Chapter 38

The RW-400—a new polymorphic data system[1]

R. E. Porter

Summary The RW-400 Data System, based upon modularly constructed, independently operating and flexibly connected components, is the logically evolved successor to conventional computer designs. It provides the means by which information processing requirements can be met with equipment capable of producing timely results at a cost commensurate with problem economic value. System obsolescence is minimized by the expandability in numbers and types of processing modules. Real time reliability is assured by component duplication at minimum cost and by the advanced design techniques employed in the system's manufacture. Man-machine communication facilities are program controlled for maximum flexibility. Parallel processing and parallel information handling modules increase the system's speed and adaptability when handling complex computing workloads. This polymorphic design truly represents an extension of man's intellect through electronics.

The RW-400 Data System is a new design concept. It was developed to meet the increasing demand for information processing equipment with adaptability, real-time reliability and power to cope with continuously-changing information handling requirements. It is a polymorphic system including a variety of functionally-independent modules. These are interconnectable through a program-controlled electronic switching center. Many pairs of modules may be independently connected, disconnected, and reconnected, in microseconds if need be, to meet continuously-varying processing requirements. The system can assume whatever configuration is needed to handle problems of the moment. Hence it is best characterized by the term "polymorphic"—having many shapes.

Rapid, program-controlled switching of many pairs of functionally-independent modules permits nondisruptive system expandability, operating reliability, simultaneous multi-problem processing capability, and man-machine intercommunication feasibility. These are only partially found in computers of conventional design.

Computer users have been forced heretofore to match problems to computer limitations. Problem changes posed serious reorientation and reprogramming difficulties. Changes from one computer to another model, due to growth in applications, often resulted in large expenditures of time and money. During maintenance or malfunction of a conventional computer its entire processing capacity is shut down. Real time processing reliability cannot be maintained on an around-the-clock basis. The conventional machine must process its problems serially. This serious limitation is only partially alleviated by time-sharing or computing-element-doubling designs. The high cost-per-hour of conventional computer operation rules out direct man-machine intercommunication during other than emergency situations.

The radically-new polymorphic design concept of the RW-400 Data System was evolved by Ramo-Wooldridge engineers to provide a practical solution to those information processing problems now inadequately handled by conventional computer designs. The RW-400 is a powerful new tool in the field of intellectronics—the extension of man's intellect by electronics.

System description

The RW-400 Data System contains an optional number and variety of functionally-independent modules. These communicate via a central electronic switching exchange. Each module is designed, within practical economic and functional limits, to maximize system adaptability over a wide range of problem types and sizes. This new design embodies the latest proven electronic design techniques, assuring high processing speeds and high equipment reliability. The RW-400's modularity assures reliable, round-the-clock processing of information with controllable computing capacity degradation during module maintenance or malfunction. Practical man-machine intercommunication is achieved in the RW-400 system by use of program-controlled information display and interrogation consoles.

Figure 1 shows the over-all system design. Modules of various types communicate through a central exchange switching center. Computing and buffering modules provide control for the system. These modules are self-controlled and make possible completely independent processing of two or more problems. One of the computer modules may be designated the master computer and

[1] *Datamation*, vol. 6, no. 1, pp. 8–14, January/February, 1960.

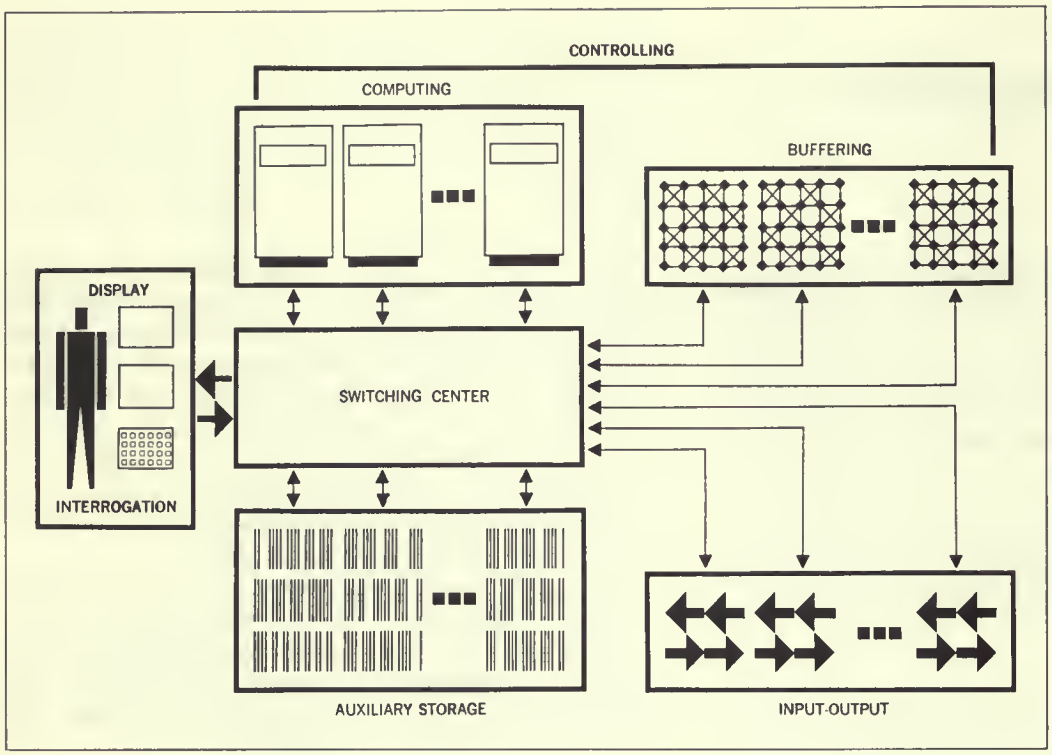


Fig. 1. The RW-400 data system.

in this role initiates and monitors actions of the entire system. An alert-interrupt network is provided to allow coordinated system action. Therefore, the system as applied to given information processing problems may change on a short range (microsecond) basis, thus providing, through programming, a self-organizing aspect to the system. In addition, the system may change through the years as the applications change. The most efficient and economical complement of equipment is applied to the problem at all times.

An RW-400 system is built around an expandable Central Exchange (CX) to which a number of primary modules may be attached. These are: Computer Modules (CM); self-instructed Buffer Modules (BM); Magnetic Tape Modules (TM); Magnetic Drum Modules (DM); Peripheral Buffer Modules (PB); and console communication Display Buffer Modules (DB). How many modules are put together in a system is entirely a function of system application. In addition to primary system modules, punched card, punched tape, high speed printing and control console devices are available. These handle nominal system input/output requirements. Additional man-machine communication devices such as interrogation, display and control consoles, may be included in the system as problem requirements dictate. A Tape Adapter (TA) module is available to provide compatibility with magnetic tape of other computers. Information generated at Flexowriter inquiry and recording stations may be directly received by the system via the Peripheral Buffer Module. This latter module also buffers the receipt of TWX and punched tape information.

The way in which a particular RW-400 Data System functions depends on the number and type of each module included. It may initially be composed of the minimum number and variety of modules needed to do a small problem or the initial part of some large but yet-to-be-defined problem. Such a system would work much like a conventional computer. It would probably include a buffer module and thus have a parallel data handling capability not found in the conventional design at a comparable price. The initial system installation may then be augmented by the timely addition of modules.

A buffer module (BM) has the capability to control its acquisition and dissemination of information independently. The buffer provides a computer module with parallel data handling capability without complicating the problem processing program with the conventional intermixture of arithmetic and housekeeping instructions. Information previously generated by the processing program may be appropriately disposed of within the system while processing continues. Data needed at a subsequent time in the processing may be retrieved from system storage in advance of need while processing progresses. The simultaneity of these operations not only materially increases over-all processing speed but also increases the practical utility of the less costly types of internal system storage such as a magnetic tape.

The computer (CM) or buffer (BM) modules, when acting in a controlling capacity, may initiate connection to an information storage or handling module during that part of the processing program when the two can work profitably in unison. The pair of modules thus interconnected neither affect nor are affected by other modules. Logical interlocks prevent unwanted cross talk among modules. An intermodule communication system lets controlling modules signal status or alert other such modules of their need to communicate. The decision by a module receiving an alert signal to permit interruption or to proceed is optional with that module. The optional interrupt feature is that needed to make the often-discussed but seldom-used program interrupt capability both useful and practical. Programs may thus permit interruptions only at convenient points in the processing sequence.

Modules may be assigned, under program control, to work together on a problem in proportion to its needs. As soon as a module's function is complete for a given problem, that module may be released for reassignment to some other task. The system is thus self-controlled to match processing capacity to each problem for the time necessary to do the job. Full system capacity may be brought to bear upon a very large problem when needed. This capacity may be apportioned among a number of smaller problems for simultaneous processing, program compilation, program checkout, module maintenance etc., when it is not needed for maximum system effort.

From the preceding system description, it is apparent that such equipment can be expanded from a modest initial installation into a very powerful and comprehensive information processing center as requirements warrant. More specific descriptions of principal system modules follow to give the reader a better feel for how this system might perform his information processing work.

The functional modules

The key to appreciative understanding of the power of the RW-400 lies in knowledge of intermodule connection. It is appropriate to describe the Central Exchange (CX) unit first, then follow with descriptions of the various modules.

The central exchange

The Central Exchange performs the vital function of interconnecting a pair of modules whenever requested to do so by either a computer or a buffer module. Since internal programmed control is only possible within a computer or a buffer module, one of the interconnected pair of modules must be either a computer or a buffer. The time in which any connection may be made or broken is about 65 microseconds. An exchange has basic capacity to connect any of 16 computer or buffer modules to any of 64 auxiliary function modules. There is nothing sacred about the number 16 since it is possible to extend the CX module's interconnection matrix through design modification when need arises. The CX is an expandable, program-controlled, electronic switching center capable of connecting or disconnecting any available pair of modules in roughly the time of one computer instruction execution. Figure 2 illustrates the permissible module interconnections within the Central Exchange.

Every intersection on the illustration represents a possible connection between modules. The "x-ed" intersections indicate typical connections in force at any point in time. The control logic of the CX module's connection table prevents more than one interconnection on any horizontal (controlling) or vertical (controlled) data path representation on the diagram. When connection is requested of the Central Exchange while one of the required modules is already carrying out a previous assignment, the requesting module can be programmed to sense this condition and wait until connection can be made without interference. Should waiting be undesirable, the requesting module can go on about its business and check back later to see when the desired connection can be made. There is an implication here, of course, that knowing the kind of a system he is dealing with, a programmer requests connections in advance of need whenever possible.

Provision for master-slave control is included via an Assignment Matrix established within the CX module by a computer module previously assigned to master status. Such a provision is necessary to preclude inadvertent connection requests from unchecked programs or malfunctioning control modules from affecting sets of modules simultaneously processing another problem. Connection requests are therefore essentially filtered through both an assignment and an interconnection validity matrix prior to being acted

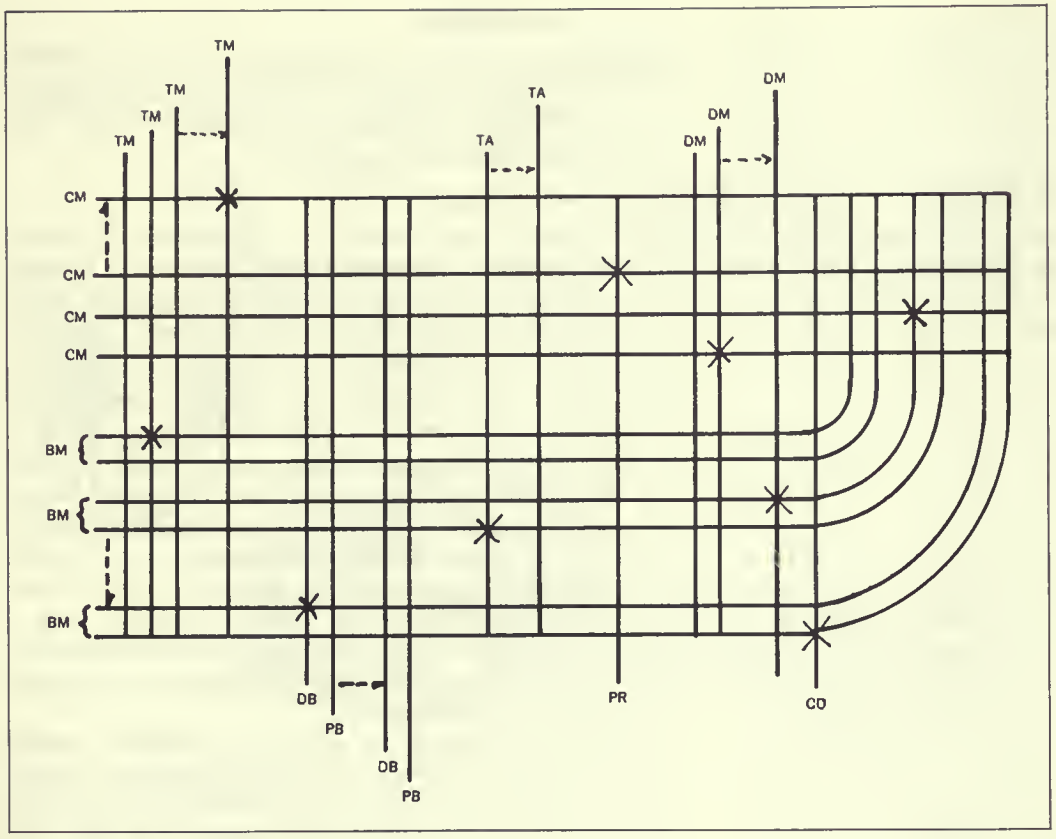


Fig. 2. The Central Exchange connection matrix.

upon by the Central Exchange. The computer module manually assigned to master status is the only one permitted to cause the interconnection of a pair of modules which does not include itself.

The computer module (See Fig. 3)

The Computer Module (CM) is a self-sufficient, general purpose, two-address, parallel word, fixed point, random access computer. Its internal magnetic core memory has a capacity of 1024 words. A computer word consists of 26 information bits and 2 parity bits. Each parity bit is associated with the 13-bit half word transferred in parallel via the Central Exchange to other system modules. The instruction repertoire of the CM consists of 38 primary instructions whose various modes effectively result in over 300 different operations. Of the 39 available CM-400 instructions, 24 may be classified as "arithmetic" and 10 as "program control" or "sequence determining" instructions. Five additional instructions may be classified as "external" or "input/output" instructions. All but three of the 24 arithmetic instructions fit into a symmetric scheme of classification wherein there are seven basic operations, each having three distinct modes. The seven basic operations are—add, subtract, absolute subtract, multiply, divide, square root and insert. The three modes are—Replace, Hold and Store. If we let the capital letter "G" identify the first operand, "H" identify the second operand, an "°" signify an arbitrary operation, the symbol "→" indicate replace, and "A" the word in the accumulator, then the three modes may be characterized as:

Replace: $H \circ G \rightarrow H, A$
Hold: $H \circ G \rightarrow A$
Store: $A \circ G \rightarrow H, A$

The three remaining arithmetic operations are Add Accumulate wherein the contents of H and G are added to the Accumulator;

Multiply Accumulate wherein the contents of H are multiplied by G and added to A; and Transmit where the contents of G are stored in H.

The ten program control instructions are Store, Store Double Length Accumulator, Load Accumulator, Insert Mask in the S Register, Stop, Link Jump, Compare Jump, Tally Jump, Test Jump and a Multi-purpose Shift.

The five external instructions are those which cause data to be transmitted to or received from a device external to the computer. Each command is multi-purpose in nature and hence equivalent to several conventional external instructions. The commands are—Command Output, Data Input, Conditional Data Input, Data Output and Character Transfer. A comprehensive discussion of the variation of each of these commands is not pertinent to this article. Suffice it to say that commands are available for carrying out a wide variety of intermodule data communication.

The interrupt capability of a Computer Module is a logical generalization of the "trapping" feature found on several conventional computers. It permits the automatic interruption of a program, at the option of the program, when the computer module receives an "alert" that a condition requiring attention has arisen. It can be used to warn the program when an error of some type has occurred, minimize unproductive computer waiting time while another module completes its task, eliminate many programmed status test instructions and provide a convenient means of subjecting one computer module to the control of another. Program control of interruptions within a CM-400 is accomplished through the sense register S. This register may be filled with an interrupt

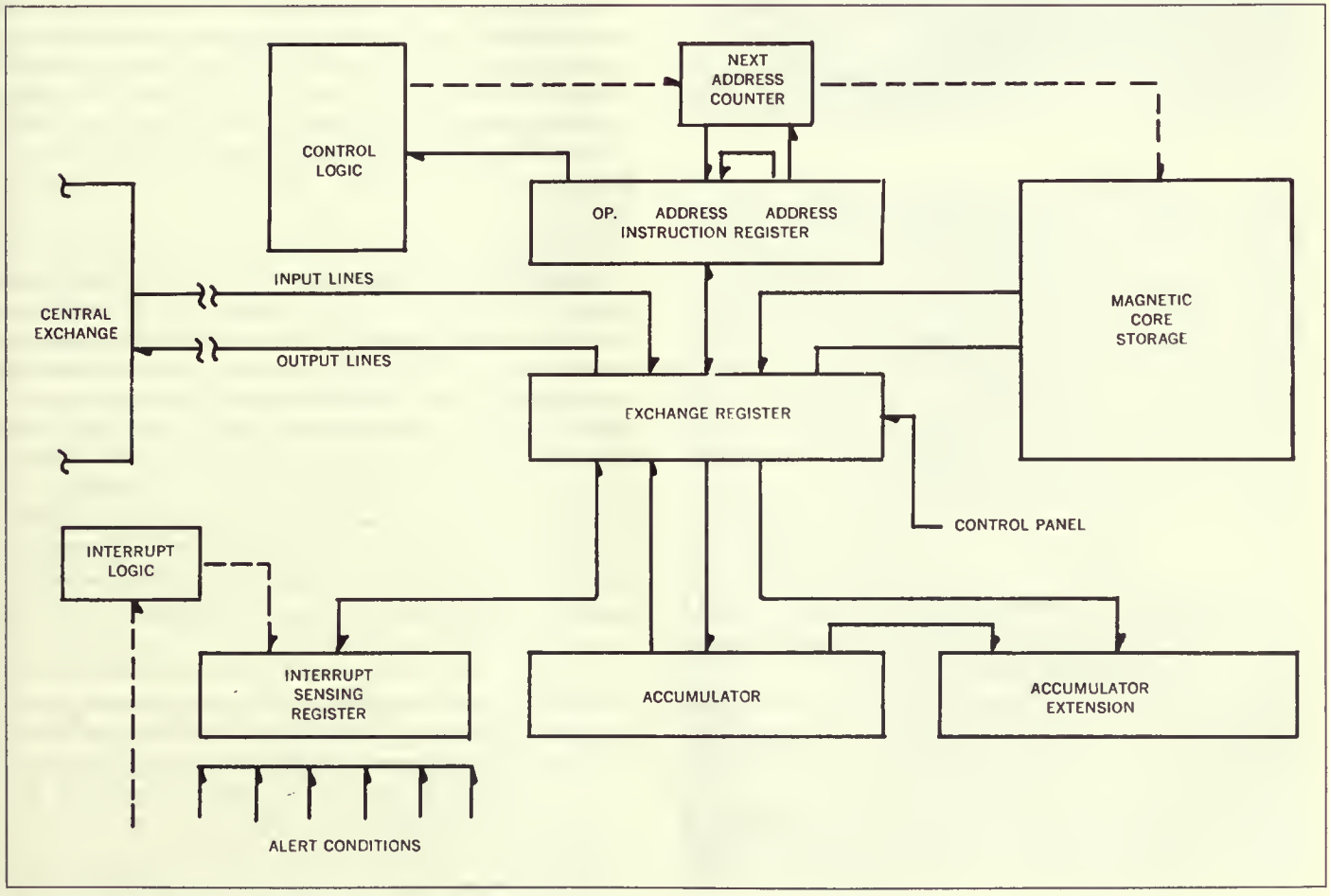


Fig. 3. The CM-400 Computer Module.

RW-400 analysis console.

mask by means of the Insert S instruction. A bit by bit correspondence exists between the S register and the interrupt register and the interrupt register I to which the alert lines are connected. A Test Jump instruction can be used to examine the coincidence between these registers of an alert signal in a bit position corresponding to a one in the S register mask. If an alert is received by the computer during the execution of an instruction, control will be transferred to memory location "O" at the end of the instruction if, and only if, (a) the sense bit corresponding to the alert is a "one," (b) the master sense bit is a "one," and (c) the instruction was not an "Insert S." The master sense bit in the S register may be programmed to permit the interrupt to take place according to the interrupt mask or to inhibit interrupt until the program can conveniently cope with it. All instructions being executed at the time an interrupt condition occurs are completed before the interruption is allowed to take place.

Figure 3 schematically illustrates the Computer Module's primary registers and the interconnecting information paths.

Typical two-address addition and subtraction times are approximately 35 microseconds including memory access time. Multiplication takes about 80 microseconds, and division and square root about 130 and 170 microseconds respectively.

Before attempting to draw a comparison between a CM and a deluxe conventional computer the reader should bear in mind the trade offs in features versus cost; parallel processing versus sequential processing; independent information handling versus program complicating "housekeeping"; and real time system reliability versus periodic inoperability. The only valid comparison is that between the RW-400 Data System and a conventional computer applied to the same task. The contribution to the RW-400 system made by the Buffer Modules can be better assessed by the reader after the following description has been considered.

The buffer module

A Buffer Module consists of two independent logical buffer units, each having 1024 words of random access magnetic core storage and a number of internal registers used in performing its functions when in the self-controlling mode. A Buffer Module may be connected to a Computer Module so that the Buffer's core storage is accessible to the computer as an extension of the computer's own storage. A Buffer may also serve as an intermediary device between a computer and another module, such as a tape or drum, to minimize time conventionally lost in data transfers. The Buffer is capable of recognizing and executing certain instructions stored in its own memory. It can therefore be left to perform data handling functions on its own while computer modules are otherwise occupied.

A Buffer Module may be connected to a Computer Module and the buffer 1024 word storage used as an indirectly addressed extension of the computer's own working storage. When the address 1023 (all ones) appears in the operand field of a computer instruction to be executed, the computer is signalled that the operand refers to some cell in buffer storage. The computer then uses the number in the buffer read register R (or in the case of a few instructions, the buffer write register W) as the effective address designated by the operand field of the instruction. Extended addressing may be used in either the first or second operand field of the instruction or in both operand fields. If extended addressing is used in only one operand field, the effective address designated by that field is the number in register R. A "1" is automatically added to the contents of the R register after the instruction is executed. If extended addressing is used in both operand fields of an instruction, the effective address of the first operand is the number in register R and the effective address of the second operand is one more than the number in register R. A "2" is automatically added to the contents of register R after the execution of this type of instruction. The R (or W) register may be preset to any desired initial condition by means of the computer's Command Output instruction. All the commands being executed by the computer must be stored within the computer

module's storage and may not be in buffer cells addressed by the computer at execution time. The extended addressing and buffer register indexing may be used to materially simplify repetitive data acquisition operations.

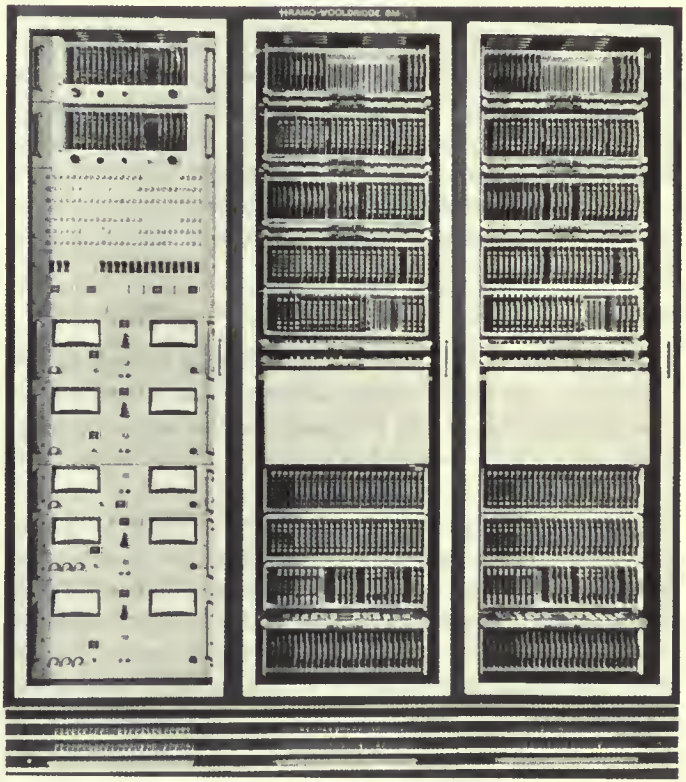
The primary function of a Buffer Module is not, however, that of an auxiliary computer storage unit. The drum and tape modules more aptly serve this function in the RW-400 system. A Buffer Module is capable of operating autonomously and of controlling other modules such as Tape Modules, Drum Modules, Peripheral Buffers, Display Buffers, Printers or Plotters. This capability enables the Buffer Modules in a system to perform routine tape searching and data transferral tasks thereby freeing the Computer Modules to do more computing. In its "self-instruction" mode, the buffer executes its own internally stored program in much the same fashion as a computer. The memory of a Buffer Module will therefore be occupied by its own control programs as well as blocks of data which it is holding for transmission to other units. The buffer is used to acquire information from the relatively slower auxiliary storage and communication modules while the computer proceeds at high speed. Blocks of information retrieved in advance of computer need by the buffer may then be rapidly transferred to the computer's own storage or operated upon as they stand in the buffer via the indirect addressing capability of the computer. Another feature of the buffer is its switching capability. Each Buffer Module is composed of two buffer units tied together. A unit function switching feature permits the employment of the two units together in an alternating mode of operation. Continuous information transfer from tape to computer, for example, may be accomplished without stopping the tape unit. A switching instruction executed simultaneously by both units of a Buffer Module causes whatever devices were connected to the first unit to be connected to the second and vice versa.

Now that the functional controlling modules and the module interconnection concept have been discussed, the more conventional auxiliary storage modules available with the system may be described to round out the processing capability of the system.

The tape modules

A Tape Module consists of an altered Ampex FR-300 tape transport plus the necessary power supplies and control circuitry to effect information reading, writing and control. One inch mylar tape is used. Information is written on 16 channels—two of which are clock channels. The remaining 14 channels consist of 13 information bits plus parity. The information reading or recording rate is 15,000 computer words per second. Data may be recorded on tape in variable blocks up to a maximum of 1024 words per block (the size of the storage available to hold the data in a sending or receiving module). Each block is preceded by a block identification which permits selective tape information searching by a Buffer Module. Single blocks imbedded in a tape file of other blocks can be overwritten. A two-stack head permits automatic verification of each block as it is written. Readback parity errors are automatically detected during the writing process. Thus drop-out areas may be determined while the data is still available in a computer or buffer for recording elsewhere.

A description of the RW-400's tape handling capability would not be complete without mentioning the Tape Adapter (TA) module. This is a self-contained unit capable of performing the reading and writing of magnetic tapes in a format acceptable to the IBM 704 and 709 systems. The TA consists of an Ampex FR-300 half-inch digital tape transport, including dual gap head and servo control system; reading, writing and control circuits; and a module housing with its own blower and power supply.



RW-400 Buffer Module.

The drum module

The Drum Module (DM) contains a magnetic drum with storage capacity of 8192 words. It may be connected to either a Computer or a Buffer Module through the Central Exchange. Average access time to the first word position on the drum is $8\frac{1}{2}$ milliseconds. Successive words are transmitted at the rate of 60,000 computer words per second. The Drum Module is conventionally used as an intermediate item storage device to minimize tape handling time.

Special system communication modules

The external data and man-machine communication of the RW-400 Data System are handled via drum buffer modules. A wide variety of asynchronously operated equipment is speed matched and program controlled through the features designed into these special system communication modules.

The Peripheral Buffer (PB) provides input/output buffers for communication between Computer or Buffer Modules and relatively slow speed external devices such as Flexowriters, Plotters, Punched Tape Handlers, Teletype Lines and Keyboard Operated Equipment. The Peripheral Buffer stores its information in four pairs of bands which operate alternately as circulating registers. Each band contains eight input and eight output buffers for a total of 32 input buffers and 32 output buffers in each Peripheral Buffer Module. Each buffer is a drum band sector 64 computer words long. Conventionally one input and one output buffer sector are connected to each external device (such as a Flexowriter) to permit two-way communication between the external device and the RW-400 system.

The display buffer

A Display Buffer (DB) acts as a recirculating storage for the cathode ray tube display units in a Display Console. Information to be displayed is sent to the DB band associated with a particular display tube via the Central Exchange. The Display Buffer sends only status information back to other system modules upon request. The information displayed on any tube is controlled by the bit pattern sent to the Display Buffer. The display pattern is regenerated 30 times per second to minimize image fading and flicker. The preceding explanation of the Display Buffer has little meaning to a reader unfamiliar with the features of the Display Console itself. This console is therefore described in more detail in the following paragraphs.

Display consoles

Display Consoles can give a problem "analyst" or "monitor" a visual picture of the status or results of any information being handled by the RW-400 system. In addition to the actual Cathode Ray Tube, numerical indicator, signal lamp and typewriter information outputs, several types of keyboard activated system control and parameter entry facilities are provided on the console. The total man-machine communication facility represented by each console is designed to be primarily a function of the computer control programs initiated by the analyst via his console.

A set of Display Control Keys generate messages which are recorded on a Peripheral Buffer sector for later interpretation and display generation by a computer program. A set of Process Step Keys are provided the analyst so that he can initiate preprogrammed system processing variations. Associated with the Process Step Keys is an overlay or "program card" which permits the assignment of a variety of meanings to the set of Process Step Keys. Insertion of the overlay by the analyst gives him a unique label for each Process Step Key and automatically cues the controlling computer to assign the corresponding set of programs to each key message. A Data Entry Keyboard is provided on the console so that the analyst can enter control parameters when asked to do so via the display devices.

A Joystick Lever affords the console operator a means of controlling the position of cross hair markers on the cathode ray display tubes. Associated with the joystick are control keys which may be used to send a message to the controlling computer specifying the coordinates of the cross hairs. Control programs may be written, for example, to act upon this information to reorient the display with respect to the area selected by the cross hair position.

A Light Gun is also provided as a means of selecting any point on the cathode ray tube displays. The gun emits a small beam of light. With the beam centered on a given point on the cathode ray display tube, pressing the trigger results in the automatic generation of a message to the Peripheral Buffer specifying the address in the Display Buffer containing the coordinates of the selected point.

A set of Status and Error lights are contained on the Display Console to provide the console operator with over-all knowledge of the system and thus minimize conflicting control requests and intermodule interference. For example, a Peripheral Buffer may not be ready to accept a console key message until after certain previously requested control actions have been completed. The Status Lights indicate this condition to the console operator so that he may act accordingly.

The printer module

The Printer Module (PR) is basically a 160 column, 900 line per minute Anelex type printer. It receives information from either a Computer or a Buffer module via the Central Exchange. Indi-

vidual characters to be printed are represented by a 6-bit code and are transmitted four to a computer word. Zero suppression, line completion and information block end codes are included for format control. A plugboard is provided for flexibility in columnar data arrangement. Paper feed is controlled by means of a loop of 7-channel punched paper tape. Control of the printing operation has been arranged so that the connected control module may send line headings from one set of memory locations, stop sending information while going to a different part of the memory, and then proceed to send data from this new set of memory locations to complete a line of print.

The punched card modules

The RW-400 Data System may be equipped with a high speed punched card reading module (CR) and an IBM card punch. The CR communicates with Computer or Buffer modules via the Central Exchange. It is capable of reading 80 column punched cards at the rate of 2,500 cards per minute. The card punch is connected to the system through the Peripheral Buffer Module (PB) since it is a relatively low speed device. Emphasis has not been placed on directly connected punched card equipment since the sources of large volumes of punched cards usually convert this data into magnetic tape form which may be more rapidly handled using the Tape Adapter Module (TA).

References

RothS59; WestG60

APPENDIX 1 RW 40 ISP DESCRIPTION

```
                                Appendix I

                          RW-40 ISP Description

The description does not include Input-Output instructions, interrupts and communication with the other computers or processors.
The description was taken from the Preliminary Manual of Information on the RW-40 and is no doubt changed in final machines.

Pc State
   A<26:1>                                   Arithmetic register
   B<26:1>                                   extension to A
   AB[0:1]<26:1>:= A□B                       Arithmetic register (double)
   P<10:1>                                   Program Counter
   Ov                                        Overflow for arithmetic shifts, +, -, and /
   SR<20:1>                                  Sense Register
   Parity error                              for Mp and transfer to other computers
   Program error                             undefined command or incorrect sequence of IO commands
   Run

Mp State
   M[1:1022]<26:1>                           Mp registers 0 and 1023 are inaccessible

Pc Console State
   CJS<8:1>                                  conditional jump switches
   Control␣panel␣test                        communication indicator

External State for IO and Other Computers
   Tape␣read                                 tape search flag
   External␣Address/EA<10:1>                 register associated Pc to address another module
   M[0:1023]<26:1>                           extra memory being accessed by External Address register
   I␣cond<19:1>                              interrupt conditions to Pc
   IO␣Select<3:1>                            1 of 8 IO devices can be selected
   IO␣Data<13:1>                             IO device Data

Instruction Format
   instruction/i<26:1>
      f/op<6:1> := I<26:21>                  function or op code bits
      g<10:1>   := i<20:11>                  first address
      J<5:1>    := g<5:1>                    test selection parameter
      h<10:1>   := I<10:1>                   second address

Operand Calculation Process
   G<26:1> := (G'; next                      first operand
      (g = 1777₈) →External␣Address ← External␣Address + 1)
      G'<26:1> := ((g = 0) → 0;
                   (0 < g < 1777) →M[q]<26:1>;
                   (g = 1777) → M[External␣Address]<26:1>)
   H<26:1> := (H'; next                      second operand
      (h = 1777) →External␣Address ←External␣Address + 1)
      H'<26:1> := ((h = 0) → 0;
                   (0<h<1777) → M[h]<26:1>
                   (g = 1777) → M[External␣Address]<26:1>)
```

```
Instruction Interpretation Process
   Run→(instruction ←M[P]: P ←P + 1; next                          fetch
      Instruction_execution)                                       execute

Instruction Set and Instruction Execution Process
   Instruction_execution := (
   Transmit                    (:= op = 27) → (H ← G):
   Arithmetic (1's complement)
   Replace Add                 (:= op = 0) → (Ov,A ← H + G: next H' ← A);
   Hold Add                    (:= op = 1) → (Ov,A ← H + G):
   Store Add                   (:= op = 2) → (Ov,A ← A + G; next H' ← A):
   Replace Subtract            (:= op = 3) → (Ov,A ← H - G: next H' ← A);
   Hold Subtract               (:= op = 4) → (Ov,A ← H - G);
   Store Subtract              (:= op = 5) → (Ov,A ← A - G: next H' ← A);
   Replace Absolute Subtract   (:= op = 6) → (A ← abs(H) - abs(G); next H' ← A);
   Hold Absolute Subtract      (:= op = 7) → (A ← abs(H) - abs(G));
   Store Absolute Subtract     (:= op = 10) → (A ← abs(A) - abs(G): next H' ← A);
   Replace Multiply            (:= op = 11) → (AR ← H x G: next H' ← A);
   Hold Multiply               (:= op = 12) → (AB ← H x G):
   Store Multiply              (:= op = 13) → (AR ← A x G: next H' ← A):
   Replace Divide              (:= op = 14) → ((H ≥ G) →Ov ← 1;
                                              (H < G) → (
                                               A,B ←H/G: next H' ← A)):
   Hold Divide                 (:= op = 15) → ((H ≥ G) →Ov ← 1;
                                              (H < G) → (A,B ←H/G)):
   Store Divide                (:= op = 16) → ((A ≥ G) →Ov ← 1;
                                              (A < G) → (
                                               A,B ←A/G; next H' ← A));
   Replace Square Root         (:= op = 17) → (A ← sqrt(H+G); next H' ← A);
   Hold Square Root            (:= op = 20) → (A ← sqrt(H+G));
   Store Square Root           (:= op = 21) → (A ← sqrt(A+G): next H' ← A);
   Accumulate Add              (:= op = 25) → (A ← Ov□A + H + G);
   Accumulate Multiply         (:= op = 26) → (A ← Ov□A + H x G);
Shift, g<10:1> is used to control the shift as follows:
   g<5:1> specifies number of shifts
   g<6> = 1 → shift left; g<6> = 0 → shift right
   g<7> = 1 → indicate an overflow
   g<8> = 1 → round the result
   g<9> = 1 → signed arithmetic; g<9> = 0 → [logical]
   g<10>= 0 → A is the operand; g<10> = 1 → AB is the operand
   Shift := (op = 30) → (Ov□A ← f(A x 2^g<5:1>, B x 2^g<5:1>, g<6:10>): next H ← A);
Logical or boolean vector data:
   Replace Insert    (:= op = 22) → (A ← (H ∧ ¬ G) ∨ (A ∧ G); next H' ← A);
   Hold Insert       (:= op = 23) → (A ← (H ∧ ¬ G) ∨ (A ∧ G));
   Store Insert      (:= op = 24) → (A ← A ∧ G; next H' ← A);
   Test Jump         (:= op = 31) → (
```

```
        (g<10:7> ≠ 0) → (
          A ← ((g<10> → I␣cond; ¬ g<10> → 177777777) ∧
               (g<9> → SR; ¬ g<9> → 177777777) ∧
               (g<8> → IO□Select□IO␣Data; ¬ g<8> → 177777777) ∧
               (g<7> → CJS; ¬ g<7> → 177777777)); next
        (g<6> ⊕ Test) → (P ← h));
    The Test condition is a selected bit of A, or other Pc or IO bits.
        Test := ((j = 0) → 0;
                 (i ≤ j ≤ 32) → A<J>;
                 (j = 33) → (Ov; Ov ← 0);
                 (j = 34) → (Parity error; Parity error ← 0);
                 (j = 35) → (Control␣panel test; Control␣panel␣test ← 0);
                 (j = 36) → (Tape␣read; Tape␣read ← 0);
                 (j = 37) → (Program␣error; Program␣error ← 0))
      Link Jump                  (:= op = 32) → ((g ≠ 0) → (P ← h; G<10:1> ← P);
                                                 (g = 0) → (P ← h));
      Tally Jump                 (:= op = 33) → ((G = ¬ 0) → (P ← h);
                                                 (G = 0) → ;
                                                 (G > 0) → (G' ← G - 1; P ← h);
                                                 (G < 0) → (G ← G + 1));
      Compare Jump               (:= op = 37) → (A ≤ G) → P ← h;
      Load A                     (:= op = 34) → (A ← 0□g□h);
      Insert S                   (:= op = 35) → (S ← (A ∧ (0□g□h)) ∨ (S ∧ ¬ (0□g□h)));
      Store AB                   (:= op = 36) → (G ← B; H ← A;
                                                 (g = 0) ∧ (h = 0) → (A ← B; B ← A))
                                                 )                  end Instruction␣execution
```