THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 56, No. 2, February 1977
Printed in U.S.A.

1A Processor:

Control System

By A. H. BUDLONG, B. G. DE LUGISH, S. M. NEVILLE, J. S. NOWAK, J. L. QUINN, and F. W. WENDLAND

(Manuscript received July 16, 1976)

This article contains a description of the central control, input/output subsystem, processor-peripheral interface, master control console, and the interunit communication bus system. The units are discussed in a manner which highlights comparison of features with No. 1 ESS and stresses those features which are most important in meeting the stringent reliability objectives.

I. INTRODUCTION

The 1A Processor control system, coupled with program, call, and auxiliary memory, is an independent, stand-alone processor with full operational and maintenance capability. It provides for execution of the processor and system program, automatic system maintenance actions, manual control and monitoring, and interfaces to memory, network, and external systems.

This processor is an outgrowth of previous Bell System switching processors. Although it contains many similar or modified features, the new processor offers significant improvements over its predecessors. In comparison with the No. 1 ESS processor,[1] several major new features and improvements are provided—for example, four to eight times faster instruction execution, writable program memory, greatly expanded program and call memory size capability, and auxiliary memory with autonomous transfer to and from program and call memory (direct memory access). These advancements are achieved while maintaining compatibility with the No. 1 ESS network and programs.

The control system consists of the central controls (CCs), communication buses, and the man-machine interface and control. The man-machine interface and control includes the master control console (MCC),

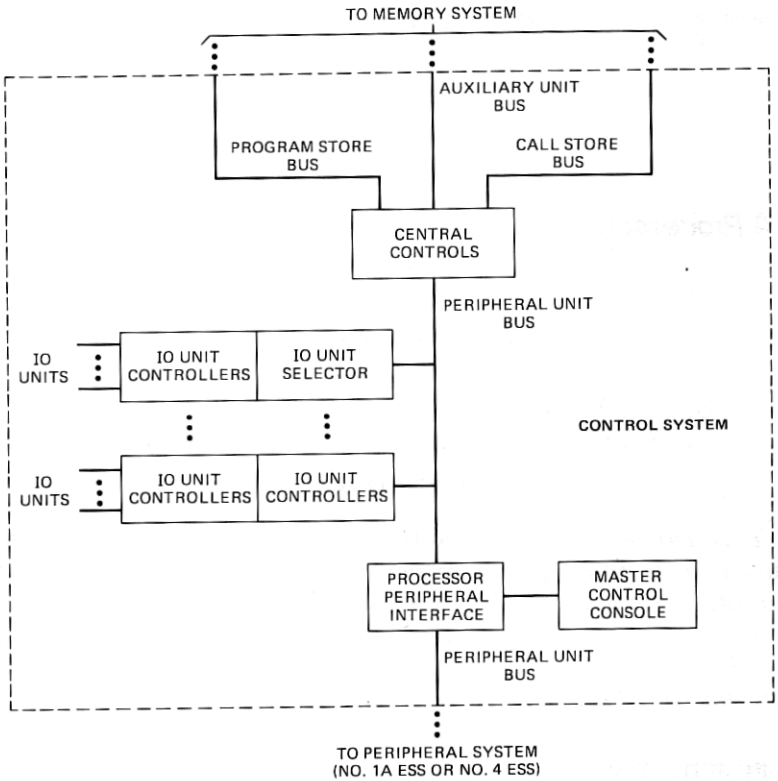


Fig. 1—Control system.

processor-peripheral interface (PPI), and the input/output unit selectors and controllers (IO). These are illustrated in Fig. 1.

This article presents an overview of the control system features and then details the operation and maintenance features of each subsystem.

II. CONTROL SYSTEM FEATURES

2.1 General

The control system coupled with the memory system constitutes a stored program processor. For switching system control, the instruction capability of the central processing unit (CC) is tailored for control and logical operations rather than mathematical operations such as multiplication, division, and function evaluation. Internal CC operations include fixed-point addition and subtraction, word and item manipulation (such as rotation and masking), sequencing to perform multiple autonomous tasks associated with each communications bus, sophisticated instruction address manipulation, and sequencing and logic operations

dedicated to maintenance functions and interrupt handling. The instruction set reflects these features and is oriented to provide efficient transfer of control, item manipulation, index register modification, and condition testing. Special instructions and instruction sequences are provided to optimize control of the switching network, and other special instructions are oriented toward maintenance actions such as interrupt handling, configuration control, and subsystem fault diagnosis.

In addition to the improvements previously mentioned, the 1A Processor surpasses the No. 1 ESS processor in other areas. Operational improvements include an expanded instruction set, more general internal CC logic, a coded enable unit selection capability on the peripheral bus, expanded IO capability for input and monitoring, and a centralized peripheral interface. Maintenance features have been improved by including more access and control, so that virtually all memory elements can be initialized and monitored. In addition, checking circuits can be exercised, all bit combinations can be transmitted over buses, and critical control signals can be sent to each unit over two completely separate paths.

2.2 Central control

The internal structure of the 1A Processor CC is similar to the No. 1 ESS CC, in part to achieve a high degree of forward compatibility. Within this constraint, a great many improvements have been made to its instruction set.

Internal registers and arithmetic logic functions have been expanded and enhanced. Processing features include generalized adder access, generalized registers, "shadow" registers which save the contents of each general register on request, and 24-bit internal data paths and registers. These changes are provided to improve the development of application and maintenance programs.

Instruction set features include generalized No. 1 ESS options, instructions with multiple functions, new instructions to utilize new internal processing features, memory protection features, and an instruction to search memory for a word matching selected register contents. Instructions are one or two 24-bit words, where appropriate, to save memory when only one word is necessary. In many cases, an instruction is available in either one- or two-word format, and the format selection is based on the length of fields or options specified by the programmer. Transfer addressing capability is improved by the use of an address stack utilizing a pointer in CC. Program and call store interfaces and internal CC logic are designed to work with either a 700-ns or 1400-ns CC-store cycle. Maintenance instructions allow communication paths and subsystem interfaces to be fully checked.

The peripheral control has been expanded beyond the central pulse

distribution (CPD) scheme used in No. 1 ESS to provide coded enabling with bus transmission checks. The coded-enable scheme accommodates a long bus without an excessive time penalty. This is possible because the CC recognizes the completion of an operation and resumes processing immediately, rather than allowing a fixed time for the full bus length. The reply bus has also been expanded from 16 to 24 bits, and nonperipheral instructions that do not interfere with peripheral registers are processed simultaneously with the completion of peripheral instructions.

The new auxiliary unit control provides for mass memory units with autonomous transfer of data between program or call memory and the auxiliary memory. This saves substantial processing time, since the transfers are not directly controlled by instructions executed in the CC. An interface controller in the CC administers the data transfer to or from auxiliary memory units on a priority basis. The priority can be altered by program control.

Enhanced maintenance features in the CC include selective inhibits of clock pulses, read and write access for nearly all memory elements including sequencer circuits, sophisticated start-stop control, and two 24-bit matchers in each CC. The CCs have access to each other over fast, dc interconnecting buses which provide control and route the CC data for matching. They also have the capability to generate maintenance and configuration signals for each unit and a circuit to check for proper operation.

2.3 Communication buses

Communication buses are used to transfer control, timing, and data signals between processor subsystems. The buses are designed to provide physical and electrical isolation, to provide for simple addition of units on an in-service basis, and to provide a pluggable connection for efficient factory testing, installation, and maintenance.

Peripheral buses are directly compatible with the No. 1 ESS peripheral system and, therefore, the timing and signal levels are constrained to be the same as No. 1 ESS. Other buses utilize advanced circuits, packaging, and cable techniques to achieve faster, lower-power operation consistent with the 700-ns CC cycle.

Each bus system is duplicated and has configuration control circuits, as in any major subsystem.

2.4 Man-machine interface and control

Normally, communication between personnel and machine is via terminals, such as teletypewriters, that interface through input/output unit selectors and controllers. Up to 96 channels are available with 110-

or 1200-word-per-minute data rates. Terminals may be local or remote, with data sets used for remote terminals.

The master control console (MCC) provides direct manual control and monitoring of key indicators. In addition to this console for the 1A Processor, a companion console is associated with each system using the 1A Processor, such as No. 1A ESS[2] or No. 4 ESS.[3]

The processor peripheral interface (PPI) provides the logic and access circuitry for the MCC. It also contains circuitry to monitor and control individual unit power switches and other manual controls.

A major function of the PPI is to provide a common unit for interface to peripheral systems such as the No. 1A ESS or No. 4 ESS network. The ability to loop peripheral bus transmissions back to the CC is provided. This allows intraprocessor bus cables and circuits to be checked independently from the peripheral units.

III. CENTRAL CONTROL OPERATIONAL HARDWARE FEATURES

Central control is a synchronous stored-program central processing unit designed with a multiplicity of hard-wired sequencers to facilitate the level of concurrent operation required. It is worst-case designed to operate on a tightly synchronized basis with main and auxiliary memory units within the 1A Processor complex. The worst-case design is based on a model of 1A technology interconnection and devices and is supported by both extensive measurements and analytical methods. Within central control, over 100 representative critical timing chains were simulated using simulation models and tools that evolved with the hardware design. With specifications on the devices and hardware, which also evolved with the design, the average worst-case maximum delay per logic stage for typical critical paths in the CC is about 12 ns, allowing about 58 stages of delay in a 700-ns cycle.

In addition to serving as an instruction execution and control unit, central control provides system synchronization. System timing originates from a 20-MHz crystal oscillator circuit in the active central control. This oscillator circuit provides signals for both the active and standby CCs, thus synchronizing them. The active CC also generates synchronization pulses for other units in the system, including main and auxiliary memory units. These synchronization pulses are sent every CC cycle (700 ns) on a per-unit basis.

Central control is designed to operate with any mixture of 700-ns and 1400-ns cycle time main memory units. In the normal system mode, the central controls are synchronized and running in step. In other words, they are executing the same instructions and operating on the same data. This is referred to as "duplex operation." In the 1A Processor system, transient data are generally duplicated, that is, stored in two physically

separate main memory units assigned the same locations in the address spectrum. Valid system configurations include duplex central controls fetching data from duplicate stores at different speeds. Each central control monitors the speed of the main store unit accessed by the mate central control, and if either CC accesses a slow (1400-ns) module, then both CCs run at the slower rate.

The instruction execution rate is strongly influenced by the speed of the main memory units since a significant proportion of the instructions are executed in one main memory cycle time. For representative code with 1400-ns memories presently available for field use, the average execution rate is about 750,000 instructions per second. Although instructions are designed to execute in an integral number of 700-ns cycles, the "average" instruction requires only 1100 ns to execute; the execution control is idle about 18 percent of the time, particularly as a result of waiting for the target of a successful transfer to be fetched.

3.1 Instruction fetching

Modern processors achieve high capacity not only because of fast hardware but also because of the increase in concurrent operation including the overlapping of instruction fetching and executing. The fetch control may be several instructions ahead of the execution control. When the fetch control system detects a conditional transfer instruction that is not yet ready to be executed, it is assumed that the transfer will not be made and straightline fetching continues. However, when an unconditional transfer is detected and a transfer is not currently being executed, no fetch operation is initiated until the target address is calculated by the execution control system (this may save a CC cycle).

To facilitate concurrent memory operations, the main memory is split into two systems: program store and call store. Typically, the program store system provides the instruction stream and the call store system provides the data. This distinction is not rigid, however, since either system can perform either function and both systems communicate with central control in similar fashion (timing, synchronization pulses, etc.). The major difference between the program and call store systems is the width of the communication path. On each read operation, the program store returns an even-odd address pair of words and central control selects the required portion, whereas the call store returns only a single word.

Since fetching program code from call store is an inherently slower operation than fetching from program store, this use is restricted and protection is provided so that unplanned transfers to call store are not allowed. Execution of program from call store is used for fault recovery and diagnostic for the program store system. These programs are in-

frequently used and therefore require a very small fraction of system real time. Gating and sequencer action for call store fetching closely emulate gating and sequencer action for program store fetching. To minimize memory access conflicts between the execution and fetch control systems, only seldom-used data are located in program store.

A program instruction may be either one or two words in length, that is, either 24 or 48 bits. The number of bits in an instruction is not directly related to the execution time, although double-word instructions tend to take more cycles to execute because they allow greater flexibility in special features and options. The two words read on each program-store fetch may be two single-word instructions, a single-word instruction and either half of a double-word instruction, halves of two double-word instructions, or one entire double-word instruction. The single-word orders save substantial program storage space since two-thirds of the instructions are typically single-word orders.

To improve efficiency in fetching long (double-word) instructions, these instructions may be aligned on even-address boundaries by inserting a special no-operation order which is not executed if encountered on an odd boundary following a short (single-word) instruction.

The instruction stack size of six (24-bit) words is chosen to best accommodate the frequency of transfers which is relatively high (about 15 percent). Because the stack is small, no attempt is made to determine if the target instruction of a transfer might already have been fetched; instead the target is fetched again.

A pictorial representation of the instruction stack is shown in Fig. 2.

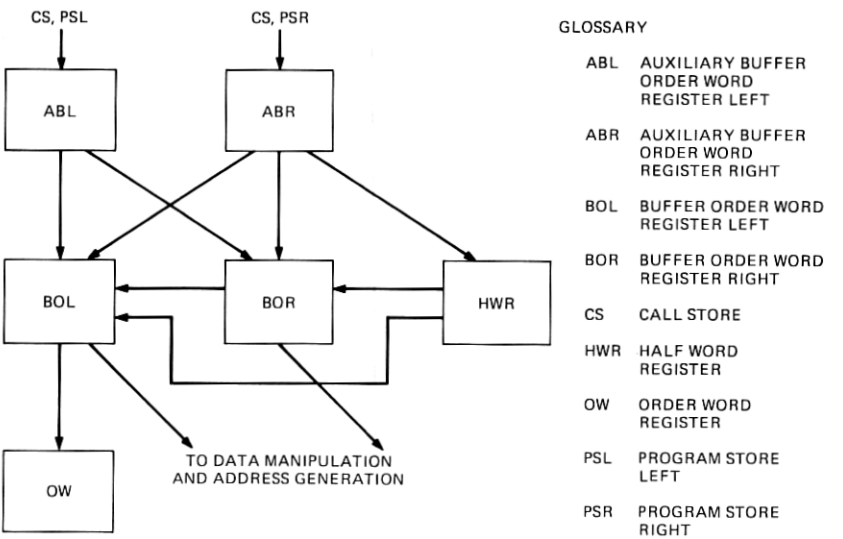


Fig. 2—Instruction stack.

A double-word program store reply is accepted into the ABL and ABR. If both words are required and if the BOL and BOR are available (empty), then the contents of the ABL and ABR are immediately gated down into position to be executed. If the BOL contains an instruction which has not yet started execution, and the BOR and HWR are available, then the contents of the ABL and ABR are gated to the BOR and HWR, respectively. If only the right half of the reply is required (for example, on a transfer to an odd address), then the ABR contents are gated to the BOL. Other gating paths within the stack are shown in the figure.

Instruction decoding and execution begin from the BOL with the data portion of a long instruction in the BOR. In the first cycle of execution, the contents of the BOL are duplicated in the OW and the data portion of the instruction is gated to the appropriate data-manipulation or address-calculation circuitry. The remainder of instruction decoding is performed from the OW with an overlap of decoding which assures continuity of gating and control signals.

3.2 Instruction execution

The 1A Processor order set is a substantially enhanced version of the No. 1 ESS order set. One of the basic requirements of the order set is to allow for the conversion of No. 1 ESS operational code into 1A Processor code via off-line translator programs with a minimum of redesign. Although the resulting order set contains some redundant features such as two address-return mechanisms, it is efficient in both real time and memory usage. The short instructions, for example, reduce required storage for programs by about 33 percent. The mask and shift facilities allow fetching a data word from memory and adjusting an item (a contiguous set of bits) within that data word in either direction with a single-word order in one memory cycle time. A double-word instruction allows detection of the least significant (right-most) one in any general-purpose register (or the logic register) and optionally clears that bit. If the register tested contains all zeros, the instruction transfers control. Conditional instructions employ the two control flip-flops storing the sign and homogeneity (all "1s" or "0s") of the result of an arithmetic or logical operation performed by either a memory read or a test instruction.

The adder and logic unit (ALU) shown in Fig. 3 is oriented to a control and decision-making type of function. The ALU performs common logical operations and fixed-point addition and subtraction using one's complement arithmetic. Arithmetic, logic, and shift functions are provided on all seven general-purpose registers. Each of these registers and the logic register, which is used both as a masking register and as a peripheral operation data reply register, has a shadow register to save data for client

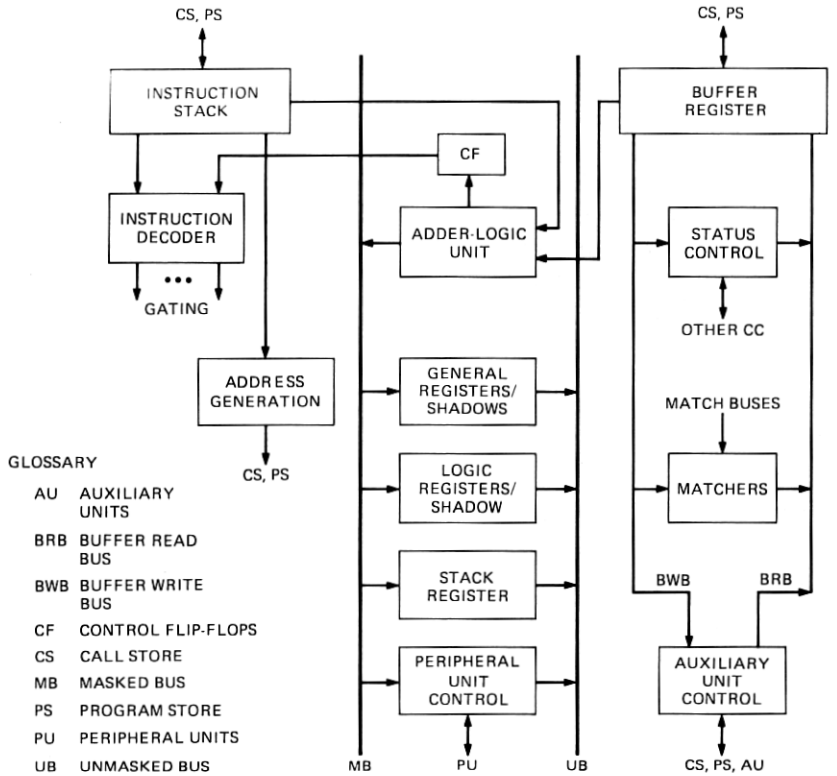


Fig. 3—Central control, block diagram.

programs. Single-cycle short instructions save or restore the contents of any combination of these registers.

To optimize the use of the available code space, particularly on single-word instructions, operation codes are of variable length and may be noncontiguous. The double-word instructions provide access to the full address spectrum, which should be large enough (over four million words) to allow using the 1A Processor in any high-capacity switching application.

3.2.1 Details of execution and timing

The 700-ns central control cycle is divided into fourteen 50-ns basic clock phases. Typical clear and gate pulses are 50 ns and 100 ns in duration, respectively.

As can be seen in Fig. 3, the logical flow of instructions is centered around the use of two major internal bus systems: unmasked bus (UB) and masked bus (MB). The CC cycle is divided into three internal bus phases (0T4, 4T8, 8T0), as indicated in Fig. 4. These bus phases are in-

tentionally unsymmetrical to allow certain data processing functions (e.g., addition) to be performed in one bus phase without penalizing the system cycle time. Even so, the longest bus phase (8T0) is extended to 6T0 for preparation of data (with parity) to be written into memory. It is desirable to execute a memory operation (read or write) within one memory cycle time. Write operations present data to the memory near the end of the first cycle of the instruction and, in the case of the 1400-ns memory, wait for a completion indication during the second cycle. In the case of a 700-ns memory, the operation is completed in one central control cycle.

The time required to perform the three basic steps of a read operation exceeds the memory cycle time by about 300 ns for either memory speed. The first step of a read operation, address generation, requires approximately 350 ns and begins by decoding the instruction as it is accepted into the instruction stack (before T12); the address is ready for transmission before T5 of the first cycle. The second step of a read operation consists of the access time of the memory system, including clock uncertainty, synchronization, cable drivers and receivers with associated logic, and bus transmission delays. For a 1400-ns memory system, this access time totals about 1050 ns; for a 700-ns memory system, about 350 ns is allocated. In either case, the data reply is available in the central control data buffer register by T12 of the appropriate cycle. The third step, processing the data reply, requires as much as 300 ns and may include performing an addition or logic operation (perhaps masking and/or complementing) and placing the result in the destination register. Thus, a read operation is not completed until almost T4 of the cycle following

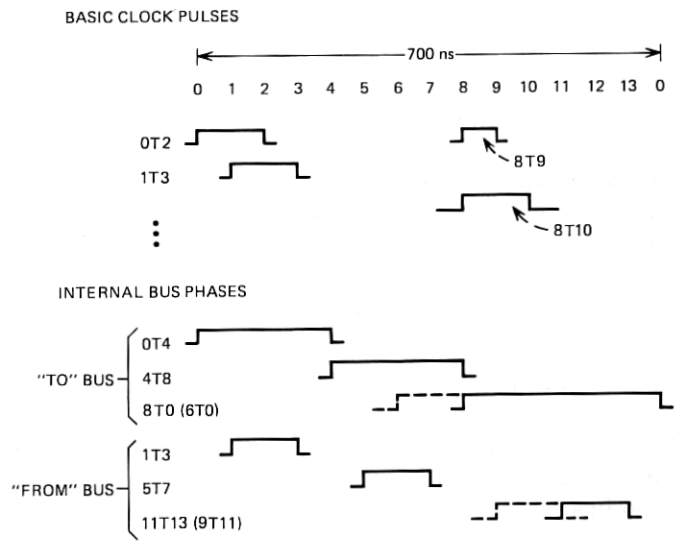


Fig. 4—Internal central control timing.

the store data reply. Decoding of a new order begins as soon as it is available in the instruction stack so that the final 300 ns of the memory read operation overlaps the first 300 ns of the following operation. This overlapping results in restrictions on the use of portions of the circuitry which may be used during this 300-ns interval by each instruction. The read operation may use the data buffer register, the ALU, the masked bus, and the destination register. The operation following the read is restricted during this interval to the unmasked bus and the address generation circuitry. If the beginning instruction requires that the data being gated into the destination register of the read instruction be used for indexing or testing, the normal sequence of accessing a register via the unmasked bus is modified. In the case of indexing, the data are loaded directly from the output of the ALU into the address-generation circuit. In the case of register testing, it is sufficient to inhibit gating the previous contents of the specified register to the control flip-flops since the read operation always loads the control flip-flops from the output of the ALU.

A beginning instruction has access to adequate resources to allow at least the beginning steps of address generation to take place during the initial 0T4 interval. This initial interval is also the time during which the preparatory steps take place for index-register modification which is completed during the 4T8 bus phase. These options are discussed later in this section. Similarly, the argument register within the ALU is commonly loaded during the 4T8 phase in preparation for the 8T0 (or 6T0) data processing phase. A mask may be prepared by gating encoded instruction data from the BOL or BOR to the size-displacement translator in the ALU during 0T4 and by gating the translated (24-bit) mask to the logic register during 4T8 for use during the data processing phase. Table I summarizes the basic operations performed in each internal bus phase.

Table I — Allocation of internal bus phases

Internal Bus Phase	Activity
0T4	Address generation Condition test Logic functions (overlap) Generate mask Preload add-one logic (A-option)
4T8	Index register modification (A-, S-, and W-options) Preload argument Return address save Move mask to logic register
8T0 (6T0)	Write data to buffer register Data word through ALU Special register, bit, or item test

The effective address is usually the sum of a data field in the instruction and either the contents of a general-purpose register (indexing) or the address of the instruction currently being executed (relative addressing). The address-generation circuitry includes a fast, carry-look-ahead, two-input adder. In the relatively infrequent case that all three of the above-mentioned components must be added to produce the effective address, a second pass through the address-generation circuitry and an additional cycle are required. A single level of indirect transfer orders is provided via hardware. For indirect transfers, the effective address generated as indicated above represents the location (i.e., address) of the address to which the instruction transfers. Since a memory read must be performed, either one or two additional cycles are required. Vector table addressing is the method of indirect addressing used to address locations external to a program unit. The location of the external address is formed as the sum of the beginning of the vector table (wired address) and the data field of the instruction that serves as an index into the vector table of addresses.

Interject transfers, like vector table transfers, employ wired addresses. The interject facility is discussed in Section 3.3.

A 64-word pushdown stack for return addresses is provided to facilitate subroutine transfers. The top of the stack is the stack (S) register in the central control, and the remainder of the stack is in a reserved location in call store. Before a return address is saved in the stack register, the previous contents of that register are stored in the call store location pointed to by the stack counter. The stack (push) option requires an extra cycle to process. The pop return-address option returns the last data stored in the call store stack to the S register. The pop option requires an extra cycle to process only if the call store is a 1400-ns unit. Automatic increment or decrement of the stack counter is performed with check for overflow or underflow each time the stack is pushed or popped.

The J return-address option, provided chiefly for compatibility with No. 1 ESS translated programs, saves the return address for successful transfer instructions in the J register, which is one of the general-purpose registers. In general, execution of the J option, which saves the address of the instruction immediately following the transfer, does not add to the number of cycles required to process an instruction.

Three index-register-modification options are available: add-one (A) option, set-register (S) option, and word (W) option. These options are available on a subset of the long instructions, some of which offer only the A option due to limited code space.

The add-one option causes the contents of the index register to be incremented after the contents are used in the indexing operation. On a store instruction, if the index-register field is null, the A option is in-

terpreted as add one to memory. In this case, the register contents to be stored are incremented before the data are stored. On certain conditional transfer orders with the A option specified, the test register is incremented after the test has been performed but only if the test is successful.

For the set-register option, the index register is not used in address generation. Instead, the specified register is set to the value of the effective address. The word option is the same as the set-register option except that the contents of the index register are used in address generation in that case.

Index-register modification does not add to the time required to execute an instruction.

3.2.2 Typical instruction timing and cycle counts

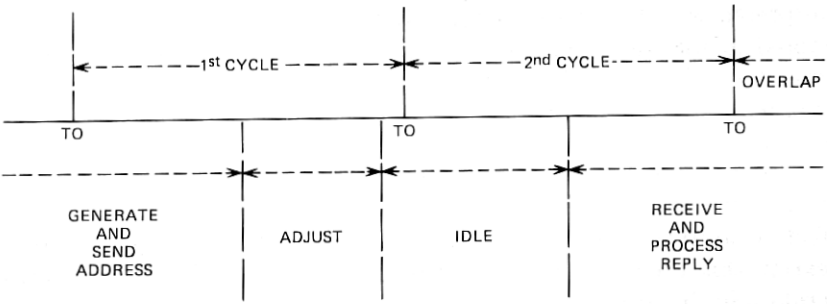
In several cases, two normally separate instructions are encoded as a single instruction. These "combined" instructions are designed to make efficient use of the time that the CC execution circuitry is normally idle on memory-access instructions to 1400-ns memory units waiting for either a data reply (read) or a completion indication (write). During the otherwise idle time, a shift or rotate operation is performed on the data in another (usually unrelated) general-purpose register.

A combined instruction may perform a load, add, compare, product, union, or exclusive-OR read operation. The data reply from the store may be product masked with the contents of the mask register and/or complemented before the main logic operation (add, compare, etc.) is performed. Alternatively, the instruction may perform a write operation with either product or insertion masking with the contents of the logic register. Figure 5 illustrates how the adjust operation is fit into the idle data processing phase.

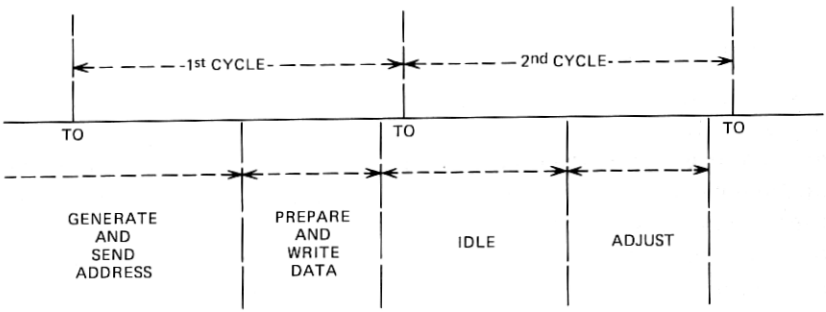
Minimum cycle counts for various types of orders are shown in Table II. The actual execution time for a program depends on conflicts between fetch and execution control, instruction mix, and auxiliary unit data transfers.

3.2.3 Typical instruction encoding

As previously mentioned, the length of an instruction is often determined by the length of the required data fields or by the particular options specified. Figure 6 shows the encoding of two typical conditional transfer orders: one short, the other long. Both of these instructions offer relative addressing, but the single-word order allows less range. The double-word order allows indexing and/or indirect addressing, while the single-word does not. Both orders provide a stack return-address option, but only the long instruction offers a J-return-address option or an

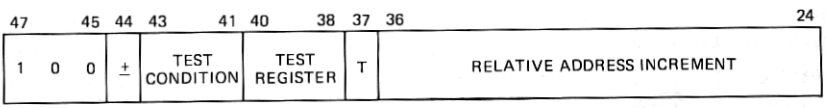


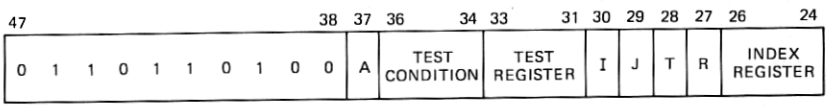
(a)



(b)

Fig. 5—Combined instruction timing. (a) Combined load. (b) Combined store.





GLOSSARY

A ADD ONE OPTION
I INDIRECT ADDRESSING
J J RETURN ADDRESS OPTION
T STACK RETURN ADDRESS OPTION
R RELATIVE ADDRESSING

Fig. 6—Short and long instruction encoding.

Table II — Typical cycle counts

Operation	1400 ns Store	700 ns Store
EA ⊛ REG → REG	1	1
MEM ⊛ REG → REG	2	1
EA → MEM	2	1
Transfer direct	2	1
Transfer indirect	4	2
Combined	2	2
Search (N words)†	2N + 1	N + 1

EA Effective address.
MEM Any memory location.
⊛ Operation.
Note: Add 1 cycle if order is both relative and indexed.
† The search instruction facilitates a sequential search of memory for a selected code.

add-one option. For the particular long transfer selected, the A option causes incrementing of the test register rather than the index register if the test is successful.

3.2.4 Peripheral operations

Central control monitors and controls the peripheral system by transmitting information to the peripheral units (PUs) which perform the functions of scanning, signal distribution, and network control. The CC communicates with the PUs by one of three methods. The first method, coded enabling, involves addressing PUs by transmitting an enable code to be accepted by the appropriate unit in a manner analogous to store operations. The second method, CPD enabling, requires a central pulse distributor such as that used in No. 1 ESS. Polling, the last method, employs a maintenance control pulse to obtain status from several PUs simultaneously.

Peripheral units may be located as far as 137 meters and 385 meters from the CC for CPD and coded enabling, respectively. Upon initiation of a PU operation, control is passed from instruction-execution control to peripheral-operation control, and nonconflicting instructions may be processed during the remainder of the peripheral operation. Conflicting instructions are those that either use or change the logic register, peripheral data register, or peripheral enable address register. If execution control encounters a conflicting instruction, it stalls until informed by peripheral operation control that the peripheral operation is complete.

Coded enable operations are performed on a variable-cycle basis, except for a limited class of orders intended for line scanning which are executed at a fixed rate. Variable-cycle operations were adopted to gain

speed during peripheral operations. The first two cycles of coded-enable operation are used to set up the enable address and peripheral data registers and may require a store access. Address and data are transmitted in the third cycle and the CC waits as long as 30 additional cycles for a response. If no completion indication is received by the end of that interval, an interrupt flag is set. CPD enable operations are performed on a fixed-cycle basis with different types of operations requiring different numbers of cycles. Operating on a predetermined cycle basis facilitates the overlap of repetitive operations. When operating in an overlap mode, repetitive scanning operations typically require either 10 cycles (trunk or line scanning) or 11 cycles (digit scanning), and nonscanner operations repeat in 14 cycles. In a typical sequence, the first two cycles are required to set up CC registers. Address information is sent to the CPD in the third cycle. The CPD transmits the peripheral address shortly thereafter, and the CC awaits completion of the operation. On scanning operations, a data reply into the logic register is expected.

Each PU which receives a polling control pulse from central control returns a status indication such as whether or not a data buffer is full. Units respond into uniquely defined positions in the reply register. A PU polling operation consumes 32 cycles unless it is terminated early by encountering a logic-register-usage instruction which, in effect, allows a software-controlled early termination.

3.3 Interrupt system

An interrupt causes program control to be passed from the current program to the program corresponding to the interrupt level. The higher priority interrupt levels are maintenance oriented whereas the lower levels are processing oriented. The immediacy of interrupt recognition by central control is a function of the type of interrupt (maintenance or processing), the level of the interrupt, whether the software-controlled inhibit is set, and the instruction sequence currently being executed. Under certain conditions, for example, couplets of instructions or AU data transfers may be allowed to complete before an interrupt is recognized.

When an interrupt is recognized by central control, the interrrupt system saves vital information in memory that might otherwise be destroyed before it could be saved via software control. This vital information includes the return program execution address, the contents of the data buffer register, and the value of the control flip-flops. The last program word fetched (ABL/ABR) is saved if the error indication points to a problem encountered on that fetch operation. On maintenance levels, the interrupt system also freezes a group of save registers which are useful for fault recognition (save data address, save program address,

save current address, and so on). When a CC mismatch is detected, the matchers are halted to preserve the mismatched data. Finally, the interrupt system initializes fetch and execution control and sets up a transfer to the appropriate interrupt level program. Having completed this sequence, the interrupt system releases fetch and execution control and allows software to perform its tasks. Next, a go-back-to-normal sequence is initiated which restores the CC to its preinterrupt state. The severity of the interrupt will determine whether or not the interrupted program is continued.

For fault situations, it is important that the level of interrupt taken be the lowest consistent with the failure indication. A main memory parity failure, for example, is not allowed to cause a CC mismatch since the latter is a higher level of interrupt.

The processing level interrupts are provided in the CC to facilitate the administration of software tasks such as input/output which must be performed within specified intervals of time. Similarly, the interject facility is used to efficiently interpose priority processing in the normal cycle of program tasks. Flags (flip-flops), which indicate that interject tasks need to be performed, may be set in the CC either by program control or by hardware in various units external to the central control (e.g., auxiliary units) that require attention. Some of these interject flags are maintenance oriented but are deferrable and should therefore not set a maintenance-level interrupt source. Certain base-level programs utilize transfer instructions which test the interject flags. An automatic check is made on this mechanism by a processing level interrupt which occurs if no interject work is performed within 10 ms.

3.4 Auxiliary unit data transfers

Direct handling of the data involved in the auxiliary unit (AU) data transfer tasks by CC execution control would require a significant amount of real time and, therefore, the control of these data transfers is handled largely by separate control on a cycle-stealing basis.

Although the AU system is connected directly to the central controls, the CCs do not have direct access to the data stored on the disks and tapes. Instead, the requests that data be transferred between bulk storage and main memory are placed in main memory by administrative software. The auxiliary unit controller receives the location of request details and initiates the data transfer. Once a job is initiated, the AU system processes the request by transferring one word at a time through the CC, allowing the CC to resolve the bus conflicts between itself and the AU. The CC gives the AU bus priority about 25 percent of the time to keep AU buffer overflows within bounds.

3.4.1 AU priority system and signaling

Central control can accommodate up to 16 auxiliary unit interface ports. In the case of multiple requests from AUs for bus usage, the CC administers bus granting on a priority structure. The AU ports are divided into two groups of eight members each. Any AU within the first group may, under program control, be assigned highest priority; members of the second group have a wired priority that is lower than that assigned to any members of the first group.

Each AU memory operation requires three cycles. The first cycle is used to establish communication between the CC and an AU. The following two cycles correspond to the actual memory operation during which the AU interface control in the CC controls buffering of the address and data information. The buffering is performed so that all memories appear to the AU to be 1400-ns memories.

Auxiliary units requiring access to the store submit requests to the CC near the beginning of the first cycle. The AU interface control in CC sends an enable to the requesting AU with the highest priority. The enabled AU replies with address information and a verify signal. If the communication path is available, a storage-access-permitted signal is sent to the AU and the AU returns a recognition signal. If the communication path is not available, a storage-access-permitted signal is not sent and the AU is blocked and must request again.

3.4.2 Bus usage administration

The operations of instruction fetching, data fetching and storing by execution control, and passing of information between main and auxiliary storage all require access to main memory. To conserve real time, execution control is normally given highest priority, fetch control is given second priority, and the AU system is normally relegated to lowest priority. However, the number of AU data-transfer tasks that must be aborted due to overflow of buffers is minimized by allowing AU system bus usage if this system is blocked in three successive attempts to obtain bus usage. The fetch or execution control may use the bus (call or program store) not required by the AU system since only a single AU task is allowed memory access at any time.

3.5 Address structure and memory protection

An address containing 22 bits allows accessing memory of up to 4.2 million words. For compatibility with No. 1 ESS programs, an alternate mode is available under program control that uses only 21 bits. In either case, the lower half of the address spectrum is assigned to call store. Except for a few thousand addresses near the top of the spectrum (which

are used for direct access of CC registers, file store controller registers, and so on), the upper half of the spectrum is assigned to program store.

It is essential to protect areas of the memory spectrum which are not duplicated within main memory such as program and translation stores, and areas which vitally affect system operation such as CC internal registers, disk controller registers, main program constants, interrupt bins, etc. Therefore, the memory address spectrum is divided into protected and unprotected areas each requiring different write instructions. An interrupt results if the address is incompatible with the instruction. Typically, only a portion of call store lies in the unprotected area. The lower- and upper-protected area registers in the CC, which define the secure and unsecure areas of the spectrum, are software controllable via secure write instructions.

IV. PROCESSOR COMMUNICATIONS

4.1 Definitions and characterization

Communication between units of the 1A Processor and between the 1A Processor and the network take place over a system of high-speed digital communication channels called buses. Each bus is made up of a number of parallel channels and each channel consists of a balanced wire pair and associated transmitting and receiving circuits. Transformer coupling is employed between the bus and the transmitting and receiving circuits.

Information on a bus is represented by the presence or absence of pulses on the individual wire pairs. Information is passed between units in parallel form with pulses sent simultaneously over a group of wire pairs.

4.2 Objectives of the bus system

The 1A Processor bus system was designed to meet two primary objectives: to provide reliable communications and to be compatible with the No. 1 ESS peripheral units. To meet these objectives, the various bus groups are duplicated as shown in Fig. 7. Bus duplication methods are covered in a later section. In addition to required information channels, each bus provides channels for parity bits which are used for error detection. Pulse transformers help to achieve reliable bus operation. These pulse transformers have electrostatic shields and high-voltage breakdown characteristics for protection against longitudinal noise signals and induced currents or voltages caused by lightning strikes or power faults. Further, failure-conducive stress conditions on bus circuit devices are avoided by using devices well within allowable operating ranges.

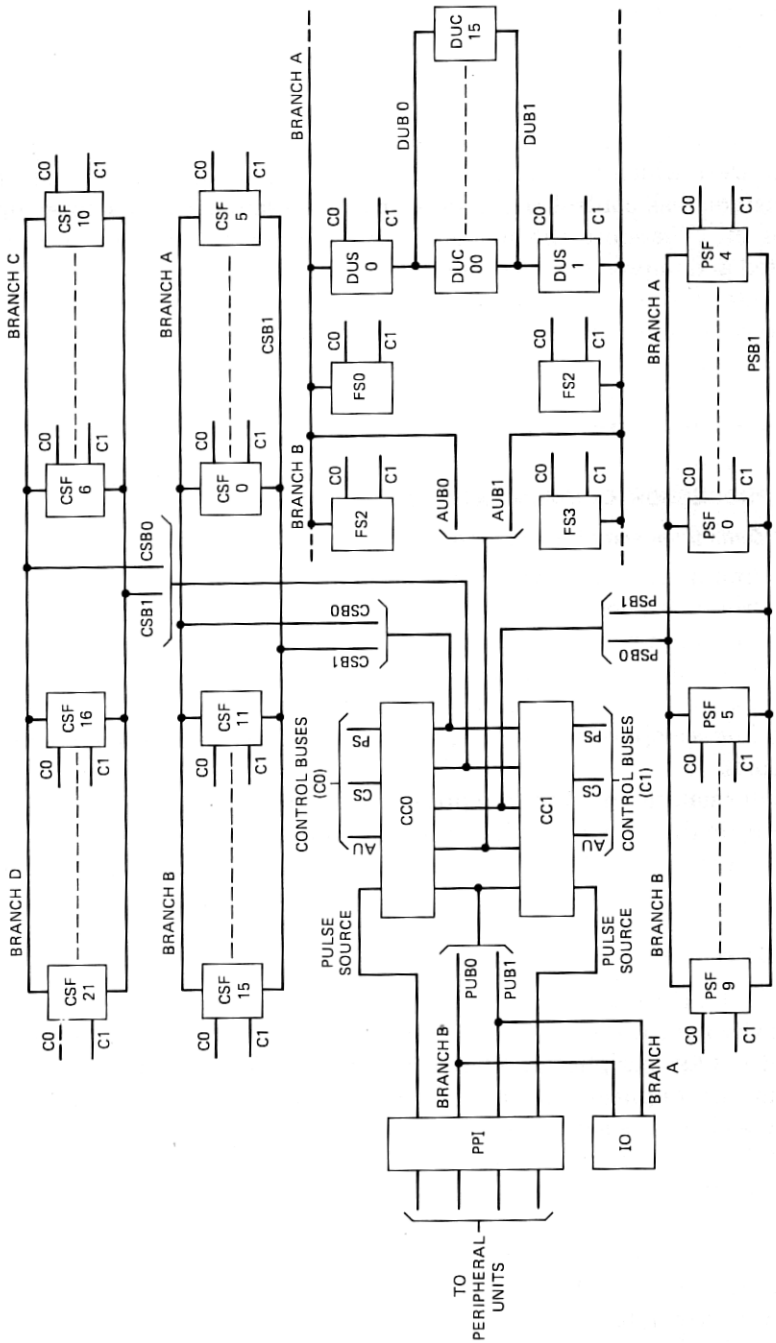


Fig. 7—Bus interconnections.

Another bus system objective is to achieve a speed compatible with the 1A Processor digital circuits. A cable with irradiated polyvinyl-chloride (IPVC) insulation is used between frames. This cable has controlled propagation delay and pulse-dispersion characteristics to minimize delay variations and degradation of pulse rise times. Within the processor frames, Teflon*-insulated cable is used for proper impedance matching and low propagation delay.

Bus driver and receiver circuits are also designed to minimize propagation delays and delay variations. This has been made possible by placing all circuit components for drivers or receivers on a circuit board and by the electrical design itself.

Another important feature is ease of bus cable installation and improved maintenance features. Use of pretested, connectorized bus cables between frames permits quick initial installation and simplifies subsequent bus growth for system additions. Connectorized cables also eliminate the common problem of wiring bus pairs incorrectly during bus installation.

Bus repair time is reduced with the use of connectorized cables and with bus circuit components mounted on plug-in circuit packs. To troubleshoot bus problems, access points are provided on bus-termination resistor assemblies. These resistor assemblies are connectorized and slip over the connectors of bus packs located at the ends of a bus.

4.3 Bus configurations

Two types of buses, shared and private, are used for communication between units of the processor. Shared buses branch out from the central control in two directions and are multipled from frame to frame, carrying data that is common to all frames on a bus. Private buses are connected between the CC and individual frames and are used for synchronization and control information. Cable assemblies consisting of eight wire pairs are used for both private and shared buses.

4.4 Bus system characteristics

The various bus groups are arranged into two main bus systems, the processor-unit bus system and the peripheral-unit bus system. Buses interconnecting units of the processor are included in the processor system. The peripheral bus system consists of buses providing communication between the processor and peripheral frames.

The processor bus system is designed for minimum delay. This objective was achieved by using the following:

* Tradename of E.I. Dupont de Nemours & Co.

(*i*) A high-speed cable driver to minimize circuit delay and recovery time.

(*ii*) A passive cable receiver capable of extracting sufficient energy from the bus to drive a low impedance without introducing an impedance discontinuity and unwanted pulse reflections.

The properties of the processor bus system are tabulated in Table III. A pulse amplitude of 7.2 V is determined by the minimum signal required by a receiver at the end of a bus having the maximum number of 10 receivers connected to it. Adequate pulse width at this receiver under worst-case timing variations is provided by a nominal 100-ns-wide pulse. The narrow pulse allows use of 700-ns memory units as well as 1400-ns memory units.

To minimize the worst-case bus cable delay on call-store and program-store buses, these cables are routed in a loop between the central control and the stores. That is, address and write cables are connected to the first store frame on a bus branch while the data-reply cables are run directly from the furthest frame. The shorter length of the reply cable compensates for the furthest store having the longest bus distance for address and write data.

The peripheral unit bus design provides an adequate signal over a long bus having up to 50 receivers connected to it. This design was implemented by the use of the following:

(*i*) A cable driver that provides a 500-ns bus pulse to compensate for pulse dispersion (rise-time degradation) and bit-to-bit timing variations inherent with long, fully loaded buses.

(*ii*) A cable receiver with an active gate for bus signal amplification, resulting in a minimum amount of energy being extracted from the bus.

Table III also lists the peripheral bus system characteristics. For compatibility with the No. 1 ESS peripheral units,[1] the 9.3-V pulse amplitude provides a sufficient signal for 50 receivers over a 137-m bus length. A minimum cycle time of 2800 ns was determined to be the time needed for bus circuits to recover, and permit system execution of peripheral instructions at maximum speed.

Bus length, as shown in Table III, may be increased by the use of a repeater for regenerating bus pulses in order to reach a maximum 385-m length with up to 50 receivers on the bus.

4.5 Bus system organization

Four bus groups make up the processor unit bus system: the call store bus (CSB), program store bus (PSB), auxiliary unit bus (AUB), and data unit bus (DUB). The call store bus group consists of two buses or four

Table III — Bus characteristics

Bus	Bus Pulse Amplitude (v)	Bus Pulse Width (ns)	Min Cycle Time (ns)	Max Loop Length* (m)
Call store	7.2	100	700	18.6
Program store	7.2	100	700	18.6
Auxiliary unit	7.2	100	700	30.5
Data unit	9.3	500	2800	143.2
Peripheral unit	9.3	500	2800	274.3†
				769.4‡

* Bus length includes path length through connectors and circuit packs.
† With 50 receivers connected.
‡ With bus repeater.

Table IV(a) — Call store bus layout

Bus Group	Bit Names	Function/Definition
Call Store Address	CSA00 → CSA15	Data location address
	CSA16 → CSA20	K-code
	CSAMP	Address parity
	CSA3T5	Timing
	CSA1T3	Timing
	CSAM	Maintenance mode
	CSAC	Control mode
	CSAW	Write mode
	CSAR	Reply mode
Call Store Write	CSWE	Write enable
	CSWP2	Data parity
	CSWP1	Data parity
	CSW00 → CSW23	Data
Call Store Reply	CSR00 → CSR23	Reply data
	CSRP2	Reply parity
	CSRP1	Reply parity
	CSRAW	All seems well
	CSRAF	All seems well fail
Call store Clock Pulse source	CS0CLK	Bus 0 synch (5T7)
	CS1CLK	Bus 1 synch (5T7)
	PSCS	Pulse source
Call Store Status display	Update	Update status display
	CSDSK0 → CSDSK4	K-code F/F's
	CSDSMT	Maintenance F/F
	CSDSR0	R0 F/F
	CSDSA0	A0 F/F
	CSDSA1	A1 F/F
	CSDSVF	Update verify

branches to accommodate the full complement of call stores. The DUB differs from the other processor buses in that it interconnects data-unit selectors within the processor and data-unit controllers that are physically located beyond the processor area. As a result, the maximum DUB

Table IV(b) — Program store bus layout

Bus Group	Bit Names	Function/Definition
Program Store Address	PSA00 → PSA15	Data location address
	PSA16 → PSA20	K-code
	PSAMP	Address parity
	PSA3T5	Timing
	PSA1T3	Timing
	PSAMT	Set maintenance F/F
	PSAM	Maintenance mode
	PSAC	Control mode
	PSAW	Write mode
	PSAR	Read mode
Program Store Write	PSWE	Write enable
	PSWP2	Data parity
	PSWP1	Data parity
	PSW00 → PSW23	Data
Program Store Reply	PSRR00 → PSRR23	Right word reply data
	PSRRP2	Right word parity
	PSRRP1	Right word parity
	PSRL00 → PSRL23	Left word reply data
	PSRLP2	Left word parity
	PSRLP1	Left word parity
	PSRAW	All seems well
	PSRAF	All seems well fail
Program Store Clock-Pulse Source	PS0CLK	Bus 0 synch (5T7)
	PS1CLK	Bus 1 synch (5T7)
	PSPS	Pulse source
Program Store Status Display	Update	Update status display
	PSDSK0 → PSDSK4	K-code F/F's
	PSDSMT	Maintenance F/F
	PSDSR0	R0 F/F
	PSDSA0	A0 F/F
	PSDSA1	A1 F/F
	PSDSVR	Update verify
Program Store Processor Configuration	VPSPCB	Set PS0 K-code
	R0PCRB	Reset R0 F/F
	R0PCSB	Set R0 F/F
	MTRCB	Reset maintenance F/F

Table IV(c) — Auxiliary unit bus layout

Bus Group	Bit Name	Function/Definition
Auxiliary Unit Address	AUA00 → AUA10	Control register address
	AUA11 → AUA15	K-code
	AUAR	Read mode
	AUAW	Write mode
	AUACTL	Control mode
	AUAPKA	Address parity
	AUASYNC	Synch
Auxiliary Unit Write	AUW00 → AUW23	Data
	AUWP2	Data parity
	AUWP1	Data parity
	AUWAWF	All seems well fail
	AUWASW	All seems well
Auxiliary Unit Store Address	AUS00 → AUS15	Data location address
	AUS16 → AUS20	Store K-code
	AUS21	CS/PS selection
	AUSR	Read mode
	AUSW	Write mode
	AUPUC	Control mode
	AUSPKA	Address parity
Auxiliary Unit Reply	AUR00 → AUR23	Reply data
	AURP2	Data parity
	AURP1	Data parity
	AURASW	All seems well
Auxiliary Unit Control	AUSP	Timing
	PCAU	Pulse source initialize
	PAUSE	Pump program store (FS0/FS1) System reinitialize (DUS0/DUS1)
	PAUMT	Maintenance
	SAP	Storage access permitted
	PAUEN	Bus enable
	AUBR	Bus request
	AUEV	Verify enable
	AUPUC	Pump complete (FS0/FS1) SR complete (DUS0/DUS1)
	AUOPIJ	Paging complete (DUS only)
	AUITJ	Maintenance request

length is much greater than for other processor buses, and the DUB characteristics are the same as the peripheral unit buses.

Within each bus group there are address, write, reply, and control buses. For the CSB, PSB, a status display bus is also included. Table IV tabulates bit layouts for each of the processor unit buses.

One bus group makes up the peripheral unit bus system but a subset of this group is used as a group in itself. This two-group arrangement permits the processor to be used with peripheral systems employing either central pulse distributor enabling or coded enabling. The bit layouts for each system are shown in Table V.

Table IV(d) — Data unit bus layout

Bus Group	Bit Name	Function/Definition
Data unit Address	DUA00 → DUA05	DUC register address
	DUAR	Read/write mode
	DUAC	Control mode
	DUAP	parity
	DUACS	Clock synch
	DUAES	End of sample
Data unit Write	DUW00 → DUW23	Data
	DUWP	Data parity
Data unit Reply	DUR00 → DUR23	Reply data
	DURP	Reply data parity
	DURAS	All seems well
Data unit Control	MI	Maintenance interject
	OI	Operational interject
	R	Read
	W	Write
	P	Parity
Data unit System Reinitialization	DUSRR	System reinitialization ready
	SRNTRY	System reinitialization not ready
	SRERR	System reinitialization error
	DUSRC	System reinitialization complete
	DUSSRA	System reinitialization activate
Data unit Select	DUS	DUC select
	DUSV	DUC select verify

4.6 Processor peripheral interface

Connections from the processor to the peripheral system are made at the peripheral processor interface (PPI) frame as shown in Fig. 7. All processors are equipped with branch B of the peripheral unit bus connected to the PPI. This provides a connection point for the peripheral system. For systems requiring bus lengths greater than that provided by one branch, branch A of the peripheral unit bus is also connected to the PPI. In addition, the PPI serves as the distribution point for the peripheral pulse source buses. These are groups of control pulse pairs. Each pair is connected to one peripheral frame in a manner similar to the private control buses used within the processor units.

4.7 Bus circuit details

Cable drivers and receivers provide the interface between logic circuits within frames and the interconnecting bus cable, as shown in Fig. 8 for central control and call store frames. Driver circuits are shunt-connected to the bus using transformer coupling. Receiver transformer connections are made in shunt or series-shunt as shown. A series-connected receiver

Table V(a) — Peripheral unit bus layout—coded enabling

Bus Group	Bit Name	Function/Definition
Peripheral Unit Enable address	PUEA00 → PUEA12	Enable address
	PUSYNC	Enable address synch
	IOCH	IO clock high
	IOCL	IO clock low
Peripheral Unit Write	PUW00 → PUW36	Data
	PUW37	*
	PUPOD	Odd address parity
	PUPEV	Even address parity
Peripheral Unit Reply	PUR00 → PUR23	Reply data
	PUASW	All seems well
	PURP	Reply data parity
	APUF	Autonomous PU F level
	APUT	Autonomous PU trouble
	APUB	Autonomous PU base level
Peripheral Unit Control	MI	Member interrogate
	GI	Group interrogate
	CLKI	Clock interrogate
	PHP	Poll high priority buffer
	PLP	Poll low priority buffer
	PSZ	Poll seizure buffer
	PDG	Poll digit buffer
	PIO	Poll input output
Peripheral unit Loop around	M0	Mapping bit 0
	M1	Mapping bit 1
	M2	Mapping bit 2
Maintenance	X	Execute

* Cabled to, but not used by system.

is used on the peripheral bus. For protection against longitudinal noise signals, a balanced bus is used.

Diodes D3, D4 provide isolation between drivers and also minimze any bus unbalance due to drivers. Circuit recovery time is controlled by D1 and D2. To protect the output transistor Q1 from burnout due to a stuck driver input, capacitor C1 is provided. With normal inputs applied, C1 is charged to a low voltage during the pulse interval and discharged between pulses by the recovery gate (REC). If an input is permanently low, C1 is charged through R1 to +3 V, back biasing Q1 and turning it off in time to protect against any damage. This protection feature is tested by holding the test (TEST) lead low, which inhibits all eight driver circuits on a pack and activates the monitor (MON) gates causing the test monitor (MON) lead to go low.

The inhibit action of the test lead also serves the useful function of controlling cable drivers when bus circuit power is turned down or turned on. This control is necessary to prevent false pulsing on the bus during these intervals. In frames that have one common power switch such as the CC, power control circuits cause the test lead to be held low prior to

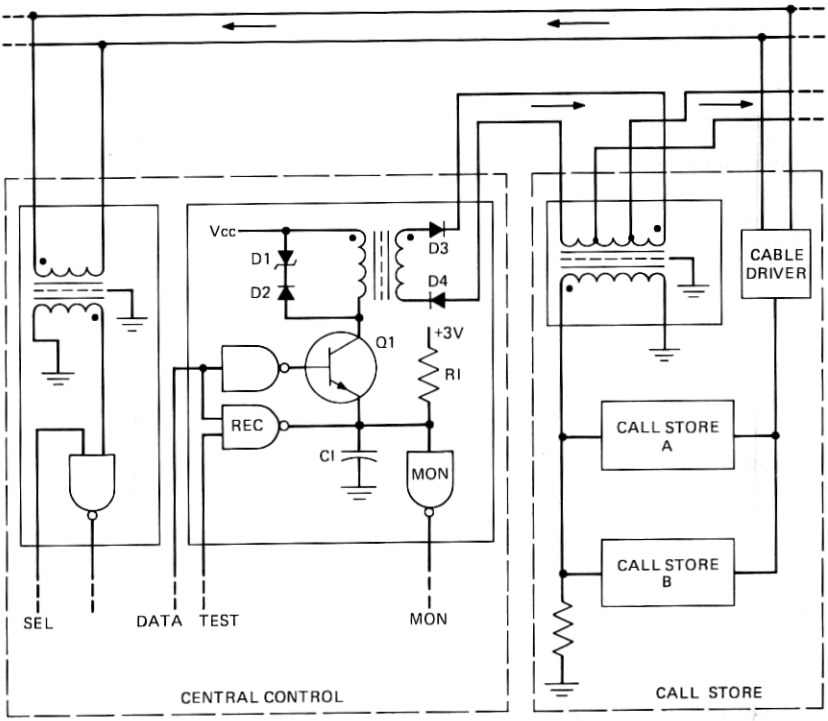


Fig. 8—Central control/call store circuits.

removal of V_{CC} or after application of V_{CC} for an interval sufficient to insure against false outpulsing. In frames that have a separate power switch for bus circuits such as CS and PS, control of the test lead is not necessary if the +3 V transition time is greater than 1 ms. In some frames, program control of the test lead is provided to select and inhibit cable drivers.

Cable receivers in processor frames other than the CC employ a series shunt-connected transformer to maintain impedance matching on the bus and also to provide drive capability into a low impedance. This is important in frames such as CS, PS where a 100-ohm intraframe bus interconnects individual memory units and the receiver output. These receivers contain eight circuits per pack.

Shunt-connected receivers are used in the CC to allow communication to one receiver from both branches of the reply buses. These receivers also contain logic gates that provide for selection of all 16 bits or individual selections of 2 bits on a pack.

To minimize the power loss on the peripheral buses, a series-connected transformer with an output buffer is used as a receiver. The high turns

Table V(b) — Peripheral unit bus layout—CPD-enabling

Bus Group	Bit Name	Function/Definition
Peripheral Unit Enable Address	PUEA00 → PUEA12	Enable address
	PUSYNC	*
	IOCH	*
	IOCL	*
	PUEA13 → PUEA31	Enable address
	T	CPD test
	R	CPD reset
	BC	CPD bus choice
Peripheral Unit Write	PUW00 → PUW37	Data
	PUP0D	*
	PUPEV	*
Peripheral unit Reply	PUR00 → PUR15	Reply data
	PUASW	All seems well
CPD Verify answer	VA00 → VA23	Verify answer
	ASWCPD	CPD-all seems well
CPD Execute	EX00 → EX15	CPD execute
CPD Diagnostic Echo	EXR00 → EXR15	CPD execute return (Echo)
	A PAR	CPD A parity
	B PAR	CPD B parity
	C PAR	CPD C parity
	M1	CPD maintenance
Peripheral unit Loop around Maintenance	PULAM0	Mapping bit 0
	PULAM1	Mapping bit 1
	PULAM2	Mapping bit 2
	PULAX	Execute

* Cabled to, but not used by system.

ratio of the transformer (1:25) reflects a very low impedance in series with the bus; this keeps the power loss to approximately 0.05 dB per receiver. A receiver pack contains eight circuits.

4.8 Cable details

An 8-pair switchboard cable with an irradiated polyvinylchloride (IPVC) insulation is used for the communication bus. Use of the IPVC

insulation provides improved thermal properties and allows cables to be soldered to connectors. The cable has a characteristic impedance of 100 ohms and a maximum propagation delay of 5.75 ns/m. Pulse dispersion of the cable is specified over a 21.3-m length for an input pulse of 5-ns rise time, with the increase in rise time of the output pulse to be less than 5.5 ns.

V. MAN-MACHINE INTERFACE AND CONTROL

The 1A Processor has been designed with a flexible, expandable control and terminal access capability and a common hardware interface for both No. 1 and No. 4 ESS peripheral networks. To tailor the control and terminal features to the specific using system, the processor utilizes two hardware subsystems. These are the processor peripheral interface and master control console (PPI/MCC) and the input/output (IO) frame. They are located on the processor peripheral bus.

Splitting IO and MCC frames is a departure from the No. 1 ESS processor which combined the maintenance TTY channel as part of the master control center. In No. 1A ESS, maintenance data are provided on several channels. Each channel is implemented through the IO subsystem, which may grow in channel capacity as needed. The AMA tape unit is not located with the 1A Processor PPI/MCC, but is supplied as a separate unit on the auxiliary unit bus system. The memory card writer system, which is part of the master control center of the No. 1 ESS processor, is not needed since all 1A Processor memory is locally writable.

5.1 The PPI/MCC

The PPI/MCC complex serves several functions. First, it provides the control panel interface containing manual keys and displays. The keys and displays of the PPI/MCC may be divided into three groups. The first group includes all critical control and display functions required for manual system configuration and recovery control during emergency conditions when the automatic features have failed. The functions include system reinitialization, hardware reconfiguration of the duplicated processor, partial software system initialization, and processor status indications. The second and third groups of keys and displays are the processor and specific using systems panels which provide direct data entry and display, bar graphs of network activity, and system and network status indicators.

A second PPI/MCC function is to act as the single processor interface to the peripheral network. All peripheral bus interconnections are terminated at the PPI, which facilitates testing, installation, and No. 1 ESS retrofit. The third major function of the PPI is to provide power scanning

and power switch lamp control, continually monitoring the status of power switches in the various processor frames. Each power switch is equipped with two scan points indicating four frame states: power on and in-service, a manually initiated state in which the frame is requested out of service (ROS), a manually powered down state, and a state indicating frame power removed by fault-sensing power circuits in the frame. Two lamps are driven from the PPI for each power switch. The acknowledge lamp (ACK) is used to indicate that the processor has recognized a manually requested change in frame state, and the out of service (OS) lamp is used to indicate that the frame has been removed from active service in the system.

The scan data are accessed routinely by the processor operating system. Thus, the frame can be smoothly moved from "in service" to "standby" status at the request of craft. Scan data are also used to initiate system reconfiguration routines to recover from the loss of an automatically powered down frame and contain the current list of available frames for use by system reconfiguration and recovery programs.

5.2 PPI/MCC configuration

The processor peripheral interface and master control console are housed in two separate frames. All keys and lamps are located on panels in the master control console and are driven and powered over cable from the PPI, which may be up to 61 m away. The MCC can be placed in a convenient location, usually with other maintenance equipment. Figure 9 shows the PPI and MCC configurations for the No. 4 and No. 1A ESS applications.

5.3 Master control console

The MCC consists of three panel sections containing keys, lamps, LED displays, and terminal connections. Of these three sections, two are common to both using systems and deal only with processor control and display functions, the processor display panel and the processor request panel. The third panel section is a system application panel, which meets the network status and key input needs of the particular using system.

All controls and displays critical to manual configuration and recovery control of the 1A Processor are located on the processor display panel, shown in Fig. 10. This panel is further divided into the processor display, processor update, override control, system reinitialization, and processor configuration sequencer sections. The processor display group of switches, lamps, and LED displays enables craft personnel to monitor basic processor configuration including CC status, bus configurations, and file store status, and request and visually display the store enable

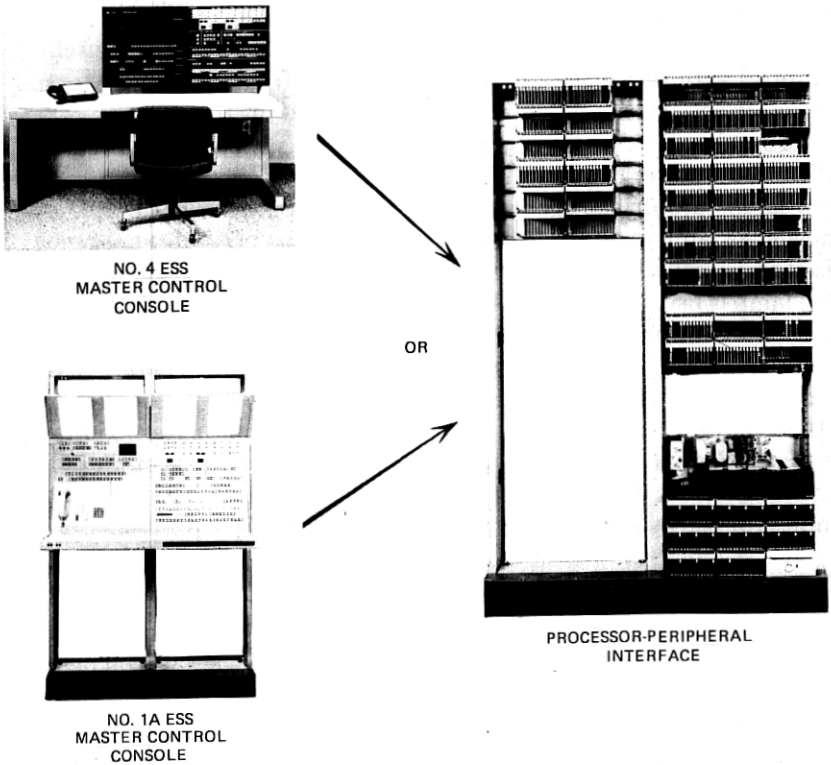


Fig. 9—Master control console and processor-peripheral interface.

code, status, and routing information of any CS/PS unit in the processor. The update group of keys and lamps enables craft personnel to observe when a program update is in progress and prevent the use of a specific file store for emergency recovery. This is used to ensure that a partially written file store is not used during system reinitialization. The override control group of switches and lamps enables craft personnel to manually activate a CC, auxiliary unit bus, program store bus, and basic program store configuration for emergency system recovery. The system reinitialization group of switches and lamps enables craft personnel to manually initiate system reinitialization in conjunction with either the override control or processor configuration sequencer group of switches and lamps. The processor configuration (PC) sequencer group of switches and lamps enables craft personnel to monitor the progress of the PC sequencer during either automatic or manual system recovery using the PC circuit in each CC and manually control the PC circuit for system recovery, or system reinitialization in conjunction with the system reinitialization controls.

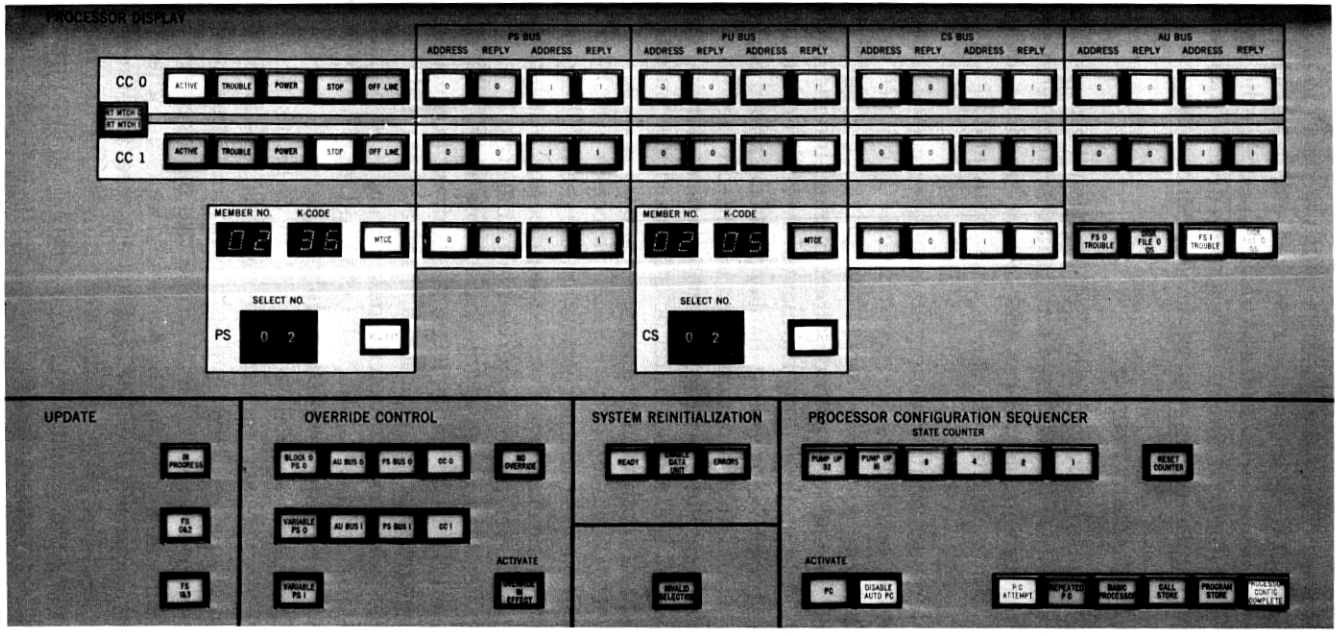


Fig. 10—Master control console display panel.

The processor request panel provides manual interaction with the system software. This panel permits craft personnel to make various A-level interrupt program requests, request and display various system data in binary or numeric form, and monitor or check the status of various MCC indicators and circuits. This panel is divided into two sections: manual interrupt program request and system display.

5.4 Processor peripheral interface

The PPI frame is divided into four functional groups: the critical control and display circuits used for manual system recovery, the peripheral bus circuits, sequencer and memory-readout logic, and the memory and circuits controlling key, lamp, scan, and signal distributor points.

The PPI frame contains memory elements, logic level drivers and receivers, and AC bus drivers and receivers, directly coupled to the central control, file store, and CS/PS stores over private bus paths. These interconnections and logic provide direct manual interaction with processor operation implementing the manual override and reinitialization functions of the MCC processor display panel described earlier. Of the displays on this panel, program access is limited to read only. This ensures that software interaction does not take place during periods when manual control is exercised. Interlocks are used to ensure that invalid manual control combinations are not selected.

Unlike other peripheral bus units, all processor peripheral bus leads terminate in the PPI (whether or not they are used). This provides a bus test capability and single interface to the processor. Much of the bus area of the PPI is, therefore, connectorized terminal strip.

The PPI itself is a coded enable logic frame as are all No. 4 ESS peripherals and the 1A Processor IO frame. Address, instruction, and data are sent down the peripheral bus, and the address is decoded. On an address match, the operation on the data is completed, and if all internal checks are satisfied an all seems well (ASW) and reply parity are returned on the answer bus, along with any reply data.

Most PPI logic is associated with two 32- by 24-bit matrices of flip-flops and status receiver logic; the access to these matrices is achieved by the peripheral bus, lamps, keys, and scan point and other status lead receivers, as represented in Fig. 11. The two matrices are completely symmetrical and, except for the read-only scan and other status indicators, are comprised of either toggle or set-reset flip-flops which are, with few exceptions, PU bus readable or writable. The first matrix is used for controlling the various lamps and storing the key inputs from the various MCC panels; it also provides PU bus read-only access to the configuration status of key processor elements. Approximately one-half

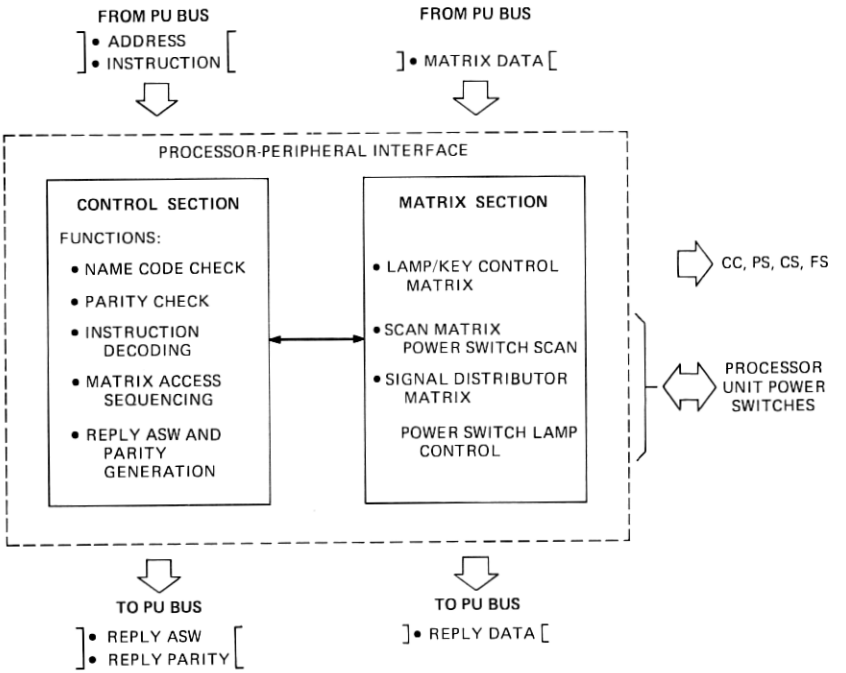


Fig. 11—Processor-peripheral interface-operational overview.

of the 672 flip-flops of this matrix are reserved for common processor panel control and display. The second half is available to the using system for its own panels.

The scan and signal distributor (SD) matrix can be subdivided into two symmetrical parts, with half the rows devoted to read-only scanner indications of the status of the various power switches in the office, and the second-half of the matrix devoted to flip-flops controlling power switch lamps on the various frames. The scan and SD matrix is also symmetrical in that corresponding bits in the scan half and SD half are assigned to the same processor unit.

Other minor functions are served by the PPI. A telephone jack circuit for the in-office telephone and an AC driven time-of-day clock are provided, as is a separate 4A timer which provides pulses at several second intervals for timing out certain displays.

5.5 Input/output subsystem features

Craft personnel usually interact with the system via the keyboard/CRT/printer interface. The 1A Processor hardware is capable of supporting up to 96 low- and medium-speed, half-duplex, asynchronous channels with up to three ASCII terminal devices per channel. Provisions

are made for either direct terminal interconnection or remote interaction via modem.

A 110 or 1200 b/s EIA RS 232 standard asynchronous terminal or the 20-mA current loop, low-speed teletype interface is currently supported. CRT, printer, keyboard, cassette tape, and paper tape terminals are used for routine maintenance of the system and network, traffic management, system status, recent change information, and retrieval of related data of other using systems.

5.6 The input/output subsystem description

The 1A Processor IO implements the terminal interface to the processor. Each frame consists of a peripheral bus interface and up to two IO unit selectors (IOUSs), which may each be equipped with up to eight IO unit controllers (IOUCs). Each IOUC, in turn, fans out to the three ports with either an EIA standard or the teletype current loop interface, as shown in Fig. 12.

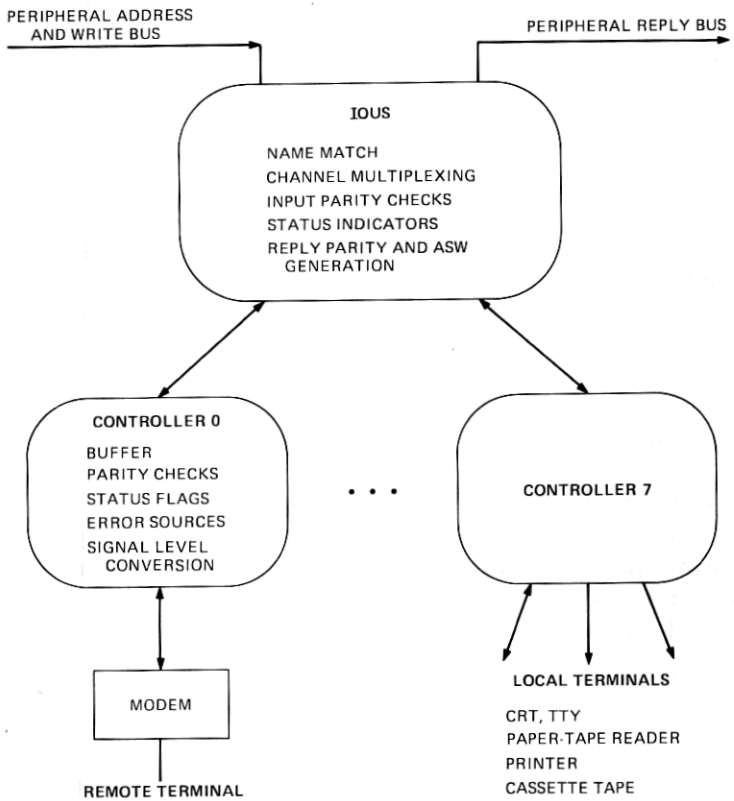


Fig. 12—Input/output frame access.

The IOUS acts as a multiplexer to the peripheral bus. Its primary functions are to provide initial decoding of address, to route data between the processor and each controller, to pass both maintenance-related and normal activity flags back to the processor on request, and to perform common maintenance tasks such as bus parity checking and generation, and validity checks on data and instructions.

The IOUC acts as the asynchronous buffer between the IOUS and a channel. Its primary functions are to buffer incoming and outgoing data, provide parity checks on the ASCII characters, generate service flags to indicate channel state changes and buffer status, and to report maintenance and service flags to the selector upon a poll request.

Three EIA interface ports are provided on each channel. Data coming in one port are echoed on the others during input to the system, and the same data go out on all three ports during system output.

Each processor includes two IOUSs within its fixed floor-space arrangement. This provides up to 16 channels with the basic processor and satisfies the minimum channel requirements for processor operation. Additional channel requirements are satisfied by adding IO frames to the peripheral bus in the network area of the office.

5.7 Input/output polling

The 1A Processor IO system is based on a 60-ms polling scheme. The polling function starts when the CC delivers a poll pulse to all equipped IO circuits. Each IOUS then distributes the pulse to its controllers and collects a 2-bit response from each controller in a poll-request register (PRR) indicating that the controller needs either service or maintenance. Service requests indicate the beginning of an input message, the end of an output message, a break, or a buffer that needs to be filled or emptied. After the polling pulse gates the service or maintenance requests into the PRR, it clears the sources of the requests in responding channels.

The logical OR function of all the requests provides a summary response, gated to the CC only when no requests exist. If any request is found, no polling response is delivered and the PRR must be read to identify the active channels. Since more than one IO selector can be equipped in an office each responds to the polling pulse simultaneously, but in a peripheral unit reply bus bit position which matches its unique enable code.

5.8 General data transfer

All data transfer is initiated by the CC and is carried out through the execution of an instruction sent by the CC. Each instruction to the IOUS begins with a start-of-sync pulse from the CC, sent along with a peripheral bus order, which clears the input registers and the main sequencer. The

input registers are then loaded in parallel with an address (or enable code), an instruction, and data. As the gating ends, the enable code is examined to see if it matches the wired address of the IO circuit. If no match occurs, the sequencer stops and waits for the next sync pulse, but if a match does occur, the sequencer continues with instruction execution.

On write instructions, the 24-bit data portion of the data input register is transferred to a destination register specified by the write instruction either in the IOUS or an IOUC. Transmit instructions transfer three 8-bit characters from the input data register to the IOUC character buffer specified by the channel decoder. Read instructions transfer data from the source register in the IOUS or an IOUC specified by the instruction to the reply register. Receive instructions transfer three characters to the reply register from the IOUC buffer specified by the channel decorder. For testing purposes, the IOUS input data register can be connected to the reply register to loop data from the write bus to the reply bus.

As with the PPI, once the operation is complete, an all seems well, parity, and reply data are gated back to the central control if all internal checks have passed.

VI. HARDWARE MAINTENANCE FEATURES

6.1 System features

The need for high reliability and maintainability in the 1A Processor control system imposes important design objectives. Many requirements that concern component selection, subsystem design, system architecture, and program structure were based on the need for very high processor dependability. Complete duplication of processing capability and memory access had to be provided because of its critical impact on system performance. All critical processor communication buses are duplicated in some manner.[4] Input/output that is critical to office integrity is duplicated or backed up with alternate IO channels. Those parts of MCC and PPI that are critical to processing capability or processor maintenance are provided with functional redundancy. The memory backup scheme is also duplicated. The overall system requirement is that all processor capabilities have subsystem backup except when the function is not vital to processor integrity.

In all of the control system units, hardware maintenance features have been provided to allow rapid detection of faults. This permits quick recovery through subsystem reconfiguration and facilitates rapid repair by good fault isolation.

Hardware and software maintenance features are designed to complement each other, with software utilizing the hardware features to achieve maintenance objectives.

6.2 Error detection and recovery

Prime error-detection techniques in the hardware are provided through redundant information (by check bits), matching internal operations, round-trip checks on data and addresses, internal operational checks, and address-range validation. The failure of any of these checks results in the transfer of processor control, via program interrupt, to one of a prioritized set of recovery programs. The level of interrupt and recovery is determined by the potential severity of the fault on system performance and the need for specialized programs to handle fault recovery for each subsystem.

6.3 Power control and alarms

All power that is critical to processor availability is duplicated. Each of a pair of duplicated units is powered from a separate distribution system. Other critical unduplicated units receive power from both power systems, either one of which is sufficient to operate the unit. Each unit has power-sequencing circuitry to allow power removal and restoral without affecting processor operations. Since all units have bus interconnections, the circuitry is required to power logic circuits before bus circuits, control critical routing flip-flops, initialize hardware sequencers, and perform any other initializations required to enable the unit for further program actions. An alarm circuit is provided in each unit to monitor and initiate visual and audible alarms and request IO teletypewriter records for all power failures or manually initiated power operations.

6.4 Central control

6.4.1 Matching

Duplicated and matched CCs form the basic program execution unit of the 1A Processor. All data movement and addressing is carried out between subsystems on duplicated communication buses by way of CC interfaces. The CCs have direct communication paths to all other processor units to effect recovery, reconfiguration, and other maintenance operations.

Normally, the CCs operate in step. Both perform identical operations with one CC active and the other standby. In this configuration, each CC matches itself with its mate to insure that both execute the same instructions and operate on identical data.

Each CC has two separate match circuits, and each circuit has the ability to match 24 internal bits to 24 bits from its mate once each 700-ns machine cycle. Each matcher has access to any one of 16 different 24-bit internal match groups. During normal in-step operation (routine

Table VI — Central control routine matching

Instruction	Quantity Matched: Active CC, Matcher 0	Active CC, Matcher 1	Standby CC, Matcher 0	Standby CC, Matcher 1
Combined load from memory and data adjust	Store address at 8T0	Data reply after processing at 0T4	Internal bus at 4T8— No data*	Internal bus at 8T0— Adjusted data
Combined store into memory and data adjust	Store address at 8T0	Data being Stored at 0T4	Internal bus at 4T8— No data*	Internal bus at 8T0— Adjusted data

* Important data appear on the internal bus at this time for some instructions of these types.

matching mode), the timing of the match and the groups to be matched in each of the four match circuits is determined by the type of instruction being executed and by the active/standby status of the CC. In this manner, the four distinct internal groups which are necessary to assure correct execution of any instruction can be matched in the same machine cycle. For example, for the instruction shown in Fig. 5, the matches would be as shown in Table VI. Routine matching can be specified in one or both CCs when they are running in step. Three other match modes are available for maintenance program usage. In directed matching, the user can specify the match group, match time, and whether the other CC or a constant should be matched. The sampled match mode is used to take a snapshot of one of the match groups; it is controlled by selection of the group match time and by specifying the machine cycle during which the snapshot should be initiated. Utility matching can be used to monitor and match memory operations and to generate a program interrupt at the detection of the match condition specified.

6.4.2 Clock synchronization and checking

The oscillator in the active CC is used as the reference source for all processor timing information. A ring counter generates all the necessary CC clock pulses. There is a maintenance clock that runs continuously and an operational clock that can be stopped in the standby CC and started under program control. Each clock phase can be inhibited, one at a time, by program. The clock output is checked for proper pulse overlap and sequencing. Oscillator level monitors and inter-CC clock synchronization monitors are also provided. A separate analog clock is provided to allow reconfiguration in the event of operational clock failure in the active CC.

6.4.3 Processor configuration

The active CC can generate pulses that control its mate CC and the configuration of all other processor units. This allows the CC to control one of the duplicated pair of memories (or backup units) that will be part of the active processor complex. CC pulses are used for changing system bus configurations during interrupt recovery and for controlling units during fault diagnosis. Internal checks of pulse circuitry are provided to verify correct circuit operation.

System troubles normally are detected by trouble-detection circuits; system fault recovery follows after being initiated by the program interrupt. Since this approach requires a sane processor, an autonomous hardware processor configuration circuit is provided in each CC to handle faults when the processor loses sanity. Normally, only the active CC activates the circuit. Reconfiguration is triggered by loss of operational

fault recovery, or configuration program sanity; clock failures; and CC configuration problems. Manual and programmed initiation of the reconfiguration circuit are also provided. The circuit consists of various timers to implement sanity checks on base (operational), interrupt (fault recovery), and configuration level programs. The outputs of the circuit control CC-to-program memory configuration, reloading program memory from disk backup when required, and isolating various subsystems from the CC until program sanity is restored.

6.4.4 System reinitialization

To allow the system to be started from the powered-up state, a facility called system reinitialization (SR) is provided in the CC, which is manually activated from the MCC. When selected, this feature provides a minimal configuration of CC, PS, and a tape unit; it also overrides most error checks. The basic block of program is loaded into PS from tape, and control is then passed to configuration level programs to assemble the remainder of the units on line, load programs and data, and restore a functioning processor.

6.4.5 Diagnostic access and control

Special hardware is provided in all 1A Processor subsystems to allow fault isolation and repair without interfering with normal system operation. A number of circuits are provided to allow the active CC to diagnose the standby without affecting its normal operation capability. Each CC, when active, can initiate many actions in its mate; for example, it can stop and start the operational clock, cause the CC to start and run n cycles, directly set or reset many internal control points, and configure the communication buses. Most registers in the standby CC can be control written and read while the operational clock is stopped. Each sequencer flip-flop can be individually controlled. These features are particularly powerful for isolating many problems, since they do not depend on the ability of the standby CC to execute instructions.

Some CC circuitry is only used when the CC is active. To facilitate fault diagnosis, these circuits can be activated in the standby mode under maintenance program control.

All bus communication circuits can be thoroughly tested since all bus bits can be individually exercised under diagnostic control. Standby CC addressing and data sending buses can be looped back to the active CC through external units.

Each clock phase of the standby can be inhibited to allow testing clock gates and clock error detectors. The matcher circuits, previously described, are used extensively for testing and fault isolation.

6.5 Master control console—processor peripheral interface

6.5.1 Master control console

The primary function of the master control console (MCC) is to provide manual control of various system features and to provide visual displays for administration and maintenance functions. Since it is not critical to system operational capability, it is not duplicated or backed up, but has been designed so that no MCC fault can cause a system outage. Diagnostic program tests are for fault detection and repair of the MCC since interrupt level programs are not required. All display and manual control features, which are interfaced to the system by the processor peripheral interface, are exercised in an interactive manner by the program under operator control.

6.5.2 Processor peripheral interface

Orders from the processor to the PPI are communicated over the peripheral bus, with parity over the address and data used as the error check. There are several PPI features to ensure availability and to isolate problems. Each peripheral bus has a dedicated PPI matrix controller. The matrix row addresses are encoded so that each address is at least Hamming-distance of two from any other. Failure to properly select a row once a PPI name code match has occurred results in an all-seems-well (ASW) failure to the CC. Other ASW components are input data parity and PPI configuration checks that insure against false responses. All of the communication and operational features of the PPI are extensively tested by the diagnostic program.

The interface portion of the PPI, which connects the peripheral bus into the application system, is provided with circuitry to loop around all processor transmit-bus bits into the receive-bus bits. This looping, which is program controlled, provides a powerful maintenance capability for isolating peripheral bus troubles.

6.6 Input/output

The input/output frame is used to provide a teletypewriter data path into and out of the processor for various administrative and maintenance tasks. Since the IO is connected to the CC by the peripheral bus, error detection makes use of such bits as the interleaved transmit parity and answer parity. The input data to each IO selector can be looped back to the reply bus. Internal maintenance circuits provide additional fault detection and isolation within each IO selector. A control pulse is provided for each IOUS to gate back to the CC all the pertinent configuration flop-flop states, routing bits, and error indicators. Internal to the IOUS, checks are made of various operations which, if all checks pass, result

in the generation of an ASW response to the CC. The items checked are: data parity, instruction validity, name-code validity, IOUS-to-IOUC bus parity, and IOUC response. Any test failure is recorded in an error-summary register for use by the appropriate fault recovery program. Checks are also made of the synchronization signals received from the CC.

Every IO is polled by CC programs every 60 ms to determine if any work should be processed. Certain IOUS failures can be detected through improper response to the polling pulse. Circuitry is also provided to test each IO controller served by an IO selector. Failure to load or unload the 24-character buffer in the IOUC results in an overflow indication administered by the input/output control program. Carrier failure (if provided for a terminal) is also detected and monitored. Under program control, a signal can be sent to the IO terminal to request an automatic response, which checks out the complete IO loop. The detection of any communication error between a CC and an IOUS will result in an immediate interrupt and entry to a fault-recovery program. In the event of an IO failure, all data is rerouted by program to specifically assigned backup channels on an operational IOUS.

6.7 Communication bus maintenance

All vital communication buses are completely duplicated, and each contains redundant information for checking validity and for data-error detection. Most processor units can generate and check error-detect information although some bits are checked in a round-trip fashion by the original sending unit. When a unit other than the CC detects an error in a bus transmission, it indicates to the CC an all-seems-well failure, which in turn results in a maintenance interrupt.

The error-check schemes on the various buses are designed to handle the expected failure modes of the various units. For instance, call and program store addressing uses a discrete code for order type, a coded unit name enable, and two parity bits over the code and store address. Each of these bits is generated over a partially overlapped, interleaved set of data bits and over the code and address. This overlapping pattern is designed to cope with particular memory circuit failures that have the potential to generate errors in more than one bit of a word. The two parity bits are stored with the data in memory. Data transmission to and from file stores is handled in a similar way.

For the peripheral bus, coded enable operations utilize two parity bits in an interleaved, nonoverlapped manner. CPD enabling has extensive checking identical to that of the No. 1 ESS. The peripheral bus is tested using the PPI loop-around feature without dependence on the application system. The other buses are tested by using the processor units as test

generators and receptors. By careful attention to test sequencing and by judicious use of the mate duplicate bus, faults can be detected and isolated.

VII. CONCLUSION

The control system has been designed with the goal of achieving maximum performance consistent with maintenance and cost objectives. A number of challenging design problems were resolved, such as maintaining software compatibility with No. 1 ESS while adding new features, achieving internal CC timing margins with a relatively short cycle and a complex decoding job, providing for compatibility with both 700- and 1400-ns memory cycle times, and meeting environmental requirements. Results of early manufacturing and field experience indicate that the goal is being realized.

VIII. ACKNOWLEDGMENTS

The work described in this article was accomplished through the combined efforts of many designers throughout Bell Laboratories and Western Electric. Without their contributions, too numerous to acknowledge individually, the goals for the control system could not have been met. The authors wish to thank J. T. Mahoney for his help in preparing this article, particularly the section on communication buses.

REFERENCES

1. "No. 1 Electronic Switching System," B.S.T.J., *43*, No. 5 (September 1964), Parts 1 and 2.
2. J. S. Nowak, "No. 1A ESS—A New High Capacity Switching System," International Switching Symposium, Kyoto, Japan, October 1976.
3. A. E. Spencer, Jr., and H. E. Vaughan, "No. 4 ESS—A Full-Fledged Toll Switching Node," International Switching Symposiu, Kyoto, Japan, October 1976.
4. P. W. Bowman, M. R. Dubman, F. M. Goetz, R. F. Kranzmann, E. H. Stredde, and R. J. Watters, "Maintenance Software," B.S.T.J., this issue, pp. 255–287.