HONEYWELL 1800-II

A Large-Scale Scientific Processor

George B. Bailey
C. Norman Canning
William S. Grinker
Rolf Kates
William L. Mellentin
Norman Nisenoff

TABLE OF CONTENTS

INTRODUCTION

SCIENTIFIC CONSIDERATIONS

A great variety of factors determine the applicability and utility of a Data Processing System with respect to the requirements of the scientific community. Any computer, no matter how fast or large, is unfinished unless its hardware is properly complemented by a full repertoire of software that is capable of economically assimilating and solving the wide range of problems encountered in a Scientific Computing Facility.

The Honeywell H-1800-II has made a significant breakthrough in total SYSTEM'S CONCEPT by combining extremely fast Central Processor Operations, flexible Peripheral Devices, completely proven and integrated SOFTWARE, and Parallel Processing at a modest Systems cost.

The factors that have prevented users from achieving total utility from competitive systems have been solved by the H-1800-II as follows:

RAW SPEED

-120,000 three address operations per second execution, the equivalent of 300,000 single address operations per second.

-10 microsecond floating binary multiplication for a full 48 bit word.

TURN-AROUND TIME

The availability of all peripheral units to the central processor through the use of the Input Output Control Center (IOCC) and hardware Parallel Processing allow AUTOMATH 1800 (FORTRAN IV) to operate in a mixture of load and go and execute later modes. This combination of hardware and software give the user IMMEDIATE turn-around.

FLEXIBILITY

The hardware features of Multiprogram Control and Buffer Traffic Control combined with the ADMIRAL Master Monitor allows complete independence of up to eight programs operating in parallel. Not only are these programs scheduled and loaded by the monitor, but its Dynamic characteristics allow for immediate emergency priorities to overide the schedule, stall programs in execution, execute the priority job and restart all stalled programs after completion of the priority job.

PRECISION

A full 48 Bit word is utilized to provide a 40 Bit Mantissa, 7 Bit characteristic and 1 bit sign for extended floating point accuracy.

ERROR SITUATIONS

The H-1800-II is completely error checked on all Input/ Output Operations, internal transfer and arithmetic operations. ORTHOTRONIC Control provides almost error free magnetic tape operation by automatically regenerating magnetic tape errors. All errors are isolated with hardware jumps to special error locations without altering the affected sequence counter.

SOFTWARE SYSTEMS INTEGRATION

All Honeywell H-1800-II Programming Systems and Routines are completely integrated under control of the ADMIRAL Master Monitor. The ability to communicate between systems and the high degree of documentation and standardization posture is a substantial operating advantage.

INSTRUCTION REPERTOIRE

A complete complement of arithmetic, logical and input/ output commands for the binary, alpha and decimal modes. The floating point orders are extensive and include hardware conversion for both the floating binary and decimal modes.

SPECIAL EQUIPMENT

Honeywell H-1800-II has a hardware organization that allows for a wide variety of special equipment to be easily integrated to meet unique customer requirements.

JOB COST

The modest cost of the H-1800-II combined with its outstanding performance generates the lowest job cost of any Data Processing System available.

Your attention is directed toward Section VI which dramatically demonstrates that the H-1800-II is superior on a price/power basis by at least a factor of 2.1 to 1 over any other presently available competitive system for a pure scientific benchmark. On a commercial benchmark of 10,000 records the H-1800-II scores at least 30% more economical than all competitive systems.

LARGE MEMORY

Up to 65,536 words of high speed core storage of 48 bits each with a 2 u s access cycle.

SUMMARY OF ADVANTAGES

SUMMARY OF ADVANTAGES

THE HONEYWELL H-1800-II AS A SCIENTIFIC PROCESSOR

The Honeywell H-1800-II is basically a binary fixed word length machine, each word consisting of 48 information bits and six parity bits. The fact that the H-1800-II has a 48 bit word length and the ability to operate in the floating binary mode, as well as the floating decimal mode makes the H-1800-II a powerful computer for processing scientifically oriented problems.

HARDWARE ADVANTAGES

1. 120,000 THREE ADDRESS OPERATIONS PER SECOND.
   2 MICROSECOND ACCESS TO A FULL 48 BIT WORD

2. FAST FLOATING POINT OPERATIONS - (A 48 BIT FLOATING BINARY MULTIPLY IN 12 US)
   48 BIT WORD FOR EXPANDED PRECISION IN FLOATING POINT OPERATIONS
   - 40 BIT MANTISSA
   - 7 BIT CHARACTERISTIC
   - 1 BIT SIGN

3. PARALLEL PROCESSING OF UP TO EIGHT COMPLETELY INDEPENDENT PROGRAMS, CONTROLLED BY HARDWARE AND REQUIRING NO PROGRAMMING OR MASTER EXECUTIVE ROUTINE.

4. MULTIPROGRAM CONTROL DIRECTS THE TIME SHARING OF THE CENTRAL PROCESSOR FOR UP TO EIGHT ACTIVE PROGRAMS AT NO COST IN CENTRAL PROCESSOR TIME. EACH OF THE PROGRAMS PROCESSED IN PARALLEL IS ASSIGNED ONE GROUP OF SPECIAL REGISTERS IN THE CONTROL MEMORY. ONCE THE PROGRAM IS LOADED, ITS CORRESPONDING GROUP DIRECTS THE SELECTION OF INSTRUCTIONS. IN THIS WAY, EACH PROGRAM MAY START, PROCEED AND STOP INDEPENDENTLY OF ALL OTHER PROGRAMS.

5. TRAFFIC CONTROL IS THE HONEYWELL 1800-II HARDWARE ELEMENT WHICH DIRECTS THE TIME SHARING OF MEMORY BY THE TAPE AND PERIPHERAL UNITS AND THE CENTRAL PROCESSOR. THE PROVISION OF HAVING SIXTEEN INPUT/OUTPUT CHANNELS MEANS AN UNUSUALLY WIDE RANGE OF SIMULTANEOUS PERIPHERAL OPERATIONS IS POSSIBLE.

6. FOR GREATER EFFICIENCY IN MAJOR INPUT/OUTPUT OPERATIONS, HONEYWELL OFFERS A VERSATILE CENTRAL PROCESSOR. A KEY FEATURE OF THIS CENTRAL PROCESSOR IS AN INPUT/OUTPUT CONTROL CENTER (IOCC) THAT PROVIDES COMPLETE CONTROL FUNCTIONS FOR A CARD READER/PUNCH, A HIGH SPEED PRINTER, AND UP TO FOUR MAGNETIC TAPE UNITS. THE IOCC COMPLETELY CONTROLS THE SIMULTANEOUS OPERATION OF THE DEVICES CONNECTED TO IT, AND, IN ADDITION, IT ALLOWS THESE DEVICES TO OPERATE SIMULTANEOUSLY WITH OTHER INPUT/OUTPUT EQUIPMENT CONNECTED TO STANDARD PERIPHERAL CONTROLS.

7. ORTHOTRONIC CONTROL IS AN EXCLUSIVE PROCESS WHICH NOT ONLY CHECKS FOR ERRORS ON TAPE AND IN PERIPHERAL COMPONENTS, BUT ACTUALLY REGENERATES INFORMATION WHICH MIGHT HAVE BECOME LOST OR DESTROYED ON THE MAGNETIC TAPE AND DISC.

8. OVERLAP OF INPUT,OUTPUT, AND COMPUTING.

9. 64 INDEX REGISTERS AND 128 GENERAL PURPOSE REGISTERS FOR INDIRECT MEMORY ADDRESSING SEPARATE FROM MAIN MEMORY.

10. COMPLETE INTERNAL ERROR CHECKING.

11. COMPLETE INDEPENDENCY OF PROGRAMS TO BE RUN IN PARALLEL TO THE EXTENT THAT A PROGRAM CAN BE DELETED FROM THE SCHEDULE AT RUNNING TIME WITHOUT AFFECTING OTHER PROGRAMS.

12. COMMUNICATIONS CONTROL ALLOWS THE CENTRAL PROCESSOR TO RECEIVE AND TRANSMIT DATA OVER TOLL OR LEASED WIRE CIRCUITS IN A VARIETY OF MODES.

SOFTWARE ADVANTAGES

1. 1000 FORTRAN IV STATEMENTS PER MINUTE COMPILATION RATE WITH AUTOMATH 1800.

   LOAD AND GO MODE WITH IMMEDIATE TURN AROUND
   EXECUTE LATER MODE
   PARALLEL COMPILATION AND EXECUTION
   DYNAMIC STORAGE ALLOCATION
   MEMORY OVERLAY
   DYNAMIC RELOCATION
   CODE OPTIMIZATION
   LOOP RESEQUENCING
   SUBSCRIPT OPTIMIZATION
   INTERSPERSED ARGUS ASSEMBLY LANGUAGE

2. ADMIRAL MASTER MONITOR FOR DYNAMIC CONTROL OF SCHEDULING, LOADING, STARTING AND EXECUTION OF ALL PROGRAMS AND SYSTEMS.

3. ARGUS (AUTOMATIC ROUTINE GENERATING AND UPDATING SYSTEM). ARGUS CONSISTS OF A MNEMONIC AND SYMBOLIC LANGUAGE FOR PROGRAM PREPARATION, AN ASSEMBLY ROUTINE WHICH TRANSLATES THIS NOTATION INTO THE INTERNAL LANGUAGE OF THE HONEYWELL 1800-II AND INSERTS EXISTING SUBROUTINES AT POINTS SPECIFIED BY THE PROGRAMMER, SORT GENERATORS CAPABLE OF TAILORING A SORT ROUTINE TO A SPECIFIC SET OF INPUT DATA, AND A PROGRAM TEST SYSTEM WHICH AUTOMATICALLY EXECUTES A SERIES OF PROGRAMS TO PROVIDE EXTENSIVE INFORMATION ABOUT THEIR OPERATION.

4. EXTENSIVE SCIENTIFIC PROGRAM LIBRARY AND UTILITY SYSTEMS.

5. FACT (FULLY AUTOMATIC COMPILING TECHNIQUE). FACT IS AN ENGLISH NARRATIVE WHICH IS EXTREMELY RICH IN LANGUAGE FORM FOR DATA HANDLING PROCEDURES. IT WILL GENERATE MACHINE CODE FOR ALL ASPECTS OF INPUT EDITING, DATA SORTING, FILE UPDATING, AND GENERATION OF PRINTED REPORTS.

6. COBOL, IMPLEMENTATION OF COBOL 61 WITH FULL OPTIONS AND TAKING ADVANTAGE OF THE PARALLEL PROCESSING CAPABILITIES OF THE H-1800-II. COMPILATION WILL EXECUTE AT APPROXIMATELY 600-800 STATEMENTS PER MINUTE.

H-1800-II

HARDWARE CHARACTERISTICS

THE HONEYWELL 1800 SYSTEM

The Equipment

A Honeywell 1800 Electronic Data Processing System consists of a central processor plus varying types and numbers of input and output units. The programmer must know the system configuration with which he is to work. An understanding of the function of each component and its relation to the entire system will make his task easier.

1. The Central Processor (1801)

The basic central processor consists of a control unit, a control or special-register memory, an arithmetic unit, and four banks of main (or high-speed) memory, each able to store 2048 Honeywell 1800 words. (A Honeywell 1800 word is composed of 48 information bits and six checking bits.) To this basic unit, additional memory banks can be added in modules of 8192 words up to a maximum of **65,536** words. An optional floating-point unit (1801-B, discussed below) is also available.

The control unit with its control memory is the nerve center of the central processor. As the site of traffic control and multiprogram control (explained below), it monitors the time sharing of the entire system to achieve maximum efficiency of operation. In addition to its multiplexing function, it is also the unit that selects, interprets, and directs the execution of instructions, and governs address selection in both control memory and main memory.

The memory cycle time is two microseconds. This is the time required to read one Honeywell 1800 word from memory (access) and to replace it in its original form (restoration).

The control memory is a magnetic-core array providing storage for 256 eighteen-bit words. The read-restore cycle of the control memory is out of phase with that of the main memory in such a way that if reference must be made to the control memory between references to main memory, it is usually possible to make such reference without loss of a main-memory cycle. As discussed more fully in Section V, the control memory contains eight identical groups of special registers such as sequencing counters, index registers,

registers used for indirect addressing, etc., the contents of which are used to select a full Honeywell 1800 word from the main memory. The offset cycle of control memory makes it possible to anticipate an address selection involving the contents of a special register and to prepare the address of a second operand while another unit is using the first operand. Because of this anticipatory technique, it is unnecessary in many cases to add memory cycles to an instruction for indexed or indirect addressing. Even when the contents of a special register are modified before they are restored, no extra memory cycle need be added, since the special register circuitry includes a separate adder, with complete and independent checking, used only for special register modification. This applies to both automatic modification, as when a sequencing counter is incremented after use, and program-controlled modification, as when an increment is specified in an address. However, while the two memory units are sufficiently out of phase to allow reading from the control memory prior to the start of a main memory cycle, a read-restore operation in which the result of an instruction is returned to a special register cannot overlap a main memory cycle; in this case, an extra cycle must be added to the instruction time.

The arithmetic unit is the portion of the central processor in which digits are combined to form new arrays in accordance with the logical rules of the command codes. The Honeywell 1800 central processor has provision for both binary and decimal arithmetic, complete logical abilities, and competent internal checking. For the interested reader, a complete description of the fixed-point addition logic can be found in Appendix A.

2. The Floating-Point Option (1801-B)

The floating-point option (1801-B) is a second control and arithmetic unit which provides the Honeywell 1800 user with 19 instructions that manipulate data in floating-point form, plus two fixed-point division instructions. The control portion of the unit selects, interprets, and directs the execution of these instructions by the arithmetic portion. In an 1800 system not equipped with an 1801-B, these instructions are not executed directly but are interpreted as pseudo instructions which call in library routines to perform the desired operations. Descriptions of this unit, its component registers, and the instructions it provides are presented in Section XIII.

3. The Console

The Honeywell 1800 console is basically a part of the central processor and

is multiplexed into the system via multiprogram control (see below). A monitor typewriter is used by the operator to communicate directly with the central processor. Manual operations on the typewriter can start and stop individual programs and interrogate Honeywell 1800 storage. Under program control, the console typewriter can also print information useful to the operator. An additional typewriter, called the console slave typewriter, can be added to the system. No manual operations can be performed from the slave typewriter, but printing can be programmed to occur on either the slave or the console monitor typewriter. In addition to the typewriter(s), the console includes display lights that give the operator an at-a-glance summary of the number of active programs, their control centers, and their progress. The console also includes a modular display panel which has indicators and displays for monitoring the status of tape units and other peripheral devices. From 1 to 45 remote inquiry stations can be included in the system for direct interrogation of stored information and printout of results.

4. Magnetic Tape Units (804) and Tape Controls (803)

The Honeywell 1800 magnetic tape units are designed for reliability and accuracy. The recording surface of the tape is untouched except by the read/record head, insuring that wear and damage to the surface are held to an absolute minimum. Vacuum is used to seat the reels on the hubs, to maintain loops in the loop chambers, and to provide contact with the capstans which cause the tape to move under the head, giving accurate control without the dangers inherent in mechanical techniques. The tape is edge guided along its entire path to protect the tape edges from damage and to avoid skew.

The system uses 3/4-inch-wide tape with Mylar[1] (polyester film) base and oxide coating. The recording portion of a full reel of magnetic tape is 2500 feet in length, ± 50 feet. A 30-foot clear leader precedes the recording portion and a 25-foot clear leader follows it. Information is written on tape in 10 longitudinal channels, eight for information bits from the word, one for a parity-checking bit, and one for a clocking indicator. One array of bits across the tape is called a frame. Information from six frames makes up the 54-bit word, which includes six parity bits. The clocking channel gives positive indication of the frame location on tape.

Variable-length recording is a basic feature of the Honeywell 1800, and

[1]Mylar is a registered trademark of E. I. du Pont de Nemours and Co., (Inc.).

records of any size may be read from or written on tape, although for control purposes a maximum size limit may be placed on records at the beginning or end of tape. Gaps between records are 0.67 inch long.

Four different tape units are available for the Honeywell 1800. Their recording and rewind speeds, recording densities, and transfer rates are shown in the table below. The first three tape systems are compatible: data recorded by one system can be read at recording density by the others.

Model	Speed (inches per second)		Recording Density (frames per inch)	Transfer Rate (decimal digits per second)
	Read/Write	Rewind		
804-1 Magnetic Tape Unit	120	360	400	96,000
804-2 High-Density Magnetic Tape Unit	120	360	555.5	133,300
804-3 Economy Magnetic Tape Unit	60	180	400	48,000
804-4 Super-Density Magnetic Tape Unit	120	360	775	186,000

When rewinding, the tape moves at three times normal speed. A small photo-electric device in the tape unit senses the presence of edge "windows" (clear Mylar) in the tape to provide beginning-of-tape and end-of-tape indications for the programmer. A physical slot in each leader is used to negate the vacuum and stop the unit when the end of tape is reached.

When a metal ring is inserted in the front of a tape reel, writing is allowed to take place on the tape. When this ring is removed, the tape is protected and cannot be written upon. The ring may be installed or removed without rewinding the tape or removing the reel. There is, in addition, a manual switch on the tape unit panel which can be set to prohibit writing.

A tape control can control up to eight tape units. The model 803-1, 803-2, 803-3, and 803-4 tape controls are used with 804-1, 804-2, 804-3, and 804-4 tape units, respectively. Each control has an input and an output channel so that one of the tape units attached to the control may be reading and another writing simultaneously. Designation of tape units as active or inactive is accomplished through the use of the tape address patchboard on the maintenance and indicator panel of the tape control. The patchboard also allows for the re-addressing of tape units; for example, the unit physically designated as number 1 may have, as a result of patchboard plugging, an effective address of 3.

Traffic Control

Traffic control is the Honeywell 1800 element which directs the time-sharing of memory by the tape and peripheral units and the central processor. Multiprogram control is the element which directs the time-sharing of the central processor by the active program control centers. A clear concept of both of these elements is basic to the understanding of parallel processing and allowable system configurations and is the key to a thorough knowledge of the Honeywell 1800.

Traffic control has as its main object the efficient use of the entire system according to a set of priorities which derive directly from the nature of the equipment and are independent of the programs. For example, an 804-1 magnetic tape unit reading at full speed assembles one Honeywell 1800 word in a one-word buffer every 125 microseconds. If instant access is not provided to memory, a second word of buffer storage must be provided to retain this word. At the end of the next 125 microseconds, another word will have been read. If the first word has not yet been placed in main memory, another word of buffer storage must be provided. Since eventually one access to memory must be made for each word to be stored, it is obviously economic to store each word as it is assembled from tape and thus reduce the required buffer storage to a minimum. However, to keep the memory continuously available to the tape buffer during a read operation would be to introduce inefficiencies in the system, for only one buffer cycle (six microseconds) is needed to store each word, and during the remaining 119 microseconds the entire system would be idle.

In the Honeywell 1800, one access to memory every 125 microseconds is guaranteed each tape unit. When a word is assembled from six frames for storage, a demand signal is generated by the buffer for one access to memory and is honored within 125 microseconds, clearing the buffer. In this case, only two words of storage are needed for each active tape unit, and the memory is utilized by the input/output operation only 6n microseconds out of each 125 microseconds, where n equals the number of active units. To achieve this time-sharing, traffic control monitors the demand signals from the buffers and arranges access to the memory within the prescribed time for each buffer demand.

As its name implies, traffic control monitors the transmission of information to and from the main memory. Its operation is represented schematically in Figure II-2. The 17 divisions of the band are called "stages" and one stage is assigned to each of the eight output

channels, each of the eight input channels, and the central processor.

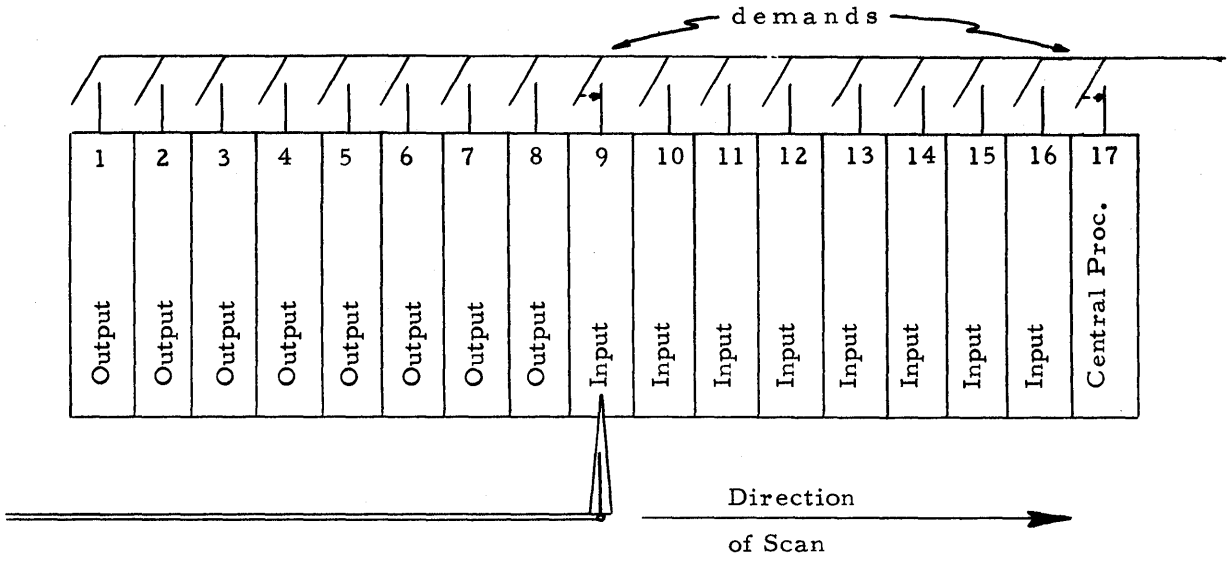


Figure II-2. Stages of Traffic Control

The creation of a demand signal by any device is represented in Figure II-2 by the closing of the switch shown in the corresponding control stage. When any program has been turned on in the central processor, the switch corresponding to the central processor stage is continuously closed. Traffic control begins each scan at the left end of the band. It proceeds to the right, ignoring all stages which show no demand signal, until a demand stage is reached. This stage is allowed access to the main memory for one buffer cycle only. Traffic control then returns to the left end of the band to begin the next scan. Because the control search is anticipatory, no system time is consumed in bypassing stages in which no demand exists.

If no input/output devices are active but a program is running, the central processor stage will have complete use of the memory since each scan of the band will find no other demand signals. If a magnetic tape unit attached to channel 1 is reading, then once every 125 microseconds a demand signal will halt traffic control at the stage marked "input 9" for one buffer cycle to allow transfer of the assembled word from buffer storage to memory. Thus, between each memory cycle of instruction execution in the central processor, 0 to 16 buffer cycles may intervene, depending on the number of devices which require access to memory.

Since, in normal read-write operations, only one buffer cycle may be allowed any stage before the next scan of the band, a maximum of 16 buffer cycles or 96 microseconds will elapse between successive interrogations of any stage. As this is within the 125 microsecond maximum, eight 804-1 tape units may be reading and eight may be writing simultaneously with central

processor operations without conflict in demands for access to memory. Furthermore, since the memory is made available to the central processor for any cycle in which it cannot be utilized by an input/output stage, no idle time is introduced as long as any program is active. Thus, traffic control insures that the system responds to input/output device demands as required without introducing idle memory cycles, and that as long as any program can proceed, useful work is being done.

The rule of operation which states that only one buffer cycle may be allowed any stage before the next scan has one exception. When a magnetic tape unit is executing a distributed read or distributed write operation, two consecutive buffer cycles are allowed a stage for modifying the appropriate address counter on recognition of the demand signal accompanying an end-of-item word. Distributed read and write instructions are more fully discussed in Section XI. Two points in connection with this exception bear mentioning here. First, the difference between the normal 96-microsecond cycle and the maximum allowable 125-microsecond cycle allows servicing of four such demands without timing conflicts. Secondly, in the improbable case that all system channels are active and five or more channels are being used in distributed tape operations, and at least five of these each assemble an end-of-item word and create a demand signal within 125 microseconds of each other, an error signal will be generated for any unit whose demand is not serviced within the required time limit. This signal will appear to the program as a normal reading error signal indicating the need for a reread from the specified unit. Complete information is available to the program so that intelligent action can be taken under program control.

The preceding discussion of traffic control is based on the use of model 804-1 magnetic tape units which have an instantaneous transfer rate of 96,000 decimal digits per second. The use of input/output units with higher transfer rates may not permit simultaneous use of all 16 input/output channels. For example, the model 804-2 magnetic tape unit, with an instantaneous transfer rate of 133,300 decimal digits per second, requires one access to memory every 90 microseconds when operating at full speed.

All existing 1800 peripheral devices are designed to be attached to standard input/output trunks with 6-microsecond buffer cycles. However, it is possible to modify any input/output trunk or any combination of trunks to operate on a 2-microsecond buffer cycle in order to attach special high-speed devices to the system. It is also possible, by means of a simple field change, to modify the traffic control priority of any input/output trunk(s) within the following restriction: If one or more 6-microsecond input trunks are assigned higher priorities than any 6-microsecond output trunk, they must be assigned higher priorities than all 6-microsecond output trunks.

Table showing typical control units, transfer rates, buffer capacities for various peripheral devices

Devices	Max. Devices Per Trunk	Channels	Rate (words/sec) Data Transmission	Control Unit Req'd.	Buffer Size (Words)	Time to Fill Buffer (usec)
Magnetic Tape Unit-Standard 803-1	8	I/O	8,000	one-804-1	1 in 1 out	125
MTU-Economy 803-3	8	I/O	4,000	one-804-3	1 I/O	250
MTU-High Density 803-2	8	I/O	11,000	one-804-2	1	90
MTU-Super Density 803-4	8	I/O	15,500	one-804-4	1	64
Printer (900 LPM)	1	Output	225	one-806-3	15	4,444
Card reader/Punch 800 cds/min	1	Input	126.6	one-807-3	10	7,500
& 250 cds/min	1	Output	41.6	one-808-3	10	24,000
Paper Tape Reader Control	1	Input	1,000 (1 char/wd)	809	1	1,000
Paper Tape Punch	1	Output	100	810	1	10,000
Real Time (non-simul)	1	I/O	X	812-1	1	varies
Real Time (simul)	1	I/O	X	812-2	1	varies
Random Access Unit	1	I/O	11,000	860-1		90
IOCC	1	I/O		integral w/CP	1	depends on peripheral unit used

Multiprogram Control

Multiprogram control directs the time-sharing of the central processor by the active programs. Each of the eight groups of special registers may direct the execution of an independent program. After a program is loaded, one of the special register groups is activated to direct the selection of instructions. This special register group, and the program it directs, is said to be "on." When the program is completed, the directing special register group is inactivated (turned "off"). Special register groups may be turned on and off independently of each other, either from the console or by program control.

Each time traffic control allows the central processor access to main memory, one memory cycle of one instruction is performed. If only one special register group is active, all cycles allowed the central processor are used in executing instructions from the active program. Since traffic control allows the central processor all available cycles except those needed to honor the intermittent demands of tape and peripheral devices, this case represents that of the conventional single-program computing machine with the ability to implement input/output operations simultaneously with computing.

When more than one special register group becomes active, central processor cycles must be shared among the several programs. The rules under which multiprogram control operates are as follows:

1. When an active special register group is selected, the next cycle allowed the central processor must be used to select the next instruction to be executed in the program controlled by that group.
2. All succeeding central processor cycles must be devoted to the execution of this instruction until it is completed. (This may range from two cycles upward.)
3. If the instruction is one which does not leave unstored information in the arithmetic or control units, its completion causes multiprogram control to scan or "hunt" for the next active special register group in sequence.
4. If the instruction does leave unstored information in the arithmetic or control unit, scanning or hunting is inhibited and the next instruction is selected from the same program. Hunting is not allowed if:
    a. the instruction generates a two-word result, of which only one word is stored in memory as the result of the instruction execution, e.g., multiply;
    b. the execution of the instruction results in a sequence change; in the case of a conditional change, hunting is inhibited only if the condition is satisfied;
    c. the instruction has an inactive C address;
    d. an unprogrammed transfer takes place as the result of executing the instruction;
    e. the instruction is capable of specifying that hunting shall not take place, and does so specify.

For example, if three programs are active, then one instruction is performed in turn from program 1, then 2, then 3, then 1, and so on, as long as all instructions selected allow hunting. Such alternation is temporarily held up if an instruction is selected which does not allow hunting, but it is resumed as soon as an instruction is selected which does allow hunting. Thus, the central processor cycles are shared on a fairly equal basis among all active programs.

The true power of multiprogram control appears when an active program attempts to execute an instruction which cannot be implemented because of the unavailability of a system component. Suppose, for example, that a write instruction calling for tape 4 is selected from an active program. The central processor, in attempting to execute this instruction, finds that either tape 4 or its associated output channel is involved in executing another instruction from the same or some other program. Seeing that the instruction cannot be executed immediately and recognizing the reason therefore, multiprogram control places the special register group in a "stall" condition. This condition indicates to multiprogram control that a) this program, although still active, shall not be allowed any central processor cycles as long as the "stall" indication remains, and b) when the channel and/or the device involved completes its present task, the stall condition shall be automatically removed and the program restored to its full active status.

Thus, when an instruction cannot proceed because of input/output conflicts with either the same or another program, the central processor cycles which it would have used are made available to the other active programs, causing them to proceed faster. The result of the operation of multiprogram control is that there is never an idle memory cycle in a Honeywell 1800 system as long as there is any active program in which an instruction can be executed.

If more than one active program is stalled because of input/output conflict, multiprogram control remembers which program was stalled first and operates on a "first off - first on" basis. Other stalled programs are activated according to the normal scanning sequence.

Although it need not concern the programmer, the reader may be interested in a short discussion of the actual machine procedure in case of program "stalls." The computer will have selected the instruction and entered the second cycle of execution before the unavailability of the input or output channel and/or device is discovered. No memory alteration will have been made, but the sequencing counter will have been advanced by one. When the stall condition is ascertained, the computer will subtract one from the sequencing counter and deactivate the program from multiprogram control. Each time that any input/output channel or device terminates an operation, all programs in a stalled state are reactivated. Multiprogram

control looks again at each reactivated program, replacing in a stalled condition those programs whose input/output demands still conflict.

Multiprogram control also receives demand signals generated by the console and by inquiry stations. The console is regarded by multiprogram control as a ninth group of special registers, except that it takes precedence over any active program in the assignment of available central processor time. A demand signal from the console is serviced at the time of the next hunt, although a non-hunting sequence of instructions will not be interrupted.

When a console demand is recognized, the computer generates an instruction to implement the activity indicated by the console command. Sufficient memory cycles are then assigned to execute this generated instruction in the same fashion as if it had been selected from memory. When the instruction has been executed, the normal hunting process is resumed. If the system includes one or more remote inquiry stations, demand signals from these stations are recognized and implemented in a similar fashion. This technique allows the operator to communicate manually with the central processor without stopping the computer, and thus allows the system full-efficiency operation even during manual manipulation on one program.

Orthotronic Control and Checking

Orthotronic control is a powerful technique, exclusive with Honeywell, which insures against loss of information from magnetic tape during writing, storage, or subsequent reading. Experienced data processing personnel know that long storage periods or inept operator handling can cause information to disappear from a tape even though the accuracy of the record was checked at the time the record was written. Even infrequent occurrences of this type can result in many man-hours and machine-hours spent in re-creation of the records. While no technique will ever completely eliminate information loss, the high reliability and accuracy of the Honeywell 804 tape units, plus the presence of orthotronic control as a standard feature of every Honeywell 1800 system, insure that such loss is eliminated as a practical problem.

Orthotronic control is based on studies of the types and extent of information losses which have occurred on magnetic tape systems. It is partly automatic and partly program-controlled. An instruction is provided which automatically creates two orthotronic words for a specified record. These words are a logical combination of all the words in the record such that only a highly unlikely periodicity of error can go undetected and uncorrected. The orthotronic words are automatically positioned to accompany the record as it is written. Read and write instructions assume the presence of the orthowords and automatically include them in the record, using them in an automatic first-level check of the correctness of the information handled. The instruction which generates the original orthowords may also be used to reconstruct missing information

if loss is detected. A full discussion of orthotronic control can be found in Appendix B.

Orthotronic control is a checking device peculiar to the magnetic tape units. In on-line operation, the central processor must be programmed to reconstruct lost data from a garbled record. When tapes are read or written in an off-line configuration, however, an off-line auxiliary control provides, in part, the central processor function. The off-line input auxiliary control automatically generates the two orthowords which must accompany each record when it is read by the central processor, and performs the first-level check on the information which involves these two words. The off-line output auxiliary control performs a check of every record read exactly as the central processor does, and is capable not only of detecting an error in the record but, in the majority of cases, of reconstructing the garbled information. The corrected information can then be printed or punched without stopping or repositioning the output device as would be necessary without such automatic error correction.

In addition to orthotronic control, and in some ways complementing it, a parity bit is written on tape accompanying each frame. The parity bit is read from tape together with the eight information bits of the frame and remains with these bits as frames are collected to form words. As each word of six frames is transmitted to memory, the accompanying parity bits are monitored to insure an error-free transmission. Each time a word is sent to or from main memory, a transmission check is performed using these six parity bits, and when the word is again written on tape, each bit accompanies its corresponding frame.

When a word is brought to the arithmetic unit, the computer generates a modulo-3 check on each frame pair (16 information bits) for use in checking arithmetic operations. The value of the 2-bit mod-3 check digit is the remainder (a value from 0 to 3, where either 0 or 3 may represent no remainder) which results when the decimal equivalent of the 16 bits is divided by 3. After the mod-3 check is generated, the parity bits are checked and then replaced by the six mod-3 checking bits. When arithmetic operations have been completed and the mod-3 check has been performed to insure that they have been completed correctly, the parity bits are again generated, replacing the mod-3 bits.

The control unit of the central processor checks the interpretation and execution of the program instructions. Selection of instructions and operand locations is checked. The checking process of an add instruction illustrates the thoroughness of the Honeywell 1800 checking system.

1. The selection of the instruction location is verified.
2. The instruction itself is verified for proper parity.
3. During the processing of the A address:

a. a mod-3 check group is attached to the address, then independently recalculated and compared with the original when this information is transferred to the memory selection circuits;

b. the selection operation is verified by comparing with the mod-3 check for the original address the special check digits delivered with the operand;

c. the operand itself is checked for proper parity when read from memory;

d. three mod-3 check digits associated with the operand are generated and stored.

4. During the processing of the B address (when the contents of the B address are added to those of the A address), steps a, b, and c are repeated for the B address. Also, three mod-3 check digits are generated as in 3d, but are added (mod-3) to the check digits previously stored.

5. As the result of the addition is transferred to memory, the C address memory selection is verified as in 3a, b, and c, and a new set of mod-3 check digits is formed from the computed sum and compared for equality with the check digits sum formed in (4) above. If the two sets of digits are equal, then the add instruction has been processed properly.

6. An example of the mod-3 arithmetic follows:

Number in A address	8426	9721	4075
The associated mod-3 check digits	2	1	1
Number in B address	1276	0216	4925
The mod-3 check digits	1	0	2
The sum of A and B is	9702	9937	9000
The mod-3 check digits	0	1	0
The mod-3 sum of the check digits is	0	1	0

In the general case where carries might occur between the operand groups, corrections of +1 and +2 are added to the appropriate check digits. Since two groups are simultaneously affected by a carry correction, any error in the addition, including the carry generation or correction process, is automatically detected.

Each of the special registers retains a mod-3 check on the 16 information bits it contains, which is used to check transmissions and arithmetic operations within the control unit. When the contents of a special register are transferred to the arithmetic unit or to main memory, they are expanded to full-word form and the mod-3 check is replaced by parity bits.

Card reading is checked not only for correct reading by the equipment, but, in the alphanumeric mode, for correctness of conversion and proper keypunching also. Card punching is checked on the 824 punches for double-punched and blank columns. Thirty columns of double-punch, blank-column detection are provided in the standard-speed punch; 80 columns are provided in the high-speed punch. The punch section of the 827 reads each punched card and checks its contents against the punch image. Printing is also checked by comparing echo pulses generated by the printer against the print image.

THE HONEYWELL 1800 WORD

The basic unit of information in the Honeywell 1800 System is a fixed-length word consisting of 54 binary digits, of which six are parity bits used by the automatic checking circuitry and 48 are information bits. Each main memory location is capable of storing one such word, and each arithmetic register is one word in length. A main memory word may represent a machine instruction or one or more pieces of data. In addition to the main memory, the central processor includes the control memory of 256 special registers, used primarily for control purposes and address modification. A special register has the capacity to store a partial word consisting of 16 information bits and two checking bits.

The check bits of the main memory and special register words are not directly available to the programmer, nor are their values subject to program control. Subsequent discussions of the Honeywell 1800 word, therefore, will refer only to the information bits, unless otherwise noted.

Data Words

A computer program generally manipulates data in one or more different forms: decimal, alphanumeric, binary, or a combination of these. The Honeywell 1800 is capable of handling all these types of information. It may interpret the 48 bits of a word in groups of four for the purpose of binary-coded-decimal operation, in groups of six for alphanumeric operation, or as individual units of information for pure binary operation. It may also interpret the 48 bits as a mantissa and an exponent for floating-point operation. Figure III-1 illustrates the structures of these different words.

A decimal word in the Honeywell 1800 contains either 11 decimal digits with a sign, or 12 decimal digits without sign. The decimal arithmetic instructions interpret all operands as a sign and 11 digits. The sign consists of four bits which may represent either the sign of the entire word or individual, 1-bit signs for as many as four different pieces of information within the word. Although a positive sign is normally represented by four binary ones and a negative sign by four binary zeros, a non-standard configuration is perfectly acceptable as input to the arithmetic unit, which interprets any combination of bits except four binary zeros as a positive sign. The sign supplied with the result of an arithmetic operation, however, is always one of the two standard conventions, either four binary ones or four binary zeros. A more detailed discussion of sign conventions can be found in Section VI.

BIT POSITION	1	5	9	13	17	21	25	29	33	37	41	45
DECIMAL	±	1	2	3	4	5	6	7	8	9	0	1
ALPHANUMERIC	R		O		B		I		N		S	O	N
ALPHANUMERIC COMPRESSED	C		.		W		E		B		B	1	7	4
BINARY	±	(44 Binary Digits)										
FLOATING-POINT DECIMAL	±	Exp'nt (7 binary digits)	Mantissa (10 Decimal Digits)									
FLOATING-POINT BINARY	±	Exp'nt (7 binary digits)	Mantissa (40 Binary Digits)									
INSTRUCTION	COMMAND CODE			ADDRESS A			ADDRESS B			ADDRESS C		
SPECIAL REGISTER									±	(15 Binary Digits)		

Figure III-1. Honeywell 1800 Word Structure

A Honeywell 1800 alphanumeric word comprises eight 6-bit groups. Each group can represent any of 26 alphabetic characters, 10 decimal digits, or 20 such special characters as punctuation marks, etc., (see Table I, page 167). Numbers may be stored in alphanumeric (6-bit) form, but the arithmetic unit cannot manipulate them as such; it handles numbers in pure binary or binary-coded decimal form. Between the central processor and the printers, information is transferred in the alphanumeric mode; between the central processor and the card equipment, information is transmitted in either the alphanumeric or the transcription mode.

The 48 binary digits of a word may also represent a pure binary number, which may be stored as a sign and 44 bits, or as 48 unsigned bits. With the exception of the instructions word add and word difference, which treat their operands as 48-bit unsigned numbers, the binary arithmetic instructions interpret operands as signed 44-bit numbers. The sign convention in binary arithmetic is identical to that described for decimal words.

The Honeywell 1800 word can also be handled as a floating-point number, composed of one-bit sign, seven-bit exponent, and 40-bit mantissa. The floating-point decimal word has an exponent that can represent a power of 10 from the -64^{th} to the $+63^{rd}$, and a mantissa that can represent a 10-digit number from .1000000000 through .9999999999 (when normalized). In floating-point binary form (represented in hexadecimal notation), the exponent can represent

a power of 16 from the -64^{th} to the $+63^{rd}$, and the mantissa a 40-bit number from .00010000... 0000 through .1111...1111 (when normalized); the mantissa can represent the equivalent of approximately 12 decimal digits. A one in the sign bit position of the floating-point word indicates that the number is positive, and a zero that it is negative. Floating-point numbers are discussed in Section XIII.

The data words described above are identified in ARGUS language by the following constant codes: DEC, fixed-point decimal number (signed or unsigned); ALF, alphanumeric word; FXBIN, fixed-point binary number; M (mixed constant), compressed alphanumeric word; FLDEC, floating-point decimal number; and FLBIN, floating-point binary number. In addition, ARGUS recognizes an octal word identified by the constant code OCT. This word contains 16 unsigned or 15 signed octal digits. If 15 signed digits are specified, the most significant digit must be less than four, since a sign is represented by four bits, leaving only two bits for the high-order octal digit.

Several differences should be noted between ARGUS notation for data words and the format shown in Figure III-1. When ARGUS notation is used for decimal words, high-order zeros in signed decimal numbers and low-order zeros in unsigned decimal numbers need not be expressed. For example, ARGUS converts the number +125 to the signed 11-digit number +00000000125 and the unsigned number 32 to the 12-digit number 320000000000. A binary word in ARGUS notation is not expressed as a 44- or 48-bit binary number, but as the decimal equivalent of the desired information bits. Therefore, a binary word in ARGUS may contain up to 14 decimal digits and a sign. For complete details on the specification of data words in ARGUS language, reference should be made to the ARGUS Manual of Assembly Language.

Special Register Words

As previously noted, a special register can store 16 information bits, or one-third of a full Honeywell 1800 word. When these bits are manipulated within the special register circuitry, the high-order bit is interpreted as a sign (1 = plus, 0 = minus). Depending upon the type of addressing used, the remaining 15 bits of a special register word may be interpreted as a main memory address, consisting of a bank indicator and subaddress, or as a special register address, consisting of a group indicator and subaddress (see Figures IV-1 and IV-2). When a special register word is modified arithmetically within the special register circuitry, the value of the sign bit determines whether it is incremented or decremented. A special register word is identified in ARGUS language by the constant code SPEC.

Instruction Words

The 48 bits of a Honeywell 1800 instruction word are interpreted as four groups of 12 bits

each. Bits 1-12 represent the command code; bits 13-24, 25-36, and 37-48 are designated as the A address group, B address group, and C address group, respectively. The address portions of instructions normally are used to designate the locations of operands and results, but in certain instructions they may contain special information such as the number of words to be moved, the number of bits to be shifted, a change of sequence counter, and so forth. A detailed discussion of addressing in the Honeywell 1800 will be found in Section IV.

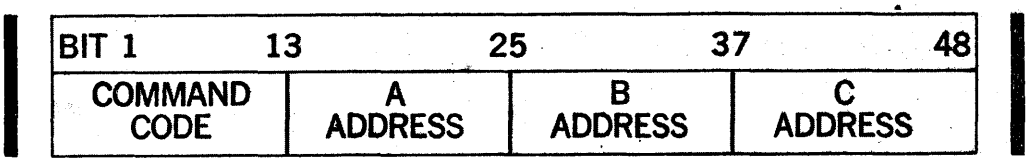
Machine instructions fall into five major categories: general instructions, unmasked and masked; inherent mask instructions; peripheral and print instructions; simulator instructions; and scientific instructions. The masked general instructions and the peripheral and print instructions are uniquely designated by six-bits — bits 7 through 12 of the instruction word. The unmasked general instructions, the inherent mask instructions, and the scientific instructions are uniquely designated by eight bits — bits 7 through 12, plus bits 2 and 3. The simulator instructions are uniquely defined by only three bits — bits 10 through 12. These groups of bits which uniquely specify the operation to be performed are called the operation code. The bits of the command code which are not used for the operation code serve various other purposes which will be described as the instruction types are discussed. A graphic summary of the format of the major command code types appears in Figure III-2. The command codes for the individual instructions, together with their mnemonic operation codes in ARGUS language, are set forth by major instruction type in Table II, page 168.

INSTRUCTIONS

In the 1800-II Honeywell three address (Fig. 1) instructions are effectively five-address functions, combining masking and bi-sequence mode.

Masking enables a fixed word machine to address a field of data as small as 1 bit without extra instructions.

The bi-sequence bit gives programmers the ability to switch to subroutines which run in the alternate sequence counter and return to the next sequenced **step** without extra instructions. Programs are simpler to code, easier to check-out, and they run faster.



In addition to the three operands, the command code can include a mask address and bit designating a transfer in control between the two sequence counters. Addressing can be Direct to any one of 2048 locations or complex, either indexed or indirect.

Indexed addressing allows relative addressing combined with indexing which is intregal with the instruction. Indirect addressing simply enables the programmer to address a series of locations in memory with a single instruction in a loop, or accumulate or transfer a series with a single instruction. Instructions can combine various types of addressing within one instruction as follows:

Type of Instruction				Masking	Addressing Types Combined			
Shift	Field	Gen'/unmasked	Peripheral	Permitted	Direct	Indexed	Indir.	SpecReg
X				X	X	X	X	
X				X	X	X	X	X
	X			X	X	X		
	X				X	X	X	X
		X			X	X	X	X
			X		X	X		

The instructions may be grouped as follows:

1) Arithmetic
2) Transfer
3) Decision
4) Shift
5) Logical
6) Peripheral
7) Conversion

Arithmetic Instructions

Honeywell 1800-II with 1801B Scientific Option operates on information in two modes, fixed point and floating point, the latter is further split into normalized and unnormalized operations, single and extended precision.

Summary of Control Memory

Control Memory consists of 256 16 bit locations and is used to store main memory addresses. Sequence counters specify the next instruction and/or subroutine to be executed, Index registers store I/O buffer or work area addresses, General Purpose Registers are used for storage of tables or temporary storage areas, and specialized counters store addresses such as a) the last instruction to cause a sequence change, b) the address of an Orthotronic checking routine, c) a group of masks, or d) the addresses used by a peripheral instruction to store I/O records.

8 groups of 32 of these completely addressable control memory registers are contained in the H-1800-II,with one group usually used by each program when programs are run in parallel. Each group of 32 control memory registers include the following:

Symbolic Name	Function
AU1 AU2	Arithmetic Counters, used to temporarily store a main memory address 00000 to 32768
SC CSC	Sequence & bisequence counters storing main memory address of next instruction in sequence. These counters, when active, automatically increment by one at execute time of each instruction
SH CSH	History registers automatically stores address of instruction which is next in sequence if 1) sequence is changed by instruction or 2) if control of program changes from SC to CSC or vice versa.

Symbolic Name	Function
UTR	Register which stores address of location in main memory to which control is transferred in case of certain errors, for programmer specified corrective action. (See---UTR, for details)
MXR	Register which stores bank # and two partial addresses of main memory locations where groups of 48 bit masks are stored. (See---MASKING)
X0,X1,X2,X3,X4,X5,X6,X7 Index Registers	used for indexed addressing and/or indirect memory addressing. Increments per programmers specification in each instruction used when used for relative addressing. When used for indexed addressing, permits relative addressing with temporary augments up to 255.
R0,R1,R2,R3,R4,R5,R6,R7 S0,S1,S2,S3	used for indirect main memory addressing. Self-incrementing per programmer's specification.
RAC,DRAC	Stores location in memory where word from read buffer is stored after each transfer. Associated with one of 8 I/O trunks.
WAC,DWAC	Stores location in memory from which word is transferred to write buffer during write operation. Associated with specific I/O trunks same as RAC,DRAC.

Details and Special Features of Control Memory

Indexing: Each program accesses 8 index registers which permit relative addressing using a base address anywhere in memory ± 255.

Ex: If in a Group 1 we have instruction:

	A	B	C	
DA	X1,1	X1,3	X1,4	defined as

Decimal adding (symbolically: A +B store in C), and if address of LOC ØØ22 in bank 5 is stored in Index register X1; then the main memory contents of LOC ØØ23 Bank 5 will be added to LOC ØØ25 and the results stored in location ØØ26 Bank 5. The digit following the comma in each operand is termed the augment. The augment is temporarily added to the base address to give the address of a location in main memory. Maximum augment is 255. The contents of the index register may be modified by a single instruction or, if desired, as part of a comparison instruction.

Masking (see below) can be incorporated in any single instruction using indexed addressing.

Indirect Addressing: Each program has up to 16 general purpose registers which can be used to store addresses of locations in memory. The registers may be addressed in such a way that the contents of the location specified may be utilized, and then the address modified by a specific increment dictated by the programmer.

Example: If it is desired to compare the contents of one location with those of a series located in alternate locations, a single instruction may be incorporated in a loop thusly:

CC	A	B	C
NN	N,R1,2	KON	TEST

Where NN specifies that if A ≠ B go to C, i.e. in this instance, if the contents of the location whose address is stored in register R1 is not equal to the contents of location KON, then change sequence by going to the instruction at TEST, otherwise proceed to the next instruction in sequence. The self-incrementing feature adds the binary value of the decimal digit (following the second comma in the A Operand in the example) to the contents of the register R1 after the use of the contents of the original address.

An example of one of the two instructions tailored for indirect addressing are the Multiple Transfer:

CC	A	B	C
MT	ZERO	50	N,R2,1

which transfers locations specified in the "A" operand to locations

whose addresses are stored in R2. Each transfer, for a total of 50, (as specified by the B operand) will cause the register to increment by 1 (as specified in C), thus transferring contents of location ZERO to 50 successive locations beginning with the initial setting of R1.

Masking Through use of the MXR

The ability of a fixed word machine to operate on less than one word thus permitting the packing of various fields of data within a word, is enhanced by the masking function. The 1800-II masks words within the instruction itself, thus eliminating extra instructions ordinarily required for such operations.

The definition of masking: If a word contains six alphabetics and two numeric characters, the latter of which are to be added to similar bit positions in another location , a mask is specified for the 2 low order digits by placing 1 bits in the bit positions occupied by these two digits. The mask is stored in a 48 bit memory location and is called into use by an instruction which stores the mask in the central processor in the mask register. Then any words transferred into or out of the accumulator by that instruction will be filtered through the mask register, only passing information in bit positions were 1's exist.

The instruction

CC	A	B	C
DA,MASK1	IN	CON	CON+1

adds the contents of IN to those of CON and stores the results in CON+1. The contents of location MASK1 effect each operand similarly, passing only information in bit positions where mask bits are stored. They key to the multiple functioning of the 3 address instruction is the use of a 5 bit mask subaddress which is combined with a partial address in the MASK Index Register to give the complete 15 bit address of the mask in memory. An average of 64 different masks may be used before a new partial address need be stored.

The UTR - unprogrammed transfer register

Use of this register permits the programmer to specify routines for certain situations which can occur in normal programs at unspecified times in the progress of the program such as beginning or end of tape, addition or subtraction overflow, parity error, read or write error, division over-capacity, Exponential underflow, and exponential overflow. The programmer merely loads the address of an even location in memory into this register, then if he wants specific sub-routines executed in instances listed above, an instruction transferring control to each desired subroutine is stored in the subsequent 15 locations. For example, addition or sub-traction overflow occurring in an instruction stored in the sequence counter always causes the machine to augment the

address stored in the UTR by 8 and execute the instruction stored at the resulting address. This instruction can take the program to a routine which is desired in cases where this condition exists, including the identification of the instruction causing the unprogrammed transfer.

SUMMARY OF INSTRUCTIONS

General Instructions - Unmasked or Masked

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
BA	Binary Add algebraically (A) to (B). Store sum in C. If overflow occurs, take next instruction from U + 8 if the sequence counter selected this instruction, or from U + 9 if the cosequence counter selected this instruction. The sign of either operand is positive if any of its four sign bits is one. The sign of the sum is 0000 if negative, 1111 if positive.	4 (See note 4)
DA	Decimal Add algebraically (A) to (B). Store sum in C. Otherwise same as Binary Add.	4 (See note 4)
WA	Word Add. Binary add (A) to (B), considered as unsigned 48-bit numbers. Store 48-bit result in C. If overflow occurs, observe same conventions as in Binary Add.	4
BS	Binary Subtract algebraically (B) from (A). Store result in C. Observe same overflow and sign conventions as in Binary Add.	4 (See note 4)
DS	Decimal Subtract algebraically (B) from (A). Store result in C. Observe same overflow and sign conventions as in Binary Add.	4 (See note 4)
WD	Word Difference. Binary subtract (B) from (A). Otherwise identical to Word Add.	4
NA	Inequality Comparison, Alphabetic. Compare (A) and (B) including sign positions. If (A) $\neq$ (B), change sequencing counter to select next instruction from location specified by C. Plus zero is not equal to minus zero.	4
NN	Inequality Comparison, Numeric. Compare algebraically (A) and (B). If (A) $\neq$ (B), follow procedure for NA. Plus zero equals minus zero.	4
LA	Less Than Or Equal Comparison, Alphabetic. Compare (A) and (B) including sign positions. If (A) $\leq$ (B), change sequencing counter to select next instruction from location specified by C. Plus zero is greater than minus zero.	4
LN	Less Than Or Equal Comparison, Numeric. Compare algebraically (A) and (B). If (A) $\leq$ (B), follow procedure for LA. Plus zero equals minus zero.	4

General Instructions - Unmasked or Masked (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
TX	Transfer (A) to C. Ignore B.	3
TS	Transfer (A) to B. If C is active, change sequencing counter to select next instruction from location specified by C.	4
HA	Half Add. Binary add without carry (A) to (B), considered as unsigned 48-bit numbers. Store 48-bit result in C. Result is zero wherever corresponding bits of (A) and (B) are identical, one wherever corresponding bits of (A) and (B) are different. This is "logical exclusive OR."	4
SM	Superimpose (A) and (B). Store result in C. Result is zero wherever bits of (A) and (B) are both zero, one everywhere else. This is "logical inclusive OR."	4
CP	Check Parity. Test (A) for correct parity. Place (A) with correct check bits in B. If (B) differs from (A), change sequencing counter to select next instruction from location specified by C.	4

General Instructions - Unmasked

BM	Binary Multiply (A) by (B). Store high-order product with proper sign in C and accumulator, low-order product with proper sign in low-order product register. Product signs are 0000 if negative or 1111 if positive.	33
DM	Decimal Multiply (A) by (B). Store high-order and low-order products as in Binary Multiply with same sign conventions.	27
BT	Binary Accumulate. Place low-order 44 bits of (A) in accumulator. Perform this transfer B' times, adding in binary (with positive sign implied) the successive 44-bit words transferred. B' (high-order 6 bits of B) = 0 to 63. Store 44-bit result, with sign of first word transferred, in C. Observe same overflow conventions as in Binary Add. Note that if A is an indirect address with non-zero increment, B different numbers are accumulated.	3 + n (See note 2)
DT	Decimal Accumulate. Same as Binary Accumulate except that transferred words are added as 11-digit decimal numbers.	3 + n (See note 2)
MT	Multiple Transfer. Transfer (A) to C. Perform this instruction B' times. B' (high-order 6 bits of B) = 0 to 63. Note that if A and C are indirect addresses with non-zero increments, B' different transfers are performed.	1 + 2n (See note 2)
TN	N-Word Transfer. Transfer B' words from consecutive locations starting at A to consecutive locations starting at C. B' = 0 to 63.	5 + 2n (See note 2)

General Instructions - Unmasked (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
CC	Compute Orthocount. Write a generated end-of-record word in the location specified by C. Orthocount the record starting at A to the end-of-record word. Store orthoword 1 in C and orthoword 2 in C + 1. Place end-of-record word in C + 2. If B is inactive, control is not changed for distributed item handling. If B is active, end-of-item words are sensed and control is changed for distributed item handling.	11 + n (See note 2)
IT	Item Transfer. Substitute an end-of-item symbol for the high-order 32 bits of (B), clearing the low-order 16 bits of (B) to zeros. Transfer words from consecutive memory locations starting with A to consecutive memory locations starting with C until an end-of-item or end-of-record word is transferred.	7 + 2n (See note 2)
RT	Record Transfer. Store an end-of-record word in B. Transfer words from consecutive memory locations starting with A to consecutive memory locations starting with C until an end-of-record word is transferred.	7 + 2n (See note 2)
MPC	Control Program. Ignore A. Place (PCR) in the location specified by C. Then alter the bits of PCR specified by bits 5-12 of B, using bits 1-4 of B to define how the bits are altered. If B address memory designator bit is 1, hunt for next program in demand. Otherwise, do not hunt.	4
PR	Proceed. If C is inactive, do not hunt for next program in demand.	2

Inherent Mask Instructions

SWS	Shift Word and Substitute. Shift right end around including sign (A) as directed by B'. Mask result and store in C (protected). B' (high-order 6 bits of B) specifies the number of 1-bit shifts.	5 + k (See note 3)
SPS	Shift Preserving Sign and Substitute. Shift right end around excluding sign (A) as directed by B'. Otherwise same as SWS.	5 + k (See note 3)
SWE	Shift Word and Extract. Same as SWS except that the unmasked portions of (C) are unprotected.	5 + k (See note 3)
SPE	Shift Preserving Sign and Extract. Same as SPS except that the unmasked portions of (C) are unprotected.	5 + k (See note 3)

Inherent Mask Instructions (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
SSL	Shift and Select. Shift right end around including sign (A) as directed by B'. Binary add to C under mask control no more than 11 low-order bits of the shifted word, with positive sign implied. Change the sequencing counter to select the next instruction from the location specified by the modified C address. B' is interpreted as in the SWS instruction.	6 + k (See note 3)
SS	Substitute. Using (B) as a mask, transfer (A) to C and protect unmasked portions of (C).	5
EX	Extract or Logical AND. Using (B) as a mask, transfer (A) to C without protecting unmasked portions of (C). Result is one wherever bits of (A) and (B) are both one, zero everywhere else.	5

Peripheral and Print Instructions

RF	Read Forward from peripheral device XX into consecutive memory locations starting with A. XX represents command code bits 1-6. Set the RAC to +A. If B is inactive, do not change control for distributed item handling. If B is active, set the DRAC to +B and sense for end-of-item words. Change sequencing counter to select next instruction from location specified by C. If end of tape is sensed, take next instruction from U + 4 if the sequence counter selected this instruction or from U + 5 if the cosequence counter selected this instruction. If a parity error was detected during the last previous read from this device, reset the parity error flip-flop, do not perform the read. Instead, take next instruction from U + 6 or U + 7. This instruction is interlocked against device XX and the associated buffer.	7
RB	Read Backward from magnetic tape unit XX into consecutive memory locations starting with A. This instruction is otherwise identical with RF except that the RAC is set to -A and if B is active the DRAC is set to -B.	7
WF	Write Forward on peripheral device XX the contents of consecutive memory locations from A through the end-of-record word. Set the WAC to +A. If B is inactive, do not change control for distributed item handling. If B is active, set the DWAC to +B and sense for end-of-item words. Change sequencing counter to select next instruction from location specified by C. If an error was detected during the last previous write to peripheral device XX, reset the parity error flip-flop, do not perform the write. Instead, take next instruction from U + 6 or U + 7. This instruction is	7

Peripheral and Print Instructions (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
	interlocked against device XX and the associated buffer. End-of-tape convention is identical with Read Forward.	
RW	Rewind tape on magnetic tape unit or paper tape reader XX. If A is active in magnetic tape rewind, set manually removable interlock after tape is rewound. B and C are ignored. If an error was detected during the last previous read or write for this magnetic tape unit, reset the parity error flip-flop and perform the rewind. In the case of a paper tape reader, do not perform the rewind but take next instruction from U + 6 or U + 7.	3
PRA, PRD, PRO	Print (A) on the typewriter and in the format specified by B. If C is active, change the sequencing counter to select the next instruction from the location specified by C. In an alphabetic print, eight 6-bit characters are printed. In a decimal print, 12 hexadecimal digits are printed. In an octal print, 16 octal numbers are printed.	5

Simulator Instructions

S	Simulator. Form a memory location address (direct or indexed) from the low-order 11 bits of the command code and store this instruction in the location thus specified. Change the cosequence counter to select the next instruction from the next higher address.	7

Scientific Instructions

FBA	Floating Binary Add. Binary add algebraically (A) to (B). Deliver result as a normalized floating-point number to C if C is active; retain result in FLAC if C is inactive. If exponential underflow occurs, take next instruction from U + 12 or U + 13. If exponential overflow occurs, take next instruction from U + 14 or U + 15.	4.95
FDA	Floating Decimal Add. Same as floating binary add, except that arithmetic is decimal rather than binary.	6.5
FBS	Floating Binary Subtract. Change the sign of the B operand and perform a floating binary add.	4.95
FDS	Floating Decimal Subtract. Same as floating binary subtract, except that arithmetic is decimal rather than binary.	6.5
FBAU	Floating Binary Add, Unnormalized. Same as floating binary add, except that result is not normalized. A 4-bit right shift is provided if necessary to compensate for mantissa overflow, but no compensating left shift occurs to renormalize a result with zero in the most significant mantissa digit.	5

Scientific Instructions (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
FDD	Floating Decimal Divide. Same as floating binary divide, except that arithmetic is decimal rather than binary.	20
BD	Fixed Binary Divide. Divide (B) by (A), handling both operands as fixed-point binary numbers. If C is inactive, retain the quotient and remainder in FLAC and FLOP. If C is active, store the quotient in C in fixed-point form; the contents of FLOP are unspecified. If the absolute value of (B) equals or exceeds the absolute value of (A), do not perform the instruction but take the next instruction from U + 10 or U + 11.	15
DD	Fixed Decimal Divide. Same as fixed binary divide, except that arithmetic is decimal rather than binary.	22
FLN	Normalized Less Than Comparison. Compare (A) with (B) algebraically. If (A) $\leq$ (B), change the specified sequencing counter to C.	4
FNN	Normalized Inequality Comparison. Compare (A) with (B) including sign positions. If (A) $\neq$ (B), change the specified counter to C.	4
FFN	Fixed-to-Floating Normalize. Form the normalized result mantissa from low-order 44 bits of (B). Result exponent is the exponent of (A) minus one for each 4-bit left shift plus one for each 4-bit right shift minus one. Result is positive if high-order 4 bits of (B) include any ones. If C is active, store result in C. If C is inactive, retain normalized double-precision result in FLAC and FLOP, where low-order 36 bits in FLOP are zeros. Sense for exponential underflow in high-order result. Set low-order underflow indicator if exponential underflow occurs in low-order result.	4
ULD	Multiple Unload. Do not reset the low-order underflow or overflow indicator. Store (FLAC) in A and (FLOP) in C. B must be inactive. If the low-order underflow indicator is set when the instruction is initiated, take the next instruction from U + 12 or U + 13. If the overflow indicator is set, take the next instruction from U + 14 or U + 15.	4
FCON	Conversion. Convert (B) as specified by (A). If C is active, store result in C. If C is inactive, retain result in FLAC and remainder in FLOP. The modulo-3 sum of (A) must be one.	
	1. Fixed Decimal to Floating Binary Conversion. If (A) is XXXXXXXX00000001 (octal), convert	20

Scientific Instructions (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
FDAU	Floating Decimal Add, Unnormalized. Same as floating binary add, unnormalized, except that arithmetic is decimal rather than binary.	6
FBSU	Floating Binary Subtract, Unnormalized. Change the sign of the B operand and perform a floating binary add, unnormalized.	5
FDSU	Floating Decimal Subtract, Unnormalized. Same as floating binary subtract, unnormalized, except that arithmetic is decimal rather than binary.	6
FBAE	Floating Binary Add, Extended Precision. Form the normalized double-precision sum of (A) and (B). If C is inactive, retain the high-order and low-order parts in FLAC and FLOP. If C is active, store the high-order part in C; the contents of FLOP are unspecified. Sense for exponential overflow or underflow on the high-order result. If exponential underflow occurs on the low-order result, set the low-order underflow indicator.	6
FBSE	Floating Binary Subtract, Extended Precision. Change the sign of the B operand and perform a floating binary add, extended precision.	6
FBM	Floating Binary Multiply. Multiply (A) by (B) to form a normalized, double-precision, floating-point product. If C is inactive, retain the high-order and low-order products in FLAC and FLOP. If C is active, store the high-order product in C; the contents of FLOP are unspecified. Sense for exponential overflow or underflow on the high-order product. If exponential underflow occurs on the low-order product, set the low-order underflow indicator.	5
FDM	Floating Decimal Multiply. Same as floating binary multiply, except that arithmetic is decimal rather than binary.	10
FBD	Floating Binary Divide. Divide (B) by (A). If C is inactive, retain the quotient and remainder in FLAC and FLOP. If C is active, store the quotient in C in floating-point form; the contents of FLOP are unspecified. The quotient will be normalized if the dividend is normalized. The remainder is not normalized. Sense for exponential overflow or underflow in the quotient. Set the low-order underflow or overflow indicator if there is underflow or overflow in the remainder. If the divisor is unnormalized or zero, do not perform the instruction but take the next instruction from U + 10 or U + 11.	14

Scientific Instructions (cont)

MNEMONIC OPERATION CODE	DESCRIPTION[5]	TIME IN MEMORY CYCLES[1]
	the low-order 44 bits of (B) to the unnormalized result mantissa. The result exponent is zero. The result is positive if the high-order 4 bits of (B) include any ones.	
	2. Floating Binary to Fixed Decimal Conversion. If (A) is XXXXXXXX00000004 (octal), convert the mantissa of (B) to a 44-bit fixed decimal result. Ignore the exponent of (B). The result takes the sign of (B).	9

NOTES:

1. One memory cycle equals two microseconds. All addresses considered active, except B addresses of CC and ULD instructions. For variations in time, see Appendix C.
2. n = number of words accumulated, transferred, or ortho-counted.
3. Values of k are based on number of 16, 4, and 1-bit shifts required. See Appendix C for table of values.
4. Under certain conditions, as explained in Appendix A, an additional one or two memory cycles may be required.
5. Where "(A)" is read "the contents of the location designated by the A address group."

CHARACTERISTICS	ADVANTAGES
I. MEMORY H-1800-II 8192-65,536 words 2 u.S Access/Word Parallel 48 Bit Transfer	MODULARITY AND COMPATIBLE GROWTH PARALLEL DATA TRANSFER

THE HONEYWELL 800 WORD

Bit Position	1	5	9	13	17	21	25	29	33	37	41	45
DECIMAL	±	1	2	3	4	5	6	7	8	9	0	1
ALPHANUMERIC	R		O	B	I		N	S		O	N	
ALPHANUMERIC COMPRESSED	C		.	W	E		B	B		1	7	4
BINARY/OCTAL	±	(44 Binary Digits)										
FLOATING-POINT DECIMAL	±	Exp'nt (7 binary digits)		Mantissa (10 Decimal Digits)								
FLOATING-POINT BINARY	±	Exp'nt (7 binary digits)		Mantissa (40 Binary Digits)								
INSTRUCTION	OPERATION CODE			ADDRESS A			ADDRESS B			ADDRESS C		
SPECIAL REGISTER								±	(15 Binary Digits)			

MAXIMUM DECIMAL COMPRESSION OF DATA

ALPHABETIC FORM

ALPHA-NUMERIC MIXED FORM

NO RESTRICTION ON CODING AS THE BINARY FORM IS ALLOWED

DECIMAL FLOATING POINT

BINARY FLOATING POINT EXPANDED PRECISION

EFFICIENT INSTRUCTION FORMAT

CHARACTERISTICS	ADVANTAGES
II. SPECIAL REGISTERS	
256 Special Registers in Addition to main memory that are separated into 8 groups of 32 Special Registers each; including:	
64 INDEX REGISTERS	INDEXED OPERATIONS
96 GENERAL PURPOSE REGISTERS	INDIRECT ADDRESSING
32 READ/WRITE ADDRESS COUNTERS	CONTINUOUS OR DISTRIBUTED READ/WRITE
16 ARITHMETIC CONTROL COUNTERS	MONITOR MULTIPLE CYCLE OPERATIONS
8 SEQUENCE COUNTERS	INDEPENDENT PROGRAM OPERATION
8 CO-SEQUENCE COUNTERS	AUTOMATIC SUBROUTINE EXITS
8 SEQUENCE HISTORY REGISTERS	AUTOMATIC SUBROUTINE RETURN
8 CO-SEQUENCE HISTORY REGISTERS	AUTOMATIC SUBROUTINE RETURN
8 UNPROGRAMMED TRANSFER REGISTERS	HARDWARE ISOLATION OF ERROR CONDITION
8 MASK INDEX REGISTERS	EXPANDS EFFECTIVE MASK ADDRESSES
III. TRAFFIC CONTROL AUTOMATIC and CONTINUOUS scanning of 16 input/output channels, interrupting processing for only one memory cycle to service a channel if it is in a demand condition.	READ/WRITE/COMPUTE IN A SIMULTANEOUS BUFFERED MODE OVERLAPS ALL INPUT AND OUTPUT OPERATIONS WITH INTERNAL PROCESSING

CHARACTERISTICS	ADVANTAGES

IV. MULTIPROGRAM CONTROL

Sequentially executes an instruction for each active program operating in parallel Automatically, such that each program has an equal opportunity to time share main memory, monitored by the Program Control Register (PCR).	PROGRAMS WRITTEN INDEPENDENTLY ARE RUN IN PARALLEL THROUGH HARDWARE CONTROL. NO SPECIAL PROGRAMMING IS REQUIRED!

V. COMMAND CODES

PARALLEL ARITHMETIC

H1800 - 120,000 Operations/second	EQUAL TO 300,000 SINGLE ADDRESS OPERATIONS/SEC
THREE ADDRESS INSTRUCTIONS	POWERFUL/COMPACT/EFFICIENT

OPERATION	A OPERAND	B OPERAND	C OPERAND

MASKING	CHARACTER, DIGIT, BIT, OR FIELD ADDRESSING
INDEXING	FLEXIBLE PROGRAMMING
INDIRECT ADDRESSING	PROGRAM INTEGRITY

GENERAL
UNMASKED OR MASKED

BA	Binary Add	BINARY ARITHMETIC
DA	Decimal Add	DECIMAL ARITHMETIC
WA	Word Add	ABSOLUTE ARITHMETIC
BS	Binary Subtract	
DS	Decimal Subtract	
WD	Word Difference	

NA	Inequality Comparison, Alphabetic	ALPHA, BINARY, AND NUMERIC LOGIC
NN	Inequality Comparison, Numeric	
LA	Less than or Equal Comparison, Alphabetic	COMBINED TRANSFER AND SEQUENCE CHANGE.
LN	Less Than or Equal Comparison, Numeric	
TX	Transfer (A) to C	
TS	Transfer (A) to B and Go to C	
HA	Half Add (Mod. 2)	
SM	Superimpose	
CP	Check Parity	

GENERAL
UNMASKED

BM	Binary Multiply	BINARY MULTIPLY
DM	Decimal Multiply	DECIMAL MULTIPLY
BT	Binary Accumulate	BINARY ACCUMULATE
DT	Decimal Accumulate	DECIMAL ACCUMULATE
MT	Multiple Transfer	MULTIPLE WORD TRANSFER
TN	N-Word Transfer	ITEM TRANSFER
CC	Compute Orthocount	RECORD TRANSFER
IT	Item Transfer	
RT	Record Transfer	
MPC	Control Program	
PR	Proceed	

SHIFT/LOGIC/MASK

SWS	Shift Word and Substitute (Protected)	LEFT AND RIGHT SHIFTING
SPS	Shift Preserving Sign and Substitute (Protected)	SELECTION OF SIGN PROTECTION
SWE	Shift Word and Extract (Unprotected)	DECIMAL, ALPHA & BINARY MODE SELECTION
SPE	Shift Preserving Sign and Extract (Unprotected)	BIT, DECIMAL, ALPHA, OR FIELD ADDRESSING
SSL	Shift and Select (Protected)	COMBINED WITH DIRECT INSERTION CAPABILITY
SS	Substitute (Protected)	
EX	Extract (Unprotected)	

PERIPHERAL AND PRINT

RF	Read Forward	FORWARD AND BACKWARD READING OF MAGNETIC TAPES
RB	Read Backward	
WF	Write Forward	
RW	Rewind	PRINTING OF ALPHA, DECIMAL, OR OCTAL
PRA	Print Alpha	
PRD	Print Decimal	
PRO	Print Octal	

SIMULATOR

SIMULATOR	SIMULATES A SUBROUTINE AND PROVIDES AUTOMATIC LINKAGE
Direct	
Indexed	

SCIENTIFIC

FBA	Floating Binary Add	FULL COMPLIMENT OF ALL SCIENTIFIC HARDWARE COMMANDS FOR BOTH <u>DECIMAL</u> AND <u>BINARY</u> OPERATIONS
FDA	Floating Decimal Add	
FBS	Floating Binary Subtract	
FDS	Floating Decimal Subtract	
FBD	Floating Binary Divide	
FDD	Floating Decimal Divide	
FBAU	Floating Binary Add, Unnormalized	
FDAU	Floating Decimal Add, Unnormalized	
FBSU	Floating Binary Subtract, Unnormalized	
FDSU	Floating Decimal Subtract, Unnormalized	
FBM	Floating Binary Multiply	
FDM	Floating Decimal Multiply	
ULD	Multiple Unload	
FBAE	Floating Binary Add, Extended Precision	
FBSE	Floating Binary Subtract, Extended Precision	
BD	Fixed Binary Divide	
DD	Fixed Decimal Divide	
FFN	Fixed-to-Floating Normalize	
FLN	Normalized Less Than Comparison	
FNN	Normalized Inequality Comparison	
FCON	Conversion (Floating Binary to Fixed Decimal, Fixed Decimal to Floating Binary)	

VI. ERROR DETECTION - Central Processor

Internal Transmission	EACH 8 BITS HAS ITS OWN PARITY
Each 48 Bit Word Has <u>6</u> Parity Check Bits That are Checked on Transmission	
2 Bit MOD-3 Check on All Operations	ADDRESS, OPERATION, CONTENTS AND ARITHMETIC
All Errors are <u>Automatically</u> Detected	ARE ALL <u>AUTOMATICALLY</u> CHECKED
and an Unprogrammed Transfer Isolates	ERRORS ARE AUTOMATICALLY <u>ISOLATED WITHOUT</u>
it for the Programmer, including:	<u>PROGRAMMING</u>

Parity
Read or Write Error
Beginning or End of Tape
Add or Subtract Overflow
Division Over Capacity
Exponential Underflow and Overflow
Control Errors

An Advanced Magnetic Tape System for Data Processing

DR. RICHARD B. LAWRANCE

Introduction and Tape Mechanism

IT IS a truism that for any but the smallest digital data processing systems major attention must be be given to the provision of an adequate magnetic tape transport, reading and writing system, and means for insuring the correctness of information all the way from the central processor to the magnetic tape and back again. This paper will describe some of the above mentioned features of the Honeywell 800[1].

Early in the layout and specification of a new system it is necessary to decide on the specifications for and approach to the magnetic tape mechanism and recording system. As regards the tape mechanism itself, our earlier experience (particularly with the DATAmatic 1000) had favorably inclined us toward the vacuum capstan approach. Our several years of experience with electrostatic clutching had led us ultimately to abandon the electrostatic approach for the DATAmatic 1000, and after re-evaluation, it was again excluded from consideration for the new system. As to the other two widely-used methods of achieving fast stop-start tape motion, we felt that the faster and more positive of these — namely the pinch-roller approach — should be the most seriously considered as an alternative to the use of vacuum capstans.

In this comparative evaluation and in the design effort which followed, we placed an overriding importance on providing the magnetic tape itself with a benign environment. This is in accord with our belief that in every stage of manual handling or manipulation by the mechanism all stresses in the tape (both during normal operation and under failure conditions) should be made zero by design or kept to demonstrably safe values.

The following tabulation compares inherent features of presently used pinch-roller mechanisms with the corresponding features of pneumatic mechanisms.

Pinch Roller

(1) In some designs (but not those in which the idler is continuously driven) the tape to be accelerated must bear not only the forces to accelerate its own mass but also the forces to give angular acceleration to the idler.

(2) The tape accelerating forces are applied in a concentrated area surrounding the line of tangency of two typically rather small cylinders (slightly spread by resiliency in the tape itself and at most one of the cylinders).

(3) In order to prevent non-simultaneous clutching across the width of the tape (with attendant tracking and skew problems) a very accurate pivoting or translatory motion is required; fast operation demands that this be designed for minimum inertia. Thickness variation across the tape is a possible source of skew.

(4) Compressive action of pinch-roller tends to emboss wear particles or other dirt into the oxide surface of the tape. (Not applicable to metal tape.)

(5) Powerful fast-pickup driving and braking mechanisms may be slow to release, placing safety restriction on minimum interval between drive and brake commands.

(6) For high performance, auxiliary air lubrication may be required.[2]

(7) Usual embodiment employs rollers of flanged-spool construction, with tape unsupported and not otherwise edge-guided over important portions of its path.

Vacuum Capstan

(1) Only the mass of the tape itself requires acceleration, thus minimizing the force transmitted to and by the tape.

(2) The tape accelerating forces are distributed over a typically fifteen-fold larger area, whose length may equal or exceed one-fourth of the capstan circumference. The capstan diameter may be conveniently large.

(3) Symmetrical engagement of the tape to capstan or brake is automatically achieved by symmetrical design of pneumatic passages. Engagement always commences along tape center line, minimizing skew. Transverse variation in tape thickness does not add to skew.

[1] Together with the paper "Control & Arithmetic Techniques in a Multi-Programmed Computer." By: N. Lourie, H. Schrimpf, R. Reach, W. Kahn, presented at this conference, the present paper forms a partial technical description of this new data processing system.

[2] R. A. Skov, "Pulse Time Displacement in High-Density Magnetic Tape." *IBM Journal of Research and Development*, April 1958.

(4) Free from dirt embossing. No material body need touch the oxide surface of the tape (although usually the magnetic head is made to do so).

(5) No restriction on interval between successive commands. Moving parts of mechanism are offset from tape path, completely covered, and cannot touch tape. No danger to tape from tug-of-war.

(6) Air lubrication of tape is a built-in feature.

(7) Essentially complete edge guiding over entire path from supply reel to takeup reel is easily incorporated.

The considerations tabulated above led us to design for the Honeywell 800 a tape mechanism utilizing the vacuum capstan principle and embodying many of the techniques and principles used in the earlier DATAmatic 1000 three-inch tape mechanism.[3]

Brief Description of Tape Mechanism

Fig. 1 shows a photograph of the Type 804 magnetic tape mechanism. The unit stands approximately 5 feet 9 inches high and occupies a floor area slightly over 2 feet square. The tape is a nominal ¾-inch wide and is moved at a normal speed of 120 inches per second in either direction as desired. High speed rewind is provided, in one direction only, at 360 inches per second.

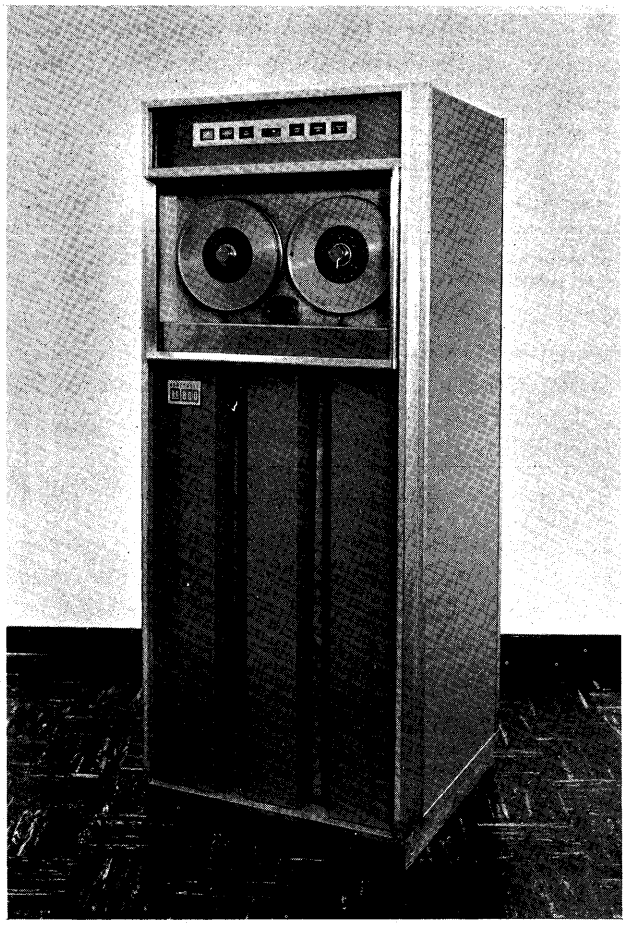


Fig. 1—Type 804 magnetic tape mechanisms.

The cabinet shown includes the separate write-amplifier final stages for the ten recording channels as well as the final stage for the AC-excited separate erase gap; the three-stage transistor preamplifiers for each of the ten playback channels; and the solid-state switching equipment for placing the read-write head and circuits in the selected mode. Also included are the power supplies, loop position sensors and servo control for the DC-operated reel motors; the power supplies for the read-write circuits; vacuum and pressure sources for the capstan, brake, and suction loop chambers; beginning-of-tape and end-of-tape sensing means; storage; and other electronic packages facilitating testing and maintenance.

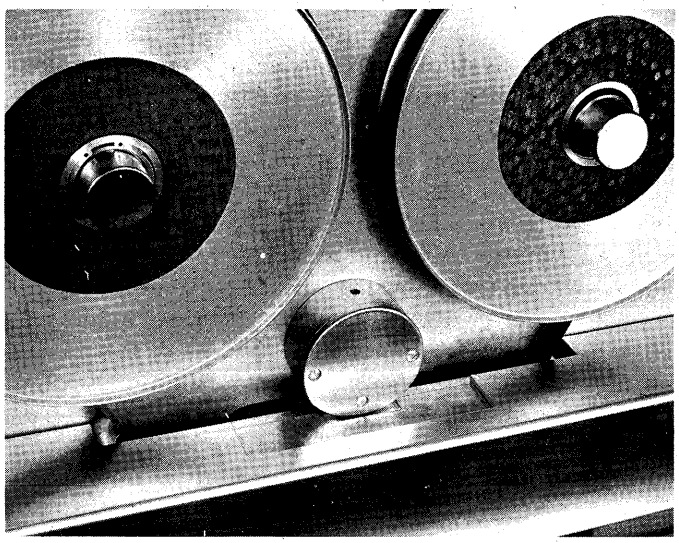


Fig. 2—Head area detail, in information transfer position.

Fig. 2 shows a closeup of the capstan area as it appears when tape is in position for information transfer. The centrally-located three-inch diameter billet contains the magnetic head assembly, and the oxide surface of the tape is uppermost. The tape lies horizontally, immediately over the two-piece horizontal vacuum brake, and thence executes a 90-degree downward turn at each vacuum capstan before dropping directly to the pneumatic loop chambers. Each capstan, when not actively engaged in driving tape, is provided with continuous air lubrication of approximately 2 psig, which effectively prevents all contact between the capstan and the tape. Unbroken edge guiding is present in the vicinity of the capstans, brake, and head, and indeed is present all the way from one reel to the other except in a space of 1¾-inches immediately next to each reel. Even in these regions back edge guiding is present, the deliberate absence of front edge guiding being in the interest of eliminating possible finger-catching accidents. It has

[3] R. B. Lawrance, R. E. Wilkins, R. A. Pendleton, "Apparatus for Magnetic Storage on Three-inch Wide Tapes." *Proceedings of the Eastern Joint Computer Conference*, 1956. Special publication T-92.

been our experience with magnetic tapes (as with other elongated flexible substrates) that continuous edge guiding is far more advantageous than guiding by periodically-spaced flanged spools, provided only that the tape be slit accurately enough. For nearly two years we have made complete and detailed observations of commercially produced magnetic tape with respect to width and the periodic curvature usually called snakiness. We can state that the snakiness can reasonably be reduced to complete insignificance while the width of slitting is held well within a total range of .002 inch.

We feel that these edge-guiding arrangements, together with accurate tape width control, yield considerable benefit in drastically reducing tracking and skew errors within the mechanism, as well as contributing to long tape life since the edges of the tape are nowhere subjected to localized sideways forces. Spring-loaded parts for exerting side-thrust on the tape are themselves subject to excessive wear, so their elimination enhances reliability.

Returning to Fig. 2 and comparing it with Fig. 1, we note that in normal operation, with the tape in the loop chambers and the head in position, the oxide surface is in rubbing or pressure contact with no parts of the mechanism except the magnetic head, bringing tape wear to a practical minimum. Fig. 3 shows the capstan and head area with the head eccentrically rotated, removing the head from contact with the tape oxide surface. Thus in high speed rewind (at whose beginning the head rotates away automatically) not even the head touches the oxide. Rotation of the head, automatically controlled, is also used during tape changing, at which time it enables the tape to slip easily over what is otherwise an unbroken edge guide.

In Fig. 3 the magnetic portion of the tape is all on the left hand supply reel and the leader of heavier-gauge clear Mylar® (permanently attached to the tape) is lying over the capstan and brake. This enables the nature of the exterior pneumatic passages of the capstan and brake to be seen. The two capstans are continuously rotated in opposite directions by individual 1200 rpm hysteresis synchronous motors, the capstan circumference being exactly 6 inches. The left and right portions of the brake (lying between the normal head location and the two capstans) are internally connected to a common working air passage which is supplied appropriately with medium suction, strong suction, or air at atmospheric pressure.

Fig. 3 also shows that no pressure pad is employed to keep the tape in contact with the magnetic head assembly. Wrapping contact between tape and head is adequately maintained by having the head press the tape down into a short and very shallow "V", the outer edges being defined by the rounded shoulders of the brake, closely adjacent. By means of this wrap, with its elimination of pressure pads, and by means of the unconventionally large radius of curvature of the magnetic head (both essentially the same in dimensions as in the DATAmatic 1000) we achieve good transient and running contact between tape and head, together with a gratifyingly low rate of head wear. Measurements carried on over more than a year's two-shift operation of a DATAmatic 1000 show for all channels of all magnetic heads a quite uniform and unexpectedly low rate of wear. The average yearly loss of material from the head under these conditions amounted to 0.0001 inch.

By implication, the tape wear produced by friction between head and tape is correspondingly small.

Fig. 3—Head area detail, with head pivoted for tape change.

Rewind and Tape Change

The central processor instructions to which the tape drive responds are Write (forward), Read Forward, Read Reverse, and Rewind. The Tape Change operation is initiated by manipulating a lever switch on the tape mechanism itself.

The position shown in Fig. 3 occurs at the termination of a tape change operation, which starts with a high speed rewind unless the tape is already rewound. Every rewind command is executed by the mechanism as a high speed rewind, and once received from the tape control unit the rewind is performed under local control until completed. During high speed rewind the tape speed is controlled by the left-hand vacuum capstan, whose motor speed is increased automatically to 3600 rpm. The tape remains in both vacuum loop chambers and accordingly receives the benefit of controlled tension and complete edge guiding. Upon rewinding past the designated beginning of tape, as sensed by a photoelectric arrangement described later, the mechanism shifts down from 360 ips to 120 ips. This latter speed endures for a fraction of a second and the tape is then stopped in normal fashion by pneumatic disengagement from

the capstan and engagement to the brake. The head, which has been automatically moved out of contact with the tape during the rewind, now rotates back into contact with the tape, and the closing of a Microswitch® signals the computer that the rewind has been completed and that the tape mechanism is again ready for instructions. At this time the magnetic head is positioned part way down the clear leader and has access to the first magnetic information location by moving the tape in the forward direction (to the right).

If it is desired to change tape, a centrally located manual switch on the control panel is thrown to the tape change position. It is irrelevant whether the tape is already rewound or not, although some head rotating operations are bypassed if the rewind is continuous with the tape change. The tape proceeds to the left, along the clear leader and at 120 ips, until a single short centrally placed slot in the leader is sensed by an orifice and vacuum switch associated with the upper end of the right hand loop chamber. When sensed this causes the tape to stop with only two or three turns remaining to be manually unwound from the right-hand reel, and the appropriate partial shutdown of the mechanism is initiated so that the reel is ready for removal.

Time taken for a rewind operation can be characterized by the equations

$$t_{\text{rewind}} \leq \frac{t_{\text{wind}}}{3} + 2.6$$

$$\text{or } t_{\text{rewind}} \leq \frac{\text{distance in ft.}}{30} + 2.6$$

in which all times are given in seconds.

Tape Reels and Mounting

Data processing magnetic tape mechanisms do not as a rule use standard reels, and the present equipment is no exception. Since a partial vacuum (about one half atmosphere absolute) is provided within the equipment for use in the clutch, it is quite natural to use this vacuum for holding the reels onto the reel mounts. This technique has already proved highly satisfactory with the three-inch-wide 23-pound reels of the DATAmatic 1000. Advantageous features include the lack of metal-to-metal contact between reel and reel mount, the fact that the reel hub is not subjected to hoop stress, and the provision of a large flat reference surface on the reel mount, which insures wobble-free rotation and accurate positioning of the reel relative to the back reference surface.

Suction is similarly used for attaching the free end of the tape leader to the right hand reel whenever a tape is loaded on the machine, as well as for initially attaching the inner end of the tape to the left-hand supply reel. It is worth mentioning that vacuum attachment of the tape to the reel makes it unnecessary to perforate the reel flanges for finger access to the hub during loading. The unperforated flanges are helpful in protecting tape from dirt and mechanical damage while in storage or during handling. Hazard to the operator is also reduced materially.

The design of the reel and reel mount involved additional factors, however. It was desired to make use of a demountable ring, capable of being stored with the reel of tape and serving by its presence or absence to enable or inhibit the recording of information on the tape. (This is in addition to a manual switch on the operator's panel.) Various embodiments of this principle have been used for several years by other manufacturers, but we believe our version has some useful and novel advantages. One desirable feature present in our arrangement is that the physical presence or absence of the write-enable ring does not need to be inferred from the status of a concealed electrical switch (which requires that electrical power be applied and that the circuit be functioning with some means of indicating its status). We have placed the write-enable ring in plain view on the front of the reel. It thus becomes easy to remove or insert the ring while the reel is in place on the mechanism, without the necessity for first rewinding the tape in order to remove the reel.

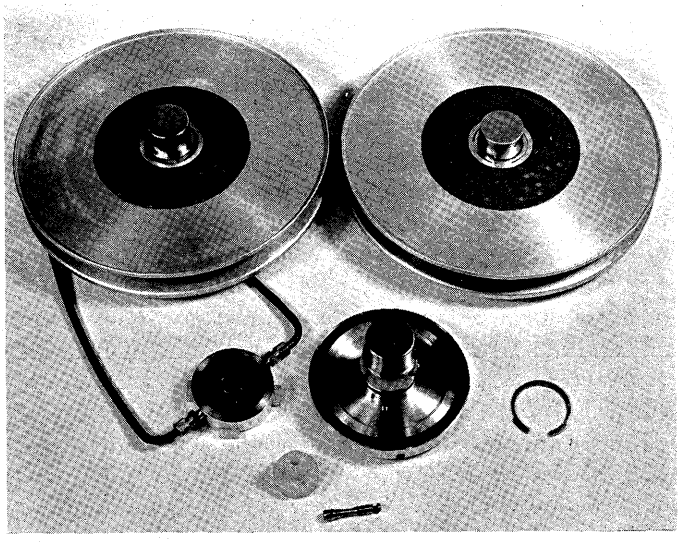


Fig. 4—Reels, reel mounts, and write-enable ring.

Fig. 4 shows a photograph of three reel mounts, one having a write-enable reel of tape mounted on it and another carrying a write-inhibit reel. The removable snap ring which converts a write-inhibit reel to a write-enable reel is shown beside the third reel mount. The principal working part of the reel mount subassembly is the central bell-shaped cylinder. Its axial motion controls three retractable nylon latches, spring-loaded radially outward, and also an internal piston, spring-loaded axially outward. To remove a tape reel from the mount the reel flanges are lightly grasped by the fingers, while the thumbs press the

central cylinder so that it moves axially inward. The three nylon latches are thus moved radially inward to the point where the reel can slide over them and be removed. In putting a reel on the mechanism, the central cylindrical bore of the reel performs a similar operation in reverse — as the reel is moved inward it presses on the nylon latches, retracting them. A small fraction of an inch before the reel is fully seated the latches snap outward and thus hold the reel in place even with no power or no vacuum. When vacuum is applied (automatically, as part of the normal cycle-up procedure) the reel is drawn into intimate sealing engagement with the rubber driving rings and is fully positioned ready for operation.

The write-enable ring operates by capturing the outer rim of the central cylinder, as the reel is pressed on. By this means the central cylinder is moved inward about ¼-inch as the reel is seated home. The internal piston-and-cylinder arrangement is thereby vented to atmospheric pressure rather than being connected to the half-atmosphere suction reservoir. The electrical image of these two pressure states is created in a stationary vacuum-diaphragm-operated Microswitch® located at the rear of the main mounting plate and sampling the pressure in the reel mount cylinder via a carbon rotary seal.

The Vacuum Clutch

Figs. 2 and 3 showed portions of the vacuum capstan and brake, and their relationship to the magnetic head as mounted in the mechanism. Fig. 5 shows an exploded view of these components of the pneumatic clutch, viewed from the side rear. Of the components all but the capstan motor are mounted to the front of the heavy flat vertical plate which serves as structural support and back edge guide, and which is omitted from the photograph. The capstans, of which only one is shown in exploded position, are directly mounted to the shafts of their respective hysteresis synchronous motors. Precision bearings are used in the motors, and capstan runout and taper are held to tight tolerances in order to achieve good tape tracking.

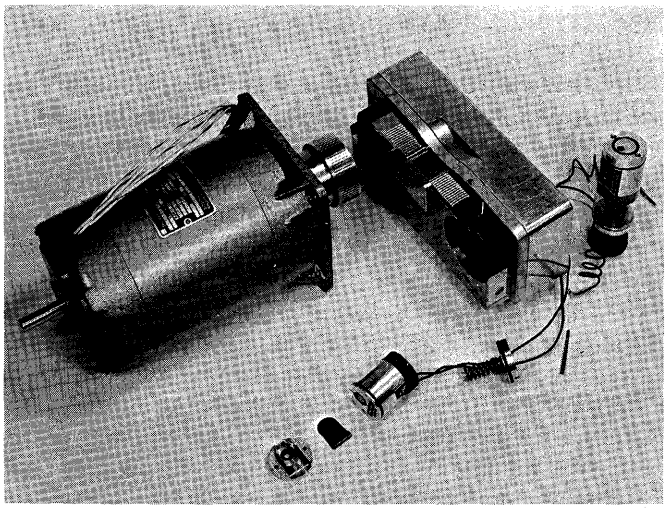


Fig. 5—Exploded view of capstans, brake and actuators.

A 90-degree segment of each capstan is connected via the working air passage to the electropneumatic valve, mounted nearby in the actuator housing body. (As shown in Fig. 5, this is bolted directly to the capstan housing body.) The fixed portion of each pneumatic commutator consists of a carbon composition cylinder which fits closely without rubbing inside the cuplike capstan. The portion of the working air passage within each carbon piece consists of a single slot centrally spanning the active arc of the capstan, and a drilled hole connecting to the actuator.

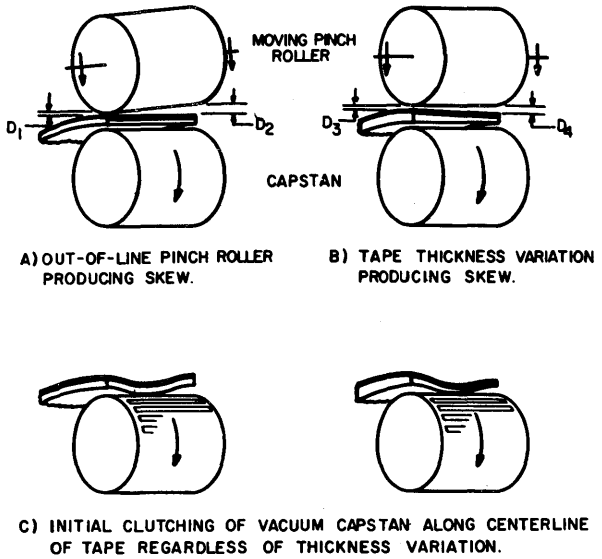


Fig. 6—Skew considerations for pinch rollers and vacuum capstans.

The location of this slot along the center line of the tape track, together with the pneumatically symmetrical design of the capstan itself, leads to what we believe is an important advantage for the vacuum clutching technique. Fig. 6 shows a series of sketches representing the clutching action and the production of skew in pinch roller and vacuum clutches respectively. Part A shows (greatly exaggerated) the engagement of a tape to a capstan when the moving pinch roller is slightly out-of-line so that distances D1 and D2 are unequal. While we have no quantitative measurements available it is not too difficult to imagine that an inequality of perhaps 0.0001 inch will result in significant time difference in the engatement of the two tape edges, to the capstan. Parallelogram distortion of the tape would then produce skew. Similarly, as shown in Sketch B, it would appear to be possible for skew to be produced even if the moving pinch roller were to be perfectly aligned with the capstan. Any thickness taper across the width of the tape will produce the same effect as an out-of-line pinch roller. Again we have no quantitative data

to support this conjecture but the point-to-point thickness tolerances to which tape backing is produced are large enough so that the possibilities of skew production from this cause should not be overlooked. The unsettling thing is that since tape is not customarily inspected for thickness uniformity it appears possible for portions of an otherwise perfect tape to produce random skew when used with a clutch of the pinch-roller type.

The situation is different with a vacuum capstan, however, as shown by experiment. A priori expectations (verified in detail by observations using a time-delayed stroboscopic flash) are that since the working air passages communicate to the underside of the tape symmetrically about the tape center line it should be the case that the center of the tape always engages first. Thereafter the region of engagement spreads symmetrically to the edges. Prior to the evacuation of the working air passage it and the underside of the tape have been supplied with air lubrication at slightly above atmospheric pressure; thus at the start of a clutching operation the underside of the tape is at rather definite and reproducible location with respect to the capstan surface. As shown in the fourth sketch of Fig. 6 it is thus to be expected that, to first order at least, any variation of tape thickness across the web will not significantly affect the symmetrical tape engagement.

Returning to Fig. 5, it can be seen that the valve actuators each contain a small but efficient electromagnet which is energized when its associated capstan is intended to drive tape. The highly effective eddy-current shielding of the aluminum actuator housing body prevents any external magnetic influence on the tape or in the head.

Fig. 7 shows a sketch of one of the electropneumatic valves, each of which is essentially a pneumatic SPDT switch. With no current in the coil the flat armature, resiliently pivoted near the right-hand end, will seal off the upper valve seat, being maintained in position by pivot bias and by air pressure differential. During this time the working air passage to the capstan is supplied with lubricating air from the pressure reservoir, at approximately 2 psig. When current is passed through the magnet winding the valve assumes the position of Fig. 7, with the armature magnetically drawn down to seal off the compressed air from the working air passage. The capstan and working air passage thus exhaust into the reservoir at half-atmosphere vacuum.

The armature is tapered slightly, as shown, to reduce inertia and speed up pull-in. Life tests on a group of similar armatures and magnets, driven at 120 operations per second for a period of over 20 months showed no measureable change in performance after 6.2×10^9 operations.

The transistor circuits which drive the actuators supply an initial high-current pulse for fast armature

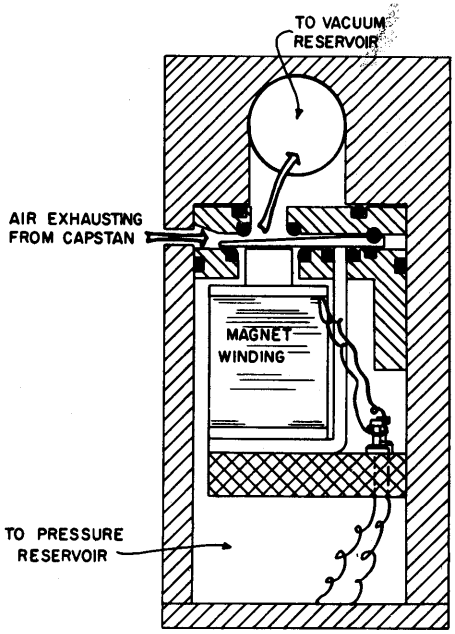


Fig. 7—Schematic of electro-pneumatic valve.

pull-in, dropping to a reduced holding current which lasts until the stop command is received. The ferromagnetic material of the magnet is an alloy with relatively low saturation flux density so that drop-out time is held to a minimum. The transistor circuits are interconnected in such a way that engagement of the tape to both capstans simultaneously is most unlikely, even under failure conditions; even if this should occur, however, the tape suffers no damage since the capstan motors will stall without the tensile elastic limit of the tape having been exceeded.

Typical curves of tape velocity versus time in starting and stopping are shown in Figs. 8 and 9. These curves were taken by recording on the tape a train of 64 pulses derived from a 5 kc keyed oscillator, turned on at the time of the stop or start command. Magnetic development with colloidal Fe_3O_4 and position measurement with a microscope and traveling micrometer table were then used to give an accurate history of tape position and velocity, relative to the read-write gap, versus time.

Fig. 8 shows that in response to a start command the tape commences to move at slightly less than one millisecond; at 2.7 milliseconds the tape has traveled 0.12 inch and is traveling at 120 inches per second. Speed fluctuations thereafter do not exceed approximately 3 or 4 percent, although the read system will tolerate many times this amount. Fig. 9 shows that in stopping, the initial deceleration occurs after about 1.2 milliseconds and that the total distance to come to rest is substantially less than 0.3 inches. As mentioned briefly earlier the 804 mechanism allows a start command to follow a stop command arbitrarily closely. The present curves indicate why the tape continues at full speed when the interval between commands does not exceed approximately

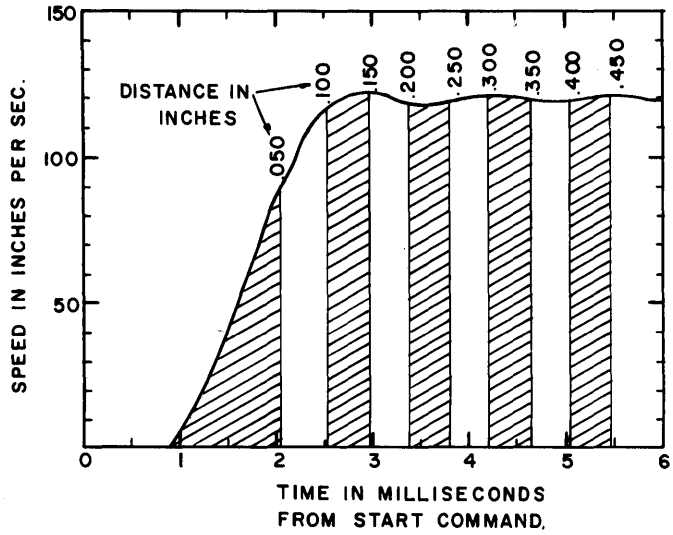


Fig. 8—Typical curve of velocity and distance *vs.* time during tape acceleration.

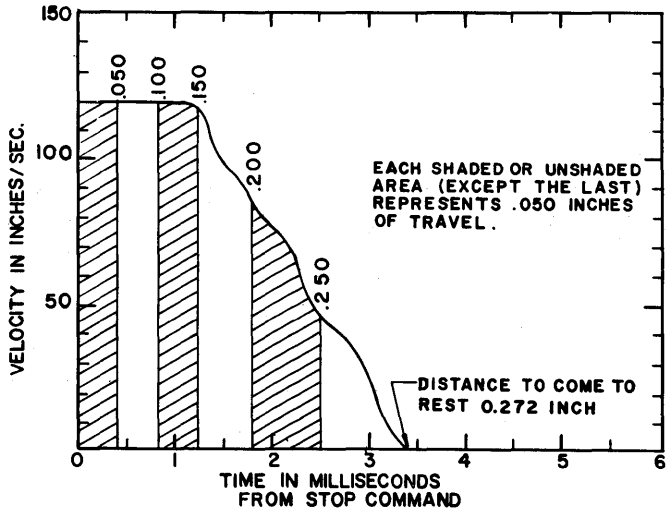


Fig. 9—Typical curve of velocity and distance *vs.* time during tape deceleration.

0.7 milliseconds; for longer intervals there is a smooth transition to the isolated-stop, isolated-start condition shown in the graphs.

Sensing of Beginning and End of Tape

In the interest of brevity we will not give a complete description of the circuit arrangements for keeping track of the position of the tape in the machine: that is, whether the magnetic head is positioned over the permanently-attached clear leader, initial information space, mid-tape information space, or one of the recognizable end-tape zones. With one exception, the task of remembering tape positions on all of the eight connected tape mechanisms is assigned to their common tape control unit. The exception is concerned with rewind operations, in which the controlling elements are completely local to the respective tape drives: a relay picks up at the start of the rewind and only releases upon sensing clear leader at the completion of the rewind.

The boundaries of all logically distinct tape regions are marked off by small windows at the front edge of the tape, created by removing oxide for a distance of 0.1 inch along the tape and .035 inches in from the edge. Since the nearest recording channel ends 0.041 inches from the edge there is no conflict between optical sensing and magnetic recording. It is possible to sense the passage of a window without interfering in any way with the execution of any write or read instruction which may be in process. Significant program advantages and time savings result from this feature.

The special illuminator contains a miniature long-life tungsten filament bulb and a one-piece optical element consisting of a lens, cylindrical barrel, and angular refracting surface. This illuminator is positioned at a fixed distance from the magnetic head near the upper end of the right-hand loop chamber, with the optical element extending at an angle through the loop chamber outer wall to a position nearly flush with the inner surface. By this means, since the angle of the refracting surface is nearly the angle for grazing refraction, a satisfactory intense light source is effectively positioned directly opposite the outer edge of the tape, yet without mechanical projection into the path of the tape.

Upon passage of one of the windows, light falls on a miniature silicon photodiode (part of the sub-assembly) which issues the window-recognition signal for interpretation and storage.

Read-Write System

In the Honeywell 800, as in nearly all other systems, the tapes are written in the forward direction only, *i.e.* with the tape moving to the right. Reading takes place in either direction as desired, and uses the same head gaps as for writing. Ten channels are used, of which eight are information channels, one is an Orthotronic parity channel, and the tenth is a clock. A separate full-tape-width erase gap, located a fraction of an inch upstream of the read-write gaps, applies AC erase to the tape at the time of recording. The read-write gaps are in-line across the tape and are spaced on 0.070 inch centers.

The AC erase serves the primary function of cleaning out the inter-record gaps and leaving the tape magnetically neutral, which facilitates record-entry recognition in bidirectional readback. NRZ1 recording (saturation-to-saturation, flux change denotes a "one") is used on the information and parity channels. The Honeywell 800 word contains 48 bits (not counting the parity bits which accompany the information on tape and in memory) so that a word occupies six frames on tape, a frame being defined as the time-simultaneous record of a bit in each information channel. The parity bit is also recorded simultaneously with the eight information bits. The frame interval is 21 microseconds, corresponding

to a frequency of 47,619 frames per second and a bit density (at 120 inches per second) of 397 per inch.

The clock channel is similarly recorded from saturation to saturation, but undergoes one flux reversal per frame. The recording of the clock is not simultaneous with the recording of the other bits of the frame but is offset by one half of the frame interval. By this means the read circuit is made self-timing, highly tolerant of speed variation in the tape mechanism, and free from one-shot circuits with their jitter and delay tolerance accumulations.

As soon as a write instruction is received the erase head is excited and remains so, independent of tape motion, until receipt of the next instruction of a different type (read, rewind, tape change). At the beginning of a record to be written, with the tape in motion and the inter-record gap just traversed, write current is initiated in all ten channels in the same standard polarity. This results in half-strength magnetic poles of known polarity being written in all channels (automatically ignored in subsequent playback). Thereafter the clock begins its 21-microsecond beat and 10.5 microseconds after the first clock beat the first frame is recorded, with flux reversals in those channels where ones are to be written. Writing continues, at six frames per word, until all words of the record have been recorded. Before cessation of writing two orthotronic words (twelve frames) and an end-of-record word are appended, after which one more clock pulse is written and all write currents drop to zero.

The construction of the orthotronic words is on a per-channel basis, roughly as follows: half the first, thirteenth, twenty-fifth . . . bits are added modulo 2 and the first bit of the orthotronic word is the complement of their sum. Similarly the second orthotronic bit is formed from the second, fourteenth . . . bits of the record, etc. The result is a very powerful check having the following properties:

1. Garbled information confined to a single channel can be recreated regardless of the length of the difficulty.
2. Garbled information extending up to twelve bits in length can be reconstructed regardless of the number of channels affected.

In playback the ten channels are connected, by means of solid-state switching, to ten individual preamplifiers located at the tape mechanism and thence are passed via the tenfold read bus to the Type 803 Tape Control Unit. Further shaping culminates in peak detection of each signal and the production of a one-half microsecond pulse essentially coincident with each flux change in the channel. These pulses set nine individual high-speed flip-flops, which accumulate the bits of the frame; the next peak-detected clock pulse (half a period later) resets all flip-flops and sends the bits into buffer storage where they reside until a complete word is available for transmission to memory.

Fig. 10 shows the appearance of playback from a single channel, and has the typical NRZ1 waveform. Because of the conservative bit density satisfactory resolution is achieved with a comparatively large head gap, minimizing signal fluctuations due to the passage of lint or other debris between the head and the tape. Fig. 11 shows the effect of a recording dropout deliberately produced by blowing fibers of cotton lint into the region between the magnetic head and the tape being written. The amplitude decrease shown is typical and produces no error in reading, as shown by the associated peak detector waveform. The read system is designed to tolerate signal decrease to well below one-fourth of normal amplitude. It is well to mention, also, that the tape mechanism incorporates the conventional positive pressurization of the region occupied by reels, capstans, head, and loop chamber entrances, thus excluding airborne dust except during necessary tape changing.

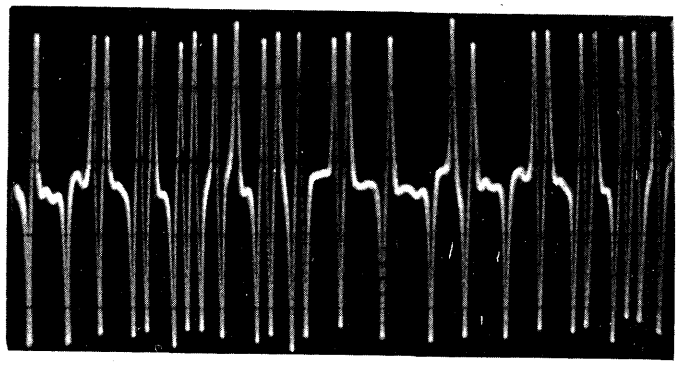


Fig. 10—Normal preamplifier output of read system.

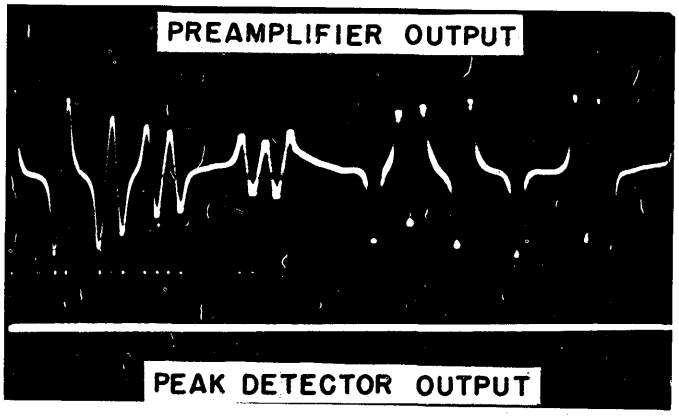


Fig. 11—Read system waveforms, including deliberate recorded dropout due to loose cotton lint.

We have not dealt at length with the internal checking of the Honeywell 800 but it is well to mention, in conclusion, two of these features associated with the magnetic tape system. Writing cannot occur (and its absence is made known) unless an enabling check shows that the erase head and the clock channel are both excited. The transmission of data to the tape

drive is checked for transverse parity at each frame and for longitudinal parity on each channel of each record.

The net result of the features described in this paper is a strong, efficient, and trouble-free tape system. The approach deliberately taken has been to design high reliability into all electrical and mechanical components, effecting error detection and correction by means of the powerful capabilities of Orthotronic control.

Acknowledgments

The read-write circuit and system development has been done by a group directed by Dr. Way Dong Woo. I am sure that the many other contributors to the design of the tape drive itself will not object to my singling out Mr. Robert A. Pendleton as major contributor.

Bibliography

[1] N. Lourie, H. Schrimpf, R. Reach, W. Kahn, "Control & Arithmetic Techniques in a Multi-Programmed Computer." Eastern Joint Computer Conference, 1959.

[2] R. B. Lawrance, R. E. Wilkins, R. A. Pendleton, "Apparatus for Magnetic Storage on Three-Inch Wide Tapes." *Proceedings of the Eastern Joint Computer Conference, 1956.* Special Publication T-92.

[3] R. A. Skov, "Pulse Time Displacement in High-Density Magnetic Tape." *IBM Journal,* April 1958.

Discussion

W. N. Papian: I wonder if you could give us a recap of the key characteristics.

Dr. Lawrance: The number of channels is ten, of which eight are information, one is the Orthotronic parity channel, and one is the clock, recorded at the same time as the other nine channels. High-frequency AC erase occurs upstream of the read-write gap at the time of writing. Each channel is recorded at 397 bits per inch (nominal 400). The three-fourths-inch wide tape moves at 120 inches per second in either direction. Rewind speed is 360 inches per second, and every rewind command is executed as a high-speed rewind.

D. Allen (Ampex): What types of errors do you typically encounter and what is their frequency? Are they so numerous as to justify correction hardware? What causes them?

Dr. Lawrance: In the oral presentation I omitted to stress sufficiently one unique feature of our mechanism. The oxide surface of the tape is in contact only with the magnetic head and does not rub or abrade against any other surfaces in the entire mechanism. We followed the philosophy that the less abuse to the oxide, the better. Familiarity with less gentle tape mechanisms may condition one to think in terms of frequent tape errors. By contrast we have had excellent tape reliability both during our extensive DATAmatic 1000 experience and in our experience to date with the Honeywell 800. Consequently, it is not correct to imply or infer that Orthotronic control is required for correction of an inordinate number of tape errors. In engineering fact, what Orthotronic control does is to take a tape system which, by all present standards, is excellent and extend its performance to still higher standards.

With regard to errors and their causes, it is extremely difficult to obtain much data without taking a long time to do so. Typically, in a fully operating computer with eight tape mechanisms, several days elapse between individual malfunctions which could possibly be tape errors. Some evidence supports our belief that airborne dirt (notably clothing lint) momentarily passing between tape and head is responsible for a significant part of these residual errors. Fig. 11 shows results obtained under these circumstances; the printed paper mentions the pressurizing used to avoid introduction of airborne dirt.

G. A. Barnard (Ampex): How many 19-inch racks, or equivalent worth, of electronic, mechanical, power supply, compressed air, and vacuum equipments are needed for the total tape system?

Dr. Lawrance: The unit shown in Fig. 1 contains everything mentioned in the question. Specifically, all actuator power supplies, reel control power supplies, compressed air and vacuum sources, erase electronics, and read-write electronics associated with the head are inside the unit. By the latter is meant the write amplifier final stages (each of ten channels), the read amplifiers (each of ten channels) and the power supplies for these circuits.

The single Type 803 Tape Control Unit, which is associated with up to eight drives, handles the data to be written (including checking) and contains the peak detecting and buffer storage for the read system. This unit, through which the eight tapes communicate with the central processor, is free standing and, with its power supplies, is almost exactly four feet wide.

D. E. Killen (Oliver Shepherd Industries): What is the maximum skew in microseconds?

Dr. Lawrance: We do not have final figures for publication yet, but I would estimate not much more than two microseconds.

H. S. Davin (Sylvania): Is there any prospect for increasing recording density on the H-800 Tape Drive?

Dr. Lawrance: We have chosen a conservative figure of 400 bits per inch (each channel) which gives the very high information rate of 96,000 decimal digits per second. If a special application required higher bit rates, I feel the density could safely be increased somewhat.

J. Hunter (Broadview Research): Are you using sandwich tape?

Dr. Lawrance: No. It is possible to do so, but our tape life and cleanliness lead us to feel that sandwich tape is not necessary.

R. Pacel (Rem. Rand): What is the tape-to-head separation?

Dr. Lawrance: The oxide surface of the tape is in contact with the head.

J. M. Kolb (Lincoln Labs.): Do you use separate Read-Write windings on heads? What happens if Start order is given before Stop is completed?

Dr. Lawrance: There is only one winding per channel, used for both reading and writing. The full-tape-width erase is applied from another gap located upstream of the read-write gap line.

With regard to the second part of the question, if a start order is given before the stop is completed, the start is executed without requiring any artificial delay. There is no restriction on the interval between commands, and the last one received prevails.

G. F. Roe (Hughes Aircraft): Do you use the "MARK" method of NRZ recording?

Dr. Lawrance: Every "one" bit is a current (and flux) reversal, putting a pole on the tape. Every "zero" bit is no change in current direction and no pole on the tape.

E. Seif (Burroughs Corp.): How much time is required to stop and then start in the opposite direction; *i.e.*, time elapsed from normal speed in one direction to normal speed in opposite direction?"

Dr. Lawrance: In reversing direction, the stop command must first be given; *i.e.* the first-existing command to drive (say forward) must be released. As far as concerns the mechanism and the tape, it is safe and permissible to issue the reverse drive command within microseconds after the termination of the forward drive command. In use with the computer, however, it is presently planned to incorporate a delay (approximately two milliseconds) so that the turnaround point is far enough down the tape to guarantee perfect reading of the just-traversed information in the second direction. To answer the question, then, the time from full speed to full opposite speed is approximately five milliseconds.

Honeywell

ELECTRONIC DATA PROCESSING

The Philosophy of Automatic Error Correction

R. M. BLOCH

AUTOMATIC computing and data-processing equipment is depended upon today to perform a substantial portion of all data-processing requirements of government and industry in this country. It is natural, then, that the issue of reliability and the related issues of error detection and error correction have assumed a position of utmost importance from the viewpoint of users of such equipment. The manufacturers of data-processing equipment, recognizing the important role of their systems in the nation today, have likewise turned their attention to these areas and are bringing their most potent forces to bear on these vital issues.

Many important technological advances which are being incorporated in present-day machines are substantially reducing the frequency of error commission. The increasing speed with which these machines operate, and the corresponding increase in work load, have made it mandatory that not only the rate of error commission be reduced, but that efficient means be found for the treatment of these errors when they do occur. It is in this latter area where the principle of automatic error correction is now coming into the forefront of consideration.

Error Detection

In any technique for automatic error correction, the prerequisite of automatic error detection must exist. In this regard the manufacturers of data-processing equipment have developed many types of automatic detection techniques during the course of the past decade. Certain of these techniques are based upon maneuvers in the area of machine programming. This mode of attack is particularly suited to processing involving a great deal of mathematical computation, where various types of check formulae can be brought to bear to detect inconsistencies in the results obtained. The correction technique generally employed here is that of rerunning a portion or all of the program in which the error or errors are suspected to have occurred.

A second technique of detection involves that of a redundancy in equipment, specifically arranged so as to yield immediate detection unless two or more units have malfunctioned at the same time and in the same fashion. Although this method has been utilized, its applicability will probably be limited in advanced equipments. This is primarily due to the costs involved in the additional equipment, as well as certain difficulties in the subsequent application of modern error correction techniques.

The detection technique which has been put into practice most frequently involves that of redundancy of information. The redundancy has ranged from complete duplication of information to simple parity checks of long trains of information. Nearly all electronic data-processing installations in use today apply this concept of redundancy in one form or another. The power of these redundancy checks has usually been evaluated on a theoretical mathematical basis. That is to say, it has generally been assumed that the frequency distribution of error patterns is based upon independent and equal probabilities of single digit errors. This premise was not unreasonable in the absence of evidence to the contrary. However, sufficient operating data on various equipments are now being assembled, which indicate that this assumption is unjustified. These data indicate that electronic equipment is, in fact, subject to certain repetitive error patterns, and that the probabilities of single digit errors are neither equal nor independent. When one considers the fact that a complex digital system is comprised of discrete electronic elements, then it becomes increasingly apparent how difficult it is to predict in advance the nature of the error patterns. Transistors, resistors, diodes, and the other basic elements of an electronic system have inherently different reliability characteristics. This in itself would indicate that certain error patterns may be much more likely than others. To make matters more complicated, different systems, indeed, different sections of a given system, make use of these components in varying configurations. To illustrate this problem, Fig. 1 shows a system wherein four parallel trunks are emanating from a memory array. These trunks are then shunted into a single operational network. It is not important for the purpose of this discussion to detail the precise

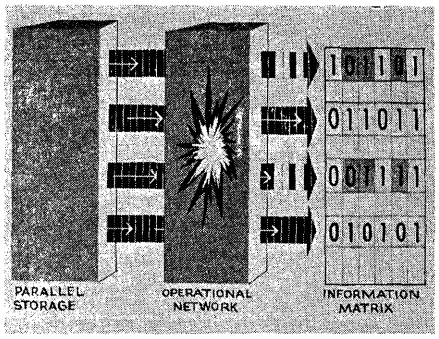


Fig. 1. Parallel information flow through a single operational network

operations being performed within this large network. It is only essential to understand that there are elements within this network which can fail in such a way so as to affect the information on several of the output lines. Furthermore, if it is assumed that each of the entering trunks contains a continual flow of information digits, it is clear that the characteristics of the circuit failure with respect to time must also have an effect on the error content of the output information. If the component were to fail permanently, one would expect the error pattern with respect to time to be perpetuated. If, on the other hand, the failure is of a transient nature and of very short duration, one would expect a corresponding error pattern of short duration to result in the exiting information.

Suppose the system design is now changed so that the four trunks emanating from memory are treated in independent operational networks, as shown in Fig. 2. A failure in one of the four networks will now presumably have no effect upon the proper operation of any of the three other networks. In particular, an error in the functioning of Network 3, as shown, will surely affect the output of this network; whereas the other three networks will be yielding correct information. Although the error patterns of a given trunk as a function of time may conceivably be similar in the two systems, it is obvious that the error pattern in Fig. 2 will generally be confined to a single trunk, whereas the pattern in Fig. 1 may frequently be distributed across several information channels. The implications of these two elementary diagrams upon the theory of error detection, and as will be seen later, upon error correction, are clear. Regardless of the error frequency, System 2 will yield a heavily biased error distribution

R. M. BLOCH is with the Datamatic Division of Minneapolis-Honeywell Regulator Company, Newton Highlands, Mass.

when viewed across the entire matrix of information.

Now if the operational network in Fig. 1 fails for an extremely short time duration, it is quite probable that a single column of the information matrix will be in error; several elements of this column, however, are likely to fail during this interval. If it can be demonstrated statistically that by far the greatest proportion of errors is of this nature, then a detection system which is capable of ferreting out a single error in a row of the matrix would be ideal. To the contrary, a detection system which will intercept a single error in a column might be very impotent indeed. If the transient failure in this network were of long duration, then single error detection in either dimension is ineffective. In the independent network array of Fig. 2, however, a somewhat different situation obtains. A transient of length equal to a digit time will likely affect but one element of the matrix and may be intercepted by any single error detection scheme. Failure of longer duration will lead to a multiplicity of errors in a single row; here, a single-error detection system in the columnwise direction will be extremely powerful, and single error detection along a row would be useless.

Empirical Concept of Design

From considerations such as these, a new approach is being taken by data-processing equipment manufacturers with regard to the design of automatic error detection and error correction networks. Heretofore, detection techniques were specified by systems personnel based upon theoretical formulations. These specifications were then transmitted to the circuit design area where the detection system became a physical reality. In this arrangement, the circuitry designers used their best reasoning and experience to assist the systems staff, but the specifications were still born of abstraction.

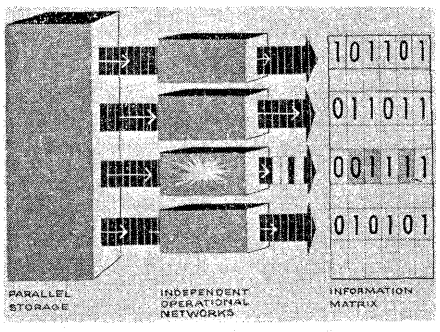


Fig. 2. Parallel information flow through independent operational networks

The new concept, which is coming into practice as of today, in effect states that the equipment itself is best able to define its own idiosyncrasies in respect to error behavior. The principle is the same as that which has been responsible for many outstanding advances in the field of science, namely, the use of empirical evidence to substantiate or refute a theory which has been propounded. Through laboratory observation of actual data-processing networks, it is possible to build up a sizable array of statistics on the error behavior of these very networks. These behavior statistics then enable the system designer to formulate a very powerful detection technique which has been tailored to the particular error pattern of the equipment. In actual practice, an interchange of information between the circuit and system groups not only may affect the detection technique used, but also may have its effect upon the original circuit design as well. Indeed, it is possible that certain changes may be made in the circuit design to yield an extremely favorable bias of error distribution for purposes of interception by a given detection system. Such a change in design is then reconfirmed by further empirical laboratory work, and thus the circle is eventually closed to yield a proven and powerful detection arrangement.

The final design that comes forth under this new concept may indeed defy the intuitive judgment of both the systems and the circuitry groups, but its validity cannot be denied; this will be borne out when the machine is put into operation in the field.

Error Correction

There are two broad classes of error correction. The first or "reversion" class consists essentially of re-executing the operation which is in error immediately upon detection. This class of correction includes all forms of program rerunning which involves reverting to some segment of the program prior to the error in question. In general, this class requires that the information in its correct form be retrieved from some previous point in the operation. It is presumed that this information in said correct form is available within some storage network of the equipment.

The second class of error correction, "deductive" correction, involves a powerful detection scheme which is capable of isolating the actual elements of information which are in error, correcting these elements, and proceeding with the program in progress. All error correcting codes have these characteristics. It is not necessary to revert to a previous point in the program, or to have access to the storage of the information in the correct form. It is literally possible to recreate the correct information by use of additional digits which have been transmitted along with the information.

The reversion class of correction has been used for some time. It is especially successful in those sections of the equipment where the retrieval of the original information is not difficult. For example, in arithmetic operational networks, once an error has been detected, say in an addition process, it is often possible to have the machine automatically retrieve the operands from memory and proceed once again with the addition process. Rereading or rewriting of magnetic tapes, once an error has been detected, are also reversion forms of correction which are being employed in various equipments today.

The second or deductive class of correction is much more difficult to deal with in practice, although it has certain outstanding advantages. Its primary importance rests in the fact that in certain areas of data-processing and computing systems it is extremely impracticable to retrieve the information in its original or correct form, and in some cases impossible to do so without a complete cessation of machine operation. It is for this reason that a great deal of attention recently has been directed to redundancy codes which can perform the function of automatic error correction. Inherent difficulties accompany the use of such codes, and these difficulties have thus far precluded the possibility of their use in any practical way in data-processing equipment. These codes generally have limitations upon the number of errors which they may detect, and if these codes are expanded so as to enable correction of a multiplicity of errors, then the redundancy required may become excessive, and the total information capac-

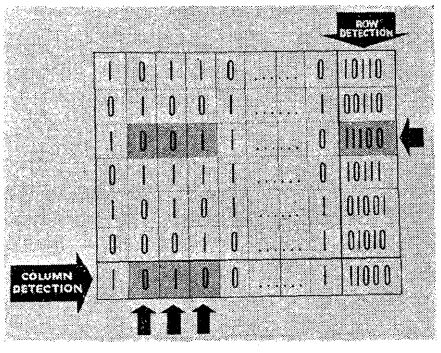


Fig. 3. Typical information array showing principle of two-dimensional detection

ity of the system is thereby severely reduced. If, on the other hand, the number of error digits exceeds the limit, then actual erroneous correction may take place; this characteristic further complicates the usefulness of such codes.

Orthotronic Control

A system of automatic error correction has recently been developed and is being put into actual use today. This system, Orthotronic Control, utilizes a strong two-dimensional detection technique capable of isolating and correcting nearly all errors committed in the system which it monitors. The new concept of design of error monitoring systems, which was discussed earlier, was put into practice in the development of Orthotronic Control. To describe this correction technique, let it be assumed that information is transmitted along parallel channels from memory, enters an intermediate buffer storage stage, and is then recorded upon magnetic tape in a multiplicity of channels. At some later time this tape is read and the information again is transmitted through parallel channels to an input buffer storage stage and thence to the main memory. The correction system is intended to retrieve automatically any information which has been lost in the course of the full transmission cycle from the time information leaves the memory to the time that it is returned to the memory; the transcription to and from magnetic tape will have occurred in the interim period. Great strides have been made in the reliability of the input-output trunk systems of data processors; yet is it still generally true that this segment of any processing equipment is probably more prone to error than all other segments combined. It is, then, in this input-output trunk network where Orthotronic error correction comes into effective operation. Several channels of information are recorded on magnetic tape, as shown in Fig. 3. Each of these channels has been independently controlled by separate sections of the output buffering system, as well as independent writing circuits. The correction system adds an additional channel, referred to as an Ortho-channel. This channel consists of a simple parity count of the corresponding digits in all of the information channels on a column-by-column basis. This Ortho-channel is automatically generated as information leaves the main memory of the system enroute to the output buffer section. The information channels, as well as the Ortho-channel, are simultaneously recorded on magnetic tape. A series of check digits which are associated with each information channel, and which were actually stored with the information in the main memory, are also recorded at this time. These check digits may be referred to as row-detection digits in contrast to the column detection digits comprising the Ortho-channel.

Experimental evidence has shown that in a system such as the aforementioned, where parallel and independent transmission is in effect, errors will tend to be localized in a single channel as a result of any given equipment malfunction. In view of this behavior pattern, only a single binary digit may be expected to be in error in any given column. However, since the error disturbance may well affect a great number of binary digits within a single row, it is necessary that a much more powerful detection scheme be used in this dimension. Here again empirical evidence performs a most useful function in determining the principle upon which the row-detection digits are to be constructed. Assume, for example, that groups of 100 binary digits are to be monitored in each row, and this comprises a block of information. If, now, it can be determined that nearly all errors are of such a nature as to convert 1's to 0's but leave the 0's undisturbed, then a set of check digits which is nothing more than a binary sum of the 1's in this particular channel will prove to be a very powerful detection arrangement. If, on the other hand, erroneous inversion of a 0 to a 1 is as likely as the opposite transformation, then a binary count technique is not as effective, and some other technique may be better employed. For example, it is possible to use different weights in successive columnar positions over any arbitrary span to form an arithmetic or logical sum of the weights corresponding to those columns in which the digit 1 exists Generally, the greater the number of digits used in the check sum, the greater the power of the detection system. It should be noted here, of course, that the use of a simple parity bit is completely ineffective in the light of the error pattern being discussed; any even number of errors within the channel would not be detected at all.

It is possible to arrange a weighting system which is distinctly biased so as to intercept the most common error patterns to be anticipated. Thus, suppose it is found that the predominant error patterns on magnetic tape involve a relatively small number of binary digits within a moderate span in a single channel. A weighting scheme has been devised for this pattern which intercepts all possible single bit, dual bit, and triple bit errors, and furthermore intercepts all but two-tenths of one percent of all other possible errors within a span of nearly 1,000 digits. This detection capability is valid whether the digit errors are due to drop-out of information, pick-up of false information, or any combination of the two whatsoever. When such a detection scheme is used, it can be confidently expected that a negligible number of errors will escape the detection network.

Once having established, then, what may be termed a strong row check, isolation of the actual erroneous digits is a rather simple matter. The row-detection digits are used to sense the channel in which the error has occurred, and the column-detection digits will indicate precisely those columns in which a digit is in error. It is then possible to have the machine automatically invert the erroneous digits and thus reconstruct the information, bringing it into its correct form. It is a characteristic of this Orthotronic system that whenever a rare error pattern occurs which is not capable of correction, the system is able to discern this fact; and fallacious correction will not occur. For example, if on occasion an error pattern were to occur extending across two or more channels within a given block of information, then it would not be possible in the system shown in Fig. 3 for correction to take place. However, in this case the two row-detection digit arrays will reflect an error, and this will be used to signal the uncorrectable condition. Another characteristic of the system rests in its ability to determine when errors have occurred in the check digits themselves. Under these circumstances the detection digits are corrected and the information is left untouched. The Orthotronic system may take various forms. Through the use of dual columnar-detection channels, for example, it is possible to detect and correct all error patterns which extend across any two adjacent channels. Again, the proper form for this Orthotronic system is dictated by the error patterns of the particular input-output system under consideration.

Future Role of Automatic Correction

The true value or merit rating of an equipment is measured by the net quantity of accurate processing which it produces within a given time period, as well as by the costs incurred in performing this processing. A certain premium will be paid for circuit designs and manufacturing techniques, especially conceived to attain an unusual level of reliability; this is particularly true in the case of certain critical

military applications. There is, however, a point beyond which further equipment improvement becomes prohibitively expensive, especially in systems placed in commerical usage. It is at this point where the advantages of automatic correction are most clearly demonstrated. With a relatively small amount of additional equipment cost, the operational merit rating of the equipment can be raised to a degree, the equivalent of which is impossible of achievement by any other means. Appropriate automatic correction techniques, when integrated with equipment of fine reliability, can be expected to usher in a remarkable new era, one in which exceptionally long periods of machine operation can be anticipated with little or no human intervention, and with unprecedented confidence in the accuracy of the final output information.

Reprinted From
Special Publication T-114
Proceedings of the
Eastern Joint Computer Conference

ORTHOTRONIC CONTROL

What It Does

Orthotronic Control is a system of error control, unique to Honeywell electronic data processing systems, which detects and corrects without human intervention mistakes occurring in magnetic tape operations.

When computers record data on magnetic tape they use computer language, a binary code. There are many binary codes. The familiar Morse Code is one and the Telegraph Code is another. In such codes, the numerical and alphabetical symbols, used by humans, are replaced by specific arrangements of dots and dashes or, more usually, dots and non-dots. If, accidentally, a recorded dot is played back as a non-dot or a non-dot as a dot, then an error has occurred.

Although modern magnetic tape handling mechanisms used with electronic computers are extremely reliable, they are sometimes subject to introduction of errors into the data being recorded or played back. These errors are not caused by the tape units or other components, but rather by extraneous sources, such as minute flaws in the coating of the tape itself, or by the transient presence of dust, hair, or lint. Such particles may be carried under the magnetic head and, by momentarily separating the tape from the head, cause errors.

Errors in magnetic tape operations are like tire punctures. They do not occur very often, but when they do .....! Some modern tires, however, do not go flat when they puncture and the journey continues without interruption. Orthotronic Control confers a similar benefit on magnetic tape operations.

All modern magnetic tape systems include what is called a Transverse Parity Check* which detects the occurrence of errors and stops the progress of the computer

program. The work of locating the error and of making the necessary correction is usually left to the computer operator. This wastes computer time and, as it requires human intervention, can sometimes introduce further errors.

Orthotronic Control eliminates the need for human intervention. Errors, when they occur, are detected by both a Transverse Parity Check and a Longitudinal Parity Check, and control of the computer is automatically switched to a special error-correcting program called the Orthotronic Routine permanently stored in the memory. This routine, making use of certain checking information contained in every record, locates the errors and corrects them. Having performed its task, the routine transfers control of the computer back to the original program which proceeds as though nothing has happened. The time taken by the Ortho-Routine to correct an error averages about five thousandths of a second!

Records are recorded one "frame" at a time transversely across the tape in eight parallel channels. So powerful is Orthotronic Control that it will reproduce a record correctly even if a channel fails to record, or play back, throughout the record.

Although errors rarely affect more than one channel at a time, Orthotronic Control will do more if necessary. For instance, if a hair lying across the tape causes all channels briefly to lose contact with the head, the Orthotronic Routine will correctly reproduce the lost data.

Orthotronic corrections take place too quickly to be noticeable. However, they can be automatically counted and the number printed out at the conclusion of each tape run. Such a procedure provides a continuous check on the condition of the tape. Frequent corrections might indicate that a tape is in poor condition or that the storage and handling environment is dusty.

If data containing numerous errors is transferred under Orthotronic Control from an unsatisfactory tape onto a new tape, the errors are all corrected. Such a process can be compared, in its level of sophistication, with the work of an accountant.

But how many accountants can locate and correct an error in five thousandths of a second? How many could find and rectify the errors in a file of 100,000 items, each of 150 numerical and alphabetical characters, in approximately five minutes? The Honeywell electronic computers can. What is more, they do it automatically each time the file is updated.

How It Does It

When a Honeywell system writes down a record of data on magnetic tape, it does so one "frame" at a time. Such a frame consists of eight information bits and one parity bit. The latter is an "odd" parity, that is to say, it takes the value 0 or 1 so that the total number of 1's in all nine bits of the frame is an odd number.

These parity bits are continuously associated with the data, not only on magnetic tape, but also in the computer memory. Every movement of data is checked by verifying that the frame parity remains odd. Any malfunction which changes a 0 into a 1 or vice versa is instantly detected. This is called the Transverse Parity Check.

Certain kinds of errors may not be detected by the Transverse Parity Check. For instance, if one or more frames of a record were accidentally omitted or if two compensating errors left the record unaffected, then the Transverse Parity Check will fail. However, such errors are detected by the Longitudinal Parity Check. This check consists of verifying that the total number of 1's recorded, in each of the nine channels of a record on tape, is an even number. Like the Transverse Check, the Longitudinal is performed on every record as it is recorded and also when it is played back from the tape.

If, when a record is played back from tape, either of these two checks indicates a discrepancy, this fact is stored and control of further operations is automatically

passed to a stored program called the Ortho Routine. The transfer of the affected record from tape to memory is completed, however, and the subsequent error analysis and correction is performed by the Ortho Routine on the record as stored in the memory.

Records on magnetic tape are of variable length and are separated by gaps of 0.65 inch. All records consist of two parts, the major part, being the data, may occupy several inches of tape. The minor part, called the Ortho Words, occupies 1/32nd inch. The Ortho Routine makes use of the Ortho Words to rectify the data.

The Ortho Words are computed automatically by the computer at the time the record is first prepared for recording on magnetic tape. They consist of two computer words which are the result of adding together the successive pairs of words of the whole record. The mode of addition used is non-carrying binary (modulo 2).

This means that if we suppose the record to consist of segments A, B, C, ... each containing one pair of words, then the Ortho Words (OW) constitute the ultimate segment. Furthermore, the Ortho Words are numerically equal to the sum of the other segments. Thus:

$$OW = A + B + C + D + \ldots\ldots \text{ Modulo } 2$$

The Ortho Routine makes use of this relationship in the following way.

First, the Routine ascertains which pair of words contains the error by examining the Transverse Parity Checks. Let us say, pair C contains the error.

Next, the Routine replaces pair C by a pair of words C' which is computed as follows:

$$C' = OW - (A + B + D + \ldots\ldots) \text{ Modulo } 2$$

The error has now been eliminated.

Similarly, if the error had been found to be in word pair A, the Routine would substitute:

$$A' = OW - (B + C + D + \ldots\ldots) \text{ Modulo 2}$$

Although for convenience of explanation the message was divided up into pairs of words, the Ortho Routine will actually replace any twelve adjacent frames centered on the error.

As demonstrated above, the Ortho Routine will correct for multi-channel errors, however, most errors are found to be confined to a single channel. For this reason, the full Ortho Routine is designed to test the record first to ascertain whether the error is single channel or multi-channel. This test is performed as follows:

When the defective record has been transferred to memory, the Ortho Routine re-computes the two Ortho Words and compares them, bit for bit, with the two original Ortho Words which came in with the record. The positions of the discrepancies indicate which channel(s) contains the error(s). If more than one channel is affected, the Ortho Routine proceeds to correct in the manner described above. If, however, only a single channel is affected, the Routine simply complements all those bits in the affected channel which lie in frames containing Transverse Parity discrepancies. The record is now free from error.

HONEYWELL 1801-II

CENTRAL PROCESSOR

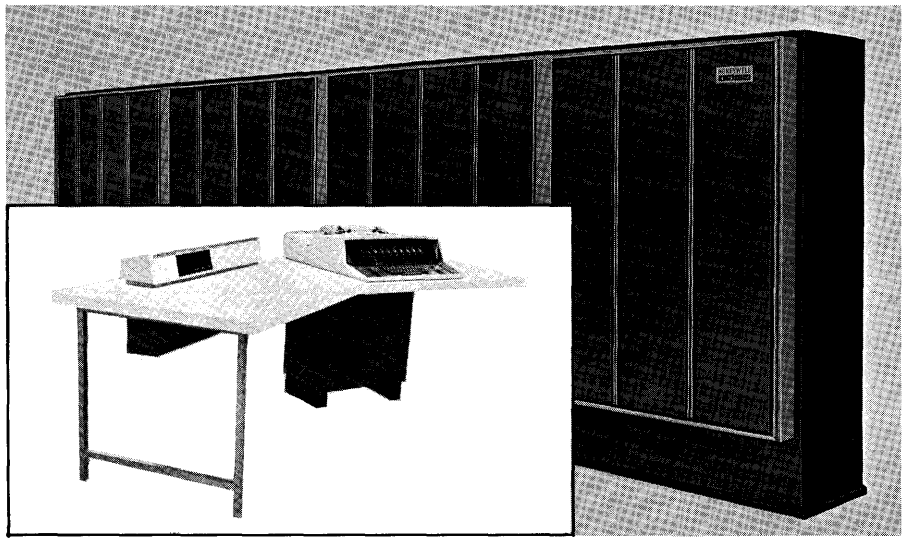
1801-II central processor consists of an arithmetic unit, a magnetic core memory, a control unit, a 256-special-register control memory, and an input/output control center. The arithmetic unit performs such operations as addition, multiplication, comparison, etc. The basic 8,192-word high-speed memory is expandable to a maximum of 32,768 words.

The 1801-II can process up to eight independent programs simultaneously. Time balance among active programs is monitored by the control unit to insure maximum operating efficiency. In addition, the control unit selects, interprets, and executes all instructions within the central processor. The execution of each independent program is controlled by one of eight identical groups of special registers in the control memory. These special registers control such activities as the sequential selection of instructions, word masking, indexed and indirect addressing, and entry to error-correcting routines.

The input/output control center controls card reading, card punching, printing, and magnetic tape operations related to input/output media conversion. One card reader-card punch, one high-speed printer, and up to four magnetic tape units can be connected to the input/output control center. A special buffer enables simultaneity of three peripheral operations within this center. The tape units connected to the input/output control center are used in independent input/output conversion operations. All other magnetic tape functions require the inclusion of an 803 tape control in the system. The card and printing equipment connected to the input/output control center can also communicate directly with main memory.

Additional peripheral devices may be connected to the central processor via standard Honeywell 800 series peripheral controls. Such devices can be operated independently of and simultaneously with the devices attached to the input/output control center.

The console is physically independent and provides for easy communication between the operator and the computer. By manually depressing keys on the console keyboard, the operator can start and stop the machine and can load and interrogate memory locations. A visual display of the status of the central processor and peripheral units is also provided. The console printer records all manual operations as well as programmed printouts on a continuous form, thus furnishing a complete log of every computer run.



OPERATING SPECIFICATIONS

WORD DEFINITION: 12 decimal digits, or 8 alphabetic characters, or 48 binary digits, or any combination of decimal digits, alphabetic characters, and binary digits totalling 48 bits. (Six parity-checking bits are appended to each word.)

MEMORY SIZE: Basic main memory—8,192 words (equal to 98,304 decimal digits or 65,536 alphabetic characters). Additional memory available in modules of 8,192 words (optional 1802 unit) up to a maximum of 32,768 words (equal to 393,216 decimal digits or 262,144 alphabetic characters). Control memory—Eight identical groups of 32 special registers. Each special register contains 16 information bits and two checking bits.

INTERNAL OPERATIONS: Binary and decimal arithmetic are standard. Floating-point arithmetic is optional and can be included by ordering the 1801-B floating-point option.

INPUT/OUTPUT: Eight pairs of general-purpose input/output trunks (of which one pair is connected to the input/output control center). One card reader-card punch, one high-speed printer, and 4 magnetic tape units can be connected to the input/output control center. Any standard Honeywell 800 series devices can be connected to the remaining general-purpose trunks.

MACHINE CYCLE TIME: 2 microseconds. (One machine cycle required to read 48 information bits and 6 parity bits from memory and restore cores to their previous states.)

SPECIAL FEATURES: Simultaneous processing of up to eight independent programs; on-line or off-line input/output conversion; simultaneous conversion, input, output, and computation; direct, indirect, and indexed addressing; and word masking.

OPERATING SPEEDS: Three-address binary addition and subtraction operations—120,000/sec. (equivalent to a speed of over 240,000 single-address instructions). Information transfer rate—420,000 words/sec.

CONSOLE: Keyboard with 64 keys—26 alphabetic, 10 numeric, 28 special symbols. Any of these characters may be typed in or printed out under program control. Console printer speed is 10-12 characters/sec.

PHYSICAL SPECIFICATIONS

CENTRAL PROCESSOR

UNIT AREA: 53 sq. ft.
DIMENSIONS: Width: 21'3"
Depth: 2'6"
Height: 6'1"
WEIGHT: 6,000 lbs.
COOLING REQ.: 26,500 BTU/hr.
POWER: 3.6 KVA

CONSOLE

UNIT AREA: 20 sq. ft.
DIMENSIONS: Width: 7'8"
Depth: 2'6"
Height: 3'0"
WEIGHT: 200 lbs.
COOLING REQ.: Included in central processor
POWER: Included in central processor

Honeywell
ELECTRONIC DATA PROCESSING
WELLESLEY HILLS 81, MASSACHUSETTS

DSI-158
5363
Litho in U.S.A.

HONEYWELL 1801-B

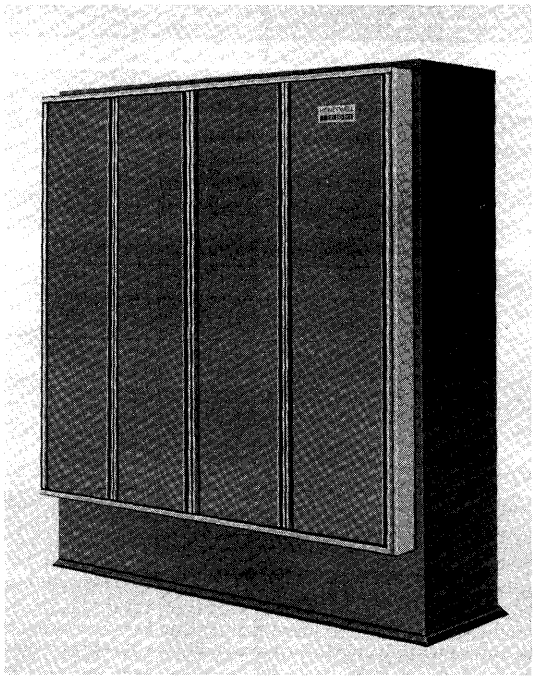
FLOATING-POINT OPTION

The 1801-B floating-point option allows the Honeywell 1800 System to carry out computations in binary and decimal floating-point arithmetic as well as in standard fixed-point arithmetic. It includes an arithmetic unit and a control unit (in addition to the arithmetic and control units located in the Honeywell 1800 central processor). The 1801-B arithmetic unit performs all floating-point instructions in accordance with the logical rules of the command codes. The 1801-B control unit selects, interprets, and directs the execution of these instructions.

To insure accuracy, the 1801-B floating-point option incorporates an extensive internal checking network which tests all data received or transmitted.

The wide range of values which can be represented by the floating-point configuration of the Honeywell 1800 word minimizes the possibility of underflow and overflow during most arithmetical computations. The values of the floating decimal word may range from 10^{-65} virtually up to 10^{63}. The 10-digit precision of the mantissa often eliminates the need for time-consuming multiple-precision computations.

The floating binary word in the Honeywell 1800 provides even greater range and precision than the floating decimal word. Its values may range approximately from 10^{-78} up to 10^{76}. The mantissa has 40 binary places, the equivalent of over 12 decimal digits.



OPERATING SPECIFICATIONS

INTERNAL OPERATIONS: Binary and decimal arithmetic.

WORD DEFINITION: 1 sign bit, 7 exponent bits, and 40 mantissa bits.

PHYSICAL SPECIFICATIONS

UNIT AREA: 15 sq. ft.

DIMENSIONS: Width 5'10"
Depth 2'6"
Height 6'1"

WEIGHT: 1,600 lbs.

COOLING REQ: 9,600 BTU/hr.

POWER: 2.5 KVA

BIT STRUCTURE OF THE FLOATING-POINT WORD

Bit	1	2 — 8	9 — 48
Decimal	±	Exponent (7 bits)	Mantissa (10 decimal digits)
Binary	±	Exponent (7 bits)	Mantissa (40 binary digits— equal to over 12 decimal digits)

48 BITS

MINNEAPOLIS-HONEYWELL REGULATOR CO.
ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

DSI-116
5262
Litho in U.S.A.

HONEYWELL

1802

ADDITIONAL MEMORY BLOCK

The basic high-speed memory contained in the 1801 or 1801-II central processor consists of 8,192 words, which is equivalent to 98,304 decimal digits or 65,536 alphabetic characters. This same memory capacity is also contained in the 1802 additional memory block. As shown in the table, up to three additional memory blocks can be added to a Honeywell 1800 or 1800-II system, increasing the high-speed memory to a maximum of 32,768 words. This in turn is equivalent to a maximum of 393,216 decimal digits or 262,144 alphabetic characters.

Equipment Complement	Words	Dec. Digits	Alpha. Characters
Basic System Memory	8,192	98,304	65,536
1 1802 Module Added	16,384	196,608	131,072
2 1802 Modules Added	24,576	294,912	196,608
3 1802 Modules Added	32,768	393,216	262,144

The Honeywell 1800 word is composed of 48 information bits plus six parity bits. It can represent 12 unsigned decimal digits, 11 decimal digits with sign, eight alphabetic characters, 48 binary digits, or any combination of decimal digits, alphabetic characters, and binary digits totalling 48 bits. It can also represent up to four separate signed numeric fields with an aggregate size of 44 bits. The six parity bits are used to verify the correctness of information transfers.

OPERATING SPECIFICATIONS

WORD SIZE: 12 decimal digits, 8 alphabetic characters, 48 binary digits, or any combination of decimal digits, alphabetic characters, and binary digits totalling 48 bits.

MEMORY CYCLE: 2 microseconds (the time required to read 48 information bits and six parity bits from memory and restore cores to their previous states).

PHYSICAL SPECIFICATIONS

The physical specifications for 1802 additional memory blocks depend not only upon the number of blocks to be added, but also upon the amount of memory previously contained in the existing system. Specifications for particular combinations will be quoted on request.

Honeywell
ELECTRONIC DATA PROCESSING
WELLESLEY HILLS 81, MASSACHUSETTS

DSI-180
5563
Litho in U.S.A.

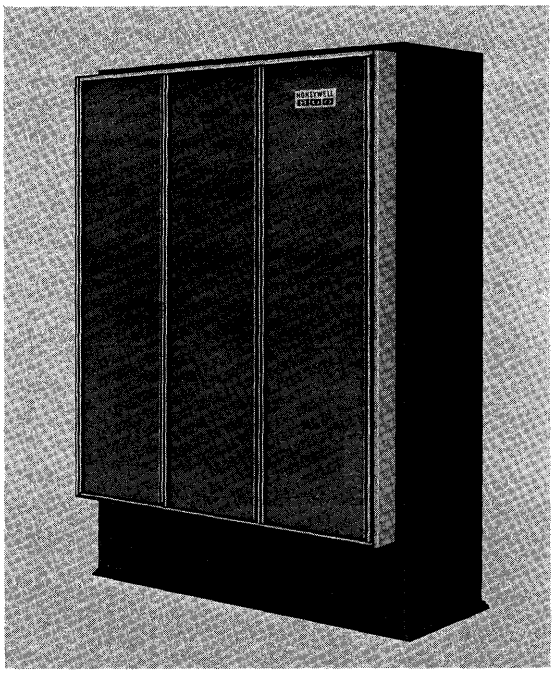
HONEYWELL 803-3

ECONOMY TAPE CONTROL

The 803-3 economy tape control links the 804-3 economy magnetic tape units with the Honeywell 800 or 1800 central processor, and handles the flow of information between them in both directions.

Buffers in the tape control reconcile the electronic speeds of the central processor and the mechanical speed of the magnetic tape units. Information is transferred between the buffers and the central processor at central processor speeds, one word at a time. When this high-speed transfer (in either direction) of a word is complete, the central processor is free to perform other data processing operations until the start of the next high-speed transfer. Automatic parity checking assures the accuracy of all information transferred.

Eight 804-3 magnetic tape units can be connected to each 803-3 tape control. Each tape control is connected to the central processor via one input and one output trunk.



Addresses of individual tape units are specified by a simple tape-address patchboard associated with the tape control.

Information transfer between each tape control and its associated magnetic tape units is independent of and simultaneous with such transfer for all other units.

OPERATING SPECIFICATIONS

Up to eight magnetic tape units can be connected to a tape control. The control permits simultaneous reading from any one of these tape units and writing on any other.

PHYSICAL SPECIFICATIONS

UNIT AREA: 11 sq. ft.

DIMENSIONS: Width 4′3″
Depth 2′6″
Height 6′1″

WEIGHT: 1,200 lbs.

COOLING REQ: 12,325 BTU/hr.

POWER: 1.3 KVA

Honeywell
ELECTRONIC DATA PROCESSING
WELLESLEY HILLS 81, MASSACHUSETTS

DSI-174
5463
Litho in U.S.A.

HONEYWELL

804-3

ECONOMY MAGNETIC TAPE UNIT

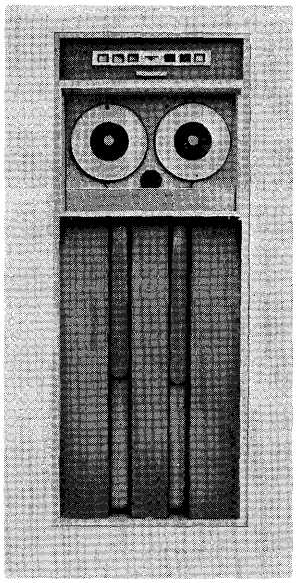
The 804-3 economy magnetic tape unit records on and reads data from ¾" wide magnetic tape moving at a speed of 60" per second. The standard recording density of 400 bits per inch yields an instantaneous transfer rate of 48,000 decimal digits per second.

In on-line use, the 804-3 is connected to an 803-3 economy tape control and operates under direct control of a Honeywell 800 or 1800 central processor. When used off-line, it is connected to a printer or card device via the appropriate off-line auxiliary control and peripheral control. At installations which include more than one Honeywell system, the 804-3 magnetic tape unit may be switched to either system via an 805 magnetic tape switching unit.

The vacuum methods used in mounting, driving, and stopping the tape avoid the danger of damage to tape inherent in the use of pinch rollers and mechanical techniques. The reading surface of the moving tape is protected against abrasion, since its only physical contact is with the read-write head.

Information on tape is doubly protected against accidental destruction: to permit recording a metal file-protection ring must be in place, and a switch on the tape unit panel must be set to PERMIT.

Outstanding automatic error detection and correction techniques, including the exclusive Honeywell Orthotronic Control, are provided to insure accuracy of recorded data.



OPERATING SPECIFICATIONS

RECORDING DENSITY: 400 bits per inch.

CHANNELS: Eight information channels, one parity channel, and one clocking channel.

TRANSFER RATE: 48,000 decimal digits/sec. (equal to 4,000 words/sec.).

TAPE MOTIONS: Reading in either direction—Writing in forward direction—Fast Rewind.

TAPE SPEED: Reading and writing at 60"/sec.—Rewinding at 180"/sec.

RECORD LENGTH: Variable.

INTER-RECORD GAP: 0.67"

DATA RELIABILITY: Parity checking plus the inclusion of two orthotronic words—logical combinations of all the data words in a record—with every record on magnetic tape for automatic error detection and correction.

REEL CAPACITY: 2,500 feet of tape.

TAPE CHARACTERISTICS: ¾" tape with Mylar base and oxide coating.

PHYSICAL SPECIFICATIONS

UNIT AREA: 6 sq. ft.

DIMENSIONS: Width 2'4"
Depth 2'4"
Height 5'8"

WEIGHT: 1,250 lbs.

COOLING REQ: 8,700 BTU/hr.

POWER: 2.8 KVA

Honeywell
ELECTRONIC DATA PROCESSING
WELLESLEY HILLS 81, MASSACHUSETTS

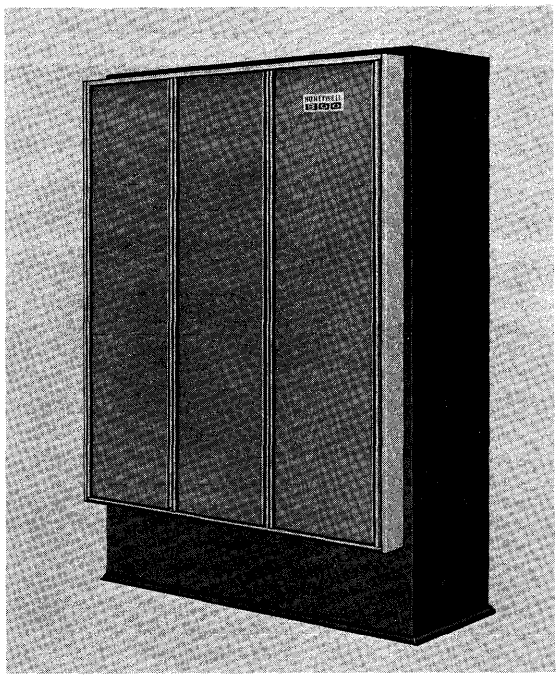
DSI-175
5463
Litho in U.S.A.

HONEYWELL

803-4

TAPE CONTROL

The 803-4 tape control links the central processor and the 804-4 magnetic tape units, and handles the flow of information between them in both directions.



Buffers in the control unit reconcile the electronic speeds of the central processor and the mechanical speed of the magnetic tape units. Information is transferred between the buffers and the central processor at central processor speeds, one word at a time. When this high-speed transfer (in either direction) of a word is complete, the central processor is free to perform other data processing operations until the start of the next high-speed transfer. Automatic parity checking assures the accuracy of all information transferred.

Eight 804-4 magnetic tape units can be connected to each 803-4 tape control. Each tape control unit is connected to the central processor via one input and one output trunk.

Addresses of individual tape units are specified by a simple tape-address patchboard associated with the tape control.

OPERATING SPECIFICATIONS

Up to eight magnetic tape units can be connected to a tape control unit. The control unit permits simultaneous reading from any one of these tape units and writing on any other.

Information transfer between each control unit and its associated magnetic tape units is independent of and simultaneous with such transfer for all other units.

PHYSICAL SPECIFICATIONS

UNIT AREA: 10 sq. ft.

DIMENSIONS: Width 4'3"
Depth 2'6"
Height 6'1"

WEIGHT: 1,200 lbs.

COOLING REQ: 6,840 BTU/hr.

POWER: 2.0 KVA

MINNEAPOLIS-HONEYWELL REGULATOR CO.
ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

DSI-118
5462
Litho in U.S.A.

HONEYWELL

804-4

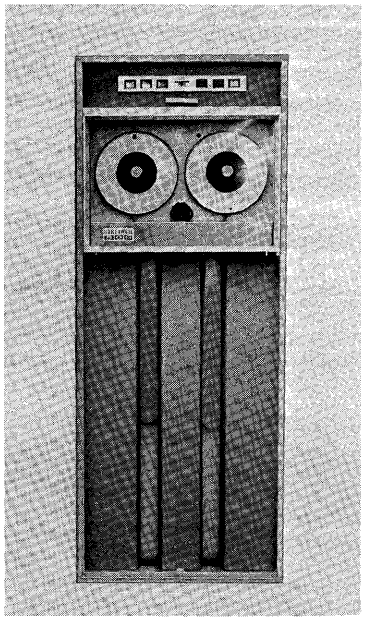
MAGNETIC TAPE UNIT

The 804-4 magnetic tape unit records on and reads data from ¾" wide magnetic tape moving at a speed of 120" per second. It is connected to the 803-4 tape control and operates under direct central processor control. The 804-4 cannot operate with an off-line auxiliary control.

The vacuum methods used in mounting, driving, and stopping the tape avoid the danger of damage to tape inherent in the use of pinch-rollers and mechanical techniques. The reading surface of the moving tape is protected against abrasion since its only physical contact is with the read-write head.

Information on magnetic tape is doubly protected against accidental destruction: to permit recording, a metal file protection ring must be in place and a switch on the tape unit panel must be set to PERMIT.

Outstanding automatic error detection and correction techniques, including the exclusive Honeywell Orthotronic Control, are provided to insure accuracy of recorded data.



OPERATING SPECIFICATIONS

TRANSFER RATE: 186,000 decimal digits/sec. (Equal to 15,500 words/sec.)

READING: In both directions.

WRITING: In forward direction.

TAPE SPEED: Reading and writing at 120"/sec. Rewinding at 360"/sec.

RECORD LENGTH: Variable.

INTER-RECORD GAP: 0.67"

DATA RELIABILITY: Automatic error detection and correction is provided by the use of two orthotronic words — logical combinations of all the words in a record — which accompany every record on tape.

Additional data checking is provided by a parity check bit for each frame (i.e., one array of information bits across the tape).

REEL CAPACITY: 2,400 feet of tape.

TAPE CHARACTERISTICS: ¾" tape, with Mylar base and oxide coating.

PHYSICAL SPECIFICATIONS

UNIT AREA: 5 sq. ft.

DIMENSIONS: Width 2'3"
Depth 2'5"
Height 5'8"

WEIGHT: 1,250 lbs.

COOLING REQ: 7,650 BTU/hr.

POWER: 2.8 KVA

MINNEAPOLIS-HONEYWELL REGULATOR CO.
ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

DSI 120
5462
Litho in U.S.A.

HONEYWELL

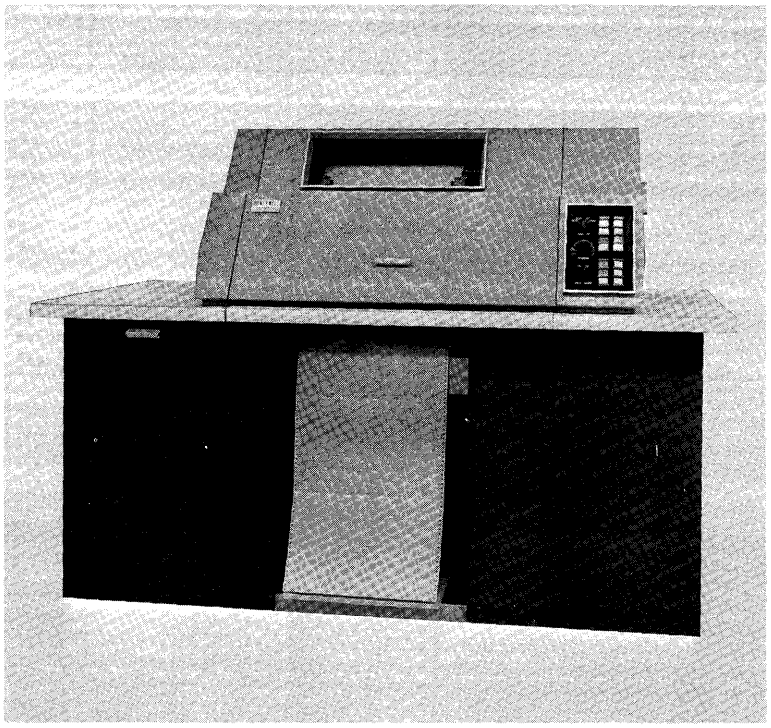
822-3

HIGH-SPEED PRINTER

The 822-3 high-speed printer operates at a speed of 900 lines per minute. Depending on paper stock being used, it produces eight to ten clear carbon copies. The printer may be used both on-line and off-line. When used on-line (i.e., directly connected to the central processor) it is under control of the 806-3 printer control or an 811-3 multiple control. In off-line operation (i.e., not directly connected to the central processor) the printer is connected to an 804 magnetic tape unit via its own control unit and an off-line auxiliary control unit.

For every line to be printed, the printer control unit receives 16 Honeywell words and decodes them into the pulses necessary to drive the appropriate print hammers. The first word is a control word which specifies the vertical format of the printout; the other 15 words specify the 120 characters to be printed in the line. A total of 160 print positions are available; of these, the 120 active positions for any given print run are chosen by plugboard wiring. Fifty-six characters are available at each print position.

Accuracy of printed data is verified by echo checking of pulses sent to the printer.



OPERATING SPECIFICATIONS

SPEED: 900 lines/min. for single spacing. 800 lines/min. for double spacing.

SPACING: Horizontal—10 columns/inch. Vertical—6 lines/inch. (Option 822-3A provides manual switch selection of 6 lines/inch or 8 lines/inch spacing.)

SKIP SPEED: 20 inches/sec. in non-printing mode.

PRINT FORMAT: 160 possible print positions. 120 active for a given run.

CHARACTER ARRAY: 56 characters available at each print position. (26 alpha, 10 numeric, 20 special.)

STOCK WIDTH: 3½" to 22" paper, edge to edge.

STOCK WEIGHT: From 11 lb. single-part form to 125 lb. continuous card stock. Adjustment for stock thickness is provided.

ADJUSTMENT OF FORM: Horizontal and vertical vernier adjustment.

SPECIAL FEATURES: Two-part heat-transfer forms and offset master forms acceptable.

PHYSICAL SPECIFICATIONS

UNIT AREA: 16 sq. ft.

DIMENSIONS: Width 6'1"
Depth 2'7"
Height 4'8"

WEIGHT: 1,000 lbs.

COOLING REQ: 4,918 BTU/hr.

POWER: 1.8 KVA

DSI-61
5361
51261
Litho in U.S.A.

HONEYWELL 860 SERIES

RANDOM ACCESS STORAGE AND CONTROLS

The 860 series of random access storage files consists of nine models of rapid-access, magnetic disc memory, along with associated control and auxiliary units necessary for operation. Each model provides large-capacity auxiliary storage for the Honeywell 800 or 1800 Electronic Data Processing System, ranging from a nominal capacity of 50 million alphanumeric characters in the model 860-1 up to 800 million in the model 860-9. This corresponds to the range from 75 million to 1.2 billion decimal digits. Starting with the 100-million-character model (860-2) the storage capability increases in increments of approximately 100 million characters for each larger model. Exact storage capacities are given on the reverse side.

To facilitate expansion of a disc-storage system, all models of the 860 series employ modular construction. Modules may be conveniently added as larger capacity is needed. Each system consists of three basic components: (1) one or more storage modules, (2) one or more auxiliary cabinets (equal to the number of storage modules), and (3) a control unit.

For capacities greater than 100 million characters, one additional 24-disc module is added to the basic 100-million-character file (860-2) for each desired increment of 100 million characters. To obtain a capacity of 800 million characters, for example, the model 860-9 employs eight 24-disc modules. Associated with each 24-disc module is a two-bay auxiliary cabinet, incorporating read, write, and control circuits. A one-bay cabinet is used with the 12-disc model (860-1).

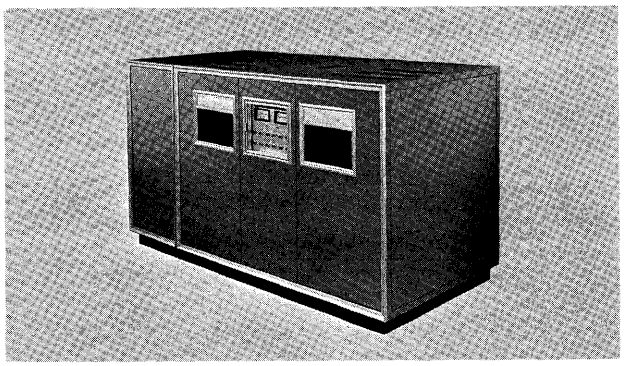
The control unit provides buffer storage, performs accuracy checks, and regulates searching, reading, recording, disc switching, head selection, and input voltages. Models 860-1 and 860-2 use a three-bay control. To the same basic unit a fourth bay is added for models 860-3 and larger. A summary of the components contained in each model of the series is given on the reverse side.

The 12 or 24 discs in a storage module rotate continuously on a common horizontal axis at a speed of 900 rpm (one revolution per 67 milliseconds). Both surfaces of each magnetic disc are used to record data.

Disc Surface Arrangement

A disc surface is divided into six concentric recording zones, each containing 128 concentric data tracks, for a total of 768 tracks per face. Each zone passes under its own read/write head, which can be positioned to any of the 128 tracks in the zone. (All read/write heads are simultaneously positioned to the corresponding tracks in their respective zones.) Data flow is serial, bit by bit, from or to a particular track, but is parallel with respect to a word of information because of the way a word is arranged for storage.

Information stored on a disc surface is arranged in the shape of a wedge which cuts across the six zones. This arrangement permits a constant transfer rate, regardless of the zone involved. Every word of 48 information bits and 6 parity check bits that is to be stored in the file is divided into two parts: One part is recorded in one zone, while the other is recorded on the corresponding track in a different but associated zone (in the same wedge portion). Zone 6, the outer zone, is associated with zone 1, the inner zone to form read-write area I. Zone 5 is associated with zone 2 to form read-write area II. And the two central zones, 3 and 4, associate to form read-write area III. The read/write heads for two associated zones are always energized simultaneously.



Twenty-four Disc Storage Module

The division of a word into two parts is done on the following basis: For reading or writing in area I, 40 bits of the word are in zone 6, and 14 bits are on the corresponding track in zone 1. For reading or writing in area II, 36 bits are in zone 5 and 18 in zone 2. In area III, the word is evenly divided, with 27 bits in zone 3 and 27 in zone 4.

From the programmer's point of view, a disc face is subdivided into 32 uniquely addressable sectors, each capable of storing 128 sixty-four word records. On a single surface 4,096 records can be stored; this is equivalent to 8,192 per disc or 196,608 per 24-disc module. Read-write areas I and II each contain 11 sectors, and area III contains 10. A read or write instruction addressed to a disc file transfers one 64-word record from or to the file at a constant rate of 5.76 milliseconds, regardless of the read-write area involved.

Access Time

The access time to any record in the file includes disc-latency, address-verification, and head-positioning times. Disc latency varies from 0 to 67 milliseconds with an average of 34 milliseconds (one half of a disc revolution). Head positioning and address verification require a minimum of 60 milliseconds, assuming that the heads have to be repositioned. Maximum head-positioning times (including address verification) are shown on the reverse side.

Continuous Addressing

In all models of the 860 series, a large number of records is available per search instruction without head repositioning. Records lying on the same relative track for all sectors on all surfaces can be continuously addressed. Thus, the more discs there are in the file, the greater the number of records available per search (see table). Up to 1,536 records per 24-disc module can be processed sequentially or otherwise for each head positioning. No more than 128 head positionings are required to go from the lowest address in a file to the highest.

Parallel Processing

All random access operations, including file searching and data transfer, can proceed in parallel with other operations. The control unit uses one pair of central processor input/output trunks. Other devices may be connected to other input/output trunks without restriction. In systems containing two or more 24-disc modules (860-3 and larger), it is possible to read from one disc file, write concurrently in a second, and simultaneously position read/write heads in any other disc modules.

Data Reliability

In addition to the storage discs, every file unit contains a non-data disc, of which one face is used for clocking tracks, the other for sector address verification. The accuracy of all data read from a disc file is verified by the use of six parity bits appended to every 48-bit word recorded in the file. Moreover, when Honeywell's Orthotronic Control technique is employed, two words of each record are used as orthowords for automatic error correction, if necessary. Error detection is performed in the control unit; automatic correction in the central processor.

Honeywell
ELECTRONIC DATA PROCESSING
WELLESLEY HILLS 81, MASSACHUSETTS

DSI-161
5463
Litho in U.S.A.

OPERATING SPECIFICATIONS

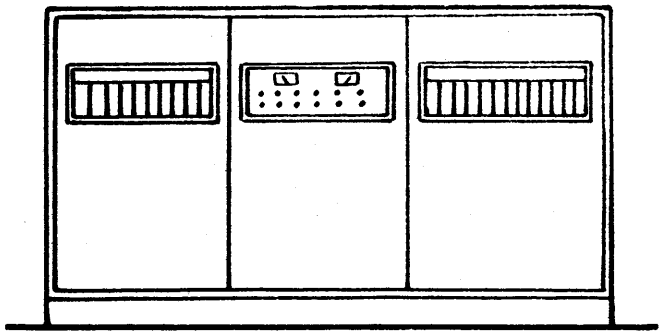
Operation: On-line
Disc Rotation Speed: 900 rpm (1 revolution per 67 milliseconds)
Recording Surfaces: 2 per disc
Concentric Recording Zones: 6 per surface
Concentric Data Tracks: 128 per zone; 768 per surface
Records Stored per Surface: 4,096
Words per Record: 64
Bits per Word: 54 (48 information and 6 parity)
Associated Zones:
1 and 6 (inner and outer) = Read-Write Area I
2 and 5 = Read-Write Area II
3 and 4 (central) = Read-Write Area III
Division of Words for Storage:
Read-Write Area I—14 bits in zone 1, 40 in zone 6
Read-Write Area II—18 bits in zone 2, 36 in zone 5
Read-Write Area III—27 bits in zone 3, 27 in zone 4
Programming Sectors: 32
Sectors per Read-Write Area:
Areas I and II—11 each
Area III—10
Tracks per Sector: 128 pairs of corresponding tracks—one track of each pair in a different, but associated, zone
Records per Sector: 128
Read/Write Heads:
For data—1 per zone; positionable to any of 128 tracks
For track verification—1 per disc module; positionable to any of 128 tracks
For clocking—2 duplicate sets of 7 fixed-position heads per disc module
Access Time: 60 milliseconds minimum; includes disc-latency, address-verification, and head-positioning times
Disc Latency Time: 0 to 67 milliseconds; 34 milliseconds, average
Range of Maximum Head-Positioning Time:
For 12-disc module—110 to 130 milliseconds
For 24-disc module—140 to 160 milliseconds
Note: Head positioning time is effectively zero if heads do not have to be repositioned in a search order.
Transfer Rate: 5.76 milliseconds per record; 10,250 words read or written per second; rate is constant, regardless of transferring zone.

	860-1	860-2	860-3	860-4	860-5	860-6	860-7	860-8	860-9
Nominal Capacity (Alphanumeric Characters)	50 million	100 million	200 million	300 million	400 million	500 million	600 million	700 million	800 million
Actual Capacity (Alphanumeric Characters)	50,331,648	100,663,296	201,326,592	301,989,888	402,653,184	503,316,480	603,979,776	704,643,072	805,306,368
Actual Capacity (Decimal Digits)	75,497,472	150,994,944	301,989,888	452,984,832	603,979,776	754,974,720	905,969,664	1,056,964,608	1,207,959,552
Data Capacity (Alphanumeric) Using 2 Words/Record for Orthowords	48,758,784	97,517,568	195,035,136	292,552,704	390,070,272	487,587,840	585,105,408	682,622,976	780,140,544
Capacity Expressed in No. of 64-Word Records	98,304	196,608	393,216	589,824	786,432	983,040	1,179,648	1,376,256	1,572,864
Maximum No. of Records/Search	768	1,536	3,072	4,068	6,144	7,680	9,216	10,752	12,288

PHYSICAL SPECIFICATIONS

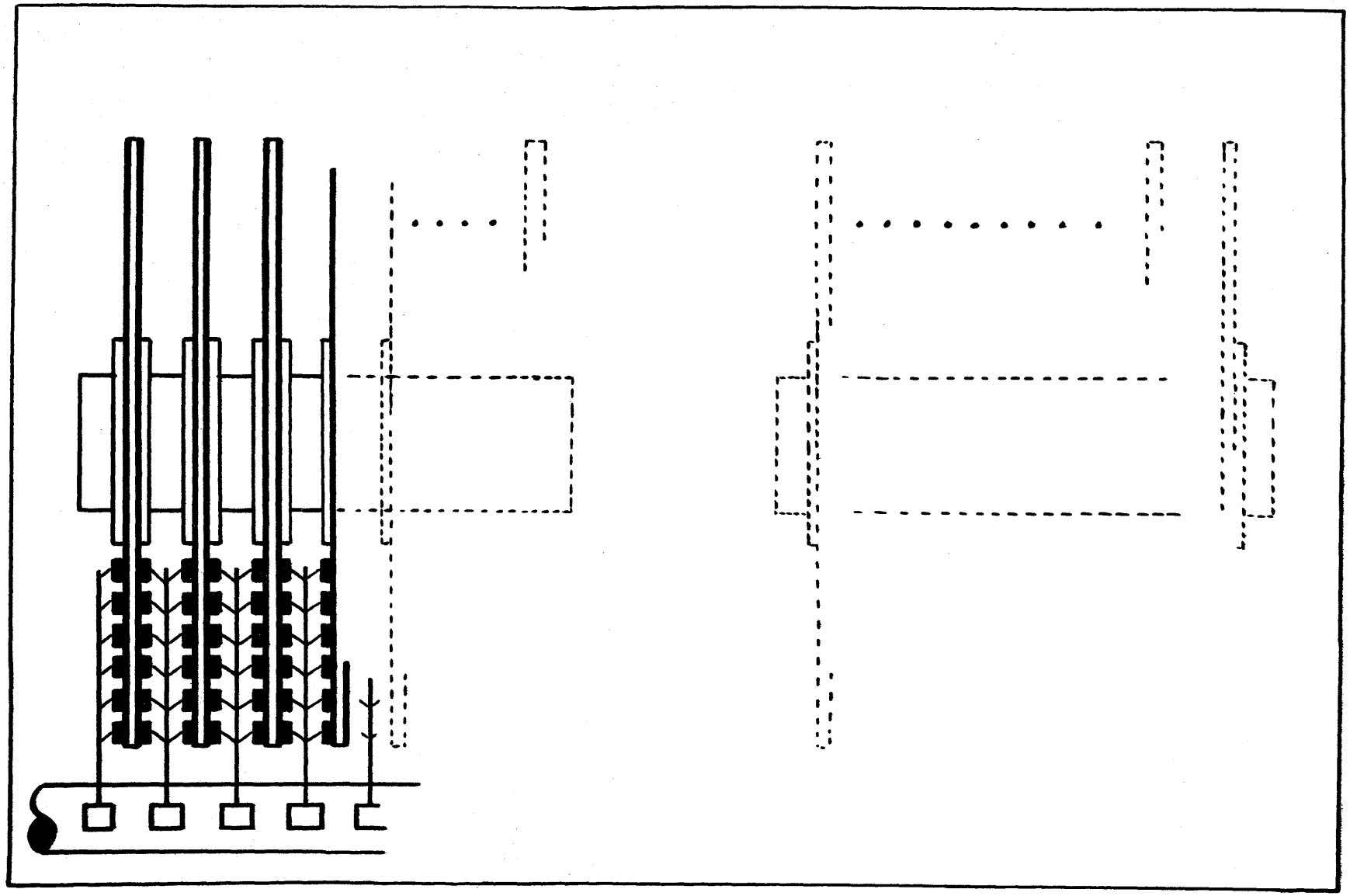
No. of Storage Discs	12	24	48	72	96	120	144	168	192
No. of 24-Disc Modules	½	1	2	3	4	5	6	7	8
No. of Two-Bay Auxiliary Cabinets	½	1	2	3	4	5	6	7	8
No. of Control Units	1 (Three Bays)	1 (Three Bays)	1 (Four Bays)	1 (Four Bays)	1 (Four Bays)	1 (Four Bays)	1 (Four Bays)	1 (Four Bays)	1 (Four Bays)
Dimensions of Disc-File Module	Width: 5′10″ Depth: 3′8″ Height: 4′4″	Width: 5′10″ Depth: 3′8″ Height: 4′4″	Two 860-2 Disc-File Modules	Three 860-2 Disc-File Modules	Four 860-2 Disc-File Modules	Five 860-2 Disc-File Modules	Six 860-2 Disc-File Modules	Seven 860-2 Disc-File Modules	Eight 860-2 Disc-File Modules
Dimensions of Auxiliary Cabinet	Width: 1′10″ Depth: 3′8″ Height: 4′4″	Width: 3′8″ Depth: 3′8″ Height: 4′4″	Two 860-2 Auxiliary Cabinets	Three 860-2 Auxiliary Cabinets	Four 860-2 Auxiliary Cabinets	Five 860-2 Auxiliary Cabinets	Six 860-2 Auxiliary Cabinets	Seven 860-2 Auxiliary Cabinets	Eight 860-2 Auxiliary Cabinets
Dimensions of Control Unit	Width: 4′3″ Depth: 2′6″ Height: 6′1″	Width: 4′3″ Depth: 2′6″ Height: 6′1″	Width: 5′10″ Depth: 2′6″ Height: 6′1″	Same as for 860-3	Same as for 860-3	Same as for 860-3	Same as for 860-3	Same as for 860-3	Same as for 860-3
Weights (lbs.): Per Disc-File Module	2,500	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000
Per Auxiliary Cabinet	375	750	750	750	750	750	750	750	750
Control Unit	1,200	1,200	1,600	1,600	1,600	1,600	1,600	1,600	1,600
Total	4,075	5,950	11,100	15,850	20,600	23,350	30,100	34,850	39,600
Input Voltage	208 Volts, 60 Cycles per Second, 3 Phases (All Models) from Power Central								
Total Power Requirements	8.25 KVA	8.25 KVA	15.25 KVA	22.25 KVA	29.25 KVA	36.25 KVA	43.25 KVA	50.25 KVA	57.25 KVA
Total Cooling Requirements (BTU/hr.)	28,180	28,180	50,060	68,840	87,620	106,400	125,180	143,960	162,740

RANDOM ACCESS STORAGE CAPACITIES



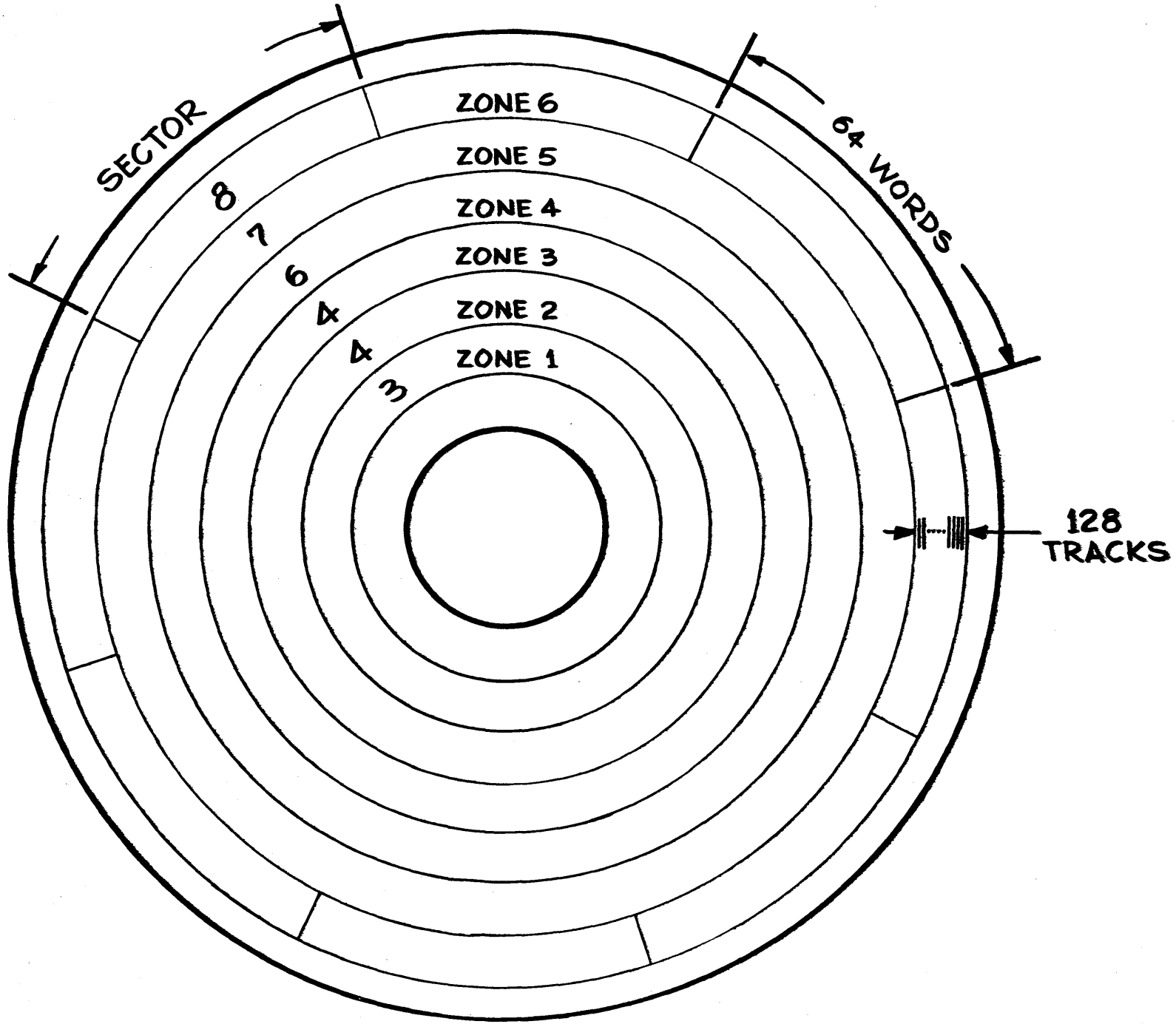
MODEL	NO. DISCS	STORAGE CAPACITY	
		NO. ALPHA. CHARS.	NO. DECIMAL DIGITS
860-1	12	50,331,648	75,497,472
860-2	24	100,663,296	150,994,944
860-3	48	201,326,592	301,989,888
860-4	72	301,989,888	452,984,832
860-5	96	402,653,184	603,979,776
860-6	120	503,316,480	754,974,720
860-7	144	603,979,776	905,969,664
860-8	168	704,643,072	1,056,964,608
860-9	192	805,306,368	1,207,959,552

RANDOM ACCESS
READ/WRITE HEADS



TOP VIEW

RANDOM ACCESS
LAYOUT OF INFORMATION ON DISCS



H-1800-II

SOFTWARE

AUTOMATH 1800

A. THE HONEYWELL APPROACH

Today, the most impressive technological advances in computer design tend to emphasize faster hardware speeds. However, equal consideration should be given to software and operating techniques. It is important to keep the "total operating system" in mind when considering overall productivity of a computer system.

Consequently, Honeywell E.D.P. has significantly reduced the time it takes to get the results of a computer run in the hands of the user. This accomplishment has been achieved by the effective use of Honeywell's unique hardware feature called parallel processing, optimization of object code, and systems design.

With the Honeywell 1800 AUTOMATH System, the engineer or scientist can receive his results in the same order of time that it takes to compile and execute his program and print out his results. Conventional Fortran systems, however, break down into three separate operations. First, all programs to be run are converted to tape on a satellite computer. Second, all programs are compiled and executed on the main frame. Thirdly, all results are printed and binary decks punched on the satellite computer. Thus, a user may wait hours even though his program could execute in seconds.

This is not the case with the Honeywell 1800 AUTOMATH System becuase these same three processes are carried out in parallel.

Two other significant features of Honeywell's system are optimization which contributes to the reduction in "turn around" time (the time between giving the input to a run until getting the results back from the run) and the overlay process which allows programs to be run on the H-1800 that could not be run

by other machines of the same memory size. The overlay process also conserves memory so that more programs can be executed in parallel than would ordinarily be possible.

With these few preliminary remarks, Honeywell welcomes the opportunity to discuss the AUTOMATH 1800 system and other prominent features of the H-1800-II.

B. AUTOMATH - THE H-1800-II FORTRAN IV SYSTEM

Automath 1800, an extremely fast compiler, (2 passes) is combined with an operating system that fully utilizes the parallel processing capabilities built into the hardware. The result is a substantial reduction in "turn-around" time. Turn-around time is minimized because input, output and central processor operations proceed in parallel. Using two input tapes and two output tapes, Automath 1800 can attain almost continuous input and output from the central processor. While one input tape is feeding the central processor for compilation and execution, the other tape is receiving input from the card reader. Similarly, while one output tape is receiving results from the central processor, the other output tape is feeding its contents to the printer. Thus operations, which would be carried out serially by conventional methods, is carried out on the H-1800 in parallel. In the Honeywell 1800 system, the operator merely stacks any new programs to be run in the card reader without any interruption of the processing already in progress.

Automath H-1800 compiles in only 16,000 words of memory. Competing systems use 32,000 words to compile FORTRAN IV programs. Thus, with half the memory the Honeywell system is surpassing the turn-around performance of competing systems and still allows the remaining 16,000 words of a 32,000 word system for parallel processing of other programs.

The outstanding weaknesses of competitive systems have been eliminated by the Honeywell 1800 Fortran IV system. In conventional systems, core storage for non-common arrays is allocated at program loading time. This is extremely wasteful. The data in non-common array storage is significant only to the subprogram defining them. Thus, if there were ten (10) subprograms, memory for the non-common arrays for all ten subprograms would be set aside at program loading time.

The AUTOMATH 1800 monitor does not allocate any non-common array storage until the subprogram defining the particular non-common array is entered. This is called dynamic allocation. The importance of dynamic allocation is twofold. First, it increases the ability to do parallel processing by not allowing any program to tie up an excessive amount of core. Second, it enables the execution of very large programs that could not otherwise be run, even if allowed the total core.

Another outstanding feature of Automath 1800 is the overlay technique for conserving memory space. This involves the "writing over" certain invariant portions of a subprogram that is not being actively executed, when the space is required by other active subprograms. The overlaid portion is then called back into memory from the program tape at the time the inactive program becomes active again.

At initial loading time, as many subprograms as possible are loaded into the available memory. As control passes to each new subprogram, a rapid check is made in the status table to determine if the program is already in memory. If it is not, the monitor determines a place for it to be loaded by first attempting to use the memory assigned to subprograms that are presently inactive. Failing to find sufficient space among the inactive subprograms, the monitor then overlays the instruction portions of the ancestors to the subprogram being called, since these portions are invariant and can be easily restored from tape if later required. (Ancestors refer to those subprograms that preceded the calling of the present one i.e. Subprogram A calls subprogram B which calls the current subprogram C. The ancestors of subprogram C are A and B in that order.)

It should be emphasized that the actual internal execution time to perform overlaying and dynamic relocation is just a few milliseconds in the most involved case. Furthermore, the

monitor performs dynamic relocations in parallel with any tape operation required to locate and load a subprogram not in memory.

It should also be emphasized that the programmer may write his programs without any consideration of the overlay process. No special control card or grouping considerations are necessary. The internal monitor automatically provides the correct amount of memory to meet the program's needs at each point of execution. Provision is also made for the programmer to designate a program as a "high-duty" program that must never be overlaid.

Turn around time is also reduced proportionately to the amount of optimization performed by the particular compiler. While certain more advanced compilers for competitive machines attempt optimization within a single statement, AUTOMATH 1800 goes much further by analyzing all statements in an entire region of statements and optimizing over the entire region. (A region is a sequence of statements extending from a point where control may enter the sequence up to the next control statement.) A large amount of time consuming repetition is thus eliminated.

For example competitive compilers might recognize that each of the subexpressions (A + B) and (D + E) appears twice in the formula: Y=Cos (A + B + C + D + E) + (B + A) **2 + (E + D) *3. Not only will Automath 1800 recognize the duplication and compute the subexpressions only once, but it will also recognize the same subexpressions and their equivalent every time they appear again in later statements in the same region.

For example, consider the region of five statements shown in the following:

```
300   A = Cos (A + B + C + D + E) + (B + A)**2. + (E + D)*3.
      Q = F + G
302   R = H + B + A
303   S = F* SIN (E + B + A + D)
      Go to 500
```

The compiler would produce the following:

```
300   Temp 1 = A + B
      Temp 2 = D + E
      Temp 3 = Temp 1 + Temp 2
      A = Cos (Temp 3 + C) + Temp 1 **2. + Temp 2 *3.
      Q = F + G
      R = H + Temp 1
      S = F* SIN (Temp 3)
      Go to 500
```

Another type of optimization performed is the resequencing of computations contained within a DO loop to positions outside the DO loop, whenever possible. The savings are especially significant if the DO loop has a large number of iterations. Furthermore, when the compiler encounters a nest of DO loops (i.e. one loop within another), it determines the level of definition of each subexpression and resequences the computation to the outermost level possible, minimizing redundant repetitions.

An example of statement resequencing is given in Figure 1. In figure 1A, two DO loops are nested one within the other, such that for each iteration of the outer (I) loop there are 100 iterations of the inner (J) loop. Since there are also 100 iterations of the outer loop, an expression in the inner loop would be repetitively calculated 10,000 times by the conventional compiler.

AUTOMATH 1800, however, by first determining the level of definition of each subexpression, recognizes that the expressions (A = R* S **T) and (A**T) consist of variables that are unaltered in the DO loop nest, and it resequences them outside of both loops as shown in Figure 1B, saving 9,999 repetitious calculations of each, because they need only be computed once. It also recognizes that the formula F(I) = G(2*I) + H(I,I) is independent of variable J and can thus be resequenced to a position outside of the J loop, as shown in Figure 1B. Thus, the formula is evaluated 100 times instead of 10,000 times. (The statement B = B +1 cannot be resequenced since its value changes with each iteration of the T loop.)

FIGURE 1

Figure 1A Before Processing

```
DO 10 I = 1,100

DO 20 J = 1,100
   A=R*S**T
   B=B+1
   C=B+A**T
   F(I) = G(2*I) + H(I,I)
20
10
```

Resequenced to:

Figure 1B After Resequencing

```
A=R*S**T
Temp = A**T
DO 10 I = 1,100
   F(I)=G(2*I)+H(I,I)

DO 20 J = 1,100
   B=B+1
   C=B+Temp
```

Still further optimization is carried out in the important area of subscripting. In performing subscript optimization, both the H-1800 and the H-800 have an advantage over other large-scale computers, because of the many special registers (256) built into the Honeywell machines. A separate special register is available to hold the address of each subscript combination, thus eliminating all the unloading and reloading of subscript addresses which must occur in machines having only a few such registers.

To prevent redundant computations, the constant to be added to the subscript combination in performing the updating is determined either at compilation time or at the outset of a loop. Updating the loop subscripts is performed at the head of the loop, rather than at the conventional time at the end of the range. This modification eliminates the one unnecessary updating that occurs when the control variable exceeds its ultimate value. It represents a meaningful saving when the last updating appears within a deeply nested series of do loops, since the updating would occur hundreds of extraneous times.

As a final point in the discussion of code optimization, it should be noted that although other compilers attempt far less code optimization, they still require three or four passes over the input tape to perform compilation, as compared with two passes for Honeywell's AUTOMATH 1800 compiler.

LANGUAGE

The Automath 1800 compiler language is based on Fortran IV language. It includes all features associated with FORTRAN IV, such as double precision, complex and logical data types and their operations, labeled common statements, adjustable dimensioned arrays, logical IF statements etc.

A. New Aspects of Fortran IV

1. Absolute value of fixed point integer constant increased to $2^{44}-1$.
2. Five new data type statements used to declare the data mode of variable or function names.
3. A new set of logical expressions is available.
4. Logical IF statement used with logical expressions.
5. Double-precision and complex arithmetic added.
6. Uniform naming convention for the 5 types of functions and subprograms.
7. Functions may be of any of the five data types.

8. Labeled COMMON statements provide separate blocks of COMMON areas.

9. Dimensioned information about arrays may be declared within a COMMON statement.

10. The COMMON and EQUIVALENCE statement inter-relationship has been changed to be more useful in altering subprograms that are associated with large numbers of other subprograms.

11. Provision for adjustable DIMENSION statements used in FUNCTION or SUBROUTINE subprograms so that the dimensions may be passed to the subprogram as arguments.

12. Generalized READ and WRITE statements incorporated, to be used with a peripheral unit designator which permits assignment of peripheral equipment at execution time.

13. New DATA statement which will initialize data areas to specified values upon program loading.

14. Certain machine-oriented statements of Fortran II relating to sense lights, sense switches, and arithmetic error conditions are replaced by subroutine subprograms.

B. Automath 1800 Language Extensions to Fortran IV

1. Octal and Hollerith constants can be used in the source program.

2. Console output statement added - TYPE.

3. An alternate form of the DATA statement (B-13 above) is available to make it an executable statement so that arrays may be reinitialized during execution.

4. EQUIVALENCE statements may be written in the normal form for subscripting multi-dimensional arrays rather than the artificial reduction to an equivalent single dimension subscript.

5. Data type statements may contain information about dimensions of arrays.

6. Adjustable arrays may have the variables denoting their size in COMMON rather than requiring them to appear as CALL arguments.

7. Methods are provided for detecting the existence of an end-of-file, end-of-tape, and uncorrectable tape error condition and transferring to an appropriate portion of the routine.

C. Fortran II to Fortran IV Translator

The Translator itself will be written in the language of Automath 800 (Fortran II) and will be capable of operating upon itself to produce an equivalent Fortran IV program to run under Automath 1800. It will be designed to operate on the minimum equipment requirements of the compiler.

Most Fortran II programs, whether or not written for Automath 800, can be successfully translated to Fortran IV using this system. However, in view of the great variety of Fortran II programs, the translator can not be expected to cover all situations.

The function of this sytem is to provide a computer translation from one language to another. It is intended to provide program diagnostic information. Programs submitted for translation should be basically sound and correct.

D. Assembly Language Processor

1. General Description

An alternate processor called in by control card will handle a subprogram written in ARGUS assembly language. The output is a subprogram in identical form to that produced from Automath 1800 source language. Thus, input to a job may consist of both Fortran IV subprograms and assembly language subprograms. These are processed in turn to produce similarly structured coding on BPT.

This assembly language will have virtually all of the machine-oriented features of ARGUS. The language is convenient for generating coding not easily produced from Fortran statements. It might be used where specific machine coding is desired or to communicate with special peripheral units not provided for in Automath. It may be used to adapt an existing ARGUS program to run with a scientific job.

Unlike Automath 800, Automath 1800 does not allow interspersing of ARGUS statements and Fortran statements within one subprogram. The availability of the separate assembler makes this unnecessary. One frequent use of interspersed ARGUS statements in Automath 800 was to provide console output from the source program. Automath 1800 has the console statement TYPE which provides this feature as part of its basic language.

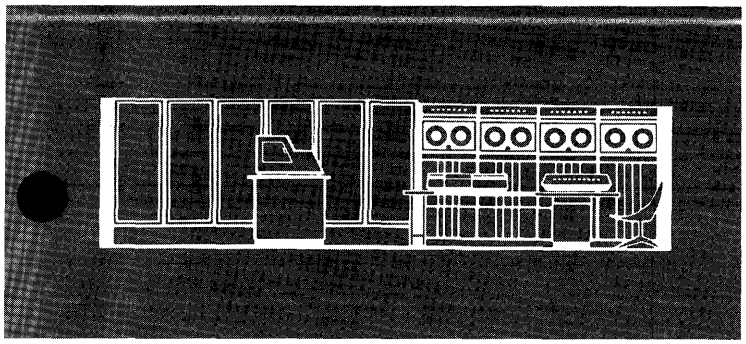
2. The Language

The assembler will accept all standard ARGUS coding related to machine instructions. ARGUS control instructions require special handling. All types of addressing will be allowed, but main memory addressing will be converted to a form of indexed addressing so that the resultant coding is fully relocatable with compiler-generated coding. Certain Fortran statements may be included in the language. These include CALL and RETURN statements as well as specifications statements such as DIMENSION, COMMON, etc. Macro calls will not be allowed in the initial version. There is a restriction of 1024 instructions and/or tags in a single program. However, when this limit is reached, one may simply write a call to another assembly language subprogram and continue the routine in the second subprogram.

DIAGNOSTIC AND CHECKOUT SYSTEMS

"Debugging" an object program written in FORTRAN IV language is simplified by an elaborate diagnostic system that detects a wide variety of source program errors and provides the programmer with a detailed explanation of the cause(s). More than 350 different English-language error messages are included, and the appropriate message is referenced directly to the specific statement in error. In addition to positively identifying "fatal" errors that prohibit compilation, the compiler also questions unusual situations which may or may not be errors. These are noted as "probable" errors, but do not inhibit execution of the program.

Checkout of an object program is facilitated by source-language derrails, which permit both dynamic and terminal dumping of tapes and memory. The resulting reports are automatically edited into the format of the variable being dumped.



Automath 1800 Compiler and Operating System

A New Approach to Rapid Scientific Processing

By: Martin N. Greenfield*

Bulletin No. 109

HONEYWELL EDP WELLESLEY HILLS 81, MASSACHUSETTS

AUTOMATH 1800, an extremely fast FORTRAN IV language compiler, is combined with an operating system that makes full use of the parallel-processing capabilities built into the hardware of the Honeywell 1800 and 800 Electronic Data Processing Systems. The result is a dramatic reduction in turnaround time to the user, without any noticeable penalty in machine utilization. All the benefits of "load-and-go" operation are retained without the disadvantages. Since AUTOMATH 1800 requires only half the memory needed by virtually all other FORTRAN IV compilers, any remaining memory capacity is available for parallel processing of other programs.

TURNAROUND TIME VERSUS MACHINE UTILIZATION

One of the most difficult problems facing the manager of a scientific computing facility is how to minimize the *turnaround time* to the user while still achieving maximum machine utilization. In a busy installation which must handle hundreds of jobs daily, this problem is especially acute, and the two goals seem to be in opposition.

To process a high volume of jobs it is necessary to employ the central processor to the fullest extent possible, reducing the number of set-ups and other interruptions to a bare minimum. This requirement has led to the widespread use of the *load-and-go* operating method, in which a substantial stack of jobs is converted from punched cards onto a common input tape and processed in one long run.

Although this method satisfies the need for processing efficiency, it is undesirable from the individual user's standpoint because he must wait a long time for results. Not only must he wait for his own job to be completed, but also for all other jobs in the stack along with it, since the results of the first job are not available until the last job is executed. Thus, machine utilization is obtained at the expense of turnaround time (the time between submission of the program and availability of results). Indeed, it is not unusual to have a program execution time of one or two minutes and a turnaround time of from two to twelve hours.

In scientific and engineering applications, high turnaround time can mean significant delays in an entire project, because the engineers, mathematicians, and scientists cannot proceed to the next phase of their work before analyzing and modifying the solutions and results of the preceding phase.

To solve the turnaround time versus machine utilization dilemma, Honeywell has taken a fresh approach to scientific and mathematical processing in the operating system for the new AUTOMATH 1800 compiler. Through full use of the parallel-processing capabilities built into the hardware of the Honeywell 1800 and 800 Electronic Data Processing Systems, very substantial reductions in turnaround time are achieved without any noticeable penalty in machine utilization. The combined hardware-software techniques used to accomplish these seemingly conflicting ends and the salient characteristics of the new AUTOMATH 1800 compiler are described in this bulletin. Since the Honeywell approach is intended to meet the needs of

* Manager, Scientific Languages Department, Honeywell Electronic Data Processing.

DSI-182-5563
Printed in U.S.A.

installations with a very large volume of scientific and engineering calculations, it should easily solve the turnaround problems of those having more modest workloads.

AUTOMATH 1800 COMPILER

The AUTOMATH 1800 compiler is based on FORTRAN IV language, the most recent version of FORTRAN. Designed primarily for the large-scale Honeywell 1800, the compiler can also be used with excellent results on the Honeywell 800. An extremely fast compilation rate—in excess of 1,000 statements per minute on the H-1800 (faster by several hundred statements per minute than any other FORTRAN IV compiler now available)—contributes materially to a reduction in turnaround time, but the story does not end there. Parallel processing permits the compilation and execution of one job to be performed simultaneously with output operations for an earlier job and input operations for a later job. All of this is done automatically and on a routine basis, without the need for human scheduling, switching, or intervention of any type. To appreciate how different the Honeywell load-and-go approach is from the conventional method, consider the following contrast.

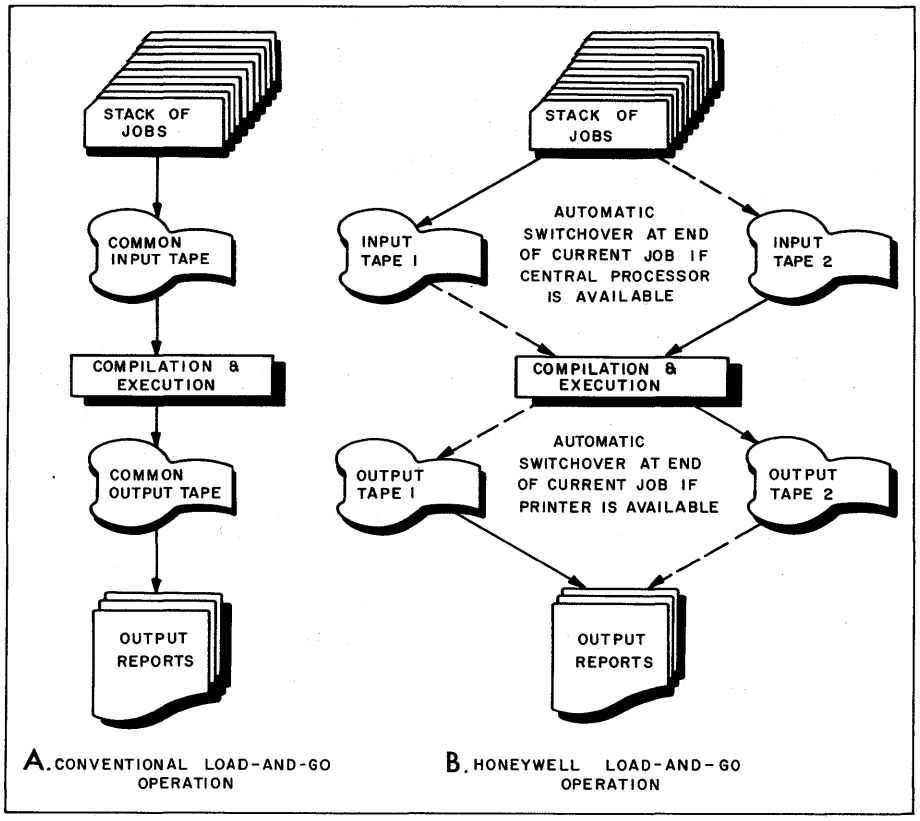


Figure 1. Comparison of Honeywell and Conventional Load-and-Go Techniques

CONVENTIONAL LOAD-AND-GO OPERATION

In conventional load-and-go operation the entire stack of jobs[1] is placed in the card reader and transferred from punched cards onto a common input tape (see Figure 1A). Compilation cannot begin until all the jobs in the stack have been transferred onto the common input tape. This means that if the stack is very large, as it generally is, a significant period of time (in terms of computer operations) may elapse between the card-to-tape conversion of the first jobs and the beginning of compilation. (It is shown below how the Honeywell load-and-go method permits compilation to start as soon as the first job has been transferred to tape, at the same time that other jobs are still being read in.)

After the first program is compiled under the conventional method, it reads its data from the common input tape, processes the data, and writes its results on a common output tape. The remaining jobs are then compiled and executed, one after the other. After all have been completed, and only then, are *any* of the results available for listing. The common output tape is listed in a tape-to-printer operation. Although the required high machine utilization is achieved, it is at the expense of turnaround time. On the other hand, the Honeywell method permits the results of a job to be printed almost as soon as they are available.

In a busy installation employing the conventional load-and-go method, the main computer is generally supported by two or more independent "satellite" computers, performing card-to-tape conversions and printing off-line. There is rarely any automatic coordination or linkage between the computers, however. The operators must either switch tapes manually at a control box or physically dismount a tape from the input satellite, carry the tape over to the

[1] A job may consist of programs in source language (FORTRAN), programs previously compiled and now stored in binary form on cards, and data.

main frame (where it takes its place in a queue of waiting work), subsequently mount the tape on the main frame, dismount it at the end of main-frame processing, carry the result over to the output satellite (where it again takes its place in a queue), and finally mount the tape for listing.

Thus, an incoming job must generally wait in at least three different queues: input, main frame, and output. Operator intervention of some type is required, whether it be switching at a control box or physical handling of tapes. The Honeywell technique obviates the need for uncoordinated satellites and eliminates all intermediate queues demanding human attention.

HONEYWELL LOAD-AND-GO OPERATION

Instead of using one common input tape and one common output tape, AUTOMATH 1800 uses two of each (see Figure 1B), switching back and forth from one input tape to the other and from one output tape to the other at appropriate times to assure an almost continuous flow of work to and from the central processor. While one input tape is feeding its contents into the central processor for compilation and execution, the other is receiving input from the card reader. Similarly, while one output tape is receiving results from the central processor, the other is feeding its contents to the printer. In this manner, input, output, and central processor operations proceed in parallel.

Switchovers are programmed to occur conveniently at the end of a complete job. On the input side they are governed by the availability of the central processor; on the output side, by the availability of the printer. When the central processor is ready for more input, it causes a switching of the input tapes at the end of the job that is presently being converted from cards to tape. Thus, the central processor is supplied with more input at the earliest convenient moment. Furthermore, there is never any interruption in input operations, since there is always a tape available to receive card images.

When the printer has exhausted the output tape which it has been listing, it causes a switching of output tapes at the end of the job that is presently being executed. Thus the printer, too, is supplied with more work at the earliest convenient moment, and the central processor is never kept waiting for an output tape. The result is that at least three independent jobs are normally in different phases of operation simultaneously. The output from the earlier jobs becomes available while other jobs in the stack are still awaiting processing.

Several other significant points of the Honeywell technique should be noted. First, turnaround time is minimized by directing each portion of the stack of jobs into the subsequent operation at the earliest possible moment. Once the run has started, no tape mounting or other operator intervention is required, except to place more jobs in the card reader and take results from the printer. Through parallel processing, the input-output operations are performed on-line without any noticeable penalty in execution time. The Honeywell central processor not only performs that part of the work conventionally delegated to satellite computers, but also does a better job because all activities are centrally coordinated. Once a job is placed in the input stack, it goes all the way through to completion in an almost continuous operation. As soon as one part of the system becomes available for additional work, it is automatically placed in service again the instant that a convenient switchover point is reached. In contrast, a satellite would remain idle until the operator supplied it with some more of the waiting work.

COMPARISON OF TURNAROUND TIMES

A comparison of the differences in turnaround time between the Honeywell technique using parallel processing and the conventional load-and-go method (on comparable-size systems) is portrayed graphically in Figure 2 for a stack of ten jobs. The top part of the diagram represents the time for conventional operation, with the ten jobs being transferred first from cards to the common input tape (top left), then compiled and executed in sequence (top center), then listed in a tape-to-printer operation (top right). The lower part of the illustration shows how several different jobs are simultaneously in different phases of processing at any instant of time with the Honeywell method. Note that the finish time for each job is very much less than in conventional operation. Note, too, that the over-all elapsed time for the complete stack of jobs is also much less.

When the conventional computer is converting job 7 from cards to tape, for example, no compilation of any program has yet begun. At the same instant of time, the Honeywell computer is also converting job 7 from cards to tape. However, it has already compiled and executed jobs 1, 2, and 3; it has already listed the results of job 1 and part of the

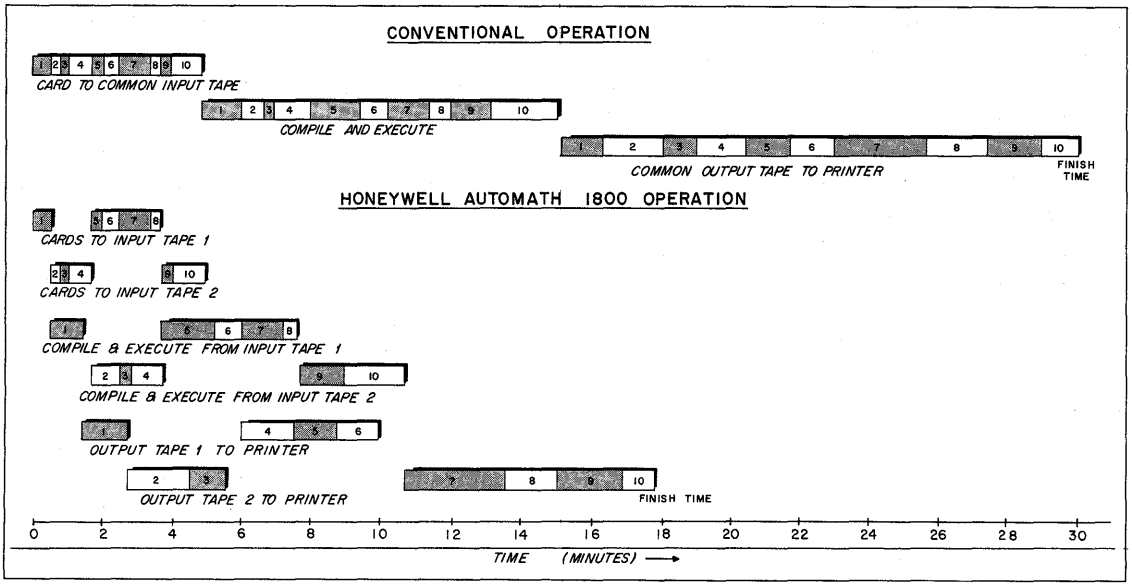


Figure 2. Timing Comparison Between Honeywell and Conventional Techniques

results of job 2; and it is presently compiling or executing job 4.

In this example, the average turnaround time for the Honeywell computer is approximately half that of the conventional computer. For a larger stack of jobs, the savings are even more dramatic. With the stack of 40 jobs shown in Table I, the average turnaround time is reduced to one-third by the Honeywell load-and-go technique. As can be seen from the table, the results of job 1 are available in 1 minute and 8 seconds from the Honeywell computer versus 66 minutes and 38 seconds from the conventional machine—a difference of more than one hour. Furthermore, the differences for later jobs in the stack are almost as great. The average turnaround time for the Honeywell computer for this stack of jobs is 28 minutes and 17 seconds as compared with 85 minutes and 42 seconds for the comparable-size system—a difference of 57 minutes and 25 seconds.

TABLE I. COMPARISON OF TURNAROUND TIMES FOR A STACK OF 40 JOBS

JOB NO.	TURNAROUND TIMES HONEYWELL (Minutes, Seconds)		TURNAROUND TIMES OTHER (Minutes, Seconds)		DIFFERENCE (Minutes, Seconds)	
1	1	8	66	38	65	30
2	2	18	67	6	64	48
3	4	3	67	51	63	48
4	5	27	68	0	62	33
5	6	8	68	53	62	45
6	8	45	68	54	60	9
7	11	6	71	15	60	9
8	13	2	71	38	58	36
9	14	18	72	39	58	21
10	18	56	77	17	58	21
11	18	30	76	33	58	3
12	19	47	77	50	58	3
13	21	15	79	18	58	3
14	21	9	79	12	58	3
15	22	18	80	21	58	3
16	23	56	81	59	58	3
17	23	42	81	45	58	3
18	26	35	84	14	57	39
19	26	26	84	5	57	39
20	27	11	84	50	57	39
21	29	48	87	27	57	39
22	29	24	87	3	57	39
23	32	53	88	20	55	27
24	32	51	88	18	55	27
25	34	12	89	39	55	27
26	35	14	90	41	55	27
27	35	27	90	54	55	27
28	39	36	94	39	55	3
29	39	45	94	24	54	39
30	39	47	94	26	54	39
31	41	3	95	42	54	39
32	44	12	98	51	54	39
33	44	3	98	42	54	39
34	45	32	100	11	54	39
35	45	33	100	12	54	39
36	47	18	101	57	54	39
37	47	32	101	59	54	39
38	49	48	103	51	54	39
39	50	35	105	14	54	39
40	50	48	105	27	54	39
AVERAGE	28 mins. 17 secs.		85 mins. 42 secs.		57 mins. 25 secs.	

THE SWITCHOVER TECHNIQUE

The details of the tape-switching technique as applied to a stack of jobs can be followed in the example of Figure 2. Initially job 1 is converted from cards onto input tape 1, and both the central processor and printer are available. As soon as the conversion of job 1 is completed, the input tapes are swapped, and the job is directed immediately into the compilation and execution stage. Job 2 is converted from cards onto the alternate input tape. Being small, job 2 is converted before the central processor finishes executing job 1. Consequently, no swapping

of input tapes occurs; job 3 follows job 2 onto the same tape, as does job 4.

At the end of job 1's execution, the printer is still available; therefore, job 1 is immediately directed into the printing operation. (Note that this job has gone through all three stages of operation – card-to-tape conversion, compilation and execution, and printing –in the minimum amount of time required to perform these processes.) While job 4 is being converted from cards to tape, the central processor again becomes available for more input; therefore, the input tapes are swapped once more, as soon as all of job 4 is on the input tape. Now jobs 2, 3, and 4 are directed in sequence into the compilation and execution phase. When job 2 has been executed, the printer is still listing the results of job 1; therefore, no switching of output tapes occurs at that time. However, the printer does become available again during the execution of job 3, and the ouptput tapes are swapped when the job is over. Jobs 2 and 3 are then channeled into the printing operation. The cycle continues in this fashion, as illustrated in the lower part of Figure 2.

TYPICAL COMPILATION RATES

To give some appreciation of the 1000-statement-per-minute rate in terms of compiling an actual program, here are some figures: A typical 80-statement program can be compiled in about 5 seconds on the Honeywell 1800; a 300-statement program takes only 20 seconds.[1] These speeds are attained through a number of advanced techniques, including syntax-directed statement analysis, high-speed table-searching methods, rapid input-output conversions, and several kinds of optimization.

PARALLEL PROCESSING OF OTHER PROGRAMS

Not only is parallel processing exploited in the operating system for the AUTOMATH 1800 compiler, but the compiler itself has been designed to meet some very stringent memory-usage specifications to allow simultaneous processing of other programs in parallel with the compiler. Consequently, AUTOMATH 1800 uses only a 16,000-word memory. This is in contrast to the 32,000-word capacity necessary for compilation of FORTRAN IV programs on competing systems. A 32,000-word Honeywell system still has half the memory available for parallel processing of other programs, at the same time that it is surpassing the turnaround performance of the competing system.

[1] Times are approximately double on the H-800.

PARALLEL PROCESSING OF TWO STRINGS

In addition to the load-and-go mode described earlier, the AUTOMATH 1800 compiler can also be operated in a batch-compile-and-execute-later mode. In this mode, all programs are compiled in a batch for execution at a later time. The option improves flexibility for certain types of applications.

Since the compiler produces object programs that can be executed in parallel with each other, the use of parallel processing can increase efficiency, even when the object programs are "compute-limited" (*i.e.*, limited by the internal speed of the central processor). Parallel execution of such programs may not appear to produce any savings, since the processing time will be the same as for serial execution (see A and C of Figure 3). However, if one of the programs being executed in parallel should be interrupted for any reason, such as the need to mount special data tapes, the interruption period—that would represent lost time on conventional systems—is entirely absorbed by the alternately running object routines (see B and D of Figure 3).

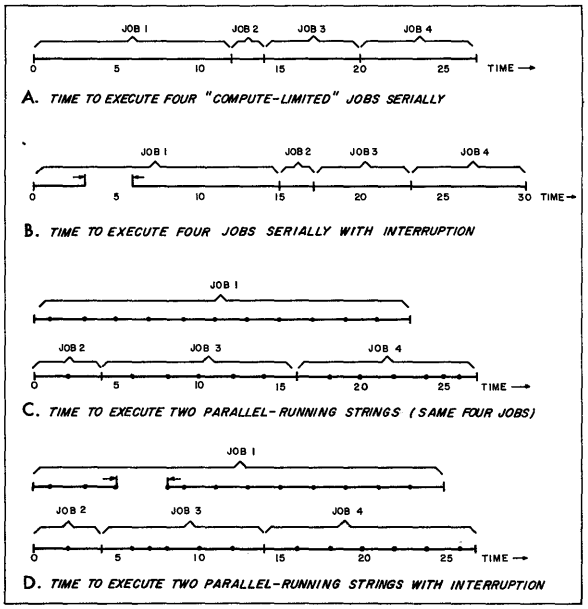


Figure 3. Parallel Processing of Two "Compute-Limited" Strings Absorbs Interruption Time

Consequently, two parallel-running strings of object programs can be executed in the minimum time to run them serially, regardless of the number of non-

coincident interruptions occurring in each of the strings.

DYNAMIC STORAGE ALLOCATION

To assure maximum use of core storage, an extensive system of segmentation control has been built into the AUTOMATH 1800 monitor. Allocations of memory for the non-common arrays and instruction strings are made dynamically rather than at initial loading time to conserve as much memory as possible until the time when it is actually needed. The overlay technique is employed wherever possible to write over inactive program segments and unused array areas that are needlessly taking up memory space.

This optimization of core storage is important for two reasons: First, it enhances the ability to do parallel processing by not allowing any program to tie up an excessive amount of memory. Second, it enables the execution of very large programs that could not otherwise be run, even if allowed the total memory.

In conventional practice, core storage for non-common arrays is allocated at program-loading time. This is wasteful because most of the storage is not actively required until particular phases of execution begin. Since the rules of FORTRAN state that data in non-common array storage are significant only to the subprogram defining them, the memory allocation is wasted until the defining subprogram is entered. Consequently, the AUTOMATH 1800 monitor does not allocate any non-common array storage until the subprogram defining the non-common array is entered. At that time, array storage is allocated from a storage pool. The allocation is maintained until a return from the defining subprogram is executed. Then the storage is immediately released and returned to the pool for availability to other subprograms.

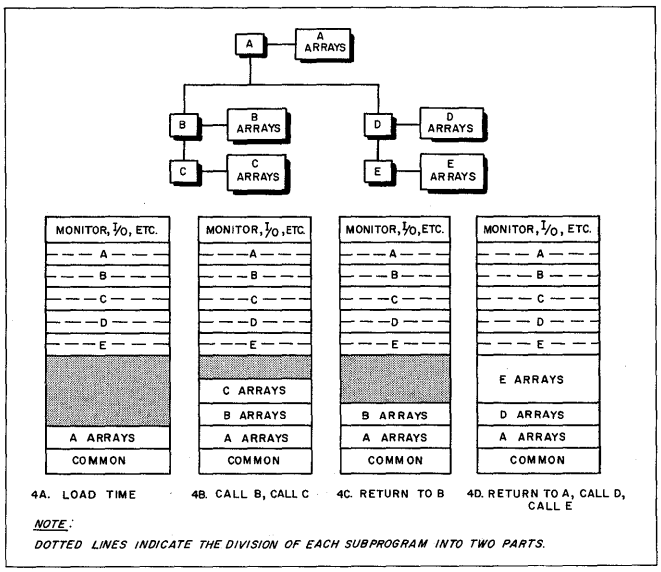


Figure 4. Dynamic Allocation of Non-Common Array Storage

The effect is that a substantial amount of memory that would otherwise have to be reserved for array storage can be saved by the dynamic-allocation and pool concepts. Instead of requiring a memory area large enough to handle all the non-common arrays simultaneously, the Honeywell technique needs only an allocation big enough to handle the arrays actively in use at any instant of time.

In summary: Memory allocations for non-common arrays are made only at the time needed, for only the amount necessary, and for only as long as they are actively used.

EXAMPLE OF DYNAMIC STORAGE ALLOCATION

An example of the dynamic-allocation and pool-storage technique is illustrated in Figure 4. The contents of memory at four instants of time are shown in Figures 4A through 4D to illustrate a simple case for five programs having non-common arays. The program hierarchy is as shown in the top part of the diagram. Program A can call subprogram B which in turn can call C, or program A can call subprogram D, which will subsequently call E. Thus, subprograms B and C will never be actively executed at the same time as D and/or E.

Initially, the five programs are loaded into memory, as shown in Figure 4A, and common array storage is reserved. However, at this time only the non-common array storage for program A is allocated (whereas the conventional method would attempt to allocate storage for all five programs, even though it was not yet required, and even though B and C would never be called at the same time as D and/or E). The shaded area in the diagram represents the remaining storage pool.

When program A calls B, array storage for program B is allocated from the central pool, and again when B calls C, more array storage is allocated. Figure 4B illustrates the memory usage at the time program C is active. When control is returned from C to B, the storage reserved for C's non-common arrays is released to the central pool, as indicated in Figure 4C. Program B subsequently returns control to A, releasing its array-storage space to the central pool, and A now calls subprogram D, which is assigned array space for the first time. D eventually calls E, whose array-storage requirements are such that they

take up all of the remaining memory in the pool (Figure 4D).

Note that it would not have been possible to execute the entire program with the given memory, using the conventional method of allocation. However, with the Honeywell technique, the program is easily executable in the same size memory.

Now consider the case where the non-common arrays require much more storage space than they did in the previous example, such that the available memory cannot satisfy all the demands simultaneously. This problem is solved by the Honeywell "overlay" technique, described below.

MEMORY OVERLAY TECHNIQUE

The overlay technique for conserving memory space involves "writing over" certain invariant portions of a program that is not being actively executed, when the space is required by other active programs. The overlaid portion is then called back into memory from the program tape at the time the inactive program becomes active again.

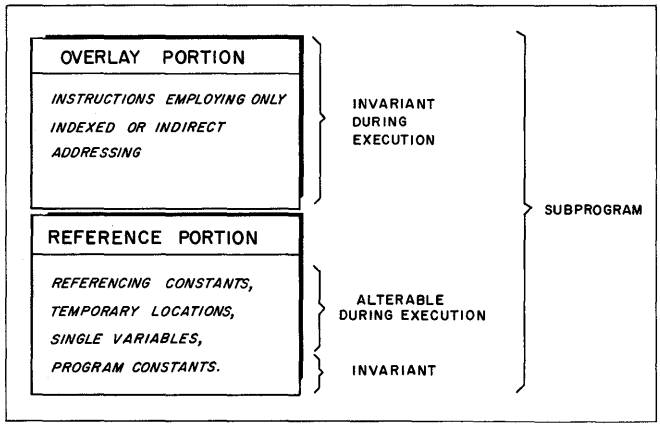


Figure 5. Subprograms Are Generated in Two Distinct Parts

To permit this type of overlaying when necessary, the compiler generates every subprogram as two distinct parts, as indicated in Figure 5. One part contains the string of instructions; the other contains the program constants and referencing (linking) constants. (The dashed lines in Figures 4 and 6 indicate the division of programs into these two parts.)

Because the instruction string does not change during the course of execution, it may be easily restored from the program tape, if it should become necessary to use the space taken up by this part of an inactive subprogram.

To determine whether a subprogram is active or inactive, the internal monitor maintains a status table, showing the presence (or absence) of any particular subprogram in memory at any instant of time, as well as its condition (active or inactive).

At initial loading time, as many subprograms as possible are loaded into the available memory. As control passes to each new subprogram, a rapid check is made in the status table to determine if the program is already in memory. If it is not, the monitor determines a place for it to be loaded by first attempting to use the memory assigned to subprograms that are presently inactive. Failing to find sufficient space among the inactive subprograms, the monitor then overlays the instruction portions of the ancestors to the subprogram being called, since these portions are invariant and can be easily restored from tape if later required.

DYNAMIC RELOCATIONS

The monitor can also, if necessary, create areas of contiguous space by performing a dynamic relocation of subprograms presently in memory to group them into adjacent spaces. The ability to perform these dynamic allocations rapidly is a direct result of (1) the large number of index and special registers in both the Honeywell 1800 and 800 and (2) instructions that can move large numbers of words. Since the instruction addresses all employ special registers for indirect addressing, they are not sensitive to being in a particular position in memory for execution. Furthermore, there is no penalty in execution time for employing indirect addressing. This contrasts with other large systems having only a few index registers. These machines must operate at reduced speeds when using indirect addressing.

EXAMPLES OF OVERLAYING AND DYNAMIC RELOCATION

An example of the overlay technique in operation is illustrated in Figure 6. The program hierarchy is exactly the same as before (Figure 4). The only difference is that the non-common arrays for programs A, B, and C now require so much space that at loading time subprogram E cannot be loaded (see Figure 6A). Program D, though presently inactive, is loaded, but there is no room for the non-common arrays of D.

As before, program A calls B; B calls C; then C returns control to B; B returns control to A, and A then calls D. The monitor recognizes that B and

C have now become inactive and overlays the area used by their non-common arrays with those of D, as shown in Figure 6B. When subprogram E is called by D, the monitor finds that there is no memory available for E's non-common arrays, nor is there any inactive-program array storage available for overlaying as there was before.

Observing that subprograms B and C are now inactive, the monitor tries to overlay their instruction strings with the non-common arrays of subprogram E. Since there is still not enough space, it next tries to overlay all of programs B and C. Still the space is insufficient, and the monitor now attempts to overlay the invariant portions of programs A and D as well. Enough total memory space is thus made available, but a problem still exists because the resulting space is not contiguous, and none of the pieces alone is large enough to hold the non-common arrays of E.

Recognizing the problem, the AUTOMATH monitor dynamically relocates the reference portions of subprograms A and D (which must remain in memory as long as those programs are active) and places them adjacent to each other in storage, as shown in Figure 6C, so that program E and its non-common arrays may now have enough contiguous space to be loaded. The desired result is thus obtained (see Figure 6C). To accomplish it, subprograms B and C were completely overlaid; the invariant portions of A and D were overlaid, and the reference portions of A and D were dynamically relocated to adjacent memory areas to create a large contiguous area for the loading of subprogram E.

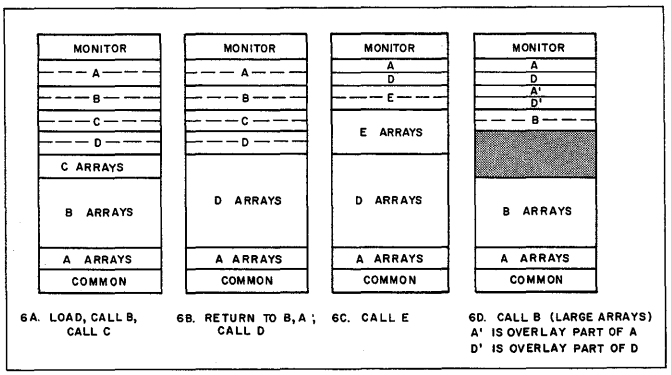


Figure 6. Dynamic Relocation, Loading from Tape, and Restoration from Tape

When subprogram E returns control to D, which then returns control to A, the invariant portion of A (*i.e.*, A′) is no longer in memory. Therefore, A′ is called back in from the program tape, and it overlays the no-longer-used non-common arrays of subprogram E, creating a storage pool with the remaining unused space. Program A next calls B, and both parts of B are reloaded from the program tape, overlaying the non-common arrays of D. Any unused memory space remaining is added to the pool. The result is as shown in Figure 6D.

It should be emphasized that although the above description is rather lengthy, the actual internal execution time to perform overlaying and dynamic relocation is just a few milliseconds in the most involved case. Furthermore, the monitor performs dynamic relocations in parallel with any tape operation required to locate and load a subprogram not in memory.

It should also be emphasized that the programmer may write his programs without any consideration of the overlay process. No special control cards or grouping considerations are necessary. The internal monitor automatically provides the correct amount of memory to meet the program's needs at each point of execution. Provision is also made for the programmer to designate a program as a "high-duty" program that must never be overlaid.

INPUT-OUTPUT PACKAGE

The emphasis on conserving memory is also apparent in the design of the input-output package. This has been written for joint service to several AUTOMATH object programs being executed simultaneously. Hence, only one copy of the package is in memory, regardless of the number of programs operating in parallel.

Double buffers for both input and output add to the speeds of these operations. While information in one input buffer is being transferred to the central processor, the alternate buffer for the same input device is automatically filled with the following record in anticipation of the next read instruction. When the central processor is ready for this record, the buffers are automatically switched, and the information is immediately available. Meanwhile, the subsequent record is read into the first input buffer. Output operations also employ the same principle. The result is that reading and writing operations are completely overlapped with computing.

Even though the physical positions of the tapes may not coincide with the programmer's concept of where they actually are (because of the anticipation activity), the programmer need not concern him-

self with this problem, since positioning control is automatically maintained for him.

The programmer has complete flexibility in designating the size of the records to be written on tape. By means of a control card, he can override the standard size used in the absence of any other information and can designate either longer or shorter records to conform with the amount of estimated memory space the program requires. The more memory space the programmer can relinquish in favor of reading and writing longer records, the faster the reading and writing operations can be performed.

Another way in which the input-output package speeds operations is by performing very rapid interpretation of FORMAT statements. In their standard form, these are rather slow to interpret as far as the computer is concerned, since they are expressed in a manner that is primarily for the programmer's convenience. On first encountering a FORMAT statement, the input-output package translates the conversion elements in the statement into a syllabic form that is much more rapidly interpretable the next time the compiler processes the statement. Thus, subsequent usage can proceed at full speed as a result of this initial translation.

PRECISION

For scientific and engineering computations, a much greater degree of precision is generally required than for other applications. Precision is not only a function of the word size, but also of the compiler's input-output routines which convert the incoming data from decimal form to floating-point binary for machine processing, then reconvert to decimal when the calculations are completed. The precision obtained on the Honeywell 1800 with the rapid-conversion input-output routines of the AUTOMATH 1800 compiler is compared below with figures for comparable-scale computers and their FORTRAN IV compilers.

Accuracy	Honeywell 1800 with AUTOMATH 1800 Compiler	Comparable-Scale Computer with FORTRAN IV Compiler
Word Size:	48 bits	36 bits
Single Precision:	12 decimal digits	8 decimal digits
Double Precision:	20 decimal digits	16 decimal digits
Integer Values:	13 decimal digits	11 decimal digits
Exponent Range:	10^{+77} to 10^{-78}	10^{+37} to 10^{-38}

CODE OPTIMIZATION

Despite the intrinsically high compilation speed, extensive code optimization is attempted to reduce the turnaround time even more. A second pass over the input tape is added primarily for this purpose. The compiler "looks ahead" to future statements to see if it can save repetitious work by performing a calculation once and using the result in subsequent expressions which will require the same calculation.

While certain of the more advanced compilers for competitive machines attempt optimization within a single statement, AUTOMATH 1800 goes much further by analyzing all statements in an entire *region* of statements [1] and optimizing over the entire region. A large amount of needless repetition is thus eliminated.

For example, many compilers can recognize that each of the subexpressions (A + B) and (D + E) appears twice in the formula:

A = COS (A + B + C + D + E) +
(B + A) ** 2. + (E + D) * 3.

Not only will AUTOMATH 1800 recognize the duplication and compute the subexpressions only once, but it will also recognize the same subexpressions and their equivalents every time they recur in later statements in the same region.

For example, consider the region of five statements shown in Figure 7.

```
              GO TO 300
300           A = COS (A+B+C+D+E) + (B+A)** 2.+ (E+D) * 3.
              Q = F+G
302           R = H+B+A                                      } REGION
303           S = F * SIN ( E+B+A+D )
              GO TO 500

        THIS REGION IS PROCESSED AS THOUGH IT WERE WRITTEN:

300           TEMP 1= A+B
              TEMP 2= D+E
              TEMP 3= TEMP 1+TEMP 2
              A=COS (TEMP 3+C)+TEMP 1 ** 2. + TEMP 2 * 3.
              Q= F+G
302           R= H+TEMP 1
303           S= F * SIN (TEMP 3)
              GO TO 500
```

Figure 7. Regional Optimization of Common Subexpressions

Control enters the region at statement 300, by means of a "GO TO 300" statement. The end of the region, defined as the next control statement, is the "GO TO 500" statement. In statement 300, AUTOMATH

[1] A *region* is a sequence of statements extending from a point where control may enter the sequence up to the next control statement.

1800 recognizes that (A + B) and (B + A) are the same thing and that (D + E) and (E + D) are also equivalent. Furthermore, it recognizes that the expression (B + A) appears again in statements 302 and 303 and that (E + D) appears again in statement 303. Thus, instead of performing four different computations for (A + B) or even three, it performs only one. Instead of doing three or even two calculations of (E + D), it does one.

RESEQUENCING OF COMPUTATIONS WITHIN DO LOOPS

Another type of optimization performed is the resequencing of computations contained within a DO loop to positions outside of the DO loop whenever possible. This means that instead of recomputing the same expression over and over again each time control passes through the loop, the compiler performs the computation only once—at the outset of the loop—and uses the result each time thereafter. A great amount of repetitive computation is thus eliminated. The savings are especially significant if the DO loop has a large number of iterations. Furthermore, when the compiler encounters a *nest* of DO loops (*i.e.*, one DO loop within another), it determines the level of definition of each subexpression and resequences the computation to the outermost level possible, minimizing redundant repetitions.

An example of statement resequencing is given in Figure 8. In Figure 8A, two DO loops are nested one within the other, such that for each iteration of the outer (I) loop there are 100 iterations of the inner (J) loop. Since there are also 100 iterations of the outer loop, an expression in the inner loop

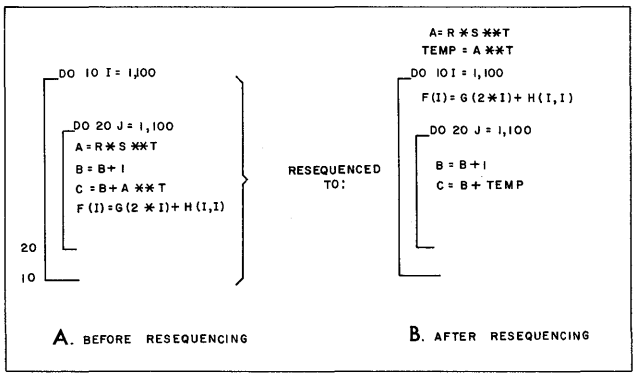


Figure 8. Resequencing Statements to Positions Outside DO Loops

would be repetitively calculated 10,000 times by the conventional compiler.

AUTOMATH 1800, however, by first determining the level of definition of each subexpression, recognizes that the expressions (A = R * S ** T) and (A ** T) consist of variables that are unaltered in the DO loop nest, and it resequences them outside of both loops, as shown in Figure 8B, saving 9,999 repetitious calculations of each, because they need only be computed once. It also recognizes that the formula

$$F(I) = G(2 * I) + H(I,I)$$

is independent of variable J and can thus be resequenced to a position outside of the J loop, as shown in Figure 8B. Thus the formula is evaluated 100 times instead of 10,000 times. (The statement B = B + 1 cannot be resequenced, since its value changes with each iteration of the J loop.)

SUBSCRIPT OPTIMIZATION

In performing subscript optimization (*i.e.*, in minimizing the number of times a subscript combination is computed), both the H-1800 and H-800 have an advantage over other large-scale computers, because of the many special registers (256) built into the Honeywell machines. A separate special register is available to hold the address of each subscript combination, thus eliminating all the unloading and reloading of subscript addresses which must occur in machines having only a few such registers (*e.g.*, seven). Furthermore, with one special register assigned to each subscript combination, it is only necessary to compute the subscript value once, since the address of the result is always available through the assigned special register.

Updating of the subscript combination can be done very rapidly by allowing the hardware to increment the special register automatically, when the value changes by a constant in the range 1 to 31. Subscript updating is stratified to occur only at the points where each control variable separately influences a change of subscript.

To prevent redundant computations, the constant to be added to the subscript combination in performing the updating is determined either at compilation time or at the outset of a loop. Updating the loop subscripts is performed at the head of the loop, rather than at the conventional time at the end of the range. This modification eliminates the one unnecessary updating that occurs when the control variable exceeds its ultimate value. It represents a

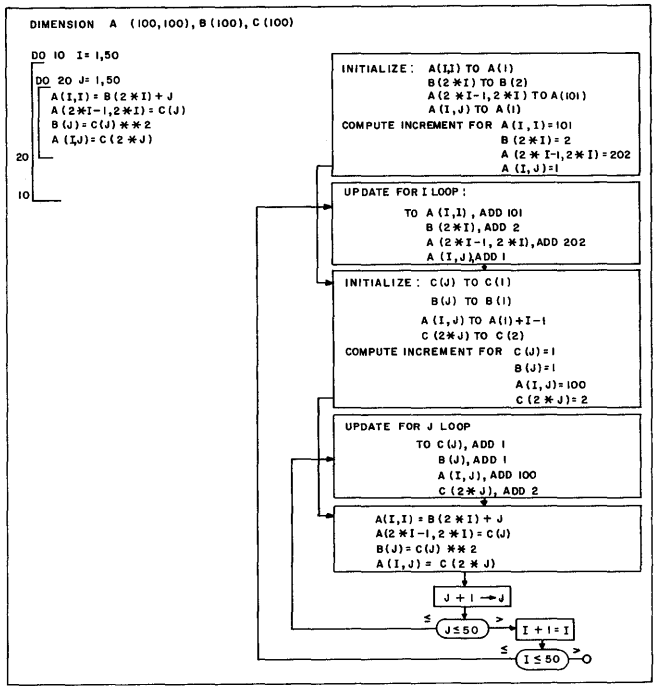


Figure 9. Subscript Optimization Example

meaningful saving when the last updating appears within a deeply nested series of DO loops, since the updating could occur hundreds of extraneous times.

EXAMPLE OF SUBSCRIPT OPTIMIZATION

A specific example of the operations performed in accomplishing subscript optimization is shown in Figure 9. Figure 9A defines two nested DO loops containing a large number of subscripted variables. The outer loop is controlled by the variable "I," and the inner loop is controlled by the variable "J." Figure 9B shows the operations performed and (by means of arrows) the sequence in which they are executed. The six major operations are: (1) initializing the "I" loop, (2) updating the subscript combinations at the end of the "I" loop, (3) initializing the "J" loop, (4) updating the subscript combinations at the end of the "J" loop, (5) executing the code, and (6) testing whether control is to return to the I loop, J loop, or main program after the code is executed.

In the first initialization area, entered at the outset of the I loop (see Figure 9B, top), the initial values for all subscripts that can be computed at this time are computed. Also computed here are the increments to be added to each subscript at the appropriate times by the update routines. These increments are determined in the initialization area rather than in the updating areas so that it will not be necessary to recompute them repetitively each time they are encountered. (Initialization is done only once, whereas updating is done every time a loop is entered, except the first time.)

Immediately following the initialization area is the update area, where the increments computed in the initialization area are actually applied for the next iteration. As mentioned earlier, its position after the initialization area is unusual, and most other compilers place it at the end of the code. However, placing it here eliminates the one unnecessary updating that occurs when the control variable exceeds its ultimate value. In this particular example, the result is a saving of 50 extraneous updatings of the J loop, which would otherwise occur for every iteration of the I loop. Notice that the updating area is bypassed the first time, since it is not necessary to update immediately after initialization.

In the second initialization area are computed all the subscript combinations which are a function of the J loop. As before, the process involves computing the increments to be added to the subscripts, as well as the initial values. The updating area following the second initialization area is also bypassed the first time, and the code is immediately executed. At the conclusion of the code sequence, the control variable J is incremented, and a test is made to determine if it has exceeded the terminal value. If it has not, control goes to the update area for J, where the already computed increments are added directly to each subscript that is a function of J. Thus, no repetitive computations are performed. The code can now be executed again. The loop continues in this fashion until the terminal value of J (*i.e.*, 50) is exceeded. Then the I loop variables are updated.

TWO-PASS OPERATION

As a final point in this discussion of code optimization, it should be noted that although other compilers attempt far less code optimization, they still require three or four passes over the input tape to perform compilation, as compared with two passes for Honeywell's AUTOMATH 1800 compiler.

READ-BACKWARD LOADING AND SEARCHING

The very high speeds of compilation achieved with the AUTOMATH 1800 compiler caused some interesting tape-manipulation problems, which are

worth mentioning because they serve to dramatize the speed of compilation. For example, the short time required to rewind the program tape was amounting to a significant percentage of the total compilation time. Even the time spent by a tape unit in resetting itself after a rewind operation was found to be unacceptable.

The first problem was solved by writing on the program tape in such a way that it can immediately be read backwards, instead of having to be rewound and read forward. The second problem was eliminated by using read-backward searching to reposition all tapes.

LANGUAGE

Based on FORTRAN IV language, the AUTOMATH 1800 compiler includes all features associated with FORTRAN IV, such as double precision, complex and logical data types and their operations, labeled common statements, adjustable-dimensioned arrays, logical IF statements, etc.

Some additions to the language are also handled. The DATA statement, conventionally used to initialize variables at loading time, is also available in an alternate form that permits arrays to be reinitialized during execution. Arrays may not only be adjustable, but also truly variable in size with their sizes set by call arguments or values in a common area. Hollerith and octal constants may be employed in the source program.

ASSEMBLY LANGUAGE PROCESSOR

Programs written in the ARGUS assembly language of the H-1800 and H-800 can be called by programs written in FORTRAN IV language. Also, compiler-originated programs can be called by those written in assembly language. Complete compatibility is attained through a separate assembly-language processor, called in by control card, which translates assembly language into a form identical to the output of the compiler. The result is complete flexibility in the intermixing and mating of programs written in either source language.

The processor permits the use of all machine-language functions (including masking) and several AUTOMATH/FORTRAN statements, such as CALL, RETURN, and the specification statements DIMENSION, COMMON, INTEGER, REAL, COMPLEX, etc. It will also accept FORTRAN input-output statements.

DIAGNOSTIC AND CHECKOUT SYSTEMS

"Debugging" an object program is simplified by an elaborate diagnostic system built into the AUTOMATH 1800 compiler that detects a wide variety of source-program errors and provides the programmer with a detailed explanation of the cause(s). More than 350 different English-language error messages are included, and the appropriate message is referenced directly to the specific statement in error. In addition to positively identifying "fatal" errors that prohibit compilation, the compiler also questions unusual situations which may or may not be errors. These are noted as "probable" errors, but do not inhibit execution of the program.

Checkout of an object program is facilitated by source-language derails, which permit both dynamic and terminal dumping of tapes and memory. The resulting reports are automatically edited into the format of the variable being dumped.

EQUIPMENT REQUIREMENTS

AUTOMATH 1800 employs 15,000 words of core storage when compiling. This includes the memory used by the card-to-tape conversion routine, tape-to-printer conversion routine, and tape-to-punch routine. In a 16,000-word system, the remaining 1,000 words of memory are available for the monitor which controls and integrates AUTOMATH with other systems.

Compile-and-execute operation employs the following tape units:

1. Common input tape.
2. Common printer output tape.
3. Common punch output tape.
4. Compiler system tape.
5. Binary program tape.
6. Work tape (also available to object programs during their execution).
7. Work tape (also available to object programs during their execution).

If coordinated input and output tape swapping is employed, two additional tapes are needed to alternate with the common input tape and common output tape. If more than one printer is employed in the coordinated operation, one additional tape must be used for each additional printer. If subprograms are to be retrieved from a library tape as an alternative to binary card input, an additional tape is required for this function. Object programs may use as many other tapes as can be connected to the system.

ADMIRAL -- A NEW H-800/1800 MONITOR SYSTEM

General Description

ADMIRAL (Automatic Dynamic Monitor with Immediate Relocation, Allocation, and Loading) is a monitor system designed to control the scheduling, loading and starting of programs and to supervise their execution on an H-800 or H-1800. The monitor is designed to remain permanently in memory and to operate in parallel with object programs. It performs the following operations:

1. Accepts scheduling data from the operator at any time. The operator tells the monitor which programs he wishes to run and the priority for running them. These programs are automatically loaded and started as hardware becomes available.
2. Allocates memory and peripheral units for all programs which are run and informs the operator of these assignments.
3. Loads each program into its allocated memory and starts it.
4. Provides dynamic diagnostic dumps during program operation and emergency dumps in case of program hangup.
5. Includes a peripheral equipment coordinator and routines to simulate off-line peripheral conversions.
6. Contains the orthocorrection routine.
7. Sets logical restart points, executes restarts, modifies hardware allocation, and provides other services at the request of running programs.

The monitor is capable of operating all systems programs (compilers, assemblers, program amintenance and selection routines), as well as object programs from a variety of programming systems such as Automath, COBOL, ARGUS and FACT. The modular design of the monitor will permit easy modification and expansion.

It is planned that the introduction of high priority and real-time jobs will automatically initiate the setting of rerun points for programs running at that time before terminating them. The terminated programs will then be restarted from these rerun points after completion of the high priority or real-time job. (A job is defined as a set of logically related programs).

Structure

The new monitor system is modular in design. Since its design allows for continual growth, it will handle all present and future Honeywell programming systems for the H-800 and H-1800. They system consists of a master (or primary) monitor and one local (or secondary) monitor for each programming system. The master monitor will control an object program through the local monitor of the programming system to which it belongs. A new programming system will merely require the addition of a new local monitor. Maintenance and refinement of in-being local monitors are also easily accomplished.

Above the minimum requirement of 1K of memory, the new monitor system will use only the amount of memory allocated to it by the operator. Optimal memory utilization is achieved by means of a dynamic segmentation technique, combined with the ability to recognize the relative frequency of use of each routine. Only the most frequently used segments are kept in memory. Those that are less frequently used are loaded as needed. The more memory the monitor receives, the more routines it can hold in memory; therefore, the more efficiently it will operate.

The Monitor System Tape will contain the following:

Master Monitor
All Local Monitors
UTILITY Programs
  Automath Compiler
  Standard Service Routines
    (e.g., THOR, editing routines,
     peripheral equipment coordinator, etc.)

Human Operator

It will be the operator's responsibility to specify the parameters for a day's run. He will introduce scheduling data to the monitor at anytime during a run (i.e., job priority, tape drive relationships, dynamic dump parameters, etc.). Emergency action may be initiated from the console. The operator will be notified by the monitor of all schedule changes and will retain the means to override the monitor's decisions.

Master Monitor

The master monitor will allocate memory, assign tape drives and peripheral equipment, and interpret monitor calls. It will schedule each program according to logical dependency, job priority and hardware requirements. Dynamic relocation is initiated by this monitor and completed by the local monitor.

Local Monitor

In addition to the machine requirements of the primary monitor, each programming system (e.g., ARGUS, COBOL, FACT, etc.) will require some additional equipment for a local monitor. The local monitor will control all object programs for that system. Local monitors will be brought in from the Monitor System Tape in overlays using memory included in the over-all requirement for the master monitor. In addition, one tape drive will be necessary for the program tape for each system (e.g., the MRT for ARGUS object programs).

ARGUS - THE HONEYWELL 800/1800 ASSEMBLY SYSTEM

Argus is a body of related programs comprising a powerful Assembly System, an outstanding Program Test System, an automatic Production Monitor, and an efficient Program Scheduler. Argus, as all software, is provided and maintained free of charge for all Honeywell 800/1800 users.

Argus, the Automatic Routine Generating and Updating System, is the core of the integrated automatic programming system for the Honeywell 1800-II. Argus is designed to minimize programmer effort and to maximize the efficiency of every phase of program preparation, from the initial coding through the checkout phase to actual production. Wherever possible, the burden of routine, clerical operations is lifted from the programmer. The file of programs approach, whereby batches of programs are assembled, tested, modified, and scheduled for production minimizes setup time by eliminating a great multiplicity of brief, repetitive computer runs. The dynamic dumping technique employed by the Program Test System enables batches of programs to be tested at full machine speed and without interruption. Diagnostic information is obtained without manual intervention, even if a programming error forces premature termination of a particular program under test. In short, ARGUS achieves a mating between the efficiency of program preparation and the remarkable efficiency of production; made possible by Honeywell parallel processing.

ARGUS is composed of the following principle elements:

1. An Assembly Program which translates symbolic coding and produces operating programs in machine language (binary) on magnetic tape.
2. A Library of Routines containing both subroutines and macro routines (i.e. Matrix manipulations, curve fitting, sort routines etc.) each thoroughly tested and capable of being incorporated into any program during assembly by the inclusion of a single pseudo instruction.

3. A Library Additions and Maintenance Program (LAMP) for adding and deleting routines and modifying existing routines in the library.

4. A Program Test System (PTS) which operates a file of unchecked programs at full machine speed, automatically obtaining requested information at points specified by the programmer for later analysis of program operation.

5. An Executive System which schedules checked-out programs for parallel processing, based on their individual hardware requirements, timing, and urgency, and then automatically loads and executes the scheduled programs.

PERT - PROGRAM EVALUATION AND REVIEW TECHNIQUE

PERT is a management control method for estimating the time required to complete a project and for evaluating the projects present status. PERT is based on the principle that a small percentage of the activities involved in a major project control the schedule for the entire project; therefore, preferred handling of these critical activities-once they are recognized-will result in better control in both time and cost.

Essentially the "Pert" program issues a report just prior to the predetermined estimated time of completion of a job indicating that this job should be completed, and if not completed, compensating action should be taken. The same report might indicate that one or more of the other jobs in the network has been completed prior to schedule, and the manpower might be put to effective use on the job that is behind schedule.

INPUT

Punched Cards - may be recorded on a master tape.

Header Card #1 - The information in this card is used for the identification of the network, the network master tape, and to control the special features of the program.

Header Card #2 - The information in this card is used for headings on each page of the output and pertains to the network to be processed.

Title Cards - The use of these cards is optional. A maximum of 4 cards supply the title that appears on the list. Cards may be in random sequence and must be between header card 2 and the detail cards.

Detail Cards - Each card contains all the information pertaining to one activity. Detail cards must be preceded by Header Cards 1 & 2. Detail cards can be in random sequence.

End Card - The program will cease processing input cards as soon as it completes reading this card.

OUTPUT

For each network processed, the program will generate the following lists:

A. Working Outputs

Diagnostics - For each error encountered in the detail cards the program will print a line to identify the activity causing the error and a second line indicating the type of error.

Markup List - A list of the entire network. The purpose of this list is for modifying the network.

Latest Modification List - It lists only those activities which have been modified during this run.

Networks on Master Tape List - This list consists of the title and file identification of every network existing in the master tape.

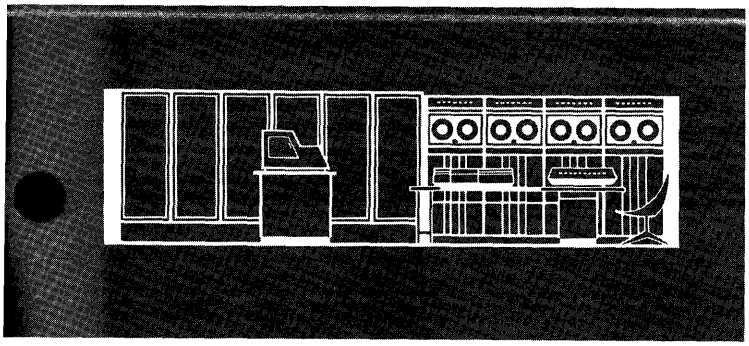
Summary List - Only those events which have been previously indicated as summary events will be listed.

Sequence MAJOR - Slack
MINOR - Expected activity completion time.

B. Final Output - The final output reports indicate the results of the calculations and the status of each activity. The following listings may be specified by programmer option:

1. Critical path - cumulative expected time. Sequence MAJOR-Slack; MINOR - cumulative expected time.
2. Succeeding event. Sequence MAJOR - Succeeding event; MINOR - Cumulative expected time

3. Cumulative expected time.
4. Active activities in sequence by slack time.
5. Department - Critical path. Sequence MAJOR-Department;
   MINOR-Slack time
6. Latest required time.



HONEYWELL EDP WELLESLEY HILLS 81, MASSACHUSETTS

ADMIRAL

A Dynamic Multiprogram Operating System for the Honeywell 800/1800

By Jacques Bouvard*

Bulletin No. 111
July, 1963

The ADMIRAL operating system harnesses the power of automatic parallel processing in the Honeywell 800 and 1800 by relieving the user of the tasks of work load scheduling and execution supervision. Working with a stack of jobs to be processed, ADMIRAL selects and executes jobs in accordance with their stated priorities and in such a way as to maintain constant optimal utilization of the available equipment configuration. Activated only when called by a running program or by the operator, ADMIRAL interrupts only the calling program; all other programs continue to operate in parallel. Efficient and flexible control is provided over all types of computer operation.

THE PROBLEM OF EFFECTIVE MACHINE UTILIZATION

The problem-solving capabilities of new computing systems have grown at a spectacular rate over the past few years. Higher internal speed and larger storage capacity have considerably expanded the realm of computer applications. Many computer users, however, have come to realize that these improvements alone are not sufficient to increase significantly their rate of throughput. Since the penalty for inefficient machine usage has drastically increased with the rising costs of these new systems, a great deal of attention must be devoted to the problem of achieving optimal system utilization. Underlying this problem are a number of factors quite heavily dependent upon the individual installation's environment. The imbalance between processing and input-output rates is in most cases the greatest single cause of inefficiency. As internal processing speeds increase, the time required to enter the data and to record the results rapidly becomes the limiting factor. During a typical run, the central processor may remain idle most of the time while input-output facilities are overloaded.

Another frequent source of inefficiency stems from the fact that system configurations are usually tailored to meet the requirements of a major application. The rest of the work load often consists of small jobs which may still account for a large portion of the total running time, even though they only utilize a small fraction of the system capacity.

Finally, extensive loss of system time may also be caused by operator intervention. Inter-job setup is particularly critical when the work load consists of short jobs. In such cases, setup time may exceed processing time. In case of hang-up or other unexpected situations, the operator may be required to analyze the error condition, obtain diagnostics, and initiate corrective action. Because the operator is not necessarily familiar with the programs in operation, this process is slow, inefficient, and prone to error which may result in further loss of time.

* Staff Analyst, Programming Systems Division, Honeywell Electronic Data Processing.

DSI-188-5863
Printed in U.S.A.

To cope effectively with these problems, Honeywell offers the concept of parallel processing. A standard hardware feature on the Honeywell 800 and 1800 Data Processing Systems, parallel processing enables the user to obtain optimal system utilization automatically, continuously, and at no extra cost. Automatic time-sharing of the central processor and the input-output channels eliminates the imbalance between input-output and processing activities. Sharing of memory space and external devices provides for the absorption of any excess capacity and eliminates idle system time during inter-job setup.

Finally, parallel processing permits the complete integration of the system so that it can be effectively coordinated from a single location. It removes the need for additional and costly satellite computers or other off-line equipment, simplifies the over-all system operation, and reduces turnaround time.

THE ADMIRAL OPERATING SYSTEM

To help the user in fully exploiting the capabilities of parallel processing, Honeywell provides an outstanding operating system called ADMIRAL. This system should not be confused with so-called super programs, which are required to achieve multiprogramming on conventional machines equipped with only one set of program control registers. The super program is in continuous operation. No such program is required on the Honeywell 800/1800, since parallel processing is purely a hardware feature which automatically controls time-sharing among programs running simultaneously.

ADMIRAL is a management program whose only purpose is to relieve the user of the burden of work load scheduling and detailed execution supervision.

DYNAMIC SCHEDULING

ADMIRAL's major function is scheduling. At the beginning of the run and at any time thereafter, ADMIRAL accepts from the human operator a simple description of the work load and of the system configuration. The job requests are automatically stacked in a queue, which is examined by the scheduling function each time a change occurs in the system utilization. Scheduling is performed dynamically. New jobs may be added to the list of waiting jobs and selected for execution without interrupting the run. Furthermore, there is no need for advance knowledge of the total work load prior to the start of execution. This feature removes the need for batching and contributes to reduced turnaround time while enhancing operating flexibility.

Scheduling is performed on a job basis rather than on a program basis. A job consists of a series of logically related programs which depend upon each other for input and output. Programs within jobs are selected for execution in a serial fashion, and thus precedence relations are easily satisfied. The selection of new jobs from the waiting list is based upon a priority coefficient which is assigned by the operator at the time the job is submitted and which can be modified at any time during the run.

The scheduling algorithm is aimed at achieving optimal system utilization while recognizing a number of practical considerations. Optimization of hardware usage is achieved by always attempting to saturate the system capacity. In general, the scheduling function attempts to run as many programs as possible in parallel. Jobs are examined in the order of their priorities. However, if a conflict develops between the hardware requirements of a given job and the current unused system capacity, this job is passed over and the scheduling function proceeds to inspect lesser priority jobs. To avoid permanently bypassing a job with large hardware requirements, the priority of such a job is automatically raised each time the job is passed over, up to a maximum priority. When such a job exists in the waiting stack, further scheduling is momentarily suspended until sufficient hardware becomes available to accommodate that job.

Another important principle is that, aside from absolute urgency considerations, once a job has been started, it should be completed as soon as possible. For this reason, whenever the scheduling function is activated following the completion of a program, it first attempts to select another eligible program from the jobs in current execution. A program is eligible for execution whenever its logical predecessors have been completed. If and only if all eligible programs in the current jobs have been started, the scheduling function selects new jobs from the waiting list. This method prevents starting too many jobs at once and having to interrupt jobs frequently during execution, which might entail a considerable amount of tape file saving and remounting operations.

Processing of high-priority jobs with no delay implies the capability to interrupt current jobs having lesser priority and conflicting equipment requirements. This is achieved by means of a set of absolute priorities which are recognized by the scheduling function. There are two types of absolute priorities: TOP and SUPER. TOP priority jobs are immediately

selected from the job list and started as soon as enough equipment becomes available. No other program in any other job can be started so long as an eligible program in a TOP priority job is waiting. SUPER priority jobs are handled in the same manner, except that the scheduling function will interrupt as many programs as necessary to insure continuous processing of such a job. This scheduling scheme insures automatic and optimal system utilization while providing the operator with the possibility of instantaneous manual override.

HARDWARE ALLOCATION

The scheduling function also involves hardware allocation for each program. As soon as a program becomes eligible, ADMIRAL automatically determines by inspecting the program the amount and nature of equipment that it requires to operate. These requirements (in terms of memory space, tape units, and other peripheral devices) are matched against the currently available system configuration to determine if the program can run now and, if so, to perform detailed allocation.

To fit a new program in memory, ADMIRAL attempts to find the smallest unused area capable of accommodating the program. This scheme avoids splitting memory into a number of small spaces. If, however, no single empty space large enough to hold the program exists and if, on the other hand, the total unused space exceeds the program's size, all other programs currently in memory are compressed toward the lower end of memory. This process creates a gap of available space at the upper end of memory into which the new program can be fitted.

In allocating tapes, ADMIRAL attempts to distribute the load among the available input-output channels. The tape allocation scheme is also aimed at minimizing the amount of tape handling by the operator between programs. When several programs of the same job manipulate the same logical file, interprogram setup can be reduced by insuring that the same physical unit is assigned to this file throughout the execution of the job. ADMIRAL interprets a statement on the job descriptor which specifies a logical equivalence between the symbolic addresses of such a file in the various programs of the job and assigns a single physical tape unit to this file.

To enhance the flexibility of the scheduling system and achieve greater equipment utilization, ADMIRAL also provides for dynamic hardware allocation throughout execution. Should the hardware requirements of a program vary during the course of execution, equipment allocation may be performed only when the equipment is actually needed, upon request of the program itself. Similarly, equipment may be released back to the system when no longer needed. For example, a program may initially require the use of a card reader to enter a few parameters, or it may use a large number of tapes during a sorting phase and only a few during a reporting phase. Such a program can request the necessary equipment when it is needed, use it only as long as necessary, and release it for use by other programs. This dynamic allocation feature is also particularly useful when dealing with programs whose requirements cannot be accurately determined in advance because they depend upon the particular data being processed.

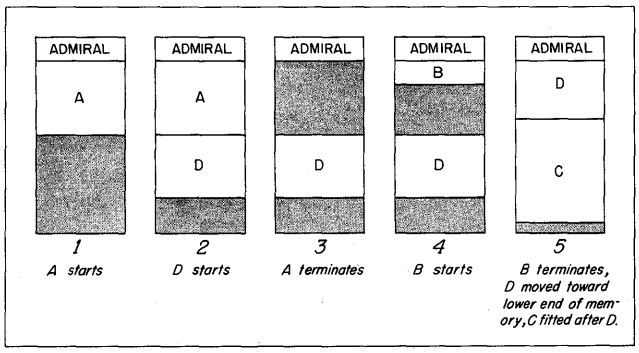


Figure 1. Dynamic Allocation of Memory

EXAMPLE OF DYNAMIC MEMORY ALLOCATION

The principle of dynamic allocation of memory can be illustrated by means of a simple example involving two jobs which are submitted at the same time. The first job consists of the three programs A, B, and C; the second job involves only the program D. Figure 1 shows the manner in which ADMIRAL allocates memory to these two jobs at five different points in time.

In the first panel, program A is selected and assigned an area of memory just after the area used by ADMIRAL. Program D of the second job is then immediately selected and allocated, as shown in panel 2. The third panel shows the release of the memory space used by program A when that program terminates. Program B of the same job is then immediately selected and fitted into the area released by program A, as shown in panel 4. However, when

program B terminates and its space is released, no single memory area is available which can accommodate program C. As panel 5 illustrates, ADMIRAL relocates program D to the lowest available memory area, creating a space at the upper end of memory which is large enough to hold program C.

CHECKOUT FACILITIES

ADMIRAL is a checkout as well as a production operating system. It includes a complete set of debugging aids such as a program correction facility, test data distribution, pinpoint dynamic dumps with flexible editing format, and memory difference dumps (in which only those registers which changed since the previous dump are edited). Dumps can be requested when the program is submitted for execution or they can be obtained upon operator command at any time during the run.

RESTARTS

ADMIRAL incorporates a number of automatic restart features and manual recovery procedures. Restart or anchor points are set upon program request when reaching convenient logical breakpoints. Intermediate results may thus be preserved as a guarantee against possible system malfunction. Restart points are also set automatically when a program must be interrupted. ADMIRAL dumps the contents of memory allotted to that program and records the current positions of the external files. Later resumption is achieved without loss of computer time by restoring the program to the status it was in at the time of setting the restart point.

COMMUNICATION WITH THE OPERATOR

To keep the human operator informed of the progress of the run, ADMIRAL logs the identification of each program at the start and at the end of the program. It instructs the operator of any decision it may take concerning job selection or hardware allocation and requests manual operations where necessary. The operator may override ADMIRAL's decisions through a set of commands issued from the console. Such commands may consist of changing job priority, adding or deleting jobs, interrupting, dumping, or restarting a program (or system of programs), modifying the system configuration, etc. ADMIRAL's response to these commands is immediate.

INTERNAL STRUCTURE

ADMIRAL is compatible with all of the Honeywell 800/1800 programming systems, including ARGUS, AUTOMATH, FACT, and COBOL. To achieve compatibility without inflicting any undue restrictions upon these systems, ADMIRAL consists of a so-called Master (or primary) Monitor which communicates with a number of local (or secondary) monitors associated with the individual programming systems. The Master Monitor is a general-purpose control package which assumes the management and supervision of the over-all system. Each local monitor is specialized for the particular needs and structure of the corresponding programming system and accomplishes detailed service functions such as loading, relocation, and segmentation control. The entire system is placed under control of a coordinating function, permanently in memory and capable of recognizing requests issued by the operator or by the running programs. Each call is analyzed and the proper function executed by one or more independent subfunctions which can be brought in and out of memory as required.

At the start of the run, the operator assigns to ADMIRAL a certain memory area which is automatically divided into three portions. One portion is allocated to the coordinator; another is used to stack up the job descriptors; while the third constitutes a storage pool to be divided among the various subfunctions and associated communication arrays. New subfunctions are placed at the highest available locations within the storage pool, while arrays are assigned starting at the lowest locations. In this manner, the available space is concentrated in the middle of the storage pool and can be used to accommodate either new subfunctions or new arrays.

Control transfers between subfunctions are accomplished through the coordinator. Data communication between subfunctions is accomplished through overlapping arrays associated with each subfunction. Storage allocation for these arrays is performed by the coordinator each time a new subfunction is activated. This scheme eliminates the need for a fixed common array. It also provides for allocating arrays only when they are needed, for only the amount necessary, and for only as long as they are actively used.

Memory usage by the Monitor is further enhanced by means of dynamic segmentation. Each subfunction consists of a string of coding which remains invariant

throughout execution. This coding only involves indexed or indirect addressing and thus is not sensitive to actual allocation. Subfunctions can therefore be retained in memory as long as sufficient space remains available and can be reactivated at will. They can also be moved in memory without any need for code relocation. These properties provide for an extremely efficient and simple dynamic segmentation scheme which reduces the cost of the Monitor in terms of memory space and machine time.

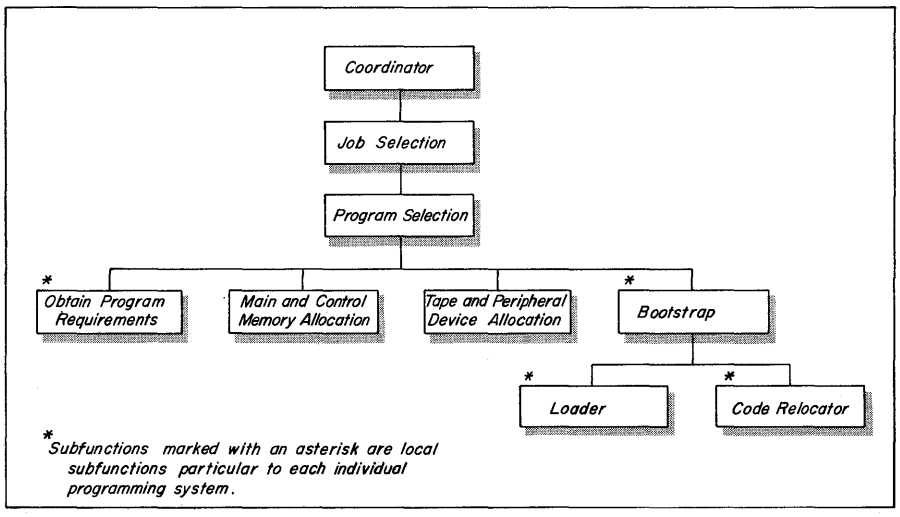


Figure 2. The ADMIRAL Scheduling Function

EXAMPLE OF ADMIRAL INTERNAL OPERATION

The interrelation between the Master Monitor and the local monitor subfunctions is illustrated in Figure 2. This example involves the operation of the scheduling function, which is activated by the termination of a program.

When a program terminates, the coordinator calls in the job selection subfunction to determine which job is to be examined first. Once a job has been chosen, the program selection subfunction checks to see if there exists any eligible program in that job. If so, a local monitor subfunction is called upon to examine the eligible program and determine its hardware requirements. Memory and peripheral allocation subfunctions are then used to decide whether sufficient equipment is currently available and to make detailed allocation. This information is then passed on to the bootstrap subfunction, which locates the first segment of the program, loads it, and relocates it by means of a local loader and relocator.

OPERATIONAL CHARACTERISTICS

The operation of ADMIRAL is quite similar to that of a real-time program: it normally remains dormant, becoming active only in response to a call issued by the operator or by one of the running programs. The cost of ADMIRAL in terms of computer time is minimized by the fact that it can operate in parallel with the rest of the work load. Only the calling program is interrupted when ADMIRAL becomes active; other programs proceed normally. The problem of queueing the calls to the Monitor is automatically resolved by the hardware with no time penalty. Whenever ADMIRAL is activated, an interlock is set which prevents further calls until completion of the current function. The method used is analogous to that employed by the machine to stack the demands on input-output channels. Calling programs are stalled as long as the Monitor remains in operation. Special provisions are made, however, to enable the operator to override this interlock when necessary.

Job descriptors are entered into the waiting list either via the console or by means of any conventional input device. Depending upon the situation, this input device may be assigned to ADMIRAL either permanently or only when the need arises to enter additional job descriptors. Full-time assignment of an input device is well suited to the case where the work load consists of a large number of short jobs; temporary assignment would probably be preferred in a production type of operation.

Job descriptors specify the identification and nature of the programs which compose the job. They may also include optional information such as absolute hardware assignments, tape consistency data, program corrections, dump requests, or test data. Test data is automatically written onto a tape assigned to the job.

The ADMIRAL system includes a library of utility programs which are available to the user for instantaneous call. These programs are scheduled and run like any other programs. They perform standard service operations such as tape handling, data editing, or peripheral conversion.

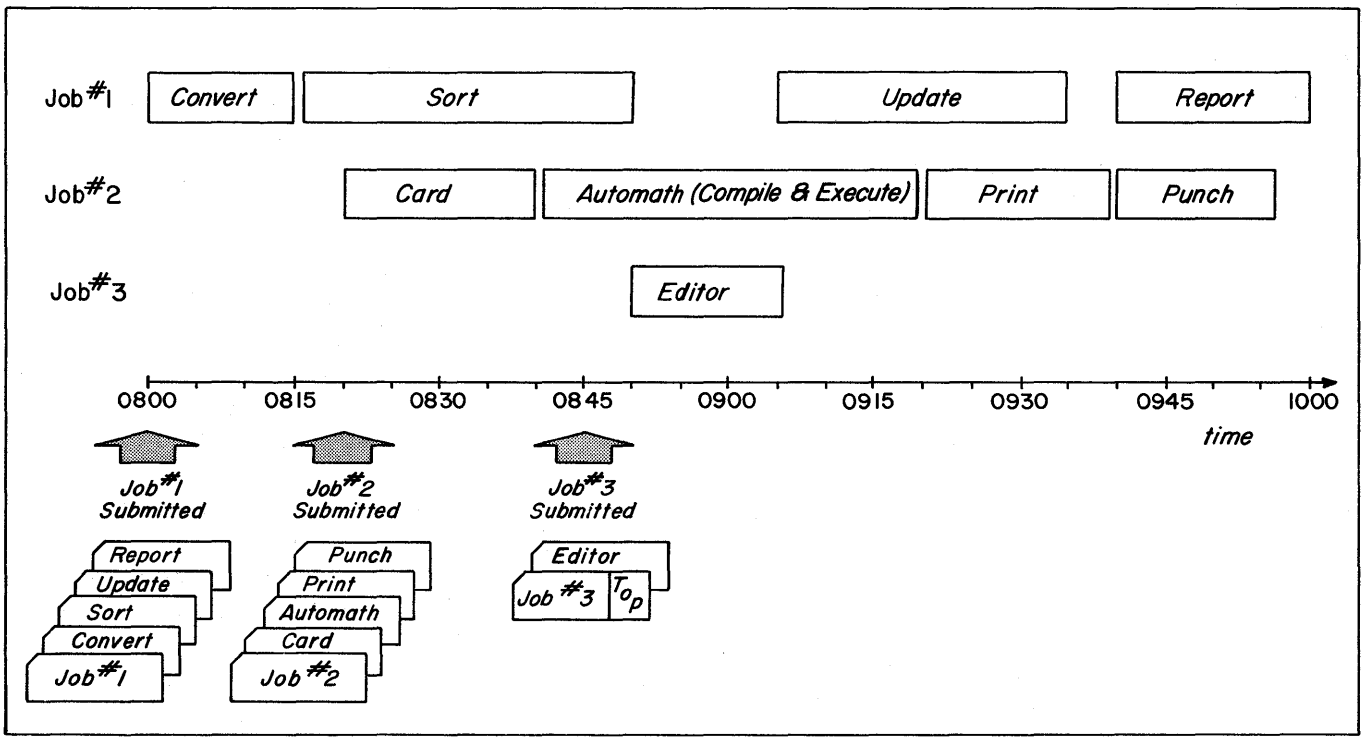


Figure 3. Example of Dynamic Scheduling

EXAMPLE OF OPERATION UNDER ADMIRAL CONTROL

The dynamic scheduling feature of ADMIRAL is illustrated in Figure 3, using a simple example which involves only three jobs. The first is a master file updating job consisting of four distinct programs: CONVERT, a card-to-tape conversion program which writes a transaction file on tape; SORT, which arranges the transactions in master file order; UPDATE, which uses the sorted transactions to produce an updated master file; and PRINT, which creates a printed report. The second job consists of a batch of Automath (scientific) subprograms to be compiled and executed. This job includes the programs CARD, which converts the source decks and associated test data onto a common input tape; AUTOMATH, which compiles and executes the source programs, writing dumps and program results onto a common output tape; PRINT, which edits and prints the output tape on the printer; and PUNCH, which punches the compiled subprograms on binary cards. The third job consists of the single program EDITOR, which edits and prints a report from a magnetic tape file.

In order to illustrate the way in which ADMIRAL handles the scheduling of these three jobs, it is necessary to make certain assumptions about the times that the jobs are submitted and about the running times of the various included programs. In Figure 3, it is assumed that the master file updating job is entered into the job list at 8:00 in the morning. Since all of the required equipment is available, the job is immediately scheduled and started. At 8:15, the card conversion is completed, and the sort is immediately selected by ADMIRAL and started.

At 8:20, the operator submits the scientific compilation job.[1] Again, all of the required equipment is available, so the card-to-tape operation is started immediately, and at 8:40 the Automath compiler begins the series of compilations.

The operator next learns at 8:45 that a report to be produced by the program EDITOR is urgently needed. The third job is accordingly submitted with

[1] This job could be performed using the input/output tape swapping method described in Bulletin 109 if additional equipment were available. However, this example assumes that conventional load-and-go operation must be used in order to run in parallel with the sort program on the available equipment configuration.

a TOP priority assignment. However, this job is temporarily delayed due to the unavailability of equipment. When the sort terminates at 8:50, the scheduling function, recognizing the existence of a TOP priority job in the waiting stack, interrupts job #1 and immediately schedules the report generator.

When, at 9:05, the TOP priority report is completed, job #1 is resumed with the program UPDATE. At 9:20, the series of Automath compilations and executions terminates, and the editing of the common output tape is started. Next, the UPDATE program is completed at 9:35; however, due to the unavailability of the printer, the program REPORT cannot be started until 9:40. At this time, the two programs REPORT and PUNCH are scheduled in parallel.

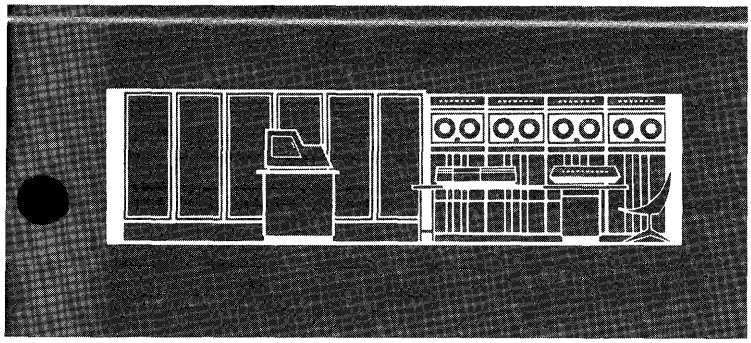
EQUIPMENT REQUIREMENTS

ADMIRAL may use a variable amount of equipment, depending upon the type of operation. It requires as a minimum one special register group (group 0), the first 1024 words of memory in any bank, and one magnetic tape unit. More equipment enables ADMIRAL to operate even more efficiently and to perform additional functions. For example, more memory can be used to hold more subfunctions in core storage and to reduce the need for access to tape. An additional tape can be used for dynamic dumping, logging, and storing restart information.

SUMMARY

The ADMIRAL operating system provides the user with a comprehensive and efficient computer-management tool. By fully exploiting the capabilities of parallel processing, optimal system utilization is continuously and automatically achieved with minimum overhead. Operational efficiency results from the extreme flexibility of the system. Dynamic scheduling eliminates the need for batching the work load, reduces turnaround time, and makes the computer more accessible to the user. Automatic operation reduces the burden on the operator and minimizes the chances of human error. Override procedures are available whenever necessary. Complete modularity in the subfunction design enhances the ease of maintenance and refinement of the system. It also provides for compatibility with a broad range of hardware configurations and applications and permits the system to meet the requirements of each particular user.

HONEYWELL
ELECTRONIC
DATA
PROCESSING
WELLESLEY HILLS 81,
MASSACHUSETTS



HONEYWELL EDP WELLESLEY HILLS 81, MASSACHUSETTS

ALPS

A New Linear Programming System for the Honeywell 800/1800

by E. E. Comerford[1]

Bulletin 113
October 1963

To assist managers in making sound, swift decisions, Honeywell offers a new Linear Programming System as a mathematical tool for operations research. ALPS (Advanced Linear Programming System), for use with the Honeywell 800 and 1800 Electronic Data Processing Systems, is not only the most versatile and extensive system of its kind today but, through the use of high-speed computation and parallel processing, the most economical and efficient. Significant, too, is the leadership shown by Honeywell EDP in implementing the most advanced variations of the Simplex Method and in providing for much larger problems than competitive systems can handle.

Note: This technical bulletin assumes some knowledge of linear programming (LP).

APPLICATIONS

Honeywell's new linear programming system is capable of solving problems in such diverse fields as industry and agriculture. Oil companies can employ ALPS to determine the most profitable mix of gasolines or the most economical way to run a refinery or series of refineries. Agriculturists can make use of ALPS to establish the most efficient division of acreage; while advertisers can use it to ascertain where and when to advertise. Anyone concerned with a mix problem can choose the most profitable blend for himself, be it of cocoa beans or caustic soda. Both the paper and the steel industry can minimize trim losses by using ALPS. These are but a few of the problems (see Table I for more) that can be solved with this powerful tool.

[1] Manager, Scientific Programming Department, Honeywell EDP.

GENERAL CHARACTERISTICS OF ALPS

ALPS is designed to be the most efficient linear programming system consistent with the state of the art. It can handle problems larger than any currently being solved by competitive systems. ALPS automatically handles both large and small problems efficiently, storing as much data as possible in the internal memory. It is a flexible tool for both the experimental analyst and the production manager. Briefly, here are some of the advantages which accrue to the ALPS user:

1. Ability to handle larger problems than other LP packages using comparable machine equipment.
2. Most efficient variations of the Simplex Method.
3. Complete repertoire of sensitivity analyses.
4. Absolute user control over the system.
5. Input compatibility with many other systems.

DSI-200-51063
Printed in U.S.A.

OPERATIONAL AREA	FACTOR EVALUATION	EFFECT OF ALPS ANALYSIS
Transportation & Distribution	Inventory costs; unit / bulk transportation costs; transit time rate; order processing rate; plant / distributor location / transportation cost comparison; rate / quantity of regional / future market demands; total / projected market requirements; total / future production capability / costs; area costs for plant / warehouse maintenance / installation.	Reduced inventory / transportation costs; increased order processing / delivery rate; reduced production costs; optimum security / service of market; optimum facilities expansion; reduced maintenance costs; accurate forecast of sales posture.
Chemical Blending	Raw material availability / costs; relative affinity / stability of individual compounds; stability / quality of final product; product quantity / market demand correlation.	Increased economy in materials purchase / production; optimum product quality; realistic production rate for market demands.
Communication Control	Quantity / type / frequency rate of incoming communication; quality / origin points of incoming communication; efficiency of communications handling; correlation of peak / slack communication load.	Increased customer service; increased rate and quality of communication flow; optimized operating efficiency; increased inventory / transportation / management control.
Personnel Evaluation	Available personnel skills / training / productivity / potential development; required skills / training of new personnel; quantity / quality / cost / time requirements of on-job training; salary / wage requirements at various job levels; salary / wage levels in industry / local area; type / number / point of hire of additional personnel for production rate increases; production rate of individual per job; number of employees required for desired production level.	Optimized use of available skills and training; cost reduction in personnel training; optimum selection of incoming personnel; decreased cost of turnover / idle personnel; decreased quantity / frequency of layoffs; increased quantity / quality of production.

TABLE I

A Few Applications of ALPS

6. Excellent output flexibility (intermediate and final results).
7. High digital precision.
8. Error detection, correction, and trouble shooting.
9. Modularity for ease of expansion.
10. Parallel processing of input, output, and computation.

SIZE OF PROBLEMS

The size of linear programming problems has been growing steadily ever since generalized systems were first introduced. Therefore, to be useful both now and in the future, an LP system must be able to handle very large problems. ALPS solves problems of more than 700 constraints using only 16,000 words of memory. With twice that amount of memory, more than 1000 constraints can be handled. ALPS utilizes any available memory in excess of the required 16,000 words to increase the efficiency of problem solving. In particular, the system uses such extra storage capacity to solve many of the smaller and medium-sized problems entirely within memory.

Most generalized packages are not even capable of dealing with problems having 700 constraints, regardless of the amount of memory used. The only other package which can handle 700 constraints in an LP problem requires 32,000 words of memory.

There is no limit to the number of column vectors which may be used with ALPS.

ALGORITHM

ALPS includes those variations of the Simplex Algorithm which recent research has proven optimal. This research, concerned with optimizing subsets of vectors, proves that handling two vectors at a time is a much more effective method of handling the product form of the inverse than the former method of first using one vector and then returning to get the other.

Several features ideally equip the Honeywell 800 and 1800 for handling a system which uses the product form of the inverse with suboptimization. Especially useful, for instance, are the automatic parallel proc-

essing ability and the characteristics of Honeywell magnetic tape units. The ability to read tape backwards greatly enhances the implementation of the algorithm because much tape movement is necessary to solve large problems using the product form. Parallel processing is used to overlap input, output, and computation as much as possible, even though the algorithm may require these operations in differing amounts. Another major advantage derives directly from the length of the machine word (48 bits). This length permits floating-point binary computations to be carried out to a precision of 11 decimal digits without the loss of time and memory space which is inherent in the use of double-precision arithmetic.

SYSTEM CONTROL

Control over the whole linear programming system is accomplished by cards called agenda. These cards allow the user to control the actions of the program, the sequence in which they are performed, and the amount of output produced. They determine whether the program will deliver one optimum solution or whether sensitivity analyses are to be performed. They can be used to interrupt a run and to restart it later. Since the user has such control over the sequence and the number of tasks performed, he can operate efficiently in both the experimental and the production mode.

For example, a user sets up a linear programming model which will determine the best kinds of insurance to buy for the least amount of money. He executes the program for a single solution by punching several agenda cards, but on the first attempt he obtains an incorrect solution because of a scaling error in the data. To find his mistake, he wishes to examine a few intermediate solutions as well as his original matrix. By adding a few agenda cards on his next run, he finds the error, corrects it, and is ready to run again. This time he wishes to obtain two solutions, one with current prices and one based on anticipated price changes, which requires a few more agenda cards. In the midst of his run, it becomes necessary to perform a high-priority program. By punching one card and placing it in the card reader, he can interrupt the program in such a way that it is possible to restart at any time. The options are nearly unlimited.

INPUT

Data are punched on cards. If a problem is quite small, it is reasonable to have the data cards read into the machine along with the agenda cards. However, the recommended procedure for most problems is to prepare a tape ahead of time, operating in parallel with some other program(s). Thus a card-image tape is produced with a minimum amount of machine time. Machine time for the system is also decreased when it receives data from tape rather than from cards.

Input formats, like the rest of the system, have been designed to be expandable in order to incorporate the results of future research. However, since there has been some standardization of card formats for LP programs, ALPS includes input routines which achieve compatibility with much data already punched on cards.

OUTPUT

A formal report can be obtained, automatically omitting printing of any intermediate data, by the use of a few agenda cards. However, when a user sets up a model, he must expect to do a certain amount of experimentation before he obtains useful answers. When he arrives at the stage where he is obtaining correct answers then, in all probability, he will wish to experiment by changing his program or by performing any of the sensitivity options explained below. Any of these experiments are performed by adding the appropriate agenda card(s) to the deck.

SENSITIVITY ANALYSIS

When a mathematical model is set up to represent a physical situation, the solution may be mathematically correct but not readily transferable to the situation represented. If linear programming techniques are used to determine which magazines should carry advertisements for a product, what happens when one of the magazines selected goes on strike? Which is the next best one in which to place the ad?

If it is possible to determine the best kinds of insurance to buy at current prices, can this optimum solution be used as a starting point to determine what to buy if prices change? The technique known as sensitivity analysis is used to find two solutions in a minimum amount of time, one for current prices and one for anticipated prices, the former being used as a starting point for the latter.

Obtaining more than one solution to a problem is necessary since the physical world is dynamic rather than static. ALPS enables the user to find answers to the questions posed above, which illustrate only two out of a complete repertoire of such capabilities. LP techniques have advanced to the stage where delivering one optimum solution to a problem is only the beginning of the requirements of a system. It is necessary to incorporate facilities, such as those built into

ALPS, for ascertaining families of optima. All of these capabilities are activated by agenda cards.

USER INTERVENTION

The composite algorithm incorporated in ALPS automatically selects and rejects vectors until an optimum feasible solution is reached, provided that one exists. Various techniques are employed throughout the system to obtain an accurate solution in a minimum amount of machine time. If the user never desires to interfere with the selection and rejection of vectors, he can be assured that his optimum solution will be obtained in a most efficient manner. However, when an experimenter has had considerable experience in solving LP problems, particularly problems quite similar in nature, he may want to change the path which would be followed if everything were automatic. He may be able to estimate which vectors are close to optimal, which will allow him to save considerable machine time in the case of a large problem. It is possible to interrupt the progress of the program and to supply his own estimates by means of agenda cards.

AUTOMATIC INVERSION

An automatic inversion is performed at intervals which are determined by the LP program and the clock. The purpose of this inversion is to minimize rounding errors and to reduce the number of computations. The user can change the interval time by means of an agenda card.

As mentioned earlier, the precision of the system also results from the size of the Honeywell floating-point binary word. The Honeywell 800/1800 floating-point unit is designed to provide maximum precision with maximum speed. It requires negligibly more time to operate with 11 decimal digits in floating-point arithmetic than it does to operate with less precision.

INTERRUPTION AND RESTART

Facilities are included for getting off the machine due to a higher-priority program or due to an emergency. Appropriate facilities for restarting are provided for each case.

ERROR DETECTION, CORRECTION, AND TROUBLE SHOOTING

An extensive set of devices for detecting machine trouble are built into the LP program and appropriate messages are stored on tape. Corrective measures are also built in. Only in those cases where the program cannot proceed further is a message printed on the console and the machine stopped. Methods for obtaining intermediate data and for examining the LP programs themselves are available in the event of program failure.

EQUIPMENT REQUIREMENTS

The minimum machine configuration required to run ALPS includes:

H-800 or 1800 central processor
Floating-point unit
4 memory modules (16,384 words)
2 groups of special registers
6 magnetic tape units
1 card reader on-line
1 printer on-line
1 programmed clock

The system can take advantage of additional memory modules to increase the size of the problems that can be handled. The availability of additional memory also permits wholly internal solutions to many of the smaller problems. Most problems can be solved using six magnetic tape units; however, the provision of additional tape units extends the combinations of optional capabilities of the system that can be called on to solve any given problem.

AGENDA CARDS

All of the agenda currently available in the ALPS system are listed in the following paragraphs. In general, the parameters listed are required to specify to the various programs such information as the tape unit containing the matrix, whether or not the tape should be rewound after use, which intermediate data should be stored, etc. Certain key agenda are described briefly; others are merely listed to give an over-all picture of the system and its mode of operation. Only a few of the innumerable options available to the user are illustrated.

Main Simplex Algorithm

NORMAL, R, B

When this agendum card is read into memory, the program finds a feasible solution if one exists. If no feasible solution exists or if an unbounded solution is found, processing is stopped for operator action. The problem is performed internally if sufficient storage is available.

The composite algorithm can handle any or all of the following conditions simultaneously and without any preliminary sign change:

1. Mixed restraints consisting of strict equalities and inequalities of both senses.
2. Mixed positive and negative right-hand-side elements.

3. Linearly dependent restraints.
4. Initial infeasible solutions.

Sensitivity Analysis
DO.D/J, R, gN, gN
COST.RANGING
COST.RANGING, NAMES
COST.RANGING, CLASS, W, F
RHS.RANGING
RHS.RANGING, NAMES
RHS.RANGING, CLASS, W, F
DO.PLP, C, T, D, K, SPEC, COMP
DO.PCR,Q, P, D
DIFFERENCE,B_1, B_2, K
SENSIT
SENSIT, K
EXTSOL
EXTSOL, K
ADDRHS, cu
RESET

As an example of sensitivity analysis, cost ranging and cost parameterization can be applied to the insurance problem previously mentioned.

Ranging consists in varying the costs of the basis vectors one at a time while holding all others constant. The cost is varied without changing the optimal basis until a maximum is found. If the vector whose cost range is being determined actually assumes the maximum value, the solution is no longer optimal and another non-basic vector must be brought in. When the ranging agendum is used, the program prints the maximum cost and the vector which would enter the basis if the maximum cost were actually assumed. The minimum value, which is ascertained in an analogous manner, is also printed along with the non-basic vector which would enter the basis if the minimum value were actually assumed. The agendum DO.PCR can be used to change all costs simultaneously, a process known as cost parameterization.

RHS.RANGING refers to ranging with regard to the right-hand side of the set of linear constraints. Ranging and parameterization can also be performed on the requirements vector.

Input
INPUT, q, n, COUNTS, REWIND, PUNCH
REVISE
REVISE, q, n, PUNCH, REWIND, LIST
CORRECT, BCD, cu, cu, cu, n, J, Y, REWIND, LOW, GENERAL
CARD/TAPE, cu, LOW

The input agenda are used not only for converting matrix elements to a form suitable for operation by other agenda, but also for innumerable editing and checking features. They allow the user to specify the number of functionals associated with the current problem. They are used to obtain lists of row names and column-vector names along with counts of non-zero elements. They also create any required slack vectors. The program checks for duplicate row names and duplicate column names.

REVISE allows modifications and/or extensions to a matrix tape which has been edited and converted. It also allows the creation of an edited and converted tape by selecting rows and columns from a card-image tape. Other agenda cards are used to correct tapes and to convert card images to tape.

Output
OUTPUT
OUTPUT, OLD
TABLEAU
TABLEAU, FROMYAAAAA
TABLEAU, INVERSE
TABLEAU, NAMES
COMPILE, cu, cu, N, REWIND
REPORT, cu, N
STACK,REPORT, cu, cu
PICTURE, HELP
TITLE
HEADING, SAVE

OUTPUT processes an intermediate tape onto which various outputs have been written during the course of ALPS processing.

TABLEAU agenda allow various vectors of the original matrix to be expressed in terms of the current basis. In order to conserve computing time and printing time, various subsets can be selected by the user. It is also permissible to obtain the inverse of the current basis by using this agendum.

PICTURE presents an over-all picture of the original matrix with one character instead of many decimal digits printed to indicate magnitude. The parameter HELP should be used only when no feasible solution is found. It causes only those vectors having entries in infeasible rows to be included.

The REPORT agenda in conjunction with COMPILE produces solutions in a formal report format. All vital data are available for incorporation into these reports.

User Intervention

INTRODUCE, R, B
NEW.BASIS
NEW.BASIS, MOD
FLAGS, mode, specs

There are several options which allow the user to control which vectors enter the basis. One method is by coding the matrix to indicate to the program that a search should be made through subsets of vectors. The code indicates that partitions exist between the vectors, and those which are "partitioned off" are not considered in any computations.

FLAGS is an agendum which suppresses vectors in a manner similar to partitioning, except that suppressed columns need not be contiguous and any flagged vector in the basis will be removed.

Another method of control is the use of INTRODUCE which introduces the vectors given on the following cards one at a time. The program proceeds to eliminate one vector in the same manner as the NORMAL agendum, and then it introduces the next vector indicated on the card following the agendum card. This can be used after an optimum solution is obtained to determine the effect on the optimum of having a particular vector in the basis solution.

NEW.BASIS is an agendum which allows changing the basis partially or fully. If the user has a basis which will decrease the number of iterations because it is closer to optimum than the current basis, he can use this agendum to save machine time.

Interruption and Restart

GETOFF
GETOFF, cu
GETOFF, PRINT
SVHREG
RESTART
RESTART, cu
LDHREG

Facilities are provided for getting off the machine due to an emergency or due to user volition. Vital data are stored when a GETOFF card is read. Restarting differs depending upon whether the GETOFF was due to an emergency or to user volition.

Automatic Inversion and Misc.

INVERT
FIX.VEC
SCALE
EPSILON,B,K_1,K_2,E

INVERT is automatically performed at intervals, which are determined by ALPS and by the clock, to reduce both rounding errors and the number of transformations. Its frequency is also controlled by the user (see Control Parameters, below).

There are other agenda which require an inversion in conjunction with their use; for instance, NEW.BASIS. Each agendum will so specify in its description.

By using FIX.VEC, vectors with lower bounds or with negative values can be handled.

SCALE allows scaling of the functional row and/or composite weighing of infeasibilities in the functional.

EPSILON is useful for problems which have a high percentage of zero elements on the right-hand side.

Control Parameters

CONTROLS
STATUS

Certain parameters and tolerances which are automatically set by ALPS can be changed by using a CONTROLS card. STATUS will read from memory all such controls and parameters for printout.

Trouble Shooting

MDUMP, CODES, REGNS, BUFFS
PATCH

The agendum MDUMP is generally used during trouble shooting to obtain intermediate data when something goes wrong. The parameter CODES can be used to print out the ALPS routines for inspection; BUFFS can be used to examine data in the buffers.

PATCH is a dynamic method of changing the routines before running.

EXAMPLE

It is desired to mix four breakfast foods in such a way as to achieve certain dietary properties while minimizing the cost of the resulting mixture. The costs and dietary properties of the individual cereals are known.

Setting Up the Problem

The costs and the sodium, protein, and caloric contents of the four constituents are shown in Figure 1. As shown in the right-hand column, the cereal mixture is to have 150 calories, sodium content not to exceed 0.2 grams, and protein content of at least 3.0 grams.

CRISPIES	CRUNCHIES	CRACKLES	CHORTLES		
4.0	7.0	8.0	6.0		COST ROW
150.0	140.0	170.0	160.0	= 150.0	CALORIES
0.1	0.1	0.3	0.3	≦ 0.2	SODIUM
2.0	4.0	5.0	3.0	≧ 3.0	PROTEIN

Figure 1. Cereal Mixture Matrix

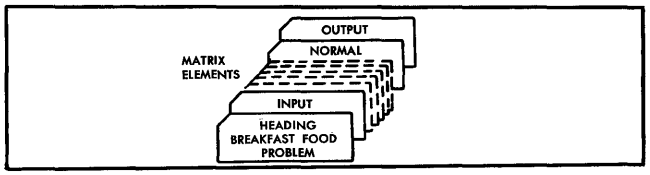


Figure 2. Sample ALPS Input Deck

These matrix elements are punched on data cards for input to ALPS. Each row and each column is assigned a 6-character mnemonic name. A code is punched for each row to indicate whether it represents an equality or a specific type of inequality.

Solving the Problem

The four agenda cards HEADING, INPUT, NORMAL, and OUTPUT are sufficient to find a solution and print out the answers. These four agenda are combined in sequence with the matrix data cards, as shown in Figure 2, and read by ALPS.

When the header card is read, ALPS stores the problem identification for later printing. Determining from the INPUT agendum that the following cards contain a matrix, the program converts the data to magnetic tape. The NORMAL agendum instructs ALPS to execute all routines which are required to solve the matrix for a minimum value of the cost function. When the solution is found, the program automatically reads the OUTPUT agendum, which causes it to print out the solution, as shown in Table II.

Large problems are solved using essentially the same method as illustrated by this simple example. The experimenter can use additional agenda cards during test runs to obtain further information, such as the density of the matrix, the ranges of the elements, etc. However, in production runs a minimum number of cards can be used.

Sensitivity Analyses

Having obtained a solution to the cereal mixture problem, the user might then wish to obtain further solutions based on anticipated price changes. If the optimum solution obtained with the card deck of Figure 2 is used with one DO.PCR agendum inserted before OUTPUT, ALPS will execute all of the necessary additional routines.

```
   PROBLEM HEADING: THE BREAKFAST FOOD PROBLEM                          PAGE 1

OPTIMAL PRIMAL-DUAL

  TOTAL     NO.     ETA     ROW     CURRENT      CHOSEN     VECTR     RHS     C/V     CURRENT D/J
  ITERS     ETAS    REC     IDENT.  VALUE        VECTOR     REMVD     NO.     NO.     THETA/PHI
    2         1       0     COST    5.26829268   CRISPI     -00000     1              .585

ROW INFORMATION

  ROW(I)          SLACK(I)          PI(I)          B(I)

  0 COST          5.26829268         1.00000000     0.00000000
  0 CALORY             -            -.00975610   150.00000000
  - SODIUM          .03170732             -        0.20000000
  + PROTEN             -           -1.26829268     3.00000000

COLUMN INFORMATION

  COLUMN          BETA(H)           D/J            A(TAU,J)

  CRISPI           .58536585            -         4.00000000
  CRUNCH               -             .56097561    7.00000000
  CRACKL           .36585366            -         8.00000000
  CHORTL               -             .63414634    6.00000000
```

Table II. Solution of Cereal Mixture Problem

HONEYWELL
ELECTRONIC
DATA
PROCESSING
WELLESLEY HILLS 81,
MASSACHUSETTS

PARTNER

"PARTNER" (Proof of Analog Results Through a Numerical Equivalent Routine), is a software package developed by Honeywell's Aero Division.

This description of "Partner" has been included to indicate the extent to which Honeywell is involved in Scientific Computing.

"PARTNER" provides rapid digital check solutions, to key analog results, in a single pass. Programs may be written directly from the system transfer functions or from the analog diagram without modifications.

Output is a time history of the control system under study. With this routine, the programmer has available all of the Common linear and non-linear functions.

Sampled data problems can be programmed, as well as continuous problems.

It has been apparent for some time that a simple and accurate method for checking analog computer solutions is highly desirable. Checking features available in present-day analog computers are quite limited in scope. In the past, a typical check solution on a digital computer involved a lengthy programming effort which produced only a program for the specific program at hand. For practical reasons, this discouraged extensive digital checking. Recently, compilers have been developed which provide a digital program from the analog diagram. However, these systems commonly require reorganizations and have several important restrictions on problem capability. In addition, they require sizable amounts of high-speed computer time and multiple passes for a final solution.

The analog simulation routine in this report has the following advantages over previous systems:

Programs are written directly from either the control system transfer functions or the analog diagram without modifications.

All of the common linear and non-linear analog functions are available to the programmer.

Sampled data problems can be programmed as well as continuous problems.

The final solution requires only one computer pass and only a nominal amount of time.

Versatility and ease of use were primary objectives in developing the analog simulation routine. With the concise programming format, only one instruction is required for any analog function. This function may be a second-order transfer function, a dead zone, or any other normal analog operation.

The routine solves the control system problem on a point-by-point basis in a pseudo-real time. All problem variables are advanced in parallel by a combination of prediction, iterations, and corrections. Pseudo-time can be stopped periodically and normal digital operations performed, simulating the effect of a perfect DA/AD linkage and high-speed digital computer. Synthetic delays are available in the routine to produce the effect of actual calculation delays in a real analog-digital system. In addition, numbers may be rounded off to study quantization effects in a problem.

HONEYWELL 800/1800

INDEX

OF

SCIENTIFIC PROGRAMS

INCLUDING

ARGUS SCIENTIFIC LIBRARY

HONEYWELL ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

C O N T E N T S

SEARCH CODE	TITLE AND DESCRIPTION
EDBGILL1	RUNGE KUTTA GILL - FLOATING BINARY METHOD OF RUNGE KUTTA GILL. EVALUATION OF THE DIFFERENTIAL EQUATIONS IS LEFT TO THE DISCRETION OF THE PROGRAMMER AND IS INCORPORATED IN THE SOLUTION AS A SUBROUTINE.
EDBCDIM1	CLIPPINGER DIMSDALE - FLOATING BINARY METHOD OF CLIPPINGER DIMSDALE WITH PROVISION FOR PRINTING SELECTED INTERMEDIATE RESULTS, AUTOMATIC ADJUSTMENT OF GRID SIZE, AND PROVISION FOR REDUCING ROUND OFF. EVALUATION OF THE DIFFERENTIAL EQUATIONS IS LEFT TO THE DISCRETION OF THE PROGRAMMER AND IS INCORPORATED INTO THE SOLUTION AS A SUBROUTINE.
EDBCDIM2	CARD INPUT SUBROUTINE FOR CLIPPINGER-DIMSDALE DIFFERENTIAL EQUATIONS SUBROUTINE. READS AND CONVERTS DATA ON CARDS FOR THE CLIPPINGER-DIMSDALE DIFFERENTIAL EQUATIONS SUBROUTINE EDBCDIM1.

SEARCH CODE	TITLE AND DESCRIPTION
EEB2EXP1	DOUBLE PRECISION E**X - FLOATING BINARY EVALUATES E**X IN DOUBLE PRECISION. ACCURATE TO AT LEAST 21 DECIMAL DIGITS.
EEB2LNX1	DOUBLE PRECISION LOGARITHM OF X TO THE BASE E - FLOATING BINARY EVALUATES THE NATURAL LOGARITHM OF X FOR A NORMALIZED DOUBLE PRECISION ARGUMENT. ACCURATE TO AT LEAST 22 DECIMAL DIGITS.
EEBATOX1	A**X - FLOATING BINARY EVALUATES A**X IF A IS GREATER THAN ZERO.
EEBEXPE3	E**X - FLOATING BINARY EVALUATES E**X. ACCURATE TO AT LEAST 10 DECIMAL DIGITS FOR X BETWEEN -32 AND +32. AT LEAST 9 DECIMAL DIGITS FOR ALL OTHER VALUES BETWEEN -128 AND +128.
EEBLNX01	LOGARITHM OF X TO THE BASE E - FLOATING BINARY FINDS NATURAL LOGARITHM OF X FOR X GREATER THAN 0. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
EEBLNX02	LOGARITHM OF X TO THE BASE 2, TO THE BASE E OR TO THE BASE 10 - FLOATING BINARY. FINDS THE REQUIRED LOGARITHM OF X FOR VALUES OF X GREATER THAN 0. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.

SEARCH CODE	TITLE AND DESCRIPTION
EEFEXPE1	E**X - FIXED DECIMAL EVALUATES E**X
EEFLNX01	LOGARITHM OF X TO THE BASE E - FIXED DECIMAL FINDS NATURAL LOG OF X FOR X GREATER THAN 0.
EEFLNX02	LOGARITHM OF X TO THE BASE 2, TO THE BASE E OR TO THE BASE 10 - FIXED DECIMAL FINDS THE REQUIRED LOGARITHM OF X FOR VALUES OF X GREATER THAN ZERO.

SEARCH CODE	TITLE AND DESCRIPTION
EHBCARP1	COMPLEX ARITHMETIC PACKAGE - FLOATING BINARY ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES COMPLEX NUMBERS.
EHBCLOG1	COMPLEX LOGARITHM OF X TO THE BASE E - FLOATING BINARY EVALUATES THE PRINCIPLE VALUE OF A COMPLEX NATURAL LOG OF THE FORM LOGZ = LOG/Z/ + IARGZ.
EHBEXPE1	COMPLEX E**X - FLOATING BINARY EVALUATES E**Z WHERE Z IS OF THE FORM X + YI.
EHBSICO1	COMPLEX SINE OR COSINE - FLOATING BINARY EVALUATES THE COMPLEX SINE OR COSINE FOR AN ARGUMENT OF THE FORM X + YI.
EHBSQRT1	COMPLEX SQUARE ROOT - FLOATING BINARY CALCULATES THE SQUARE ROOT OF AN ARGUMENT OF THE FORM X + YI.
EHFPACK1	COMPLEX ARITHMETIC PACKAGE - FIXED DECIMAL DECIMALLY ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES COMPLEX NUMBERS.

SEARCH CODE	TITLE AND DESCRIPTION
EIBLAGR1	LAGRANGIAN INTERPOLATION - FLOATING BINARY LAGRANGES INTERPOLATION FORMULA IS USED TO EVALUATE A FUNCTION ANYWHERE IN A TABULATED SET OF UNEQUALLY SPACED POINTS.
EIBLAPL1	LAPLACE-EVERETT CENTRAL DIFFERENCE INTERPOLATION - FLOATING BINARY LAPLACE-EVERETT CENTRAL DIFFERENCE FORMULA IS USED TO EVALUATE A FUNCTION NEAR THE MIDDLE OF A TABULATED SET OF EQUALLY SPACED POINTS.

SEARCH CODE	TITLE AND DESCRIPTION
EMBASSM1	MATRIX ADDITION, SUBTRACTION OR SCALAR MULTIPLICATION - FLOATING BINARY ADDS, SUBTRACTS, MULTIPLIES BY A SCALAR, OR ANY COMBINATION OF THESE FOR M*N MATRICES.
EMBASSM2	SYMMETRIC MATRIX ADDITION, SUBTRACTION AND SCALAR MULTIPLICATION - FLOATING BINARY ADDS, SUBTRACTS, MULTIPLIES BY A SCALAR OR ANY COMBINATION OF THESE FOR SYMMETRIC MATRICES. UPPER OR LOWER TRIANGLES ONLY STORED. SEE REMARKS - PAGE 23, #2
EMBCINV1	MATRIX INVERSE CORRECTION ROUTINE - FLOATING BINARY CALCULATES THE INVERSE OF MATRIX,A,USING EMBSLEQ1 AND CORRECTS THIS INVERSE UNTIL EACH ELEMENT HAS AT LEAST 10 SIGNIFICANT DIGITS OR UNTIL 5 ITERATIONS ARE MADE.
EMBEIGA1	EIGENVALUES OF A SYMMETRIC MATRIX BY GIVENS METHOD - FLOATING BINARY CALCULATES BY GIVENS METHOD THE EIGENVALUES OF A REAL SYMMETRIC MATRIX OF WHICH THE UPPER TRIANGLE IS STORED IN MEMORY.
EMBEIJT1	EIGENVALUES AND EIGENVECTORS BY THE JACOBI THRESHOLD METHOD - FLOATING BINARY COMPUTES THE EIGENVALUES AND EIGENVECTOR OF A REAL SYMMETRIC MATRIX OF WHICH THE UPPER TRIANGLE IS STORED IN MEMORY.

SEARCH CODE	TITLE AND DESCRIPTION
EMBMATR1	MATRIX TRANSPOSE - FLOATING BINARY FINDS THE TRANSPOSE OF A GIVEN MXN MATRIX A.
EMBMULT1	MATRIX MULTIPLICATION - FLOATING BINARY AN M*N MATRIX IS POSTMULTIPLIED BY AN N*P MATRIX TO OBTAIN AN M*P MATRIX. MATRICES STORED BY COLUMN.
EMBPOMT1	POSTMULTIPLICATION OF A MATRIX BY ITS TRANSPOSE - FLOATING BINARY AN M*N MATRIX,A,IS POSTMULTIPLIED BY ITS N*M TRANSPOSE TO OBTAIN AN M*M MATRIX,C. A AND C ARE STORED BY COLUMNS. THE TRANSPOSE IS NOT STORED SEPARATELY. SEE REMARKS - PAGE 23, #3
EMBPOTV1	POSTMULTIPLICATION OF THE TRANSPOSE OF A MATRIX BY A VECTOR - FLOATING BINARY THE TRANSPOSE OF AN M*N MATRIX,A, IS POSTMULTIPLIED BY AN M VECTOR TO OBTAIN AN N VECTOR,C. A AND C ARE STORED BY COLUMN. THE TRANSPOSE IS NOT STORED SEPARATELY. SEE REMARKS - PAGE 23, #4
EMBPRMT1	PREMULTIPLICATION OF A MATRIX BY ITS TRANSPOSE WITH THE ENTIRE RESULT STORED - FLOATING BINARY AN M*N MATRIX,A, IS PREMULTIPLIED BY ITS N*M TRANSPOSE TO OBTAIN AN N*N MATRIX, C. A AND C ARE STORED BY COLUMNS. THE TRANSPOSE IS NOT STORED SEPARATELY. SEE REMARKS - PAGE 23, #5

SEARCH CODE	TITLE AND DESCRIPTION
EMBPRMT2	PREMULTIPLICATION OF A MATRIX BY ITS TRANSPOSE WITH THE ENTIRE OR LOWER TRIANGULAR MATRIX STORED - FLBIN AN M*N MATRIX,A, IS PREMULTIPLIED BY ITS N*M TRANSPOSE TO OBTAIN AN N*N MATRIX,C. A IS STORED BY COLUMN AND THE TRANSPOSE IS NOT STORED SEPARATELY. C IS STORED BY COLUMN AS EITHER A WHOLE MATRIX OR A LOWER TRIANGULAR MATRIX. SEE REMARKS - PAGE 23, #6
EMBSDI01	SOLUTION OF LINEAR EQUATIONS, DETERMINANT AND INVERSE - FLOATING BINARY A VARIATION OF THE GAUSS ELIMINATION METHOD FOR SOLUTION OF LINEAR EQUATIONS AND MATRIX INVERSE. THE DETERMINANT IS CALCULATED DURING THE ELIMINATION PROCESS. THE ORIGINAL MATRIX IS REPLACED BY THE SOLUTION. DESIGNED FOR MINIMUM SPACE REQUIREMENTS.
EMBSINC1	SYMMETRIC MATRIX INVERSE AND CORRECTION - FLOATING BINARY COMPUTES THE INVERSE OF A SYMMETRIC MATRIX WHEN ONLY THE UPPER TRIANGULAR PORTION OF THE MATRIX IS STORED AND TO CORRECT THIS INVERSE TO FULL WORD ACCURACY GIVING AS A RESULT THE UPPER TRIANGULAR PORTION OF THE INVERSE.
EMBSLEQ1	SOLUTION OF LINEAR EQUATIONS, DETERMINANT AND INVERSE - FLOATING BINARY USES GAUSS ELIMINATION METHOD WITH ROW INTERCHANGE TO POSITION MAXIMUM PIVOT ELEMENT. THE ROUTINE WILL SOLVE A SET OF LINEAR EQUATIONS AND EVALUATE THE DETERMINANT, FIND THE INVERSE AND DETERMINANT OR FIND ALL THREE. DESIGNED FOR ACCURACY. SEE REMARKS - PAGE 23, #7

SEARCH CODE	TITLE AND DESCRIPTION
EMBSYDI2	INVERSION OF A SYMMETRIC MATRIX - FLOATING BINARY INVERTS A FLOATING BINARY SYMMETRIC MATRIX WHEN ONLY THE UPPER TRIANGLE IS STORED IN MEMORY BY COLUMN. THE MORRIS-ESCALATOR METHOD IS USED, WITH THE RESULT STORED OVER THE INPUT. THE VALUE OF THE DETERMINANT IS ALSO CALCULATED.
EMBSYMM1	SYMMETRIC MATRIX MULTIPLICATION - FLOATING BINARY MULTIPLIES A BY B TO OBTAIN C WHERE A, B AND C ARE SQUARE MATRICES OF ORDER N.
NOTE	EXCEPT WHEN OTHERWISE STATED, ALL MATRICES ARE STORED IN HIGH SPEED MEMORY.

SEARCH CODE	TITLE AND DESCRIPTION
EPBCPOL1	COMPLEX POLYNOMIAL EVALUATOR - FLOATING BINARY EVALUATES A POLYNOMIAL FOR A GIVEN COMPLEX VALUE OF THE INDEPENDENT VARIABLE.
EPBEVAL1	REAL POLYNOMIAL EVALUATOR - FLOATING BINARY EVALUATES A POLYNOMIAL FOR A GIVEN VALUE OF THE INDEPENDENT VARIABLE.
EPBCROT1	ROOTS OF A POLYNOMIAL - FLOATING BINARY FINDS THE ROOTS OF A POLYNOMIAL OF DEGREE N.
EPBRLRT1	REAL ROOTS OF A POLYNOMIAL - FLOATING BINARY FINDS THE REAL ROOTS OF A POLYNOMIAL OF DEGREE N.
EPFEVAL1	REAL POLYNOMIAL EVALUATOR - FIXED DECIMAL EVALUATES A POLYNOMIAL FOR A GIVEN VALUE OF THE INDEPENDENT VARIABLE.

SEARCH CODE	TITLE AND DESCRIPTION
ERB2SQR1	DOUBLE PRECISION SQUARE ROOT - FLOATING BINARY COMPUTES THE NORMALIZED DOUBLE PRECISION SQUARE ROOT OF A NORMALIZED DOUBLE PRECISION FLOATING BINARY ARGUMENT N. ACCURATE TO AT LEAST 21 DECIMAL DIGITS, 3.3 MILLISECONDS AVERAGE.
ERBCBRT1	CUBE ROOT - FLOATING BINARY FINDS THE CUBE ROOT OF A FLOATING BINARY ARGUMENT USING NEWTONS METHOD OF SUCCESSIVE APPROXIMATIONS. AT LEAST 9 SIGNIFICANT DECIMAL DIGITS, 3.9 MILLISECONDS AVERAGE. THIS CUBE ROOT ROUTINE SHOULD BE USED WHENEVER THE ABOVE ACCURACY IS SUFFICIENT AND THE SIMULATED SCIENTIFIC OPTION SUBROUTINE IS USED INSTEAD OF THE FLOATING POINT HARDWARE.
ERBCBRT2	CUBE ROOT - FLOATING BINARY FINDS THE CUBE ROOT OF A FLOATING BINARY ARGUMENT USING NEWTONS METHOD OF SUCCESSIVE APPROXIMATIONS. AT LEAST 10 SIGNIFICANT DECIMAL DIGITS, 2.3 MILLISECONDS AVERAGE. THIS CUBE ROOT ROUTINE SHOULD BE USED WHENEVER THE FLOATING POINT OPTION IS AVAILABLE.
ERBSQRT1	SQUARE ROOT - FLOATING BINARY FINDS THE SQUARE ROOT OF A FLOATING BINARY ARGUMENT USING NEWTONS METHOD OF SUCCESSIVE APPROXIMATIONS. AT LEAST 10 SIGNIFICANT DECIMAL DIGITS, 3.0 MILLISECONDS AVERAGE. THIS SQUARE ROOT ROUTINE SHOULD BE USED WHENEVER THE SIMULATED SCIENTIFIC OPTION SUBROUTINE IS USED INSTEAD OF THE FLOATING POINT HARDWARE.

SEARCH CODE	TITLE AND DESCRIPTION
ERBSQRT2	SQUARE ROOT - FLOATING BINARY FINDS THE SQUARE ROOT OF A FLOATING BINARY ARGUMENT USING NEWTONS METHOD OF SUCCESSIVE APPROXIMATIONS. AT LEAST 10 SIGNIFICANT DECIMAL DIGITS, 2.2 MILLI-SECONDS AVERAGE. THIS SQUARE ROOT ROUTINE SHOULD BE USED WHENEVER THE FLOATING POINT OPTION IS AVAILABLE.
ERDSQRT1	SQUARE ROOT - FLOATING DECIMAL FINDS THE SQUARE ROOT OF A FLOATING DECIMAL ARGUMENT USING NEWTONS METHOD OF SUCCESSIVE APPROXIMATIONS. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
ERF2SQR1	DOUBLE-PRECISION SQUARE ROOT - FIXED DECIMAL CALCULATES THE SQUARE ROOT OF A DOUBLE PRECISION FIXED DECIMAL ARGUMENT USING NEWTONS METHOD OF SUC-CESSIVE APPROXIMATIONS. ACCURATE TO AT LEAST 21 DECIMAL DIGITS.

SEARCH CODE	TITLE AND DESCRIPTION
ESFDIV01	DIVIDE WITHOUT REMAINDER - FIXED DECIMAL CALCULATES AN ELEVEN DIGIT QUOTIENT. SEE REMARKS - PAGE 23, #8
ESFDIV02	DIVIDE - FIXED DECIMAL CALCULATES AN ELEVEN DIGIT QUOTIENT AND AN ELEVEN DIGIT REMAINDER. SEE REMARKS - PAGE 23, #9
ESMESS01	SIMULATED SCIENTIFIC OPTION A PACKAGE OF ROUTINES WHICH SIMULATE THE SCIENTIFIC OPTION HARDWARE.
ESXDIV01	DIVIDE WITHOUT REMAINDER - FIXED BINARY CALCULATES A FORTY-FOUR BIT QUOTIENT. SEE REMARKS - PAGE 23, #10
ESXDIV02	DIVIDE - FIXED BINARY CALCULATES A FORTY-FOUR BIT QUOTIENT AND A FORTY-FOUR BIT REMAINDER. SEE REMARKS - PAGE 23, #11

SEARCH CODE	TITLE AND DESCRIPTION
ETB2ATN1	DOUBLE PRECISION ARC TANGENT - FLOATING BINARY FINDS ARCTANX FOR /X/ LESS THAN 16**11. ACCURATE TO AT LEAST 22 DECIMAL DIGITS.
ETB2HTN1	DOUBLE PRECISION HYPERBOLIC TANGENT - FLOATING BINARY FINDS TANHX USING E**X (DOUBLE PRECISION) SUBROUTINE. ACCURATE TO AT LEAST 20 DECIMAL DIGITS.
ETB2SNC1	DOUBLE PRECISION SINE OR COSINE - FLOATING BINARY FINDS SINX OR COSX FOR /X/ LESS THAN 2PI*16**9 RADIANS. ACCURATE TO AT LEAST 22 DECIMAL DIGITS.
ETBASCS2	ARC-SINE OR ARC-COSINE - FLOATING BINARY CALCULATES THE PRINCIPAL VALUES OF ARC-SINE OR ARC COSINE. ACCURATE TO AT LEAST 10 DECIMAL DIGITS. SEE REMARKS - PAGE 23, #12
ETBATAN1	ARC-TANGENT - FLOATING BINARY FINDS ARCTANX FOR /X/ LESS THAN 16**11. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
ETBATXY1	ARC-TANGENT X/Y - FLOATING BINARY COMPUTES ARCTAN X/Y FOR X/Y BETWEEN -1.5574 AND 1.5574 OR AN ANGLE LESS THAN ONE RADIAN. ACCURATE TO AT LEAST 10 DECIMAL DIGITS. SEE REMARKS - PAGE 23, #13

SEARCH CODE	TITLE AND DESCRIPTION
ETBHSAC1	HYPERBOLIC SINE OR COSINE - FLOATING BINARY FINDS SINHX OR COSHX USING E**X SUBROUTINE. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
ETBHTAN1	HYPERBOLIC TANGENT - FLOATING BINARY FINDS TANHX USING E**X SUBROUTINE. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
ETBSNCS1	SINE OR COSINE - FLOATING BINARY FINDS SINX OR COSX FOR /X/ LESS THAN 2PI*16**9 RADIANS. ACCURATE TO AT LEAST 9 DECIMAL DIGITS.
ETBSNCS2	SINE OR COSINE - FLOATING BINARY FINDS SINX OR COSX FOR /X/ LESS THAN 2PI*16**9 RADIANS. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
ETBTANX1	TANGENT - FLOATING BINARY FINDS TANGENT FOR /X/ LESS THAN OR EQUAL TO 1.25 RADIANS. ACCURATE TO AT LEAST 9 DECIMAL DIGITS.
ETBTCOT1	TANGENT OR COTANGENT - FLOATING BINARY CALCULATES TANX OR COTX FOR A FLOATING BINARY ARGUMENT. ACCURATE TO AT LEAST 10 DECIMAL DIGITS.
ETFSNCS1	SINE OR COSINE - FIXED DECIMAL CALCULATES SINX OR COSX FOR A FIXED DECIMAL ARGUMENT

SEARCH CODE	TITLE AND DESCRIPTION
EVBCHI21	CHI SQUARED - FLOATING BINARY EVALUATES CHI SQUARED IN FLOATING BINARY
EVXRANU1	RANDOM NUMBER GENERATOR - FIXED BINARY THIS ROUTINE GENERATES RANDOM BINARY NUMBERS OF FIXED LENGTH.

SEARCH CODE	TITLE AND DESCRIPTION
E1AMCED1	CARD EDIT TRANSFERS AN ALPHANUMERIC FIELD FROM ONE AREA IN MEMORY TO ANOTHER AND PERFORMS THE REQUESTED EDITING. ORDINARILY THE INPUT AREA WILL BE A CARD READER INPUT AREA.
E1FAMED1	MONEY EDIT TRANSFERS A DECIMAL FIELD FROM ONE AREA IN MEMORY TO ANOTHER AND CONVERTS THE FIELD TO ALPHANUMERIC FORMAT. ORDINARILY THE OUTPUT AREA WILL BE A PRINTER OUTPUT AREA. THE FIELD IS ASSUMED TO BE A DOLLAR AMOUNT AND THE DOLLAR SIGN AND COMMAS ARE SUPPLIED AUTOMATICALLY.
E1FBC2M1	CARD TO MEMORY EDIT - FLOATING BINARY CONVERTS CARD INPUT TO FLOATING BINARY AND STORES THE RESULTS IN MEMORY.
E1FBC2T1	CARD TO TAPE EDIT - FLOATING BINARY CONVERTS CARD INPUT TO FLOATING BINARY AND STORES THE RESULTS ON TAPE
E1FDC2M1	CARD TO MEMORY EDIT - FLOATING DECIMAL CONVERTS CARD INPUT TO FLOATING DECIMAL AND STORES THE RESULTS IN MEMORY

SEARCH CODE	TITLE AND DESCRIPTION
E1FDC2T1	CARD TO TAPE EDIT - FLOATING DECIMAL CONVERTS CARD INPUT TO FLOATING DECIMAL AND STORES THE RESULTS ON TAPE
E1MAPED1	PRINT EDIT TRANSFERS A FIELD FROM ONE AREA IN MEMORY TO ANOTHER, CONVERTS THE FIELD TO ALPHANUMERIC AND PERFORMS THE REQUESTED EDITING. ORDINARILY THE OUTPUT AREA WILL BE A PRINTER OUTPUT AREA.

SEARCH CODE	TITLE AND DESCRIPTION
E2BPACK1	DOUBLE-PRECISION PACKAGE - FLOATING BINARY ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES TWO TWENTY DIGIT FLOATING BINARY OPERANDS TO OBTAIN A TWENTY DIGIT FLOATING BINARY RESULT.
E2BPACK2	DOUBLE-PRECISION PACKAGE - FLOATING BINARY ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES TWO TWENTY DIGIT FLOATING BINARY OPERANDS TO OBTAIN A TWENTY DIGIT FLOATING BINARY RESULT. THIS SUBROUTINE GIVES GREATER ACCURACY THAN E2BPACK1. MAY ONLY BE USED WITH THE FLOATING POINT HARDWARE.
E2DPACK1	DOUBLE-PRECISION PACKAGE - FLOATING DECIMAL ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES TWO TWENTY DIGIT FLOATING DECIMAL OPERANDS TO OBTAIN A TWENTY DIGIT FLOATING DECIMAL RESULT.
E2FPACK1	DOUBLE-PRECISION PACKAGE - FIXED DECIMAL ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES TWO TWENTY-TWO DIGIT FIXED DECIMAL OPERANDS TO OBTAIN A TWENTY-TWO DIGIT FIXED DECIMAL RESULT. SEE REMARKS - PAGE 23, #1
E2FTMPY1	TRUNCATED DOUBLE-PRECISION MULTIPLY - FIXED DECIMAL DECIMALLY MULTIPLIES TWO, TWENTY-TWO DIGIT OPERANDS TO OBTAIN A TRUNCATED TWENTY-TWO DIGIT PRODUCT.

SEARCH CODE	TITLE AND DESCRIPTION
ECBDCON1	FLOATING BINARY TO FLOATING DECIMAL CONVERSION CONVERTS A FLOATING BINARY NUMBER IN MEMORY TO A FLOATING DECIMAL NUMBER IN MEMORY.
ECDBCON1	FLOATING DECIMAL TO FLOATING BINARY CONVERSION CONVERTS A FLOATING DECIMAL NUMBER IN MEMORY TO A FLOATING BINARY NUMBER IN MEMORY.
ECFSEXC1	DEGREES TO RADIANS OR RADIANS TO DEGREES - FIXED DECIMAL CONVERTS A SIGNED FIXED DECIMAL WORD OF DEGREES, MINUTES AND SECONDS TO RADIANS OR A SIGNED FIXED DECIMAL WORD OF RADIANS TO DEGREES, MINUTES AND SECONDS
ECFXFRA1	CONVERSION OF A FIXED DECIMAL FRACTION TO A BINARY FRACTION CONVERTS A FIXED DECIMAL FRACTION IN MEMORY TO A FIXED BINARY FRACTION IN MEMORY.

SEARCH CODE	TITLE AND DESCRIPTION
ECFXINT1	CONVERSION OF A FIXED DECIMAL INTEGER TO A FIXED BINARY INTEGER CONVERTS A FIXED POINT DECIMAL INTEGER IN MEMORY TO A FIXED POINT BINARY INTEGER IN MEMORY.
ECMMFUN1	FLOAT AND UNFLOAT CONVERSION CONVERTS EITHER A FIXED POINT FRACTION IN MEMORY TO A FLOATING FRACTION OR A FLOATING FRACTION IN MEMORY TO A FIXED POINT FRACTION.
ECXFFRA1	CONVERSION OF A FIXED BINARY FRACTION TO A FIXED DECIMAL FRACTION CONVERTS A FIXED POINT BINARY FRACTION IN MEMORY TO A FIXED POINT DECIMAL FRACTION IN MEMORY.
ECXFINT1	CONVERSION OF A FIXED BINARY INTEGER TO A FIXED DECIMAL INTEGER CONVERTS A FIXED POINT BINARY INTEGER IN MEMORY TO A FIXED POINT DECIMAL INTEGER IN MEMORY.

SEARCH CODE	TITLE AND DESCRIPTION
EUBPACK1	EXTRA-PRECISION PACKAGE - FLOATING BINARY ADDS, SUBTRACTS, MULTIPLIES, OR DIVIDES TWO NORMALIZED EXTRA PRECISION FLOATING BINARY NUMBERS TO OBTAIN A NORMALIZED EXTRA PRECISION FLOATING BINARY RESULT. ACCURACY IS NOT AS GOOD AS THAT OBTAINED WITH EITHER DOUBLE PRECISION PACKAGE.

REMARKS

VARIOUS CHANGES HAVE BEEN MADE IN SOME OF THE PRECEDING SUBROUTINES. BECAUSE THESE WERE ONLY MINOR CORRECTIONS, ONLY THOSE PAGES IN THE WRITE-UPS WHICH WERE ACTUALLY AFFECTED WERE REPRINTED. THE FOLLOWING INFORMATION IS GIVEN- NAME OF THE SUBROUTINE, DATE ON THE CORRECTED PAGES AND THE PAGE NUMBER.

#1 E2FPACK1 - PAGES 10, 11 AND 12 - 1/7/63

#2 EMBASSM2 - PAGES 7 AND 8 - 10/25/62

#3 EMBPOMT1 - PAGE 5 - 8/10/62

#4 EMBPOTV1 - PAGE 5 - 8/10/62

#5 EMBPRMT1 - PAGE 5 - 8/10/62

#6 EMBPRMT2 - PAGE 6 - 8/10/62

#7 EMBSLEQ1 - PAGE 8 - 8/10/62

#8 ESFDIVO1 - PAGE 6 - 8/10/62

#9 ESFDIVO2 - PAGE 7 - 8/10/62

#10 ESXDIVO1 - PAGE 5 - 8/10/62

#11 ESXDIVO2 - PAGE 7 - 8/10/62

#12 ETBASCS2 - PAGE 10 - 6/15/62

#13 ETBATXY1 - PAGE 6 - 8/10/62

L I N E A R P R O G R A M M I N G P A C K A G E

THREE LINEAR PROGRAMMING PACKAGES ARE NOW AVAILABLE FOR USE ON THE H800/H1800 SYSTEMS. ALL ARE CODED TO BE USED WITH THE AUTOMATH COMPILER. THEY ARE CALLED RSMA, RSMB AND RSMC RESPECTIVELY, THE FIRST THREE LETTERS IN EACH CASE STANDING FOR REVISED SIMPLEX METHOD. VERSIONS A AND C ARE WRITTEN IN SINGLE PRECISION WHILE B EXISTS IN BOTH A SINGLE AND A DOUBLE PRECISION VERSION, THAT IS RSMB1 AND RSMB2. RSMA, RSMB1 AND RSMB2 ALL USE THE EXPLICIT INVERSE OF THE BASIS. RSMC USES THE PRODUCT FORM OF THE INVERSE.

RSMA, AS A SUBSET OF RSMB, PERFORMS ONLY THE BAREST MINIMUM THAT RSMB DOES, BUT IS THE ONLY ONE OF THE GROUP THAT CAN BE RUN ON A 4K MACHINE (SMALL PROBLEMS). RSMB1 (AND RSMB2) ALLOWS FOR MORE VARIATION IN INPUT, CONTROL PARAMETERS, OUTPUT, ETC. THAN RSMA. HOWEVER IT USES 1.4K MORE MEMORY AND THUS CANNOT BE RUN ON A 4K MACHINE.

RSMC IS SOMEWHAT SLOWER THAN RSMB ON THE SET OF PROBLEMS TESTED. HOWEVER, THIS IS COUNTERACTED BY THE FACT THAT LARGER PROBLEMS CAN BE RUN ON RSMC AS THE MEMORY REQUIRED FOR DATA IS LESS THAN RSMB.

PLEASE NOTE THAT ALL OF THE ABOVE MENTIONED PROGRAMS ARE DESIGNED TO KEEP ALL DATA IN MEMORY ALL THE TIME AND DO NOT USE ANY TAPES FOR COMPUTATIONS. TAPES ARE USED ONLY FOR INITIAL INPUT AND FINAL INPUT (INTERMEDIATE OUTPUT CAN BE PUT ON A TAPE IF NEEDED). THUS THEY ARE FASTER THAN TAPE ORIENTED PROGRAMS EVEN FOR INCORE PROBLEMS, I.E., PROBLEMS THAT ARE SMALL AND DO NOT NEED TAPES FOR INTERMEDIATE COMPUTATIONS.

THE FOLLOWING TABLES SHOW MEMORY REQUIREMENTS AND AN ESTIMATE OF PROBLEM SIZE-

LINEAR PROGRAMMING PACKAGES - MEMORY REQUIREMENTS

PROGRAM NAME	NO. OF SUB.	USUAL	MINIMUM	MAXIMUM	DATA REGION
RSMA	14	3596	3407 REMOVE- KAR	3596	M + 5M + 3N + 2A
RSMB1 S.P.	21	4896	4218 REMOVE- EFM, ERR, HOT, KAR, UNL	4861 REPLACE HOT1 BY HOT3	M + 5M + 3N + 2A
RSMB2 D.P.	22	5279	4601 REMOVE- EFM, ERR, HOT, KAR, UNL	5294 REPLACE HOT1 BY HOT3	2M + 7M + 3N + 2A
RSMC	22	5057	4397 REMOVE- EFM, ERR, HOT, KAR, UNL	5122 REPLACE HOT1 BY HOT3	.5M + 7.4M + 3N + 2A

WHEN THE FOLLOWING SUBROUTINES ARE REMOVED, THE FUNCTIONS MENTIONED BESIDE EACH WILL NOT BE PERFORMED-

KAR - INPUT DATA ON CARDS. (USE TAPE INPUT)

EFM - UNDERFLOW/OVERFLOW SETUP. USES THE ORDINARY AUTOMATH SETUP.

ERR - COMPUTING MAXIMUM ERROR ON ROW AND COLUMN.

HOT - HORIZONTAL OUTPUT.

UNL - DUMP INTERMEDIATE DATA ON TAPE 3 AND USE THIS TAPE LATER.

SIZE OF PROBLEMS- AN EXAMPLE, VALUES OF MAXIMUM M

PROGRAM NAME	MEMORY AVAILABLE				
	4K	8K	12K	16K	32K
RSMA M + 5M + 3N + 2A	15	54	76	93	143
RSMB1 M + 5M + 3N + 2A	-	46	70	84	140
RSMB2 2M + 7M + 3N + 2A	-	32	52	55	105
RSMC .5M + 7.4M + 3N + 2A (APPROXIMATELY)	-	50	85	103	174

THE VALUES OF M GIVEN ABOVE ARE COMPUTED BY ASSUMING THAT N = 2M, A = $\frac{NM}{12}$

P E R T

THE PERT 800/1800 PACKAGE CONSISTS OF THREE SEPARATE DIFFERENT PROGRAMS WHICH MUST BE RUN SERIALLY IN THE ORDER PERTIME1, PERTIME2 AND PERTIME3. THESE PROGRAMS ARE WRITTEN IN ARGUS AND MUST BE RUN UNDER CONTROL OF THE HONEYWELL EXECUTIVE SYSTEM.

THE PERT NETWORKS, IN THE FORM OF PUNCHED CARDS, ARE PROCESSED BY THE PROGRAM AND THE RESULTS OF THE CALCULATIONS ARE REPORTED IN THE FORM OF PRINTED OUTPUT. THE PERT PROGRAM IS DESIGNED TO SIMPLIFY THE TASKS OF DATA PREPARATION FOR INPUT AND INTERPRETATION OF OUTPUT. EACH INPUT DATA CARD (EXCEPT TWO HEADER CARDS) CONTAINS ALL THE INFORMATION PERTAINING TO ONE ACTIVITY AND EACH LINE OF OUTPUT INDICATES THE RESULTS OF THE CALCULATIONS CORRESPONDING TO ONE ACTIVITY.

THE MINIMUM HARDWARE AND SOFTWARE REQUIREMENTS ARE AS FOLLOWS-

(1) 4096 WORDS OF CORE STORAGE MEMORY

(2) 5 TAPE UNITS (THIS INCLUDES THE PROGRAM TAPE)

(3) 1 CARD READER

(4) 1 HIGH SPEED PRINTER

(5) ARGUS SYSTEM BOOTSTRAP AND LOADER

(6) ARGUS LIBRARY SORT #1 AND SORT #2

(7) ARGUS ORTHOCORRECTION ROUTINE

SOME OF THE MORE IMPORTANT PERT 800/1800 SYSTEM CAPABILITES ARE AS FOLLOWS-

(1) THE PROGRAM IS CAPABLE OF PROCESSING UNLIMITED NUMBERS OF ACTIVITES EFFICIENTLY (APPROXIMATELY 100/MINUTE).

(2) EVENT CODES OF 16 DIGITS OR LESS MAY BE ASSIGNED IN RANDOM ORDER. THE FIRST SEVEN ARE USED FOR NETWORK IDENTIFICATION AND ARE COMMON THROUGHOUT THE NETWORK.

(3) THE PROGRAM WILL PROCESS NETWORKS WITH MULTIPLE START AND END EVENTS. THESE MAY OR MAY NOT HAVE SCHEDULE OR ACTUAL DATES. WHEN DATES ARE NOT SPECIFIED, THE REFERENCE DATE IS ASSIGNED TO START EVENTS AND THE CALCULATED VALUE OF T_E IS ASSIGNED TO END EVENTS.

(4) THE PROGRAM HAS THE ABILITY TO MAINTAIN PERT NETWORKS ON A MASTER TAPE AND UPDATE THESE NETWORKS AS REQUIRED. CARD FILE MAINTENANCE MAY ALSO BE USED IF DESIRED.

(5) THIRTY-FIVE ALPHANUMERIC CHARACTERS ARE AVAILABLE TO DESCRIBE THE EVENT OR ACTIVITY TITLE.

(6) SCHEDULE AND/OR ACTUAL COMPLETION DATES IN THE FORM OF MMDDYY MAY BE ASSIGNED TO EACH EVENT.

(7) THE PROGRAM HAS THE CAPABILITY OF PROCESSING BOTH ACTIVITY ORIENTED AND EVENT ORIENTED NETWORKS.

(8) COMPLETED EVENTS (OR ACTIVITIES) ARE RETAINED IN THE NETWORK MASTER FILE FOR HISTORIC INFORMATION, BUT ARE ELIMINATED FROM THE FINAL OUTPUT.

(9) CALENDAR CONVERSIONS FROM NUMBER OF WEEKS TO CALENDAR DATES ARE MADE ON A FIVE DAY WORK WEEK BASIS. SATURDAYS, SUNDAYS AND HOLIDAY DATES DO NOT APPEAR IN THE OUTPUT.

(10) THE PROGRAM HAS THE CAPABILITY OF REFERENCING THE T_E CALCULATIONS EITHER TO TIME NOW OR PROJECT START DATE.

(11) THE PROGRAM WILL COMPUTE SLACK FROM ANY POINT IN THE NETWORK AS DESIGNATED BY THE USER.

(12) THE PROGRAM HAS THE ABILITY TO CALCULATE EITHER THE LONGEST PATH OR THE SHORTEST PATH THROUGH A GIVEN NETWORK.

(13) THE USER HAS THE CHOICE OF USING EITHER THREE TIME ESTIMATES OR ONE TIME ESTIMATE FOR ACTIVITY DURATION.

(14) THE PROGRAM WILL CALCULATE STANDARD DEVIATION AND PROBABILITY IF REQUESTED BY THE USER.

(15) THE PROGRAM STARTS ALL CALCULATIONS WITH THE LAST COMPLETED EVENTS AND THEREFORE ELIMINATES ALL UNNECESSARY CALCULATIONS.

(16) TO MAKE EFFICIENT USE OF HIGH SPEED MEMORY DURING THE T_E AND T_L CALCULATIONS THE PROGRAM ASSIGNS BINARY PSEUDO NUMBERS TO THE EVENT CODES IN ORDER TO REDUCE THE LENGTH.

(17) ACTIVE ACTIVITIES ARE DESIGNATED BY A (#) SIGN TO THE LEFT OF EACH SUCH ACTIVITY IN THE REGULAR OUTPUT.

(18) THE PROGRAM GENERATES A MAXIMUM OF EIGHT DIFFERENT TYPES OF OUTPUT.

(19) THE PROGRAM WILL ACCEPT THE INPUT DATA CARDS IN RANDOM ORDER.

(20) THE PROGRAM HAS THE CAPABILITY TO AUTOMATICALLY SUMMARIZE DETAIL NETWORKS AND GENERATE HIGHER LEVEL OUTPUT.

(21) THE PROGRAM WILL TRY TO ISOLATE ALL POSSIBLE INPUT ERRORS IN A SINGLE RUN. IF ANY OF THE INPUT ERRORS ARE PROHIBITIVE AND FURTHER CALCULATIONS ARE NOT POSSIBLE THE RUN WILL BE DELAYED. A LISTING OF ALL ERRORS WILL AUTOMATICALLY BE GENERATED.

STATISTICS PACKAGE

THE STATISTICAL PACKAGE CONTAINS THREE PROGRAMS WHICH INVOLVE CORRELATION, REGRESSION AND CURVE FITTING.

(1) MEVACOP - MEAN, VARIANCE AND CORRELATION PROGRAM

(2) STRAP1 - STEPWISE REGRESSION ANALYSIS PROGRAM

(3) LESCUF - LEAST SQUARES CURVE FITTING PROGRAM

ALL OF THESE PROGRAMS ARE WRITTEN IN AUTOMATH.

MEVACOP IS USED TO COMPUTE THE MEAN, VARIANCE, COVARIANCE OR CORRELATION COEFFICIENT OF THE VARIABLES WHICH ARE STORED IN TWO GROUPS ON TAPE, EG. THE X GROUP AND THE Y GROUP. THE PRINTOUT OF COVARIANCE OR CORRELATION IS OPTIONAL.

THE MINIMUM SYSTEM COMPONENTS REQUIRED ARE AS FOLLOWS-

(1) 4096 WORDS OF CORE STORAGE MEMORY

(2) ONE OFF-LINE PRINTER

(3) 3 TAPES- PTS, OUTPUT TAPE AA, INPUT DATA TAPE AB

INPUT DATA IS READ FROM TAPE. OUTPUT IS ON TAPE AA AND MAY BE PRINTED ON-LINE OR OFF-LINE.

STRAP1 IS USED IN DATA ANALYSIS TO OBTAIN THE BEST FIT OF A HOMOGENEOUS SET OF OBSERVATIONS OF INDEPENDENT AND DEPENDENT VARIABLES BY AN EQUATION OF THE FORM

$$Y = B_0 + B_1 X_1 + B_2 X_2 + \cdots + B_{m-1} X_{m-1}$$

WHERE Y IS THE DEPENDENT VARIABLE, X_1, X_2, ... ARE THE INDEPENDENT VARIABLES AND B_0, B_1, ... ARE THE COEFFICIENTS TO BE DETERMINED.

IN THE STEPWISE PROCEDURE THE INTERMEDIATE RESULTS ARE USED TO GIVE VALUABLE STATISTICAL INFORMATION AT EACH STEP. THESE INTERMEDIATE ANSWERS ARE ALSO USED TO CONTROL THE METHOD OF CALCULATION. THE FINAL REGRESSION INCLUDES ONLY THOSE VARIABLES WHICH ARE SIGNIFICANT.

THE MINIMUM SYSTEM COMPONENTS REQUIRED ARE AS FOLLOWS-

(1) 4096 WORDS OF CORE STORAGE MEMORY

(2) ONE ON-LINE CARD READER

(3) ONE OFF-LINE PRINTER

(4) 3 TAPES- PTS, OUTPUT TAPE AA, WORK TAPE AD

DATA MAY BE READ FROM CARDS OR FROM TAPE. OUTPUT IS ON TAPE AA, WHICH IS EVENTUALLY USED FOR OFF-LINE PRINTING.

LESCUF IS USED TO FIT A POLYNOMIAL (OR A WEIGHTED POLYNOMIAL) WITH A GIVEN SET OF DATA BY THE METHOD OF LEAST SQUARES.

THE MINIMUM SYSTEM COMPONENTS REQUIRED ARE AS FOLLOWS-

(1) 4096 WORDS OF CORE STORAGE MEMORY

(2) ON-LINE CARD READER

(3) OFF-LINE PRINTER

(4) 3 TAPES- PTS, OUTPUT TAPE AA, WORK TAPE AD

INPUT DATA IS READ FROM CARDS. OUTPUT IS ON TAPE AA WHICH IS SAVED FOR OFF-LINE PRINTING.

H-1800-II

SPECIAL EQUIPMENT

COMMUNICATIONS AND THE HONEYWELL 1800-II

Honeywell several years ago recognized the future impact of data transmission on EDP and has devoted considerable effort to the research and development of equipment which would enable Honeywell computers to receive and transmit vast quantities of data over a variety of communications capabilities ranging from an inexpensive single channel unit for teletype grade circuits (the H-881), to a unit for high capacity multiplexing of many low- and medium-speed circuits (the H-884), to a high-speed, error detecting unit for transferring information at the full capacity of a voice circuit - 2400 bits per second (the H-880). The following paragraphs briefly describe the properties of each of these units.

The H-881 is a relatively simple, one channel device which enables the H-1800 to be utilized as follows:

1. The H-1800 may be a way station on a half-duplex teleprinter private (leased) line.
2. The H-1800 may be hooked up to any of the Bell or Western Union leased communication networks, such as 83B2 or Plan 115, etc.
3. On the TWX or TELEX system, the H-1800 may be a subscriber station.
4. Utilization of the 400 series Dataphone with either a voice private line, or the general switched telephone network will allow transmission at the speed of 75 characters per second.
5. Any teletype speed - 6, 7.5 or 10 characters per second may be used.
6. Any code may be used, as translation is programmed.
7. Any of the new billing plans may be applied - such as WATS or WADS.

The H-884 is essentially a multi-channeled H-881, with the added feature that translation is made automatically from either Baudot or 7-Channel ASA code to 6-bits Honeywell code.

The H-880 is used with the 200 Dataphone series to provide a single-channel high-speed (up to 300 characters per second for a voice line) for communicating with high-speed terminal devices. Besides being able to communicate with other Honeywell computers, the following equipment may be used as terminal devices:

IBM 1009	Data Transmission Unit
IBM 1013	Card Transmission Unit
IBM 7701,7702	Magnetic Tape Transmission Unit

The H-880 employs a large buffer (512 characters) to temporarily store the high volume of information before transferring it into the computer. The code used is the error-detecting 4 of 8 code, which the IBM equipments employ. The code is automatically translated to Honeywell 6-bit code in the 880.

Beyond these various standard equipments provisions can be made in conjunction with the H-1800-II to accommodate translation for other codes, interface with other data transmission systems at comparable rates (such as Kineplex), for off-line transmission of magnetic tape or other media, and for connection to high speed transmission over private microwave or Telpak systems including Type D. In addition, Honeywell is continuing to investigate new common carrier and private transmission systems as they become available so as to recommend and provide the most efficient system for any specific requirements.

Publicly demonstrated proof of Honeywell achievement in communications was the use of TELSTAR for a computer-to-computer link via TELSTAR example last year, and hooking the computer up to a public TELEX station located in Paris. Honeywell is continuing a study effort on advanced programming techniques as well as hardware that will more easily process priority message handling, act as a switching center, or solve other common systems problems.

HONEYWELL
DATA COMMUNICATIONS

MODEL 880 COMMUNICATIONS CONTROL UNIT:

COMPATIBLE WITH:

DATAPHONE	201A	2000 BPS
	201B	2400
	202A or 202B	600 or 1200

MODEL 881 COMMUNICATIONS CONTROL UNIT

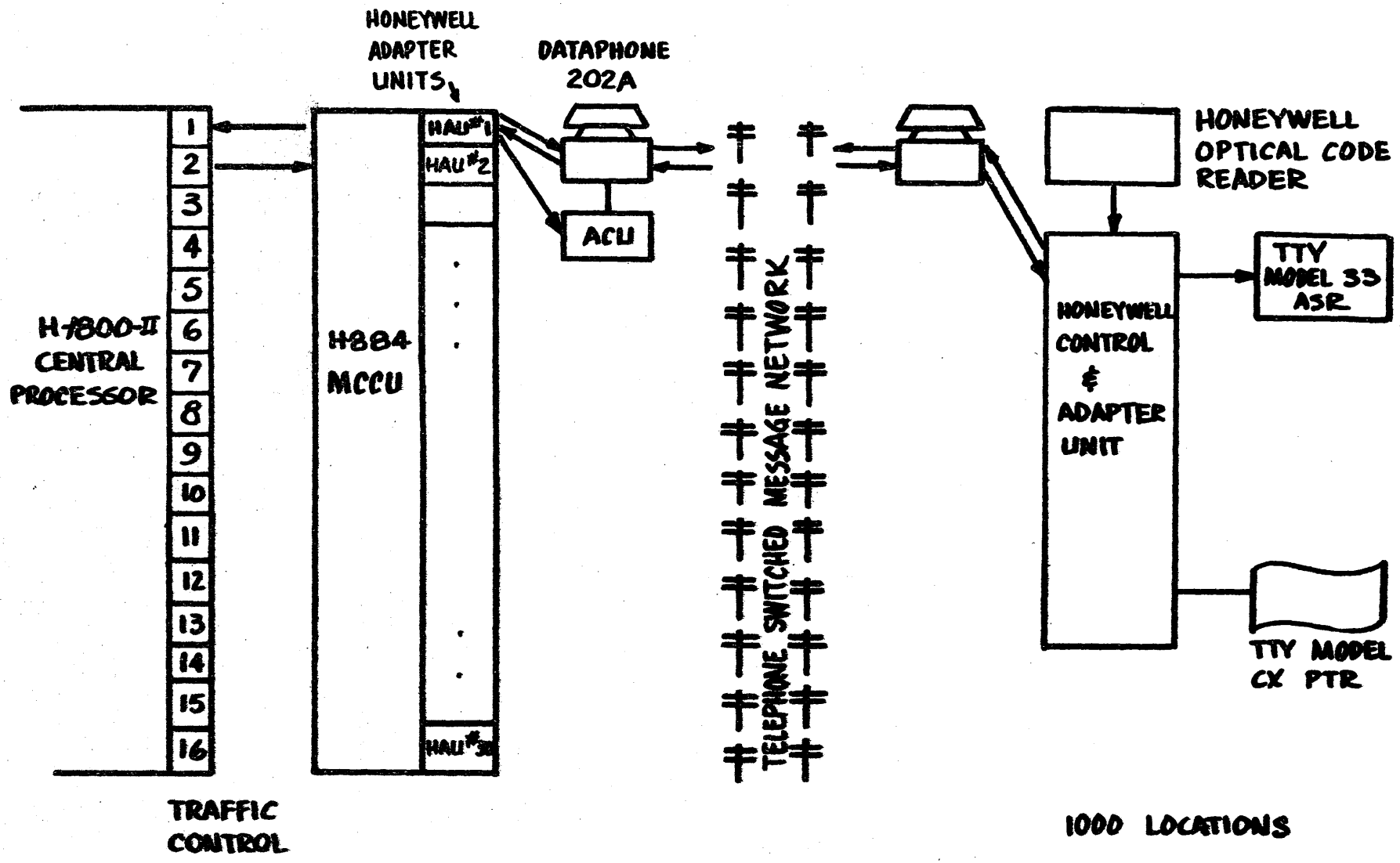
MODEL 884 MULTIPLE COMMUNICATIONS CONTROL UNIT

BOTH COMPATIBLE WITH:

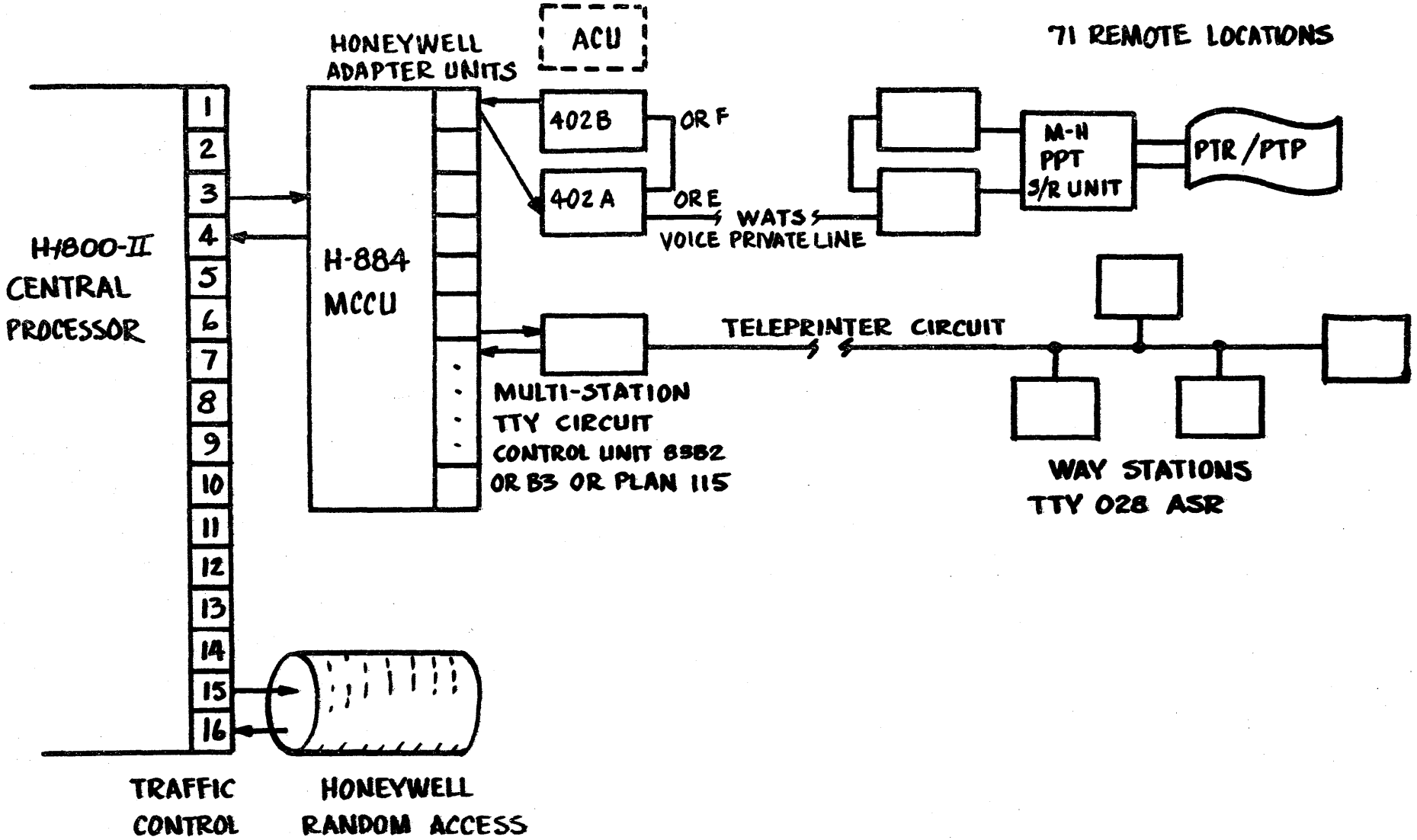
60, 66, 75 wpm TELEPRINTER CIRCUITS

VOICE GRADE LINES UP TO 75 cps

TYPICAL
PROPOSED HONEYWELL COMMUNICATIONS SYSTEM



TYPICAL
PROPOSED HONEYWELL COMMUNICATIONS SYSTEM



GENERAL PURPOSE ON-LINE REAL TIME CONTROL

Special external devices, except those which present a tape or standard communications type interface, are normally connected to the H-1800-II system through the general purpose Real Time Control Units (RTCU), Models 1812-1, 1812-2, 812-1 and 812-2. The basic design of these units is the same; however, the 1812 transfers a full computer word (48 bits plus parity) in 6 microseconds compared to the 812's 18 microseconds. The 1812-1 and 812-1 allow two-way nonsimultaneous transmission with a time-shared one-word buffer, while the 1812-2 and 812-2 allow simultaneous transmission with two buffers.

The choice between these various models depends upon a systems analysis of the external equipment requirements. External equipment can be anything from a single high speed broadband microwave system to a number of analog converters, non-computer digital sources (such as radar output), or telemetry reception and conversion equipment. The standard, general purpose multiplexer and distributor are described below. Honeywell readily can provide other designs to suit the specific requirement.

Transfers with external equipment are asynchronous while transfers from the RTCU control unit to the computer are synchronous. Input and output record lengths are variable. The RTCU generates new parity for transfers to the computer and checks parity on output. A test mode of operation is provided to simulate the central processor for off-line testing, in addition to programmable exercising of external devices.

A typical application of this equipment is the Submarine FBM Training Facility, Second Increment, which Honeywell has installed at New London, Connecticut and a similar facility at Charlestown, South Carolina. The central control is performed by an H-800 which communicates with simulated and operational-type training equipment through an 812-2. A multiplexer

receives some 128 inputs and a distributor routes over 600 outputs. External equipment includes instructor's control consoles, operational-type fire control systems, large screen graphic displays, radar and sonar simulators, and fully simulated periscope systems.

ON-LINE REAL TIME PROGRAM PROTECTION

A feature to extend the multiprogramming capabilities of the H-800 and H-1800 is called memory lockout. This feature allows not only the normal parallel operation of production programs with on-line real-time applications, but also permits program checkout to be run in parallel. This feature, when activated, will cause any program attempting to address the portion of the computer reserved for on-line real time use to be turned off, a special indicator to be activated at the operators console, and preserves the contents of the addressed location.

PROGRAM ADDRESSABLE TAPE SWITCHING UNIT

A modified 805 or 1805 tape drive switching unit is available which permits the release of a tape for off-line usage or the demanding of an off-line tape to an on-line status under program control. This capability can usefully increase by reducing delays in tape handling.

MULTIPLEXER

PURPOSE

Connects an 812-2 Real Time Control Unit to a minimum of 4 independent input channels, each channel representing a conversion of data formats to a 36-bit word that is presented to the Multiplexer. This word size is arbitrary and can be modified, but is designed to allow the addition of 12 bits of source address and still provide for a full word from a 36-bit computer or tape interface or several analog to digital converter outputs. It is also ample to handle 1 or 2 normal telemetry words.

DESCRIPTION

The Multiplexer is designed to control and buffer a minimum of 4 independent input channels. Individual additional channels can be added. Each channel is serviced as it becomes ready in a completely ansynchronous manner. The time required to transfer a word into memory from any channel is 6 micro-seconds per word.

The Multiplexer can be expanded to control and buffer additional independent channels by simply adding an additional 36 bit buffer register for each additional input channel. This expansion is almost unlimited from a hardware viewpoint, i.e., recognizing requests and accepting data. Rather, the maximum load would be determined from programming considerations and data source requirements.

DISTRIBUTOR

PURPOSE

Connects an 812-2 Real Time Control Unit to a minimum of 4 output channels. In each case up to 36 bits of data are transferred to the channel selected by the bits indicating channel address. Up to 12 bits are available for addressing. The 36 bits may be used at will as described under "Multiplexer."

DESCRIPTION

The Distributor is designed to control a minimum of 4 output channels. Each channel can be addressed by the programs in either central processor and up to 36 bits in parallel can be transferred to the selected channel. Channels can be added in units of 8.

The number of channels addressed by the distributor can be expanded up to 2^{12}.

The actual maximum would be determined by programming considerations and the characteristics of the terminal devices being addressed.

ADDITIONAL INPUT-OUTPUT TRUNKS

The standard H-1800 II provides 8 pairs of input-output trunks. If the amount of peripheral equipment requires 8 more trunk pairs can be provided. The existing addressable Distributed Read and Write address counters are used as Read and Write address counters for the additional 8 pairs of trunks. Program selection of the additional trunks is controlled by an additional set of peripheral instructions. Priority and order of servicing all devices is determined by the trunk to which the device is attached.

CLOCKS AND TIMERS

Two standard programmed clocks, the 1813-3 and 1813-4, are available. The 1813-3 increments any specified one of the unused read or write address counters once each second. The counter may be set to any initial value, including zero, and then automatically proceeds up to the maximum of 9.1 hours of operation. The contents of the counter may be accessed by any program for determination of elapsed time.

The 1813-4 provides hour-minute-second information through the 1801 console. The information is supplied as six binary-coded decimal digits (2 for each unit) in the low order of any specified main memory location as programmed.

In addition to these standard units, provision can be made to utilize an external facility time reference or to provide access from an incremental timer at the end of fixed or programmed intervals. Resolutions of up to one millisecond are possible if required.

ANALOG INPUT AND OUTPUT

Analog information may be entered into the H-1800 at rates and accuracies compatible with both modern day analog information generating systems. It is possible to enter over a million samples per second, each of the samples being accurate to plus or minus 1 percent, into a H-1800 system. More accurate information may be transferred at an appropriate decrease in information transfer rate. For instance, over three hundred thousand samples per second may be entered into an H-1800-II system with each sample having an accuracy of plus or minus 1/100 of one percent. An appropriate number of analog digital orders must be used to achieve these rates.

The above remarks are not meant to intimate that only one single in ut would be sampled at the quoted rates with the quoted accuracies. Rather, this is a composite rate at a given accuracy. This rate may be achieved by the use of multiplexing equipment feeding into an analog-digital converter which in turn is feeding the H-1800 system. Such multiplexing equipment could be used to sample, at varying rates, many input lines, each having input rates varying at much lower rates than the total input rate achievable with the H-1800 system.

Synchronization of the analog-digital converters and multiplexers and other equipment such as sample and hold devices necessary for these input configurations is available in the form of various control pulses. The input information need be presented to the H-1800-II for only approximately one third of the total transfer time. The remaining two thirds of the time is available for the external equipment to modify and to do other processing with the register on the input to the computer.

Analog outputs are available from a H-1800-II system at similar rates and accuracies to those mentioned above in conjunction with analog inputs. The outputs, again, can be generated at up to a million outputs per second with one percent accuracy. Such outputs require only the use of telemetry and digital to analog converters.

Telemetry inputs may be accepted by an H-1800 by the use of appropriate controls to assemble standard IRIG message formats into 48 bit or less bits per word combinations. This information in assembled form can be accommodated at rates approaching 8 million bits per second. An H-1800-II system may be used in conjunction with telemetry inputs to produce quick look outputs through the above described analog output systems to create partially pre-digested information on tapes for later more complete processing, and to do some of the pre-digestion on line during the acceptance of the telemetry input information. Decomputation of complex multiplex inputs is easily achieved and can greatly expand the flexibility of most telemetry systems.

Many other special type inputs can be processed by H-1800-II systems. This might be illustrated by radar data take-off systems. The outputs of such systems normally present a digital representation of the radar output to the computer. In the case of a typical search radar data processor signals are entered and two basic decision processes take place.

1. A decision on the validity of target information; that is, whether the target is to be accepted or rejected as a false target.
2. A determination of the target position perameters such as azimuth, range-rate.

This results not only in processing of information but also in condensation of the information. Such information might be used to generate an output which is fed to a tracking radar which can then lock on to a target which has been acquired by the search radar in conjunction with the data processing system.

Plotters and other display devices both incremental and XY plotters may be connected on line to an H-1800-II system. In both cases, ample control has provided for information interchange into the plotting device. The plotting device may control the length of time during which information is presented to it. Incremental plotting devices and XY plotting devices may be driven with ease from an H-1800 system and of course will not take valuable processing time from basic processing capabilities

of such a system because of the traffic control and parallel processing features of the H-1800-II.

Cathode ray tube devices may also be connected to an H-1800-II system. It is possible to utilize the computer directly to generate movement of the electron beam within the Cathode ray tube display device. It is equally feasible to utilize the computer to generate messages to Cathode ray tube display systems, which in turn, present complete displays including maps and locations of targets as well as alpha-numeric messages associated with these targets.

The above cited examples in the field of analog inputs-outputs and special applications are merely a few of the specialized on-line type applications for which the H-1800-II is extremely well suited.

MAGNETIC DRUM STORAGE

Magnetic drum storage can be provided with an H-1800 II where faster access times for from one-quarter to eleven million digits outweigh the higher cost per bit as compared to magnetic disk storage. The equipment required consists of a control unit for up to eight separate drum units. Since several cathode ray tube displays and other special systems provide a magnetic drum interface, this same basic type of control unit is applicable.

Many sizes and types of the available magnetic drums can be accommodated. A highly satisfactory type which has been investigated in detail provides 1,440,000 alphanumeric characters per unit. Eight of these units would provide 11,520,000 characters. The drum operates at 1800 RPM with 825 heads, up to four heads can be connected in parallel to provide a transfer time as low as 37.5 microseconds per word. Two heads would normally be satisfactory. Average access time would be 17 milliseconds.

It should be pointed out that disks can provide lower than normally quoted access times if less than the maximum storage is used or if high priority files are specially arranged.

SPECIAL CONSOLES

In any complex system it is often desirable to have a master control console to display status and allow control of all equipment. The H-1800 II provides a console directly associated with the central processor. This console provides an input-output typewriter, control switches, and status indicators for each program control group and peripheral device. Honeywell has an active console design group who can provide consoles with status display and control (either manual or computer override) of any number of devices and switching components. In addition, elements such as CRT displays or

inquiry stations can be incorporated. In any case, human factors engineering is applied to design to aid efficient and accurate use. The standard consoles have won a number of engineering awards for design.

DUAL CENTRAL PROCESSOR CONTROL

All of the standard peripheral control units and many special control units can be modified to allow the sharing of these units by two central processors. This Dual CP Control allows independent addressing of a control unit with priority given to the Central Processor that requests first. Once initiated, a peripheral operation maintains control until the operation is terminated. Through program control, either Central Processor can maintain control over the control unit for a series of operations when necessary for priority processing. From all operational aspects, a control unit with Dual CP Control appears to each computer as if it were connected to only that computer.

All intregal aspect of Dual CP Control is Inter-Computer Control. This Control allows the two Central Processors to communicate core-to-core so that the assemblage of peripheral equipment can be logically distributed between two Central Processors and specific assignments known to both.

IBM COMPATABILITY
HONEYWELL MODEL 836 TAPE CONTROL

The inclusion of the Honeywell Model 836 Tape Control into a Honeywell system permits a tape-to-tape means of communicating between Honeywell and IBM equipment.

The Model 836 Tape Control Unit, and its associated 729-II tape drive will read tapes which have been written on an IBM 727, 729-II, 729-IV or 7330 tape drive, and will write tapes which are readable on any of these units.

When the 836 and its associated tape unit is used in a program it is addressed the same as any of the Honeywell drives.

H-1800-II

TYPICAL EQUIPMENT CONFIGURATIONS

TYPICAL CONFIGURATIONS

1800 II

Minimum Automath System - 16K - 48DKC Tape Drives (8)

Qty.	No.	Description	Rental	Purchase Price
1	1801-II	Central Processor 8192 Words Input/Output Control Center Console	18,000	858,000
1	1802-II	Additional Memory 8192 Words	3,200	153,600
1	1801B-II	Floating Point	4,300	206,400
1	803-3	Economy TCU	2,000	96,000
8	804-3	Economy MTU 48 DKC @550/mo. @26,400/Purchase	4,400	211,200
1	822-3	High Speed Printer 120/160 - 900 LPM	1,550	74,400
1	827	Card Reader/Punch 800/250 CPM	550	30,000
			$34,000	$1,629,600

Large System - 32K - 186DKC Tape Drives (12)

Qty.	No.	Description	Rental	Purchase Price
1	1801-II	Central Processor 8192 Words Input/Output Control Center Console	18,000	858,000
3	1802-II	Additional Memory 8192 Words @3200/mo. 153,600/Purchase	9,600	460,800
1	1801B-II	Floating Point	4,300	206,400
1	803-4	Super Density TCU	4,100	196,800
12	804-4	Super Density MTU 186 DKC @900/mo. @43,200/Purchase	10,800	518,400
1	822-3	High Speed Printer 120/160 - 900 LPM	1,550	74,400
1	827	Card Reader/Punch 800/250 CPM	550	30,000
			48,900	2,344,800

This addition announces the availability of H-1800 memory beyond 32,768 words. The maximum storage limitation has now been raised to 65,536 words, with the second 32,768 available in 16,384 word modules. Thus an 1800 system may have the following memory capacities: 8,192; 16,384; 24,576; 32,768; 49,152; or 65,536 words.

Pricing information:

	Monthly Rental	Hourly Rental Charge for Extra Shift Use	Purchase Price	Purchase Option Credit (Percent)
1802-1 Additional Memory Block (16,384 words/ block)	$7,500	$17.04	$360,000	70

Maintenance	0-36 Months	37-72 Months	73-108 Months
1802-1	$375.50	$337.50	$319.00

Honeywell

ELECTRONIC DATA PROCESSING

HONEYWELL 800-II EQUIPMENT PRICE SCHEDULE

REVISED OCTOBER 1, 1963

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	CENTRAL PROCESSOR				
801-II	Central Processor (4,096 words) Console Power Unit	$10,500	$23.86	$498,000	70
801B-II	Floating-Point Option	2,100	4.77	100,800	70
802-II	4,096 Word Additional Memory Blocks (Additional memory blocks are available only in units of 4,096 words up to a maximum of 28,672 words)	1,600	3.65	76,800	70
	MAGNETIC TAPE UNITS AND CONTROLS				
803-1	Tape Control	2,000	4.54	96,000	70
803-2	High Density Tape Control	3,100	7.05	148,800	70
803-3	Economy Tape Control	2,000	4.54	96,000	70
803-4	Super Density Tape Control	4,100	9.35	196,800	70
804-1	Magnetic Tape Unit	900	2.05	43,200	50
804-2	High Density Magnetic Tape Unit	900	2.05	43,200	50
804-3	Economy Magnetic Tape Unit	550	1.25	26,400	50
804-4	Super Density Magnetic Tape Unit	900	2.05	43,200	50
805	Magnetic Tape Switching Unit	75	.17	3,600	70
	PERIPHERAL CONTROLS				
	Printer Controls				
806-1	Printer Control (for 822-1)	1,050	2.39	50,400	70
806-2	Printer Control (for 822-2)	1,250	2.84	60,000	70
806-3	Printer Control (for 822-3)	1,450	3.30	69,600	70
	Card Reader Controls				
807-1	Card Reader Control (for 823-1)	950	2.16	45,600	70
807-2	Card Reader Control (for 823-2)	1,100	2.50	52,800	70
807-3	Card Reader Control (for 827)	1,100	2.50	52,800	70

HONEYWELL ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

DSA-81A
1963
Litho in U.S.A.

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	CARD PUNCH CONTROLS				
808-1	Card Punch Control (for 824-1)	$1,050	$2.39	$50,400	70
808-3	Card Punch Control (for 827)	1,150	2.61	55,200	70
	PAPER TAPE UNITS (including Control)				
809	Paper Tape Reader and Control (1000 FPS)	690	1.57	33,120	70
810	Paper Tape Punch and Control (110 FPS) (specify model 1 for 11/16" tape or model 2 for 7/8" or 1" tape)	690	1.57	33,120	70
	MULTIPLE TERMINAL UNIT CONTROLS				
811-1	Printer – Card Reader – Card Punch Control (for use with 822-1, 823-1 or 823-2, 824-1 or 824-2)	1,700	3.86	81,600	70
811-2	Printer – Card Reader – Card Punch Control (for use with 822-2, 823-1 or 823-2, 824-1 or 824-2)	1,850	4.20	88,800	70
811-3	Printer – Card Reader – Card Punch Control (for use with 822-3, 823-1 or 823-2, 824-1 or 824-2)	1,950	4.43	93,600	70
811-4	Printer – Card Reader – Card Punch Control (for use with 822-1, 827)	1,700	3.86	81,600	70
811-5	Printer – Card Reader – Card Punch Control (for use with 822-2, 827)	1,850	4.20	88,800	70
811-6	Printer – Card Reader – Card Punch Control (for use with 822-3, 827)	1,950	4.43	93,600	70
	REAL TIME CONTROLS				
812-1	Real Time Control Unit (non-simultaneous input-output)	1,250	2.84	60,000	70
812-2	Real Time Control Unit (simultaneous input-output)	1,800	4.08	86,400	70
813-3	Programmed Elapsed Time Clock	35	.08	1,600	70
813-4	Programmed Real Time Clock	155	.35	7,500	70

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	OFF-LINE TERMINAL UNIT CONTROLS				
815	Off-Line Output Auxiliary Control	$ 700	$1.59	$33,600	70
816	Off-Line Input Auxiliary Control	700	1.59	33,600	70
817	Off-Line Input-Output Auxiliary Control	950	2.16	45,600	70
818	Off-Line Printer Control (for use with 822-3 and 804-1, 804-2 or 804-3)	1,550	3.52	74,400	70
	TERMINAL UNITS				
	PRINTERS				
822-1	Standard-Speed Printer (150 LPM) (407)	800	2.27	42,000	55
822-2	Bill-Feed Printer includes basic unit (408 model A1) equal-unequal compare (15 positions) carriage storage (15 positions)	1,175	3.34	70,125	55
822-3	High-Speed Printer (900 LPM)	1,550	3.52	74,400	70
822-3A	Vertical Spacing Option for the Model 822-3 (allows spacing of six lines per inch or eight lines per inch) An installation charge will be made if this feature is field installed	100	.23	4,800	70
822-3B	Two Speed Printing Option (600 or 900 lines per minute)	40	.09	1,920	70
	CARD READERS AND CARD PUNCHES				
823-1	Standard-Speed Card Reader (240 CPM) (085)	125	.35	7,700	55
823-2	High-Speed Card Reader (650 CPM) (088III)	325	.92	14,700	60
824-1	Standard-Speed Card Punch (100 CPM) includes basic unit (519 Model II) summary punch feature 45 columns of comparing offset stacker 30 columns double-punch blank-column detection	154	.44	7,881	50
824-1A	Heavy Duty Power Supply for the Model 824-1 (required for transcription mode punching)				545 (one time charge)
827-1	Card Reader – Card Punch (800 CPM/250 CPM) (1402) (includes early card read feature)	560	1.27	30,215	60

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	MISCELLANEOUS UNITS				
833	Magnetic Ink Character Sorter-Reader Input Control Unit	$ 1,300	$ 2.96	$ 62,400	70
	The customer may provide the terminal unit himself, or we will procure (where possible) and re-sell (not rent) to him. Only minor modifications are necessary to adapt the terminal unit to our model 833. The charges for these modifications may be quoted as not to exceed 5% of the purchase price of the terminal item.				
836	Tape Control Unit (controls one 729II Tape Unit)	1,950	4.43	93,600	70
840	Optical Scanning Unit and Control	2,530	5.75	121,440	50
860-1	Random Access Storage and Control (50 million characters)	6,100	13.90	275,000	50
860-2	Random Access Storage and Control (100 million characters)	8,100	18.50	365,000	50
860-3	Random Access Storage and Control (200 million characters)	12,500	28.50	560,000	50
860-4	Random Access Storage and Control (300 million characters)	16,900	38.50	760,000	50
860-5	Random Access Storage and Control (400 million characters)	21,300	48.50	960,000	50
860-6	Random Access Storage and Control (500 million characters)	25,700	58.50	1,160,000	50
860-7	Random Access Storage and Control (600 million characters)	30,100	68.50	1,360,000	50
860-8	Random Access Storage and Control (700 million characters)	34,500	78.50	1,560,000	50
860-9	Random Access Storage and Control (800 million characters)	38,900	88.50	1,760,000	50
872	Slave Console Typewriter	300	.85	14,400	50
880	Communications Control Unit	990	2.25	47,520	70
	BEFORE UNITS LISTED BELOW CAN BE QUOTED, HOME OFFICE APPROVAL IS REQUIRED				
	AMA Group Control Unit (handles 1-8 individual control units)	750	1.71	36,000	70
	AMA Individual Control Unit (does not include reader)	400	.91	19,200	70
	Prices for non-standard connection cables will be furnished on request.				

All prices are f.o.b. the Honeywell plant in the Greater Boston area of Massachusetts, and are subject to change without notice. Any applicable taxes involving the sale, lease, or maintenance of the above equipment are to be added to the listed prices. The purchase option credit is stated as a percentage of primary-shift rentals paid.

Honeywell
ELECTRONIC DATA PROCESSING

HONEYWELL 800-II MAINTENANCE CHARGE SCHEDULE

REVISED OCTOBER 1, 1963

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
	CENTRAL PROCESSOR				
801-II	Central Processor (4,096 words) Console Power Unit	$525.00	*1.79*	$472.50	$446.50
801B-II	Floating-Point Option	100.00	*.34*	90.00	85.00
802-II	4,096 Word Additional Memory Blocks (Additional memory blocks available in units of 4,096 words up to a maximum of 28,672 words)	80.00	*.27*	72.00	68.00
	MAGNETIC TAPE UNITS AND CONTROLS				
803-1	Tape Control	100.00	*.34*	90.00	85.00
803-2	High Density Tape Control	155.00	*.53*	139.50	132.00
803-3	Economy Tape Control	100.00	*.34*	90.00	85.00
803-4	Super Density Tape Control	205.00	*.70*	184.50	174.50
804-1	Magnetic Tape Unit	155.00	*.53*	139.50	132.00
804-2	High Density Magnetic Tape Unit	155.00	*.53*	139.50	132.00
804-3	Economy Magnetic Tape Unit	100.00	*.34*	90.00	85.00
804-4	Super Density Magnetic Tape Unit	155.00	*.53*	139.50	132.00
805	Magnetic Tape Switching Unit	5.00	*.02*	4.50	4.50
	PERIPHERAL CONTROLS				
	PRINTER CONTROLS				
806-1	Printer Control (for 822-1)	55.00	*.19*	49.50	47.00
806-2	Printer Control (for 822-2)	100.00	*.34*	90.00	85.00
806-3	Printer Control (for 822-3)	115.00	*.39*	103.50	98.00
	CARD READER CONTROLS				
807-1	Card Reader Control (for 823-1)	50.00	*.17*	45.00	42.50
807-2	Card Reader Control (for 823-2)	60.00	*.20*	54.00	51.00
807-3	Card Reader Control (for 827)	60.00	*.20*	54.00	51.00

HONEYWELL ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

DSA-78A
1963
Litho in U.S.A.

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
	CARD PUNCH CONTROLS				
808-1	Card Punch Control (for 824-1)	$ 60.00	.20	$ 54.00	$ 51.00
808-2	Card Punch Control (for 824-2)	65.00	.22	58.50	55.50
808-3	Card Punch Control (for 827)	65.00	.22	58.50	55.50
	PAPER TAPE UNITS (including Control)				
809	Paper Tape Reader and Control	69.00	.24	62.00	59.00
810	Paper Tape Punch and Control (Model 1 or 2)	69.00	.24	62.00	59.00
	MULTIPLE TERMINAL UNIT CONTROLS				
811-1	Printer – Card Reader – Card Punch Control (for use with 822-1, 823-1 or 823-2, 824-1 or 824-2)	85.00	.29	76.50	72.50
811-2	Printer – Card Reader – Card Punch Control (for use with 822-2, 823-1 or 823-2, 824-1 or 824-2)	120.00	.41	108.00	102.00
811-3	Printer – Card Reader – Card Punch Control (for use with 822-3, 823-1 or 823-2, 824-1 or 824-2)	150.00	.51	135.00	127.50
811-4	Printer – Card Reader – Card Punch Control (for use with 822-1, 827)	85.00	.29	76.50	72.50
811-5	Printer – Card Reader – Card Punch Control (for use with 822-2, 827)	120.00	.41	108.00	102.00
811-6	Printer – Card Reader – Card Punch Control (for use with 822-3, 827)	150.00	.51	135.00	127.50
	REAL TIME CONTROLS				
812-1	Real Time Control Unit (non-simultaneous input-output)	62.50	.21	56.50	53.00
812-2	Real Time Control Unit (simultaneous input-output)	90.00	.31	81.00	76.50
813-3	Programmed Elapsed Time Clock	2.00	.01	1.50	1.50
813-4	Programmed Real Time Clock	8.00	.03	7.50	7.00

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
	OFF-LINE TERMINAL UNIT CONTROLS				
815	Off-Line Output Auxiliary Control	$ 50.00	.17	$ 45.00	$ 42.50
816	Off-Line Input Auxiliary Control	50.00	.17	45.00	42.50
817	Off-Line Input-Output Auxiliary Control	70.00	.24	63.00	59.50
818	Off-Line Printer Control	120.00	.41	108.00	102.00
	TERMINAL UNITS				
	PRINTERS				
822-1	Standard-Speed Printer	147.50	.50	176.75	208.75
822-2	Bill-Feed Printer (basic unit plus recommended attachments)	190.00	.65	223.25	260.75
822-3	High-Speed Printer	310.00	1.06	279.00	263.50
822-3A	Vertical Spacing Option for the Model 822-3 (allows spacing of six lines per inch or eight lines per inch)	20.00	.07	18.00	17.00
822-3B	Two Speed Printing Option (600 or 900 lines per minute)	8.00	.03	7.20	6.80
	CARD READERS AND CARD PUNCHES				
823-1	Standard-Speed Card Reader	15.00	.05	22.50	28.50
823-2	High-Speed Card Reader	52.25	.18	78.00	103.25
824-1	Standard-Speed Card Punch (basic unit plus recommended attachments)	39.50	.13	49.75	60.00
824-1A	Heavy Duty Power Supply for the Model 824-1 (required for transcription mode punching)	—		—	—
824-2	High-Speed Card Punch (basic unit plus recommended attachments)	35.75	.12	43.00	53.00
827-1	Card Reader – Card Punch (includes early card read feature)	45.00	.15	61.25	84.75
	MISCELLANEOUS UNITS				
833	Magnetic Ink Character Sorter-Reader Input Control Unit	87.50	.30	79.00	74.50
836	Tape Control Unit	195.00	.67	175.50	166.00

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
840	Optical Scanning Unit and Control	$ 380.00	*1.30*	$ 342.00	$ 323.00
860-1	Random Access Storage and Control (50 million characters)	730.00	*2.49*	657.00	620.50
860-2	Random Access Storage and Control (100 million characters)	920.00	*3.14*	828.00	782.00
860-3	Random Access Storage and Control (200 million characters)	1,500.00	*5.11*	1,350.00	1,275.00
860-4	Random Access Storage and Control (300 million characters)	2,100.00	*7.16*	1,890.00	1,785.00
860-5	Random Access Storage and Control (400 million characters)	2,600.00	*8.86*	2,340.00	2,210.00
860-6	Random Access Storage and Control (500 million characters)	3,100.00	*10.57*	2,790.00	2,635.00
860-7	Random Access Storage and Control (600 million characters)	3,700.00	*12.61*	3,330.00	3,145.00
860-8	Random Access Storage and Control (700 million characters)	4,140.00	*14.11*	3,726.00	3,519.00
860-9	Random Access Storage and Control (800 million characters)	4,670.00	*15.92*	4,203.00	3,969.50
872	Slave Console Typewriter	60.00	*.20*	54.00	51.00
880	Communications Control Unit	90.00	*.31*	81.00	76.50

The monthly maintenance charges shown on this schedule apply to prime-shift usage, five days per week, Monday through Friday, excluding holidays.

Honeywell
ELECTRONIC DATA PROCESSING

HONEYWELL 1800-II EQUIPMENT PRICE SCHEDULE

REVISED OCTOBER 1, 1963

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	CENTRAL PROCESSOR				
1801-II	Central Processor (8,192 words) Console Power Unit	$18,000	$41.00	$858,000	70
1801B-II	Floating-Point Option	4,300	9.80	206,400	70
1802-II	Additional Memory Blocks (8,192 words per block) (maximum of three additional memory blocks per system)	3,200	7.27	153,600	70
1802-II-1	16,384 Word Additional Memory Block (Maximum of two additional blocks per system. Basic requirement of three 1802-II blocks.)	7,500	17.04	360,000	70
1813-3	Programmed Elapsed Time Indicator	35	.08	1,600	70
1813-4	Programmed Real Time Clock	155	.35	7,500	70
	MAGNETIC TAPE UNITS AND CONTROLS				
803-1	Tape Control	2,000	4.54	96,000	70
803-2	High Density Tape Control	3,100	7.05	148,800	70
803-3	Economy Tape Control	2,000	4.54	96,000	70
803-4	Super Density Tape Control	4,100	9.35	196,800	70
804-1	Magnetic Tape Unit	900	2.05	43,200	50
804-2	High Density Magnetic Tape Unit	900	2.05	43,200	50
804-3	Economy Magnetic Tape Unit	550	1.25	26,400	50
804-4	Super Density Magnetic Tape Unit	900	2.05	43,200	50
805	Magnetic Tape Switching Unit	75	.17	3,600	70
	PERIPHERAL CONTROLS				
	Printer Controls				
806-1	Printer Control (for 822-1)	1,050	2.39	50,400	70
806-2	Printer Control (for 822-2)	1,250	2.84	60,000	70
806-3	Printer Control (for 822-3)	1,450	3.30	69,600	70
	Card Reader Controls				
807-1	Card Reader Control (for 823-1)	950	2.16	45,600	70
807-2	Card Reader Control (for 823-2)	1,100	2.50	52,800	70
807-3	Card Reader Control (for 827)	1,100	2.50	52,800	70

HONEYWELL ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS

DSA-82A
1963
Litho in U.S.A.

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	CARD PUNCH CONTROLS				
808-1	Card Punch Control (for 824-1)	$1,050	$2.39	$50,400	70
808-3	Card Punch Control (for 827)	1,150	2.61	55,200	70
	PAPER TAPE UNITS (including Control)				
809	Paper Tape Reader and Control (1000 FPS)	690	1.57	33,120	70
810	Paper Tape Punch and Control (110 FPS) (specify model 1 for 11/16" tape or model 2 for 7/8" or 1" tape)	690	1.57	33,120	70
	MULTIPLE TERMINAL UNIT CONTROLS				
811-1	Printer – Card Reader – Card Punch Control (for use with 822-1, 823-1 or 823-2, 824-1 or 824-2)	1,700	3.86	81,600	70
811-2	Printer – Card Reader – Card Punch Control (for use with 822-2, 823-1 or 823-2, 824-1 or 824-2)	1,850	4.20	88,800	70
811-3	Printer – Card Reader – Card Punch Control (for use with 822-3, 823-1 or 823-2, 824-1 or 824-2)	1,950	4.43	93,600	70
811-4	Printer – Card Reader – Card Punch Control (for use with 822-1, 827)	1,700	3.86	81,600	70
811-5	Printer – Card Reader – Card Punch Control (for use with 822-2, 827)	1,850	4.20	88,800	70
811-6	Printer – Card Reader – Card Punch Control (for use with 822-3, 827)	1,950	4.43	93,600	70
	REAL TIME CONTROLS				
812-1	Real Time Control Unit (non-simultaneous input-output)	1,250	2.84	60,000	70
812-2	Real Time Control Unit (simultaneous input-output)	1,800	4.08	86,400	70

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	OFF-LINE TERMINAL UNIT CONTROLS				
815	Off-Line Output Auxiliary Control	$ 700	$1.59	$33,600	70
816	Off-Line Input Auxiliary Control	700	1.59	33,600	70
817	Off-Line Input-Output Auxiliary Control	950	2.16	45,600	70
818	Off-Line Printer Control (for use with 822-3 and 804-1, 804-2 or 804-3)	1,550	3.52	74,400	70
	TERMINAL UNITS				
	PRINTERS				
822-1	Standard-Speed Printer (150 LPM) (407)	800	2.27	42,000	55
822-2	Bill-Feed Printer includes basic unit (408 model A1) equal-unequal compare (15 positions) carriage storage (15 positions)	1,175	3.34	70,125	55
822-3	High-Speed Printer (900 LPM)	1,550	3.52	74,400	70
822-3A	Vertical Spacing Option for the Model 822-3 (allows spacing of six lines per inch or eight lines per inch) An installation charge will be made if this feature is field installed	100	.23	4,800	70
822-3B	Two-Speed Printing Option (600 or 900 lines per minute)	40	.09	1,920	70
	CARD READERS AND CARD PUNCHES				
823-1	Standard-Speed Card Reader (240 CPM) (085)	125	.35	7,700	55
823-2	High-Speed Card Reader (650 CPM) (088III)	325	.92	14,700	60
824-1	Standard-Speed Card Punch (100 CPM) includes basic unit (519 Model II) summary punch feature 45 columns of comparing offset stacker 30 columns double-punch blank-column detection	154	.44	7,881	50
824-1A	Heavy Duty Power Supply for the Model 824-1 (required for transcription mode punching)				545 (one time charge)
827-1	Card Reader – Card Punch (800 CPM/250 CPM) (1402) (includes early card read feature)	560	1.27	30,215	60

MODEL NUMBER	DESCRIPTION	MONTHLY RENTAL	HOURLY RENTAL CHARGE FOR EXTRA SHIFT USE	PURCHASE PRICE	PURCHASE OPTION CREDIT (PERCENT)
	MISCELLANEOUS UNITS				
833	Magnetic Ink Character Sorter-Reader Input Control Unit	$ 1,300	$ 2.96	$ 62,400	70
	The customer may provide the terminal unit himself, or we will procure (where possible) and re-sell (not rent) to him. Only minor modifications are necessary to adapt the terminal unit to our model 833. The charges for these modifications may be quoted as not to exceed 5% of the purchase price of the terminal item.				
836	Tape Control Unit (controls one 729II Tape Unit)	1,950	4.43	93,600	70
840	Optical Scanning Unit and Control	2,530	5.75	121,440	50
860-1	Random Access Storage and Control (50 million characters)	6,100	13.90	275,000	50
860-2	Random Access Storage and Control (100 million characters)	8,100	18.50	365,000	50
860-3	Random Access Storage and Control (200 million characters)	12,500	28.50	560,000	50
860-4	Random Access Storage and Control (300 million characters)	16,900	38.50	760,000	50
860-5	Random Access Storage and Control (400 million characters)	21,300	48.50	960,000	50
860-6	Random Access Storage and Control (500 million characters)	25,700	58.50	1,160,000	50
860-7	Random Access Storage and Control (600 million characters)	30,100	68.50	1,360,000	50
860-8	Random Access Storage and Control (700 million characters)	34,500	78.50	1,560,000	50
860-9	Random Access Storage and Control (800 million characters)	38,900	88.50	1,760,000	50
872	Slave Console Typewriter	300	.85	14,400	50
880	Communications Control Unit	990	2.25	47,520	70
	BEFORE UNITS LISTED BELOW CAN BE QUOTED, HOME OFFICE APPROVAL IS REQUIRED				
	AMA Group Control Unit (handles 1-8 individual control units)	750	1.71	36,000	70
	AMA Individual Control Unit (does not include reader)	400	.91	19,200	70
	Prices for non-standard connection cables will be furnished on request.				

All prices are f.o.b. the Honeywell plant in the Greater Boston area of Massachusetts, and are subject to change without notice. Any applicable taxes involving the sale, lease, or maintenance of the above equipment are to be added to the listed prices. The purchase option credit is stated as a percentage of primary-shift rentals paid.

Honeywell
ELECTRONIC DATA PROCESSING

HONEYWELL 1800-II MAINTENANCE CHARGE SCHEDULE

REVISED OCTOBER 1, 1963

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
	CENTRAL PROCESSOR				
1801-II	Central Processor (8,192 words) Console Power Unit	$900.00	*5.11*	$810.00	$765.00
1801B-II	Floating-Point Option	215.00	*1.22*	193.50	183.00
1802-II	Additional Memory Blocks (8,192 words per block) (maximum of three additional memory blocks per system)	160.00	*.91*	144.00	136.00
1802-II-1	16,384 Word Additional Memory Block (Maximum of two additional blocks per system. Basic requirement of three 1802 blocks.)	375.50	*2.13*	337.50	319.00
1813-3	Programmed Elapsed Time Clock	2.00	*.01*	1.50	1.50
1813-4	Programmed Real Time Clock	8.00	*.05*	7.50	7.00
	MAGNETIC TAPE UNITS AND CONTROLS				
803-1	Tape Control	100.00	*.34*	90.00	85.00
803-2	High Density Tape Control	155.00	*.53*	139.50	132.00
803-3	Economy Tape Control	100.00	*.34*	90.00	85.00
803-4	Super Density Tape Control	205.00	*.70*	184.50	174.50
804-1	Magnetic Tape Unit	155.00	*.53*	139.50	132.00
804-2	High Density Magnetic Tape Unit	155.00	*.53*	139.50	132.00
804-3	Economy Magnetic Tape Unit	100.00	*.34*	90.00	85.00
804-4	Super Density Magnetic Tape Unit	155.00	*.53*	139.50	132.00
805	Magnetic Tape Switching Unit	5.00	*.02*	4.50	4.50
	PERIPHERAL CONTROLS				
	PRINTER CONTROLS				
806-1	Printer Control (for 822-1)	55.00	*.19*	49.50	47.00
806-2	Printer Control (for 822-2)	100.00	*.34*	90.00	85.00
806-3	Printer Control (for 822-3)	115.00	*.39*	103.50	98.00
	CARD READER CONTROLS				
807-1	Card Reader Control (for 823-1)	50.00	*.17*	45.00	42.50
807-2	Card Reader Control (for 823-2)	60.00	*.20*	54.00	51.00
807-3	Card Reader Control (for 827)	60.00	*.20*	54.00	51.00

HONEYWELL ELECTRONIC DATA PROCESSING DIVISION
WELLESLEY HILLS 81, MASSACHUSETTS
DSA-79A
1963
Litho in U.S.A.

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
	CARD PUNCH CONTROLS				
808-1	Card Punch Control (for 824-1)	$ 60.00	.20	$ 54.00	$ 51.00
808-2	Card Punch Control (for 824-2)	65.00	.22	58.50	55.50
808-3	Card Punch Control (for 827)	65.00	.22	58.50	55.50
	PAPER TAPE UNITS (including Control)				
809	Paper Tape Reader and Control	69.00	.24	62.00	59.00
810	Paper Tape Punch and Control (Model 1 or 2)	69.00	.24	62.00	59.00
	MULTIPLE TERMINAL UNIT CONTROLS				
811-1	Printer – Card Reader – Card Punch Control (for use with 822-1, 823-1 or 823-2, 824-1 or 824-2)	85.00	.29	76.50	72.50
811-2	Printer – Card Reader – Card Punch Control (for use with 822-2, 823-1 or 823-2, 824-1 or 824-2)	120.00	.41	108.00	102.00
811-3	Printer – Card Reader – Card Punch Control (for use with 822-3, 823-1 or 823-2, 824-1 or 824-2)	150.00	.51	135.00	127.50
811-4	Printer – Card Reader – Card Punch Control (for use with 822-1, 827)	85.00	.29	76.50	72.50
811-5	Printer – Card Reader – Card Punch Control (for use with 822-2, 827)	120.00	.41	108.00	102.00
811-6	Printer – Card Reader – Card Punch Control (for use with 822-3, 827)	150.00	.51	135.00	127.50
	REAL TIME CONTROLS				
812-1	Real Time Control Unit (non-simultaneous input-output)	62.50	.21	56.50	53.00
812-2	Real Time Control Unit (simultaneous input-output)	90.00	.31	81.00	76.50

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
	OFF-LINE TERMINAL UNIT CONTROLS				
815	Off-Line Output Auxiliary Control	$ 50.00	.17	$ 45.00	$ 42.50
816	Off-Line Input Auxiliary Control	50.00	.17	45.00	42.50
817	Off-Line Input-Output Auxiliary Control	70.00	.24	63.00	59.50
818	Off-Line Printer Control	120.00	.41	108.00	102.00
	TERMINAL UNITS				
	PRINTERS				
822-1	Standard-Speed Printer	147.50	.50	176.75	208.75
822-2	Bill-Feed Printer (basic unit plus recommended attachments)	190.00	.65	223.25	260.75
822-3	High-Speed Printer	310.00	1.06	279.00	263.50
822-3A	Vertical Spacing Option for the Model 822-3 (allows spacing of six lines per inch or eight lines per inch)	20.00	.07	18.00	17.00
822-3B	Two Speed Printing Option (600 or 900 lines per minute)	8.00	.03	7.20	6.80
	CARD READERS AND CARD PUNCHES				
823-1	Standard-Speed Card Reader	15.00	.05	22.50	28.50
823-2	High-Speed Card Reader	52.25	.18	78.00	103.25
824-1	Standard-Speed Card Punch (basic unit plus recommended attachments)	39.50	.13	49.75	60.00
824-1A	Heavy Duty Power Supply for the Model 824-1 (required for transcription mode punching)	—		—	—
824-2	High-Speed Card Punch (basic unit plus recommended attachments)	35.75	.12	43.00	53.00
827-1	Card Reader – Card Punch (includes early card read feature)	45.00	.15	61.25	84.75
	MISCELLANEOUS UNITS				
833	Magnetic Ink Character Sorter-Reader Input Control Unit	87.50	.30	79.00	74.50
836	Tape Control Unit	195.00	.67	175.50	166.00

MODEL NUMBER	DESCRIPTION	MONTHLY MAINTENANCE CHARGES BASED ON AGE OF MACHINES IN MONTHS 0-36	Hourly Extra Shift Use Charge 0-36 Mos.	37-72	73-108
840	Optical Scanning Unit and Control	$ 380.00	*1.30*	$ 342.00	$ 323.00
860-1	Random Access Storage and Control (50 million characters)	730.00	*2.49*	657.00	620.50
860-2	Random Access Storage and Control (100 million characters)	920.00	*3.14*	828.00	782.00
860-3	Random Access Storage and Control (200 million characters)	1,500.00	*5.11*	1,350.00	1,275.00
860-4	Random Access Storage and Control (300 million characters)	2,100.00	*7.16*	1,890.00	1,785.00
860-5	Random Access Storage and Control (400 million characters)	2,600.00	*8.86*	2,340.00	2,210.00
860-6	Random Access Storage and Control (500 million characters)	3,100.00	*10.57*	2,790.00	2,635.00
860-7	Random Access Storage and Control (600 million characters)	3,700.00	*12.61*	3,330.00	3,145.00
860-8	Random Access Storage and Control (700 million characters)	4,140.00	*14.11*	3,726.00	3,519.00
860-9	Random Access Storage and Control (800 million characters)	4,670.00	*15.92*	4,203.00	3,969.50
872	Slave Console Typewriter	60.00	*.20*	54.00	51.00
880	Communications Control Unit	90.00	*.31*	81.00	76.50

The monthly maintenance charges shown on this schedule apply to prime-shift usage, five days per week, Monday through Friday, excluding holidays.

H-1800-II

A COMPETITIVE ANALYSIS

DEMONSTRATING A CLEAR CUT PRICE/POWER SUPERIORITY

H-1800-II

Price/Power Superiority

The charts in this section demonstrate a substantial competitive advantage for the H-1800-II over all other systems in absolute job cost.

SCIENTIFIC BENCHMARK

Based on ten floating point adds, one floating point multiply and one 120 character transfer the H-1800-II will execute in 179 microseconds versus from 238 to 939.7 microseconds for competitive systems. Even more dramatic is the relative cost ratios for this benchmark based on a per microsecond rental cost. The H-1800-II has a superior job/cost ratio of from 2.1 to 7.3 over all competitive systems.

10,000 RECORD PROCESSING BENCHMARK

Based on the processing of 10,000 records of 1920 characters each on magnetic tape the H-1800-II will be from 30% to 360% lower in job cost. With a complete overlap of Read/Compute/Write and the utilization of 186,000 Decimal Digits per second tape transfer rate the H-1800-II represents a powerful processor for data manipulation operations.

PRICE/POWER SUPERIORITY

SYSTEM	GENERAL: Effective System Rental Range ($K)	GENERAL: Primary Market	CENTRAL PROCESSOR: Memory Min.-Max. Published	CENTRAL PROCESSOR: Memory Min.-Max. Equivalent Alpha Characters	CENTRAL PROCESSOR: Word Size	CENTRAL PROCESSOR: Cycle Time Published	CENTRAL PROCESSOR: Cycle Time Adjusted 48 bit word	CENTRAL PROCESSOR: Instruction Type	CENTRAL PROCESSOR: Index Registers (Maximum)	TIMING (USECS) FIXED A+B=C 5 Digits	FIXED A+B=C 8 Digits	FIXED A+B=B 5 Digits	FIXED A+B=B 8 Digits	DECIMAL A X B=C 5 Digits	DECIMAL A X B=C 8 Digits	Fixed Binary (Word) A+B=C	Fixed Binary (Word) A+B=B	Fixed Binary (Word) A X B=C	Floating Binary (Word) A+B=C	Floating Binary (Word) A X B=C	Internal Transfer 120 Char.	Tapes-Maximum Number	Simultaneity & Other Features	Benchmark 10 Floating Point Adds 1 Floating Point Multiply 1 120 Char. transfer usec	Average Monthly Rental $ K/Mo.	Hourly Agreement	Cost/usec 10^{-10}	Benchmark Cost 10^{-10}	Ratio
Honeywell 1800	$27-$60	Mixed	8k - 32k	64k - 256k	48 Binary 12 Decimal 8 Alpha	2	2	3-Address	64	8	8	8	8	54	54	8	8	66	9.9	10	70	64	MRWC	179	43.5	200	.6	107.5	1
IBM 7094	$55-$75	Scientific	32k - 32k	192k - 192k	36 Binary	2	2.7	Single Address	7							10	10	16	12	16	102	80	MRWC	238	65	176	1.05	250	2.3
7090	$50-$70	Scientific	32k - 32k	192k - 192k	36 Binary	2.16	2.91	Single Address	3							13.08	13.08	34.01	22.67	32.7	264.4	80	MRWC	523.8	60	176	.95	499.7	4.6
7080	$40-$70	Business	80k - 160k	80k - 160k	Character	2 (1)	16	Single Address 5-Char. Instruction		35*	47*	24*	33*	123*	193.6*						201*	40	MRWC	No Floating Point	55	176	.87	-	-
7074	$20-$40	Business	5k - 30k	25k - 150k	10 Decimal 5 Alpha	4		Single Address	99	24	24	20	20	72	72				32	72	204	40	MRWC	596	30	176	.47	281	2.7
7044	$20 $55	Scientifid	8k - 32k	48k - 192k	36 Binary	2.5	3.3	Single Address	3							15	15	40	24	35av.	105	50	MRWC	380	37.5	176	.59	224	2.1
Philco 2000/210	$35 $70	Scientifid	8k - 32k	64k - 256k	48 Binary	10	10	Single Address 2/word	32							45	34.8	122.2	51.9	99.9	320.8	64	MRWC	939.7	52.5	176	.83	780	7.3
CDC 3600	$40-$75	Scientific	32k-262k	256k-2096k	48 Binary	1.5	1.5	Single Address Half or full word	6							4.4	4.4	8.3	6.3	8.3	465*	over 512	MRWC	536.3	57.5	176	.91	488	4.5
1604A	$25-$45	Scientific	8k-32k	64k-256k	48 Binary 8 Alpha	6.4	6.4	Single Address, Half Word	6							21.6	20.4	39.6 to 77.2	33.2	50.4	300*	32	MRWC	682.6	46.0	176	.55	375	3.5
Univac 1107	$32-$60	Scientific	16k-65k	96k-390k	36 Binary	4.0	5.3	Single Address	5							24	24	32.7	26	32.7	164*	84	MRWC	456.7	46.0	176	.73	333	3.1

*ESTIMATED

(1) 7080 Has a 1 usec Control Memory of 1000 Characters

H-1800-II

PRICE/POWER SUPERIORITY

10,000 RECORDS PROCESSED

INPUT/OUTPUT CAPABILITIES ARE BASED UPON THE MAXIMUM NUMBER OF TAPE UNITS OPERATING SIMULTANEOUSLY AT FULL SPEED PROCESSING CAPABILITIES ARE BASED UPON THE TIME AVAILABLE WHILE MAINTAINING MAXIMUM INPUT/OUTPUT (RECORD) AND PROCESS ONLY (HOURLY).

EDPM MANUFACTURER MACHINE	RENTAL (176 HRS.) I/O MONTHLY / HOURLY	RENTAL (176 HRS.) PROG MONTHLY / HOURLY	RENTAL (176 HRS.) TOTAL MONTHLY / HOURLY	INPUT/OUTPUT (HOURLY) PER TAPE UNIT OPERATING SIMULTANEOUSLY	PER TAPE UNIT RECORDS	PER TAPE UNIT CHARACTERS	TOTAL RECORDS / CHARACTERS	COST/10,000 RCDS. INPUT/OUTPUT	PROC	TOTAL	EQUIPMENT RENTAL INPUT/OUTPUT	EQUIPMENT RENTAL PROC	EQUIPMENT RENTAL TOTAL	RATINGS (IBM-7090 BASE) INPUT/OUTPUT (Total RCDS)	PROCESSING (FP) HOURLY RECORD A + B→C	A · B→C	A ÷ B→C	COST/10,000 RECORDS INPUT/OUTPUT	PROC	TOTAL
Burroughs 5000	18,750 / 106.53	20,950 / 119.03	39,700 / 225.56	4-422TU	102,701	197,185,920	410,804 / 788,743,680	2.59	2.90	5.49	.57	.50	.53	1.42	1.64 / 1.17	1.22 / .87	.92 / .66	.40	.35	.37
CONTROL Data Corp. 3600	23,505 / 133.55	38,770 / 220.28	62,275 / 353.83	3-606TU	128,377	246,483,840	385,131 / 739,451,520	3.47	5.72	9.19	.71	.91	.83	1.33	3.28 / 2.54	3.66 / 2.83	2.31 / 1.79	.53	.68	.62
Honeywell 1800	28,800 / 163.64	32,100 / 182.38	60,900 / 364.02	4-804IVTU 2-729IITU	170,164 64,188	326,714,880 123,240,960	809,032 / 1,553,341,440	2.02	2.25	4.27	.87	.76	.81	2.80	2.30 / .73	2.75 / .87	1.85 / .58	.31	.27	.28
IBM 7044	23,435 / 133.18	23,045 / 130.94	46,480 / 264.09	3-729VITU	128,571	246,856,320	385,713 / 740,568,960	3.45	3.39	6.84	.71	.54	.61	1.33	.95 / .69	.94 / .68	.67 / .48	.53	.40	.46
IBM 7090	32,800 / 186.36	42,200 / 239.77	75,000 / 426.13	3-729IVTU	96,283	184,863,360	288,849 / 554,590,080	8.45	8.30	14.75	1.00	1.00	1.00	1.00	1.00 / 1.00	1.00 / 1.00	1.00 / 1.00	1.00	1.00	1.00
IBM 7094	34,005 / 193.21	48,150 / 273.58	82,155 / 466.79	3-729VITU	128,571	246,856,320	385,713 / 740,568,960	5.01	7.09	12.10	1.03	1.14	1.09	1.33	1.91 / 1.41	2.06 / 1.52	1.54 / 1.14	.77	.85	.82
PHILCO 212	34,300 / 194.89	56,075 / 318.60	90,375 / 513.49	3-334TU	274,290	526,636,800	822,870 / 1,579,910,400	2.37	3.87	6.24	1.04	1.32	1.20	2.84	4.60 / 1.63	4.12 / 1.47	2.31 / .82	.36	.46	.42
Univac 1107	21.450 / 121.88	31,450 / 178.69	52,900 / 300.57	2-IIIATU 1-IIATU	194,594 96,283	373,620,480 184,863,360	485,471 / 932,104,320	2.51	3.68	6.19	.65	.74	.70	1.68	.95 / .55	1.43 / .82	1.02 / .59	.38	.44	.41

HONEYWELL SCIENTIFIC USERS

HONEYWELL SCIENTIFIC USERS

The following H800 installations are a representative group of H800 customers that utilize their equipment wholly or in part on scientific applications.

Ames Research NASA
Moffet Field
California
- Data Reduction - Wind Tunnel
- Pert
- Numerical Analysis

Army Map Service
Washington, D.C.
- Secor Project - Satellite Mapping
- Triangulation Adjustment
- Coordinate Conversion

Chrysler Corp.
Engineering Department
Detroit, Michigan
- Dynamical Analysis
- Data Correlation
- Turbine Testing Synthesis and Analysis
- Structural Analysis

General Motors
Delco-Remy
Anerson, Indiana
- General Scientific

General Motors Corp.
3044 West Grand Boulevard
Detroit, Michigan
- General Scientific

Honeywell - Aero Division
Minneapolis, Minnesota
- Guidance and Control Systems for Space Vehicles
- Formula Evaluation
- Partner - Analog Simulation

Honeywell - Aero Division
St. Petersburg, Florida
    Vehicle Error Analysis
    Control Loop Analysis
    Equipment Simulation

Massachusetts Institute of Technology (H-1800 Nov. 1963)
Instrumentation Laboratory
Cambridge, Massachusetts
    General Scientific

National Institute of Health
9000 Wisconsin Avenue
Bethesda 14, Maryland
    Kroll Problem - Statistical Analysis
    Cancer Studies - Statistical Analysis
    Physiological Research - Data Reduction

National Library of Medicine (Medlars)
8600 Wisconsin Avenue
Bethesda, Maryland
    Information Retrieval
    Document Indexing (Index Medicus)

National Weather Records
Asheville, North Carolina
    Data Reduction
    Frequency Distributions
    Multiple Regression
    Correlation
    Curne Fitting
    Harmonic Analysis

A. C. Nielsen
2100 West Howard Street
Chicago, Illinois
    Error Studies

Northern Illinois Gas
Aurora, Illinois
    Power Engineering Computations
    General Scientific and Statistical

Philadelphia Electric
Edison Building
Philadelphia, Penna.
    Electrical Load Flow
    Transmission Line
    Steam Station Performance
    Power Engineering Computations

Submarine Base, U.S. Navy
Naval Training Device Center
New London, Connecticut
    Simulation - War Games

University of Southern California
University Park
Los Angeles, California
    Management Games
    General Scientific

HONEYWELL'S TOTAL CORPORATE SYSTEM'S CAPABILITY

SECTION I

INTRODUCTION

A. Background

The Minneapolis-Honeywell Regulator Company was organized for the specific purpose of producing automatic controls and automatic control systems. The company has grown into a corporation which employs approximately 47,000 people with 45 plants throughout the world. In 1962 Honeywell sales increased to more than 595 million dollars, of which approximately one third was military sales.

Although the total sales for 1962 represented a broad line of components, devices and systems, control and control technology was basic to a large majority of these. This was not only true in the commercial area, but particularly so in the area of military systems and programs.

B. Organization

An over-all corporate organization chart is shown in Figure 1. This chart includes a very brief description of some of the prime capabilities of each of the major divisions of the company. A more detailed discussion concerning these capabilities will be found in Section II.

The unique combination of capabilities which is presented in Figure 1 provides the Honeywell organization with a broad spectrum of capabilities to tackle any major systems management program which may arise. One example of this type of capability was demonstrated by the Honeywell systems

management of the Navy ASROC System. The ASROC program, as well as several other similar efforts, will be discussed in some detail in Section II.

SECTION II

DIVISIONAL CAPABILITIES

A. Introduction

As previously noted, Honeywell is organized into several major product groups. Each group, in turn, is composed of several divisions. For the purposes of this presentation, only three specific divisions will be described, the Ordnance, Aeronautical, and EDP Divisions. In addition, comments will be made concerning the MPG Research Organization located in Minneapolis.

B. Aeronautical Division

The Aeronautical Division is composed of three major subdivisions. These are located in Minneapolis, Minnesota; St. Petersburg, Florida; and Boston, Massachusetts. The Division in concerned with: airborne control systems, guidance flight controls, human factors, inertial guidance, and navigation and guidance computers.

The Aeronautical Division has performed many major programs for the Defense Establishment. In the following pages several of the more significant programs, with their goals and a performance summary, will be presented.

1. Project Mercury

Honeywell's major contribution to the Mercury Program included an automatic stabilization and control system, a rate stabilization and control system, and an attitude and rate indicating system.

In addition to these systems and other equipments, Honeywell has provided the basic man-machine human factors analysis for the Mercury Program.

To insure success, several back-up control systems were provided which permitted the astronaut to select either complete manual take-over or semi-automatic operation. The primary back-up is a manual flight control system which has an independent set of reaction jets to further guarantee the success of the flight. An additional secondary back-up system utilizes switches on the pilot's control stick to allow "fly-by-wire" control through the reaction jet system.

The systems performance was considered excellent, as was pilot acceptance of the system. In actual test the system reliability exceeded the predicted value.

2. X-20 (Dyna-Soar)

Under the Dyna-Soar program, Honeywell is responsible for the primary glider guidance sub-system. The electronic circuitry for this system is based upon high frequency magnetic amplifiers developed for the F-101 adaptive control system. This system was also employed on the X-15 Aero Space test vehicle.

One interesting aspect of the X-20 Program has been the integration of the PERT control networks employed for the development and test of both the airborne and ground equipment with the associated contractors. The Honeywell PERT networks must interface perfectly with the PERT networks of the

Boeing Company, the systems contractor. Such integration has proven to be most advantageous in promoting the establishment of compatible program planning for an expeditious effort.

3. Wag-Tail

Under a company sponsored internal effort a completely new concept for a missile weapons system was postulated and investigated. This program--known as Wag-Tail--was for a highly maneuverable jet-vane controlled air-to-surface missile. Extensions of the original program were funded by the Air Force Armament Center. Under that extension a guidance and control system, airborne structure design, dynamic analysis, as well as maneuverability, low and high speed wind tunnel studies and ground support equipment were developed.

This program was considered an outstanding example of the industrial -- military team approach to a problem. During the testing, no failures occurred in any of the systems.

4. Apollo Spacecraft Program

The Honeywell Aeronautical Division is currently carrying out a program asa major sub-contractor to the North American Aviation Company to provide a complete Apollo system. Under this contract Honeywell is responsible for the closed loop flight control system and its performance. This involves the design and provisioning of the manual controls, the interfacing of the signals to the jets, the pulse modulation and fuel management schemes, as well as the sizing of the valves and jets themselves. The sensing of the

attitude movement and the conversion to signals involves the utilization of a family of sensors. Honeywell is responsible for the provisioning of the sun-seeker, horizon scanner, accelerometers, and gyros. This attitude reference system includes a sophisticated inertial reference package. The transmission of the output signals of these devices to a display system is also a Honeywell responsibility. The design and provisioning of these displays include the necessary attitude indicator, switching modes, and associated flight control instruments required for the crew.

The study of the crew work pattern, both continuous and discontinuous, is also a responsibility of Honeywell, as is the integration of this information to provide an optimal man-machine control system.

The Apollo program, from its inception as a pre-proposal effort, has incorporated intensive and careful management analysis and control practices. This has included PERT as well as other accepted practices. The primary purpose of the regularly generated PERT reports is to provide information and guidance to first level management personnel, and indicate as early as possible to higher management, possible danger areas. This permits remedial action to be instituted, at the appropriate level, prior to any damaging effects to the over-all goals of the program.

C. Ordnance Division

The Ordnance Division is composed of three separate facilities. These are located in Hopkins, Minnesota; Duarte, California; and Seattle, Washington. The operation in Hopkins is concerned with the development

of nuclear and non-nuclear armaments and armament systems, missile stabilization systems, and associated and related check-out and support equipments. The Duarte installation has been concerned with systems management programs, the development of complex electronic systems and trainers. Of particular note is the development, at this installation, of the ASROC system, and the Fleet Ballistic Missile Training facility. The Seattle Development Laboratory has been primarily concerned with sonar underwater systems, and communications investigations.

1. ASROC

Honeywell is the system prime contractor for the design and development of the ASROC anti-submarine warfare weapons system. This responsibility has included the direction of the efforts of many contributing groups, the scheduling, reporting, and documenting of all phases of the project. Finally, the design and fabrication of not only the missile itself, but also of a prototype launcher and the underwater fire control system was required. Honeywell's contribution to the ASROC Program has included, in addition to the major responsibilities noted above, the following tasks:

a. Production engineering, including tool design for fabrication and assembly.

b. System engineering, program planning, and scheduling.

c. Field and laboratory testing, including static and live firings.

d. Design, fabrication, installation and operation of telemetry equipment.

e. Test data reduction and analysis.

f. Integration of other anti-submarine warfare systems into the total ASROC fire control complex.

2. ASROC Shore-Based Trainer

The ASROC trainer currently in operation at the Norfolk Naval Base provides a synthesized operational system having the same external appearances and functional features as the actual equipment. Computational and equipment responses are performed with units designed to simulate, rather than duplicate, the operational system. The shore-based trainer is capable of simulating a tactical problem involving one ASROC ship, two support units, and two target submarines. An instructor's console provides complete control over the exercise, its duration, complexity, and conduct and permits evaluation of personnel performance.

3. Fleet Ballistic Missile Training

Honeywell is the prime contractor for the design and development of the Fleet Ballistic Missile Training Facility which is now being installed at New London, Connecticut. This facility contains three simulated attack centers representing the equipment and lay-out found aboard three different types of nuclear propelled submarines. Each attack center realistically simulates all equipment that is normally manned in that attack center of each of these submarines. Special interface devices allow the Honeywell H-800 computer to accurately produce the operational functions of each piece of equipment. Peripheral components in the attack centers include the fire-control system, and associated units such as radars, periscope,

communications system, underwater sound telephone, and various types of navigation and plotting equipment. In addition, sonar displays are incorporated in three program operator consoles, one of which is located in each attack center.

A separate and distinct facility provides geographic and data displays for monitoring and evaluating the operation of the three attack centers. A master instructor's station permits on-the-spot modification of a particular problem. This particular command center may be employed by higher echelon personnel, to not only direct portions of the strategic and tactical problem but also, to permit the evaluation of personnel in training.

4. Standard Communications Rating System

Under an Air Force contract, an investigation was undertaken to develop a Standard Communications Rating System useful for comparison and evaluation purposes. During phase one of this program examinations of basic rating factors common to all classes of communications equipment was undertaken. This examination included the data transmission rates, message priorities, information reliability, economics, as well as other characteristics considered by the contractor to be necessary to adequately rate communications systems. The fundamental communications parameters, such as bandwith, power, range, antennas, modulation, and detection serve as the basis for this formulation. Consideration was also given to such problems as interference and jamming characteristics in the development of this rating technique.

During Phase two a standard communications rating technique for comparison and evaluation of communications systems was formulated, and a standard rating chart generated.

5. Semi-Automatic Teletypewriter Message Distribution System (AN/FGC - 65).

This program required the design, development, prototyping, and delivery of a Semi-Automatic Teletypewriter Message Distribution System (AN/FGC--65) to facilitate the prompt delivery of a large volume of teletypewriter message traffic in a major Army headquarters. The system accepts messages from three teletypewriter trunk lines and delivers these messages within minutes to teletypewriter line printers in one or more of twenty-five designated offices upon command of an operator. The system eliminates the need for time-consuming manual processing which may cause delays in the delivery of messages. These delays could, in a manual system, amount to many hours. High priority messages receive special treatment from the system to insure immediate delivery.

D. MPG Research

The MPG Research Organization, which is located in Minneapolis, Minnesota, provides the necessary advanced technology required by the various divisions of the Military Products Group. Many advanced studies and investigations have been carried on by the scientists of this organization, however, only two programs which are directly related to the EDP area will be presented here.

1. Techniques for Evaluating Command and Control Systems.

In march, 1961, the MPG Research Department undertook a program for the

Air Force to develop techniques to permit the evaluation of command and control systems. The objectives of this program were:

a. To gain a more complete understanding of the problems involved in the analysis of this general class of systems.

b. To establish the feasibility of extending the known techniques developed for control system analysis to analyzing, optimizing, and simulating large complex systems.

c. To formulate approaches for the development of these techniques if such a development was deemed feasible.

During the life of this program the primary functions of command and control systems were defined in several different mathematical frameworks and were shown to be amenable to analysis through the use of network topology, linear graph theory, optimum flow theory, linear and dynamic programming techniques, combinatorial techniques and queuing theory.

2. Analysis and Optimization of Command and Control Systems

A program is currently underway to apply and utilize the results of the previously described effort. One of the objectives of the current program is to develop an abstract mathematical model of a specific Sage Direction Center. Techniques will then be established for analysis and synthesis of the information handling and decision making functions utilizing the models developed. It is anticipated that improved systems configurations will be the result of this study, as well as the generation of advanced techniques for the analysis of other command and control systems.

This particular study is not directed toward the elimination of the human operator nor the human decision maker. Its prime objective is to determine a more satisfactory merging of both the man and machine complexes to provide optimal operation in a large, complex and intricate command and control system.

SECTION III

COMPUTER DESIGN CAPABILITIES

A. SYSTEMS DESIGN PROGRAMS

1. Early Computer Experience

In 1955, Minneapolis-Honeywell (with 60 percent ownership) joined with the Raytheon Corporation to found the DATAmatic Corporation to develop and market large electronic data processing systems. The merger combined Honeywell's marketing and management experience with Raytheon's technical and engineering strength in the computer field. Practically all of DATAmatic's original 200 employees were engineers, scientists, and mathematicians, who had been brought together by Raytheon to work on various computer projects. These scientists, who were active in the development of such pioneer systems as Harvard's Mark I, II, and III computers, Hurricane, RAYDAC, ENIAC,

EDVAC, ORDVAC, and other early systems, are still with the EDP Division in responsible administrative and staff positions.

The Electronic Data Processing Division of Honeywell is firmly established in the data processing field. An ability to originate, design, build, and maintain complex solid-state digital data processing equipment has been clearly demonstrated by the successful installation of more than one hundred medium and large scale solid-state digital systems for both commercial and military applications. The EDP Division has gained a solid reputation with its customers for the highly reliable equipment supplied to specified schedule demands.

2. Honeywell Commercial Computer Developments

a. DATAmatic 1000

This large-scale vacuum tube system was designed and produced during the mid-1950's. Engineering breakthroughs introduced with this system included Orthotronic Control, the first automatic record correction rather than pure error detection system (to be discussed below). Three inch magnetic tape units resulted in transfer rates for exceeding the state-of-the-art at that time.

As a tribute to the Electronic Data Processing Division's ability to build and maintain large-scale digital systems, several D-1000 users, upon re-evaluating their current requirements, have stayed with Honeywell. Both the U.S. Treasury at Parkersburg and The First National Bank of Boston have installed H-800 solid-state systems because of the highly reliable performance of their earlier generation D-1000 equipments. All other D-1000 installations are still in use, and none have been replaced by competitors' later model systems.

b. Honeywell 800

The Honeywell 800 was the first of the large-scale solid-state data processing systems introduced by Honeywell. It featured such design innovations as parallel processing and other economy techniques. This system is now in its third year of production, and it has been produced and tested in the EDP plants at a substantial volume rate.

Introduced with the Honeywell 800 was the concept of employing a computer to design a computer. The D-1000 was used to supply not only design layout and construction information, but also to generate directly usable processing data and test routines. Breakthroughs in test technology have been made which allow a large volume of complex equipment to be reliably tested in a comparatively short period of time.

Among the approximately fifty installations of H-800 systems for commercial and military use, there are at least two where this system has been accepted and integrated into a real-time complex approved to Military Standards and Specifications. These are the Submarine FBM Training Facilities at New London, Connecticut and Charleston, South Carolina for the Navy. Other Government applications of H-800 equipment include NAS, Alameda, California; Army Map Services; the Army Finance Center; The National Institutes of Health; The National Library of Medicine (MEDLARS system); and, The National Weather Records Center.

c. Honeywell 400

The Honeywell 400 was announced in 1960 and first shipments

were made in late 1961. This system is a powerful solid-state digital system basically for business applications. It incorporates significant new advances in the application of magnetic core logic which have been implemented very successfully in manufacturing.

3. Magnetic Tape Subsystem

The USAEPG Magnetic Tape Subsystem has been designed and developed by the Electronic Data Processing Division for the U.S. Army Electronic Proving Ground, Fort Huachuca, Arizona, under Contract No. DA-36-039-SC-80854 and Technical Requirement AD-16-62.

The Magnetic Tape Subsystem is a solid-state electronic data storage subsystem of one or two identical vans each containing four Honeywell Model 404-1 Magnetic Tape Units, a Magnetic Tape Control Unit (Input/Output Buffer Unit), and a DC Power Supply Unit. This data storage subsystem (see Figure IV-1) is completely contained within a government furnished, lightweight, insulated type V-51.

The Magnetic Tape Subsystem is designed to operate with the FIELDDATA family of automatic data processing equipment as a data storage, input/output system. The four Magnetic Tape Units of the subsystem may be controlled by a BASICPAC, INFORMER, or MOBIDIC central processor through the subsystem Magnetic Tape Control Unit. Information is recorded on, or read from 3/4-inch wide magnetic tapes moving at 120 inches per second. The magnetic tapes are read, written, and rewound under central processor control. The subsystem provides the FIELDATA family of computers with a flexible, high-speed, input/output and data storage capability adaptable to the wide range of data storage problems associated with tactical and combat support activities such as Fire Control, Intelligence, Logistics, and Personnel Accounting.

4. Digital Communications - Store and Forward Studies

Several studies concerning the systems organization for digital communications (generally referred to as "Store and Forward" systems) have been carried on by Honeywell recently. These studies have been of an interdivisional nature, drawing team members from corporate divisions who could substantially contribute. The objectives of these studies have been to establish optimum systems configurations for the users various requirements.

Specific study groups have investigated the application of:

a. conventional data processing systems;

b. semi-modular systems; and

c. distributed modular systems.

The results of these investigations are of a very highly proprietary nature, however, one study report has been abstracted for inclusion in this proposal. The author examined a semi-modular organization and reached the conclusion that from an economic standpoint, such functions as storage and memory should be centrally located.

5. PICO

The St. Petersburg facility of the Aeronautical Division has developed a unique spaceborne computer which has been designated - PICO. The PICO computer program was entirely funded with corporate funds.

Designed for advanced aero-space applications, the computer is completely housed in less than 0.4 of a cubic foot, weighs approximately 20 pounds, and requires 46 watts of power. Since PICO was intended for extremely broad environments, it is not suprising to find that:

a. it will operate over a temperature range from -55°C to +100°C;

b. it will withstand 20g's of vibration and 50g's of shock.

The computer employs a 24 bit word and can perform an addition in 12 μs. Although the basic system does not contain a built-in multiply or divide, these are available as an option.

6. Development of a Standard Communications Rating System

The Seattle Development Laboratory of Honeywell's Ordnance Division has recently completed an Air Force contract to develop a standard communications rating system. The objective of this contract (AF 30(602)-2390) was to determine the feasibility of a single, or a small number of rating systems useful for comparison and evaluation of communication systems. The rating system applied to all classes of communication systems and considered user requirements and restrictions. Initially only transmission and reception facilities for providing point-to-point service were included. Extension to cover relaying and terminal equipment was later considered. All types of communication requirements except for hardened command and control and personal rescue communications, were included as factors influencing the rating of communication systems.

Mathematical criteria for optimization of the configuration of any propagation mode were established. The tropospheric scatter mode of propagation was examined in detail to ascertain that it did meet the requirements for optimization. Examination was also made of other modes. Communications equipment cost and physical data were processed to determine the form of the empirical equations necessary for the optimum configuration.

Feasibility of constructing a nomograph for specifying the optimum system parameters of any propagation mode meeting the requirements of the mathematical model was established. A model of the nomograph for use with tropospheric scatter was demonstrated.

7. Systems Studies

The inertial guidance, control, and computational requirements for numerous ballistic, orbital, and lunar missions have been studied by Honeywell as a prime or team-member participant. Some of these programs are: SR-183 (Lunar Guidance and Control), SR-199 (Orbital Bomber), SAINT (Satellite

Rendezvous), Centaur (Lunar Guidance and Control), Apollo (Manned Lunar Mission), Dynasoar (Orbital Glide Bomber).

In addition to the system and computer development studies performed at the Honeywell Florida Guidance Center, emphasis has been placed on the theoretical aspects of these programs in the MPG Research Department. Particular emphasis has been placed on trajectory optimization studies, optimal fire-control systems, error analysis techniques, guidance system sensitivity studies, measurement errors, control system optimization, and concepts.

8. Studies of the Theory of Optimal Automatic Control

Honeywell has been conducting for more than one and one-half years an internally-funded, comprehensive research study of advanced control techniques for high-order systems with multiple control inputs. This study has concentrated on the development of techniques that are applicable to a wide class of control problems and that minimize the trial and error factor so common to most classical methods of analysis and synthesis. Results to date include significant development in the area of optimal state vector control (time optimal, minimum energy, etc.), system parameter determination (an integral part of adaptive control), and control of nonlinear systems.

B. COMPUTER CIRCUITRY DEVELOPMENT INVESTIGATIONS

1. Introduction

The following few paragraphs can only introduce the extensive knowledge and experience which exists within Honeywell in general, and the EDP Division, in particular, in the area of advanced computer circuitry development. These particular projects have been selected for the purpose of demonstrating the extent and depth available internally at Honeywell in this area.

2. Micro-electronics

This broad-based investigation has been carried on by several separate

organizations within the corporate structure, at several levels. These levels refer to time-near, medium- and long-range utilization in the product line. A most significant study is a joint effort which has been undertaken by the Corporate Research Center (located in Hopkins, Minnesota) and the Semiconductor Division, which is located in Riviera Beach, Florida. The reason for this dual approach has been to integrate both the production techniques, which will be required, and the training of production personnel with the advanced techniques which exist and will be developed and required. Preproduction lots have been, and are currently being, tested by many of the divisions of the company throughout the country.

3. Magnetic Thin Film Studies

Investigations of thin magnetic film techniques by use in both future, high-speed, fast memories, and logical arrays have been carried on for several years. Most of the basic materials studies have been performed by staff personnel of the Hopkins Research Center. The products of these studies have been employed by EDP and the Military Products Group in the design and testing of various systems for analysis.

The techniques which have been employed are not restricted to the production of flat film memory structures - but have tended toward more advanced concepts.

4. Glass Delay Line Storage

A recent development within EDP has been concerned with the application of glass delay lines for storage. This effort has generated a lively interest since the lines present certain unique properties which can not be matched by any other technique. For example, storage of information, although serial in a single line, is at a bit rate substantially above a megacycle. Data packing of several thousand bits per line has been shown to be feasible.

The delay line concept has been examined for several applications, among these is the concept of a medium-size, serial-parallel associative search memory. The most attractive aspect of this concept, and by far not the only one, is the extremely low cost involved in the actual construction of a complete system.

5. PICO Circuitry and Packaging

The PICO computer has already been mentioned earlier in this chapter. It is referred to again here since the circuitry and packaging are also a unique feature which should be examined and considered on their own.

The over-all system was designed for a mean time between failure of 10,000 hours. Thus far, the arithmetic unit has been in life test for approximately 6,000 hours, with no failures occurring during that period of time. During that period, the processor has been cycled from minus 55 degrees to plus 100 degrees centigrade once a week. The memory has been tested for somewhat less time, but with the same results - no failure.

An advanced system is being designated which employs a method of circuit redundancy which should provide twice the MTBF postulated for the original PICO.

6. Tunnel Diode Arithmetic Unit

Honeywell engineers, always intent upon exploiting the most advanced techniques - if they can provide the high level of reliability and performance required - began a study of tunnel diodes and their application sometime ago. The results of these efforts has been the design of a tunnel diode arithmetic unit for the Honeywell 1800 system.

7. All-Fluid Logic Building Blocks

All-fluid logic building blocks with no moving parts are presently being developed at Honeywell. Various configurations of the fluid amplifier invented

at the Diamond Ordnance Fuse Laboratory are being investigated to provide the switching and memory functions necessary in digital logic.

Some of the advantages expected for such devices are small size, minimum weight, low cost, and very high reliability. It is also expected that such devices will operate reliably in extreme environments, and will tolerate high radiation levels.

While it is unlikely that such building blocks will be sufficiently fast to serve in the high-speed portions of high-speed systems, they hold potential and will be investigated for application in peripheral equipments.

Attainable goals for an all-fluid logic system in terms of speed and packing densities are:

a. Speed — 10 to 100 usec switching time

b. Packing Density — 400 elements (logic generating devices or flip/flops) per cubic inch.

8. Self-Healing Circuitry

The final program to be discussed in this section concerns an investigation currently supported by the U.S. Air Force. This program, Developments of Self-Repair Procedures, is concerned with the examination of methods of causing electronic components to repair themselves by the natural behavior of the atoms within the components. The most promising approach appears to be through the utilization of crystalline anisotrophy, in which a given metal or alloy in crystalline form tends to add atoms in a preferred direction — namely in the direction of a given break in a given component. This effort is being performed at a low level but could produce major results in a two to five year time frame.

MISCELLANEOUS