-Output Orders*

Program access to the central pulse distributor, peripheral unit address, and scan answer buses is required for control of, and communication with, the peripheral units. General purpose instructions as well as special purpose orders, that are designed to conserve real time when repetitive access is required, are provided. The peripheral unit order structure is compatible with No. 1 ESS frames and additional orders are included to provide faster and more efficient communication with serial receivers of central pulse distributor signals such as the No. 2 ESS network frames and the peripheral decoder.

The No. 1 ESS frames require information on the peripheral unit address bus in one-out-of-*N* form to avoid the replication of translators in the peripheral frames and to check for errors in the received information. In No. 2 ESS, wired translators for access to the peripheral unit address bus are generally not provided except where real-time considerations are paramount, as for the ferrod scanner. The scanners, for example, require two groups on one-out-of-eight to select one of 64 scanner rows.

A scan order is provided which activates the central pulse distributor based on the high ten bits of the enable address register, translates the low six bits into two one-out-of-eight groups pulsing them onto the peripheral unit address bus, receives a scanner reply from the scan answer bus, and places the result in the logic register. For general use, an order is provided which allows the contents of the logic register, general register, and the low six bits of the enable address to be pulsed directly onto the peripheral unit address bus. In this way, the format of the information being sent is controlled by the program and translated using the general purpose program instructions. Economy and flexibility is thereby gained at the expense of real time. In addition, as mentioned in the central pulse distributor description (Section 5.1.2), an order is provided for pulsing central pulse distributor points, either those contained in the local distributor or those in a supplementary distributor.

The scan order just described is convenient for handling a single row of ferrods. Another typical call processing function would be to sequence through a number of scanner rows, comparing them with last-look bits in call store and indicating when a change is detected. In addition, it is desirable to sometimes mask out the status of certain ferrods for maintenance reasons or, for example, when the ferrod

status definition changes during the course of a call, as for the by-link trunk. The by-link trunk ferrod provides supervisory status as well as dial pulse information; therefore, the supervisory scan program must ignore the status of the ferrod when dialing is in progress (the ferrod is then being scanned by an originating register). To accommodate this need, a macroinstruction is provided which uses the enable address register to address the scanner, a call store address register (CA) to direct the reading of status bits, and optionally a second call store address register (AA) to direct the reading of a mask. These are combined in a logic function which causes the instruction to terminate if the condition is met or else repeats after incrementing the enable, call, and addition address registers until the cycles as specified by the kall record counter are completed.

Additional flexibility is provided by options in the instruction, which determines the increment for the enable address and call store registers.

Two other macroinstructions are provided, both of which are designed for serial communication over central pulse distributor pairs. The bit pattern in the logic register is shifted with each instruction loop, the low bit controlling the polarity of the central pulse distributor pulse over the pair defined by the enable address register. Typical central pulse distributor action requires four machine cycles in order to include the various checks that are performed during, and as a result of, a central pulse distributor order. However, after sending an order with each polarity which includes all central pulse distributor checks, the remaining bits can be sent at a rate of one cycle per bit by excluding further checks and reducing the pulse width to improve transformer recovery. Thus, one macroinstruction transmits a bit per four cycles with central pulse distributor checks and the other instruction transmits a bit per cycle with no distributor checks. The faster order is restricted to the local central pulse distributor and is used for sending network orders and orders to peripheral decoders controlling high usage service circuits such as multifrequency transmitters.

5.5 *Input-Output Error*

In sending an order to the periphery, the input-output control performs error checks that are designed to detect faults in itself, in the buses, and in the peripheral units. The reaction to an input-output error is not like other processor error conditions, which simply switch processors, since the location of the fault is not known. Here a special

program must be summoned to isolate the trouble and establish a working mode.

5.5.1 *Check Circuits*

There are three input-output errors that can result from a peripheral action: (*i*) central pulse distributor error, (*ii*) scanner ALL-SEEMS-WELL error, and (*iii*) enable verify error.

The central pulse distributor contains check circuits which monitor the enable access circuitry to see that one and only one access switch in each of the three groups is enabled. The magnitude of the current pulse into the matrix is monitored to determine that it is not excessive and the pulse in the primary leg of the final transformer stage is monitored to determine that its current magnitude is sufficient. These checks encompass many faults such as pulser failures, open transformers, shorted diodes in the matrix, shorted output pairs, translator failures, and so on. The supplementary central pulse distributor performs similar checks and replies to the input-output control with a central pulse distributor ALL SEEMS WELL signal when all checks pass. Absence of the ALL SEEMS WELL will cause the input-output control to indicate a central pulse distributor error, just as in the case of the local central pulse distributor.

The presence of a scanner ALL SEEMS WELL is expected with all scan orders. The scanner checks the validity of the information it receives, monitors its own operation, and replies with an ALL SEEMS WELL signal when the operation appears to be correct. The input-output control will indicate an ALL SEEMS WELL error when the reply is missing.

The enable verify check is intended to determine whether or not the frame received an enable signal. The input-output control checks for the enable verify reply on a common bus similar to the ALL SEEMS WELL bus, both of which are considered members of the scan answer bus.

5.5.2 *Reaction to Errors*

With the occurrence of any of these errors, the input-output control reacts differently depending upon whether the error was produced by a program order, an autonomous line scan order, or an autonomous digit scan order. Program orders encountering input-output errors will set the condition flip-flop and terminate the order if it is a macrotype. The condition flip-flop is investigated after all input-output program orders to determine whether to go to an appropriate working mode program.

An autonomous line scan order encountering an error will result in stopping the line scanner as though an origination was detected. Subsequent rescan by the program will result in the error being treated as a program scan order. The autonomous digit scan error results in more immediate action since digits can be lost or the call store mutilated. The error in this case causes the input-output control to freeze its timing and generate a high priority maintenance interrupt. This also results in a working mode program but, in this case, the identity of the trouble producing order is contained in the input-output digit logic. In all cases, the working mode program seeks to isolate the fault and might result in a reconfiguration of peripheral controllers or central processors.

VI. MAINTENANCE AND ADMINISTRATION CENTER

6.1 *Introduction*

In order to maintain and administer the system, a means must be provided to allow the maintenance man to communicate with the system. The maintenance and administration center provides that means. Through it, the man may request such functions as system status printouts, diagnostics, verification of translation data, or updating of magnet cards. The goal in designing the center was to build a fairly compact, highly intergrated center which is closely tied into the central processor, rather than build several pieces of test equipment to do specific jobs. The center is a useful tool in factory testing, installation, system maintenance, and laboratory debugging.

The center includes the maintenance center, teletypewriter control circuits, maintenance teletypewriter, single card writer, and trunk test panel.

The first four units occupy one double bay frame, and the trunk test panel occupies a single bay frame. Reference 1 has photographs of these frames. The maintenance teletypewriter is in the center bay directly below the maintenance center panel. The bay to the left contains the single card writer and maintenance center logic. The trunk test panel is in the bay to the right. Such proximity requires little movement of the maintenance man to perform all the functions.

6.2 *Maintenance Center*

6.2.1 *Introduction*

The maintenance center consists of several almost independent functional blocks which can be interconnected in different configura-

tions to accomplish the desired functions. These blocks are: a display buffer, internal registers for status display and control, a comparator, a central processor stop mechanism, a central processor interrupt mechanism, the control unit control circuit, a margin control circuit, and the emergency action panel.

The display buffer and three other internal maintenance center registers are connected to a maintenance center gating bus using the same concept as the program gating bus in the control unit. Also, coupled to the maintenance center gating bus by dc communications are the program gating bus, program address register, call store input register, and call store address register of each control unit. The display buffer can be used to automatically display the contents of the program address register, call store address register, program store, or call store, or it can be gated to by program. The display buffer can be "frozen" by hardware or program to permit an address to be trapped, or data to be displayed without being overwritten. The display buffer is shown on the maintenance center panel, Fig. 13, as a 22-bit register, marked in various ways to facilitate reading the information it contains. The lighted pushbuttons, which can be used to set the individual bits, are colored in groups of three for easy reading as octal numbers. Omitting bits 21 and 0, which indicate the transfer allow check bit and parity bit, the remaining 20 bits are indicated in groups of five by a white background mask for easy reading as one 5-bit or two 10-bit operation codes. The 5-bit groups are further divided to indicate the octal breakdown of the operation code.

The comparator will match two inputs of up to 22 bits each, with both a match and mismatch output. Inputs are from the maintenance center gating bus, the comparator input switches, and call store input register of each control unit. The comparator output can be used to trigger one of several functions, allowing the craftsman to capture events or conditions such as a register equal to a given value, or a logic gate becoming active. The two most common uses are to trigger the stop or interrupt mechanisms.

The main application for which the maintenance center was designed is as a system maintenance tool in an operating office. In this capacity, the maintenance center serves in four roles:

(*i*) An in-service monitor of the status of the system,
(*ii*) The test and control center for routine functions,
(*iii*) A backup means when planned diagnostic procedures fail, and
(*iv*) An emergency control in extreme situations.

As an installation tool, the maintenance center takes the place of

external test equipment in previous systems. With the maintenance center, it is possible to manually control, access, and monitor the control unit thus insuring the system's proper manufacture and installation. Its use as an installation tool (as well as at the factory) is in close harmony with the test specifications and initial test (X-ray) programs.

Also it is an aid in program development in the system laboratory. In this capacity, an individual programmer is permitted manual access to most control unit registers as well to the control logic.

The usefulness of the maintenance center in each of these applications can best be explained by describing its separate features.

6.2.2 *Operational Features*

The features of the maintenance center fall into three categories: (*i*) control until matching, (*ii*) routine operation, and (*iii*) special purpose operation.

Under certain conditions (explained later), the maintenance center

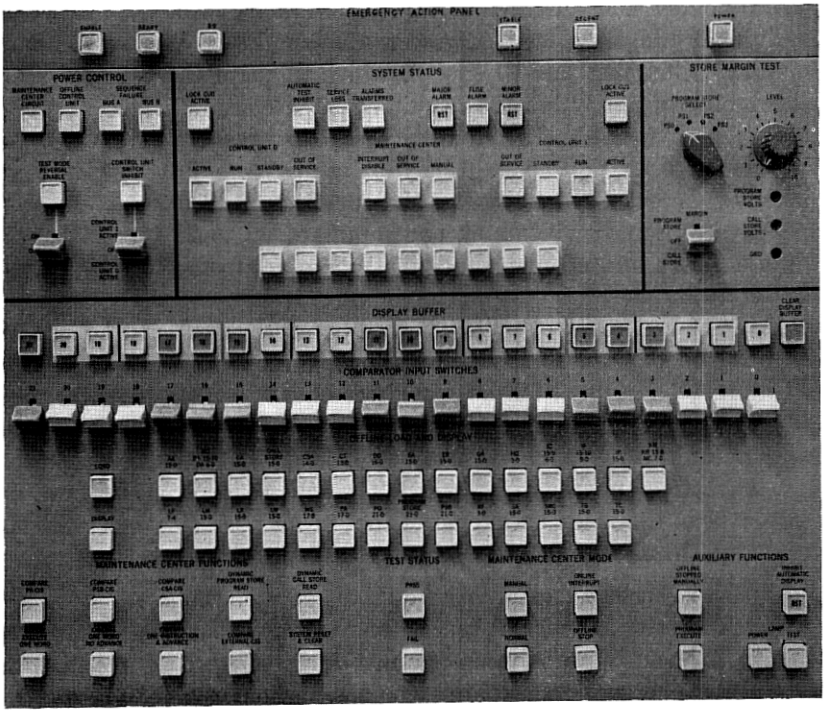

Fig. 13 — Maintenance center panel.

acts as a master check circuit, matching the results of the duplicated control units. Routine operations include those automatic or manual features which are used or observed routinely by the office personnel, or operated automatically by the system. The third category includes those features which are used when manual intervention is required by the office personnel, usually in a nonroutine capacity.

The maintenance center has two operational modes, NORMAL and MANUAL. The MANUAL mode is required in most cases of human intervention; however, this disables certain automatic features which are performed when the center is in the NORMAL mode. Use of the MANUAL mode does not necessarily imply an abnormal or emergency situation. Several day-to-day operations require the MANUAL mode because they use circuitry used for other purposes in the NORMAL mode.

6.2.3 *Control Unit Matching*

Without the intervention of a maintenance man or some detected trouble, both control units are usually running simultaneously, executing the same instructions in synchronism, with only the outputs of the off-line control unit to the periphery and to the maintenance center being inhibited. If the maintenance center is in the NORMAL mode, this state is said to be the NORMAL-COMPARE state, which enables the processor match circuit. In this mode, the contents of the call store input register in each control unit are being matched in the maintenance center. Since the call store is used quite frequently to store temporary results by the program (the call store input is used on both read and write), any trouble that is not detected by check circuits should cause the call store inputs to differ within a short time. This mismatch causes an interrupt signal to be sent to both control units, activating a mismatch detection monitor program in an attempt to locate the trouble. When the mismatch occurs, the on-line call store address register is gated to the display buffer, then "frozen." This aids the maintenance programs in the detection and recovery process.

Whenever the maintenance center is in the NORMAL-COMPARE mode and no mismatches have occurred, the current program address is shown in the display buffer. In the event of an automatic switch of the control units, the program address location at the time of the switch is trapped in the display buffer for use by the maintenance programs.

6.2.4 *Routine Operations*

Features that are part of routine operation are the system status displays, trouble recording, and control of the on-line—off-line status

of the control units. These features are active regardless of the MANUAL or NORMAL mode of the maintenance center. A fourth use of the maintenance center in routine operation is in conjunction with the magnet card updating and writing process using the single card writer. This use is explained later.

As can be seen from the photograph of the maintenance center panel (Fig. 13), the system status displays indicate the current alarm, control unit, and maintenance center status, as well as the status of the maintenance and call processing program capabilities. A line of lights at the bottom of the system status display area is permanently assigned by the particular application of the No. 2 ESS. These indicate status of unique programs or devices (for example, certain peripheral units).

In addition to these status indicators, the maintenance center contains a register which monitors the control unit check circuits. This register is called the error register.

The on-line—off-line status of the control units is normally determined by a flip-flop in the maintenance center, as controlled by the program timer or the maintenance programs. In emergency situations it can be controlled by the craftsman.

6.2.5 *Special Purpose Operation*

The special purpose operations include most of the manually operated functions, except for card writing, and require the maintenance center to be in the MANUAL mode. They are used mainly for installation, program debugging, and diagnostic work. Probably the most useful of these features are the "compare" and "load and display" features. The compare feature uses the same comparator as the processor matching operation already described. Inputs to this comparator allow matching the following items:

(*i*) program address versus a set of comparator input switches on the panel,

(*ii*) call store address versus the comparator input switches,

(*iii*) program address versus the display buffer (can only be set up up by program),

(*iv*) program store output buffer versus the comparator input switches,

(*v*) any external signal (up to 22 bits) versus the comparator input switches.

When a match occurs, the match signal may be used for the following, depending upon the function and upon the control unit selected (on-line or off-line). It may cause:

(*i*) an interrupt in the on-line control unit whenever any of the above matches occur,

(*ii*) the off-line control unit to be stopped whenever any of the above matches occur,

(*iii*) the on-line or off-line program store contents to be displayed in the display buffer whenever the program address matches the comparator input switches (dynamic program store display), and

(*iv*) the on-line or off-line call store contents to be displayed in the display buffer whenever the call store is accessed and the call store address matches the comparator input switches (dynamic call store display.)

If the off-line control unit has been stopped by a match, the display buffer, which has been displaying the program address, is "frozen" and contains the last program address, thus effecting a "from address trap." This is most useful in tracing the progress through a program. After the desired operations have been performed, the control unit can be started again and allowed to run in any of the following modes: (*i*) continuously (remove the match function), (*ii*) up to another match, which could have been changed while the control unit was stopped, or (*iii*) for only one instruction.

The incrementing of the program address register can also be inhibited, allowing repetition of the same word, such as when it is necessary to look at timing signals with an oscilloscope. Two modes are available:

(*i*) execute one word (one or two instructions) continuously, and
(*ii*) execute one word at a time.

When the off-line control unit is stopped (on a match as above, or by depressing a pushbutton on the panel), several avenues of investigation are open to the maintenance man. The first is the ability to load or display any register in the stopped off-line control unit. This is accomplished by first selecting a LOAD or DISPLAY function, then selecting the appropriate register from the set of pushbuttons (see Fig. 13). Registers are displayed in the display buffer, and are loaded from the display buffer, which itself can be preset from the maintenance center panel, if desired. Call store can be read or written and program store can be read using the same mechanism, if the appropriate address register is loaded first.

Frequently it is desirable to be able to look at or trap conditions in the on-line processor without disturbing its call processing operation. For this purpose, the on-line processor may be interrupted (whereas

the off-line processor was stopped). The interrupt is used to trigger one of a set of utility programs which have been previously requested via the maintenance teletypewriter. Among the utility routines are the ability to do the following:

(*i*) read or write any on-line or off-line call store word,
(*ii*) read any on-line or off-line program store word, or
(*iii*) read or load most on-line or off-line control unit registers.

Contents of the program store, call store, or registers can be typed out on the teletypewriter either on a one-shot basis or whenever the data changes, or can be displayed dynamically in the display buffer. The program can turn off the interrupt, as in the case of a request for a single printout, or when the teletypewriter printout buffer area in the call store becomes full. Only one interrupt request can be active in the system at a time.

The above utility requests may also be base level requests. A base level request asks that the function be performed immediately upon receipt of the input message, or once per base level scan if a repetitive function is desired. These base level requests are recognized regardless of the mode of the maintenance center, and can be acted upon simultaneously with an interrupt request.

The remaining special purpose features are the store margin control, power control, and emergency action control. The store margin control allows margins to be checked manually on any one of the four program stores or on the call store, with the threshold voltage of the sense amplifier available at test points on the maintenance center panel. (This is independent of the preset automatic margin tests controllable by program.) The power control provides power on-off functions and a system override control which allows the flip-flop which controls the active status of the control units (program control flip-flop) to be bypassed. The override is to be used by the maintenance man only in emergencies, such as repairing the program control flip-flop. Normally, a control unit can be locked active by depressing the LOCK CU—ACTIVE pushbutton, which does not bypass the flip-flop but still holds that control unit active. The emergency action control allows the maintenance man to call in an initialization program manually. This program will initialize the control unit and, depending upon the status of two associated pushbuttons (STABLE and RECENT), zero the nonstable areas of call store, and neither, either, or both of two areas of call store: (*i*) recent change buffer area and (*ii*) stable call store (all except recent change).

6.3 *Teletypewriter Facilities*

The teletypewriter facilities contained in the maintenance and administration center include the teletypewriter control circuits, as well as the maintenance teletypewriter. Up to eight teletypewriter channels can be accommodated by the No. 2 ESS, with a minimum of two channels required. Each channel is controlled by one control circuit. Any standard teletype device can be attached to a channel, either singly or in combination, such as a send-receive teletypewriter, receive only teletypewriter, paper tape punch, paper tape reader, or automatic send-receive teletypewriter. Long distance channels can be handled with a 108/820 data set combination over a private line.

Each channel control circuit is an independent parallel-to-serial converter, accepting the 7-bit teletypewriter code from the program, and pulsing out the eleven 9.09 millisecond pulses necessary to control the teletype device. Data communication to the circuit is over a dc connection from the program gating bus, with the individual circuit selected by one of eight central pulse distributor points. When receiving information from the channel, the circuit acts as a serial-to-parallel converter with communication from the circuit to the program over a set of ferrods. There are seven data ferrods, one signal present ferrod, and several status ferrods—low paper, teletypewriter disconnect, data set disconnect, and so on—depending on the particular channel.

The layout of the control circuit consists of one common tray of logic providing timing selection (the on-line clock is used), power-off relays and data selection. The individual circuits grow in modules of two circuits per logic tray, but only one set of cards needs to be supplied at a time. The circuit trays can be added "in service" as needed.

The teletypewriter program routes messages to, and accepts messages from, the correct channel depending upon the content of the message. If that channel is out of service or does not exist, the message is routed to or accepted from its backup channel designated by the operating company. No backup can be designated if so desired.

A maintenance detection check is made on each output message to guarantee the communication loop. This check consists of sending a "who are you" code to the teletypewriter which responds with a fixed answer. If the answer is not received, the message is routed to the backup, and diagnostics are run on the original channel. Diagnostics and restoration can also be requested manually.

6.4 *Single Card Writer*

6.4.1 *Application*

The single card writer is the device by which permanent magnet twistor cards are magnetized in the smaller No. 2 ESS offices. As the name implies, it magnetizes one card at a time, rather than an entire module as does the No. 1 ESS card writer. The single card writer has been designed as simply as possible to meet moderate change activity requirements and still keep the cost low so as to be attractive in the smallest No. 2 ESS offices.

6.4.2 *Procedure*

When a card is to be written, a message is typed on the teletypewriter indicating the card to be magnetized, or asking for the next card to be magnetized. Upon the acknowledgement, the card is placed in the single card writer and the WRITE button depressed. This starts the magnet head moving and saturates a ferrod causing the program to gate 22 bits of information to the display buffer. The program notifies the single card writer via a central pulse distributor point that the display buffer is loaded. The polarity of the pulse is checked against the card type (A or B) which is in the writer, and a WRONG light indicates an incorrect card type.

When the single card writer head senses the first initializing magnet, slightly before the first row of magnets, it takes the low 11 bits from the display buffer. When the second initializing magnet is sensed, the high 11 bits from the DB are magnetized into the low 11 bits of the second row of magnets and the single card writer asks for more information. This sequence is repeated 32 times in all, with 33 milliseconds between each word request, then an end-of-row ferrod is saturated. The program checks that the proper timing exists between the 32nd word request and the end-of-row ferrod, and that 32 requests have been received. The process is then repeated for the next three passes. If the entire sequence is completed satisfactorily, an END light is operated.

The procedure for magnetizing a set of cards involves pulling the desired cards from a spare set, magnetizing them, placing them in the off-line store, and verifying the magnetization. This is then repeated with the on-line control unit switched off-line. The time required to magnetize 37 cards on both sides has been demonstrated to be one hour.

6.5 *Trunk Test Panel*

Manual testing of lines and trunks are an important part of every office. In the No. 2 ESS, most line testing will be accomplished from a standard No. 3 local test cabinet. This provides for continuity, leakage, foreign potential, and subscriber line checks. All manual trunk testing, as well as transmission measurements on lines, will be conducted from the trunk test panel.

The trunk test panel will occupy a bay adjacent to the maintenance center frame. This facility provides for switched-up access to any trunk, service circuit, or line terminated in the No. 2 office. It enables the following checks to be performed:

(*i*) operational and transmission tests on trunks and service circuits,
(*ii*) leakage and continuity checks on both lines and trunks,
(*iii*) transmission checks on subscriber lines and PBX-type trunks,
(*iv*) removal of trunk from service and restoral to service under key control, and
(*v*) voice communication via private or regular telephone channels.

Test connections are made via one of three access trunks established under key control and with the aid of a panel mounted *Touch-Tone*® telephone. The access trunks allow for:

(*i*) originating and terminating service,
(*ii*) unrestricted test access to all circuits, even under line load control and busy conditions,
(*iii*) connection, under key control, to a voltmeter test circuit, transmission and signaling test circuitry, and make-busy arrangements,
(*iv*) jack access to allow any portable instrument connection,
(*v*) holding a circuit while testing another, and
(*vi*) the ability to place a trunk in any of its operational states.

VII. SUMMARY

This article has described the structure and relationship of the units which unite to form the major system control element, the No. 2 ESS control unit. The details of the implementation of the logic and circuit designs have been avoided but, instead, the general structure and features of the system have been stressed. The control unit design is a product of the needs for an economic and efficient stored program control system for a particular class of offices. The influences of both the call processing design and the maintenance plan are evident when

viewing the details of the system. Repetition and overlap between companion articles in this issue has been avoided whenever possible. As a result, further details and philosophy concerning the control unit design are in the associated articles, especially those concerned with call processing and maintenance.

REFERENCES

1. Lonnquist, C. W., Manganello, J. C., Skinner, R. S., Skubiak, M. T., and Wadsworth, D. J., "Apparatus and Equipment," B.S.T.J., this issue, pp. 2817–2863.
2. Andrews, R. J., Driscoll, J. J., Herndon, J. A., Richards, P. C., and Roberts, L. R., "Service Features and Call Processing Plan," B.S.T.J., this issue, pp. 2713–2764
3. Digrindakis, Jr., Freimanis, L., Hofmann, H. R., and Taylor, R. G., "Peripheral System," B.S.T.J., this issue, pp. 2669–2712.
4. Meinken, R. H., "Ferrite Sheet Memory," Proc. 1960 Elec. Components Conf., Washington, D. C., May 10–12, 1960.
5. Meinken, R. H., "A Memory Array in a Sheet of Ferrite," Proc. Magnetism and Magnetic Materials Conf., October 16–18, 1956, p. 674.
6. Genke, R. M., Harding, P. A., and Staehler, R. E., "No. 1 ESS Call Store—A 0.2-Megabit Ferrite Sheet Memory," B.S.T.J., *43*, No. 5, Part 1 (September 1964), pp. 2147–2191.
7. Ault, C. F., Gallaher, L. E., Greenwood, T. S., and Koehler, D. C., "No. 1 ESS Program Store," B.S.T.J., *43*, No. 5, Part 1 (September 1964), pp. 2097–2146.
8. Connell, J. B., Hussey, L. W., and Ketchledge, R. W., "No. 1 ESS Bus System," B.S.T.J., *43*, No. 5, Part 1 (September 1964), pp. 2043–2045.
9. Freimanis, L., Guercio, A. M., and May, H. F., "No. 1 ESS Scanner, Signal Distributor, and Central Pulse Distributor," B.S.T.J., *43*, No. 5, Part 2 (September 1964), pp. 2270–2277.