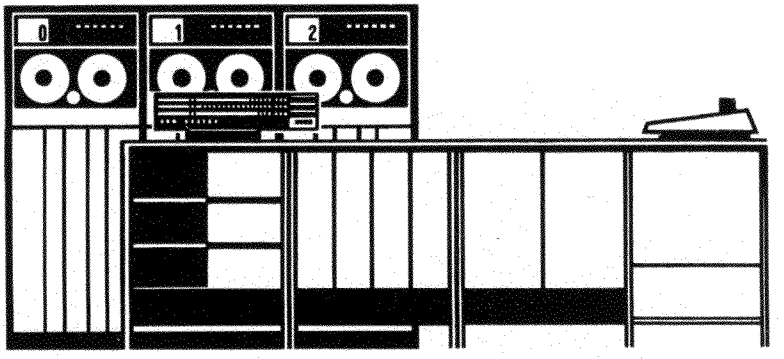
HONEYWELL 200

SUMMARY DESCRIPTION



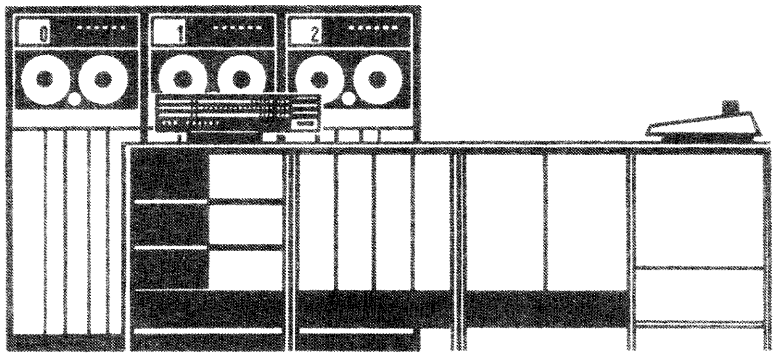
Honeywell

ELECTRONIC DATA PROCESSING

Litho in U.S.A.
NP 2655

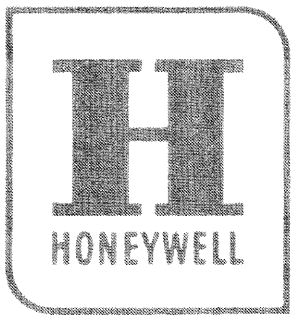
DSI—218
51163
51263
5164
75C264

HONEYWELL 200
SUMMARY DESCRIPTION



Honeywell
ELECTRONIC DATA PROCESSING

Questions and comments regarding this manual should be addressed to:
Honeywell Electronic Data Processing
Merchandising Division
60 Walnut Street
Wellesley Hills 81, Massachusetts



FIRST EDITION
First Printing, November, 1963
Second Printing, December, 1963
Third Printing, January, 1964
Fourth Printing, February, 1964

Table of Contents

List of Illustrations

FOREWORD

This summary description of the Honeywell 200 Electronic Data Processing System is intended for those having a general familiarity with data processing. Machine characteristics and programming aids are described in terms which should aid comparisons between the Honeywell 200 and competitive equipment. The equipment characteristics reported herein remain subject to minor revision in order that design improvements may be incorporated.

1 Introduction

A basic data processing objective in business and industry is to maximize record-keeping efficiency while at the same time minimizing the cost of record-keeping operations. Electronic data processing systems have enabled many businesses to achieve a *partial* realization of this objective; that is, they have been able to maximize record-keeping efficiency but at a sacrifice in cost. There are many other businesses, however, which have not been able to take advantage of high-speed electronic data processing because for them the cost-performance ratio associated with presently available systems does not satisfy their requirements.

The Honeywell 200 Electronic Data Processing System was developed to enable present as well as prospective users of electronic data processing systems to achieve maximum efficiency at minimum cost. The significance of this new system is characterized by four key words:

Throughput
Modularity
Simplicity
Compatibility

THROUGHPUT

In most business data processing applications, a high level of throughput is more significant than high computational speeds. Therefore many presently available data processing systems compromise internal speed in favor of increased input/output flexibility.

The H-200, on the other hand, provides for both high internal computation speeds and simultaneity of input/output operations. Specifically, the combination of a high-speed magnetic core main memory (two-microsecond memory cycle), a control memory with an access time of 250 nanoseconds, and an outstanding input/output traffic control feature, enables four simultaneous input/output operations to be performed concurrently with computing. For example, in one minute this throughput potential can be used to

read or write 4,360 records of 500 characters each,
punch 250 cards,
read 800 cards,
print 900 lines of 120 characters each, and execute approximately 1,000,000 instructions – all at the same time.

Perhaps even more significant than this combination of input/output simultaneity and high operating speed is the fact that this capacity is built into every H-200; it does not depend on complex software or expanded system configurations.

MODULARITY

A basic requirement for the potential user of an electronic data processing system is the ability to start with a small system, which may be easily expanded to meet growing data processing requirements. The H-200 provides for system expansion through the availability of such peripheral devices as paper tape units, random-access disc and drum memories, additional magnetic tape units, additional high-speed printers, and a comprehensive array of data communication control units. These devices can be joined in various combinations to form a balanced system which satisfies the input/output requirements of virtually any application.

System expansion is complemented by a high degree of internal modularity including such features as a group of advanced programming instructions, a modularly expandable main memory, indexed and indirect addressing features, and an automatic program interrupt feature.

SIMPLICITY

The speed, simultaneity, and modularity of the H-200 are enhanced by programming and operating simplicity. Since the system's high level of throughput

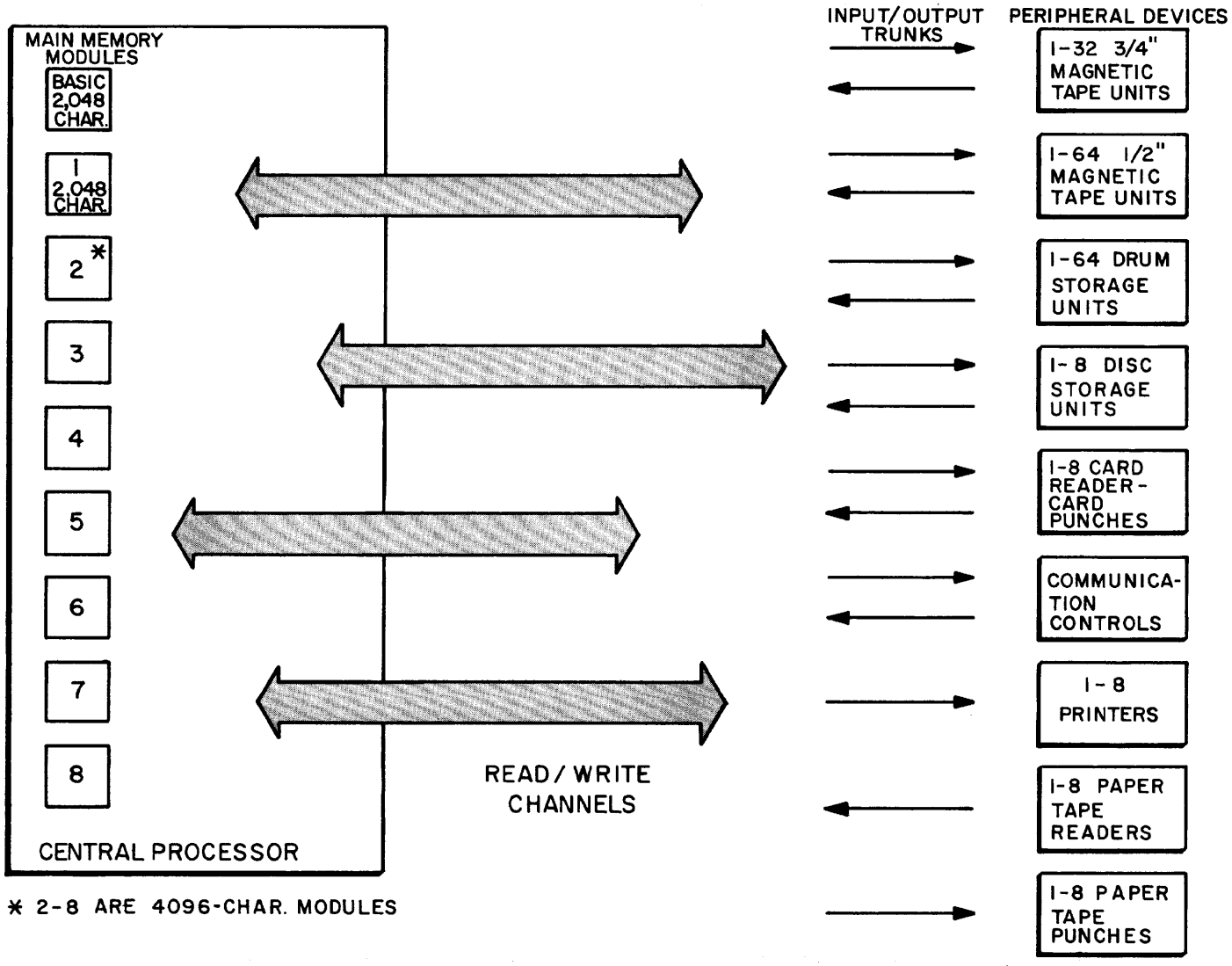
is achieved automatically rather than by elaborate program control, the power of the system's comprehensive array of software can be applied directly to the job being processed.

The Easycoder programming and operating system assists H-200 users in all phases of their data processing activities. It significantly reduces both the time required to prepare H-200 programs and the machine time required to execute them.

COMPATIBILITY

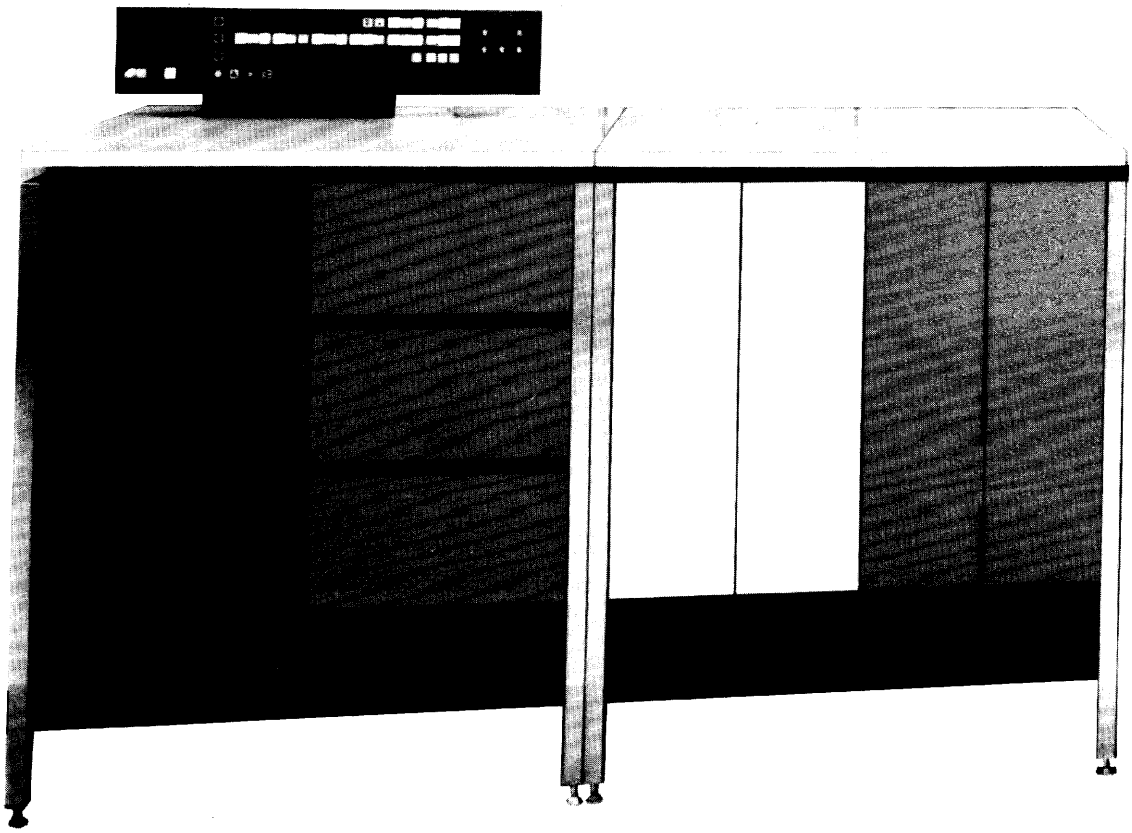
The H-200 is the only electronic data processing system currently available that is program-compatible with several widely used competitive systems. It can accept programs written for these systems and, by means of automatic conversion software, produce programs which can be executed on the H-200. This means that users of these systems can take advantage of the H-200's superior data processing capabilities without incurring the prohibitive cost of reprogramming.

In the sections that follow, the H-200 is described in terms of its central processor, peripheral devices, and associated programming aids. The information presented in these pages clearly demonstrates the power and capabilities of this new data processing system.



Honeywell 200 System Components

2 Central Processor



The Model 201 Central Processor contains a magnetic core memory, control unit, arithmetic unit, and special-purpose control elements which, under the direction of an internally stored program, perform the arithmetic, logical, and input/output functions of the Honeywell 200 Data Processing System.

A major feature of the central processor's structural design is the use of integrated system modules. Each module contains all of the circuitry required for a particular system function; for example, one module contains all of the printer control circuitry, another contains the components of the arithmetic unit, and so on.

This feature greatly enhances the expandability of the system; in most cases, expansion involves little more than plugging in additional modules. The reliability of components within each module has been maximized through the use of silicon semiconductors.

Logically, the central processor is divided into five basic units as shown in Figure 1. The central processor characteristics are summarized in terms of these five units in Figure 2.

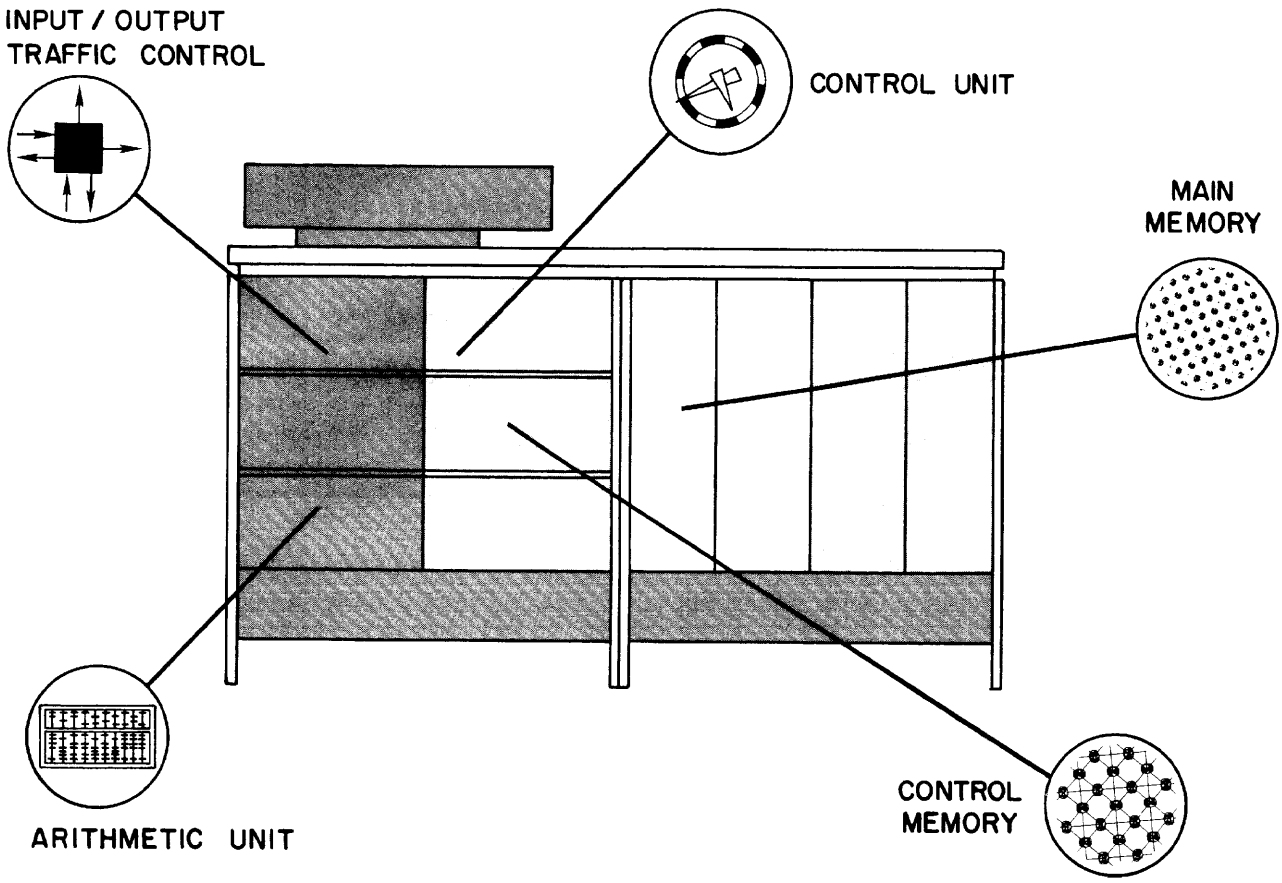
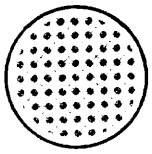


Figure 1. Logical Division of the Central Processor.



Main Memory

The main memory is a magnetic core, random access device which provides storage for instructions and data during the performance of programs. The memory unit supplied as part of the basic central processor has a capacity of 2,048 characters, each of which is stored in a separate, addressable, memory location. This capacity may be expanded in modular increments by adding one 2,048-character module and additional 4,096-character modules. A memory cycle, that is, the time required to transfer a character from a memory location and restore the cores to their original state, is two microseconds.

An indexed addressing option provides the facility of using six groups of three memory locations each as index registers. When these locations are not being used for indexing operations, they can function as normal storage locations.

There are no reserved input/output areas in the main memory. The programmer has complete freedom in specifying the locations and sizes of such areas to meet the needs of any program. This allows both a high degree of programming flexibility and an economical usage of memory.

REPRESENTATION OF INFORMATION IN THE MAIN MEMORY

VARIABLE-LENGTH FIELDS

Information is stored in variable-length memory areas called *fields*. A field is defined as a group of consecutive memory locations whose contents are treated as a unit. Each location within a field stores either six binary digits or one alphanumeric character. Since fields can be of any length (from one memory location up to virtually the maximum number of locations), information units of varying lengths can be stored without wasting memory capacity.

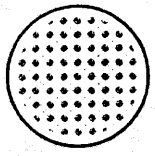
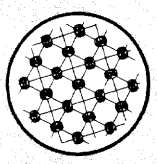
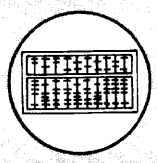
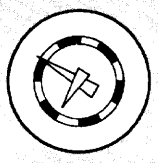
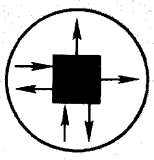
 **MAIN MEMORY**	BASIC MEMORY — 2,048 characters. ADDITIONAL MEMORY — One 2,048-character module and additional 4,096-character modules. PROCESSING UNIT — Six-bit character. Variable-length groups of consecutive characters form instruction and data fields. INSTRUCTION FORMAT — Variable. Typical configuration: operation code, two addresses, and variant character. ADDRESSING MODES — Direct, indirect, indexed. INDEX REGISTERS — Six, each capable of storing three six-bit characters. MEMORY CYCLE — Two microseconds to read and restore one character.
 **CONTROL MEMORY**	MEMORY CAPACITY — 16 control registers, each capable of storing the address of a character position in the main memory. CONTROL REGISTERS — Basic configuration: two operand-address registers, two instruction address registers, and up to eight read/write channel counters. ACCESS TIME — 0.25 microseconds. MEMORY CYCLE — 0.5 microseconds.
 **ARITHMETIC UNIT**	OPERATIONS — Decimal arithmetic, binary arithmetic, logical operations. TYPICAL OPERATING SPEEDS — 5-digit decimal add (A + B ⟶ B) 44 microseconds. 5-digit compare (A:B) 34 microseconds.
 **CONTROL UNIT**	PARITY CHECKING — One parity bit with each character stored in memory. PROGRAM CONTROL — Sequential selection, interpretation and execution of all stored program instructions. CONTROL PANEL — Control and display functions.
 **INPUT/OUTPUT TRAFFIC CONTROL**	READ/WRITE CHANNELS — Three channels standard; auxiliary channel optional. I/O instructions designate channel connections. INPUT/OUTPUT TRUNKS — Basic configuration of four input and four output trunks; expandable to eight input and eight output trunks. PERIPHERAL SIMULTANEITY — Up to four peripheral transfer operations simultaneous with computing.

Figure 2. Summary of Central Processor Characteristics.

ITEMS

Consecutive fields can be combined to form a larger unit of information called an *item.* Grouping fields to form an item simplifies the manipulation of related data fields, and minimizes the number of instruction executions required to move consecutive fields within the main memory.

RECORDS

Any unit of information that is to be transferred between the main memory and a peripheral device is defined as a *record.* Records, like fields, can be of any length.

PUNCTUATION BITS

The use of a variable-length data format requires that there be a method of indicating the actual length of a unit of information. This requirement is fulfilled by two *punctuation* bits associated with each memory location. These bits can constitute a *word mark* – used to define the length of a field; an *item mark* – used to define the length of an item; or a *record mark* – used to define the length of a record. Figure 3 provides a summary of data storage unit characteristics.

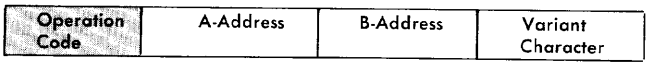
FORMAT	DEFINITION	LENGTH DEFINED BY
FIELD	A group of consecutive characters which are treated as a unit.	Word Mark
ITEM	One or more consecutive fields.	Item Mark
RECORD	A unit of information to be transferred between a peripheral device and the main memory.	Record Mark

Figure 3. Data Storage Characteristics.

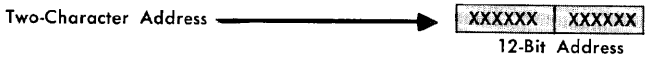
In addition to defining the lengths of data fields, word marks are also used to define the lengths of instructions in memory.

REPRESENTATION OF INSTRUCTIONS IN THE MAIN MEMORY

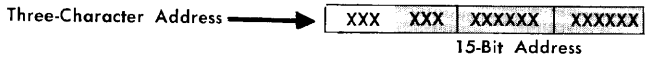
Instructions, like data units, are variable in length. A typical H-200 instruction appears in memory as:



The operation code is stored as a single six-bit character. Each address portion is stored as either two or three characters, depending upon the mode of instruction assembly selected by the programmer. In either case, the address portions are interpreted as pure binary numbers. A two-character address is interpreted as a continuous 12-bit binary number (addresses in the range $0\text{-}4095_{10}$).



A three-character address is interpreted as 18 binary digits in which the rightmost (low-order) 15 bits represent the address of a location (addresses in the range $0\text{-}32768_{10}$) and the remaining three bits specify whether the address is direct, indirect, or indexed.



In an indexed address, the three high-order bits specify one of six main memory index registers whose contents will be used in the indexing operation. A significant feature of indirect addressing is the fact that an indirect address can specify another indirect address, and so on through any number of repetitions.

The use of binary addressing simplifies the modular expansion of memory because it eliminates the necessity of complex machine-language coding schemes in order to represent memory addresses in expanded configurations.

The variant character is appended to certain instructions to modify the operation code. When it is used, it always appears as the last character in the instruction. Figure 4 illustrates the six basic formats in which instructions are stored in the main memory. Two additional and slightly different instruction formats, used to initiate input/output operations, are described in the summary of H-200 instructions.

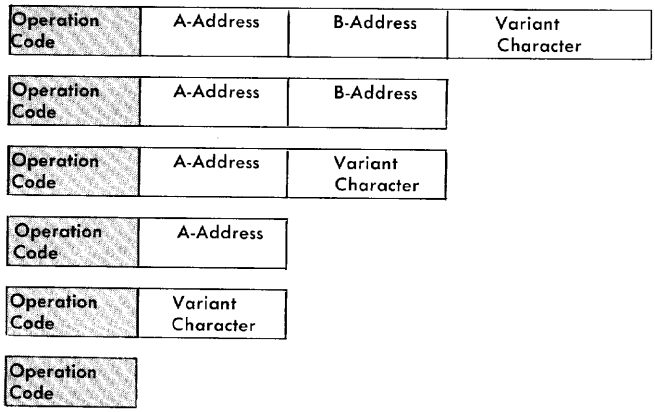
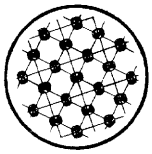


Figure 4. Instruction Formats.



Control Memory

The control memory is a magnetic core storage unit consisting of 16 control registers. These registers are functionally divided into three categories: instruction address registers, operand address registers, and read/write counters. Each register can store the address of a main memory location. During a program run, the control registers are used to store the addresses that describe the retrieval and execution of all instructions. Access to a control memory register requires 250 nanoseconds.

INSTRUCTION ADDRESS REGISTERS

The basic central processor is equipped with one instruction address register. Its function is to direct the sequential retrieval of instruction characters from their storage locations in the main memory.

In systems equipped with the second instruction address register, a single instruction can interchange the contents of these two registers. This function provides a convenient means, using only two instructions, of branching to and returning from any position in a program.

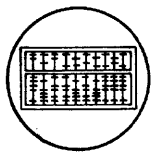
Program interrupt, an H-200 feature used in conjunction with communication controls, includes the addition of another control register, the interrupt register. In a system equipped with this register, the controlling program is interrupted automatically upon receipt of a signal from a communication control, and the contents of the interrupt register and the sequencing instruction address register are interchanged. Thus, in one operation, the machine can enter a routine which services the communication control, storing the current location of the controlling program for a subsequent return to normal program sequence.

OPERAND ADDRESS REGISTERS

Two operand address registers direct the execution of all instructions during a program run. Normally, these registers sequentially address the memory locations containing data referred to by the address portions of an H-200 instruction.

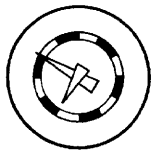
READ/WRITE COUNTERS

Data is transferred between the main memory and peripheral devices via read/write channels. Associated with each channel are two read/write counters. These counters store the starting and current addresses of the data being transferred by the read/write channel.



Arithmetic Unit

The arithmetic unit performs arithmetic and logical operations. Result conditions such as overflow and zero balance cause the setting of indicators whose status may be used to initiate a change in program sequence. The arithmetic unit performs both decimal and binary arithmetic. It can perform multiplication and division operations directly if equipped with the multiply/divide option. In systems not equipped with this option, multiply and divide operations are performed by subroutines supplied by Honeywell.



Control Unit

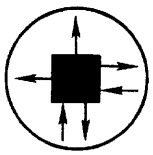
The major functions of the control unit are the selection, interpretation, and execution of the instructions in a program run. In effect, the control unit interprets the programmer's instructions and initiates the machine subcommands necessary to execute each instruction. The information used by the control unit in performing these tasks is stored in the registers of the control memory. (For example, the first instruction address register provides the control unit with the information necessary for the sequential selection of instructions.)

CONTROL PANEL

Associated with the control unit is a control panel (Figure 5) which provides a visual indication of the status of the entire system and permits manual intervention in the operation of the system. By using the various control switches, the operator can start and stop the machine and can load and interrogate both main and control memory locations. The panel is also equipped with a minimum of four "sense" switches

Figure 5. Control Panel.

which may be used in conjunction with programmed instructions to stop processing or to select predetermined program paths. The use of these switches increases the flexibility of a program by allowing it to be used in several different applications.



Input/Output Traffic Control

An outstanding feature of the Honeywell 200 is the input/output traffic control, which directs the time sharing of the main memory by the various peripheral devices and the central processor. This control unit makes possible the simultaneous performance of central processor computing and up to four input/output operations. For example, the H-200 can simultaneously print reports, read cards, read or write tape, and compute.

READ/WRITE CHANNELS

The degree of peripheral simultaneity in any H-200 system depends on the number of read/write channels in the system. A read/write channel is a bi-directional data path across an interface between the main memory and a peripheral device. Whenever an input/output operation is to be performed, a programmer-assigned read/write channel completes the path between the required peripheral device and the main memory.

The basic H-200 is equipped with three read/write channels, enabling the basic system to compute while simultaneously performing three input/output operations which utilize any peripheral devices in the system. Peripheral simultaneity can be expanded by adding one auxiliary read/write channel, thus permitting the simultaneous performance of computing and four input/output operations. For example, in round figures, the H-200 can read or write 4400 tape records of 500 characters each, punch 250 cards, read 800 cards, print 900 lines of 120 characters each, and perform one million average computing operations – all in one minute.

INPUT/OUTPUT TRUNKS

The basic system is equipped with four input and four output trunks, each of which is connected to a peripheral control. Data is transferred between main memory and a trunk (and thus a peripheral device) via the read/write channel specified in the instruction which initiates the transfer.

Additional peripheral devices can be connected to the system simply by adding input/output trunks. As many as four more input and four more output trunks can be added to the basic configuration. Figure 6 illustrates the connections between the central processor and the peripheral devices.

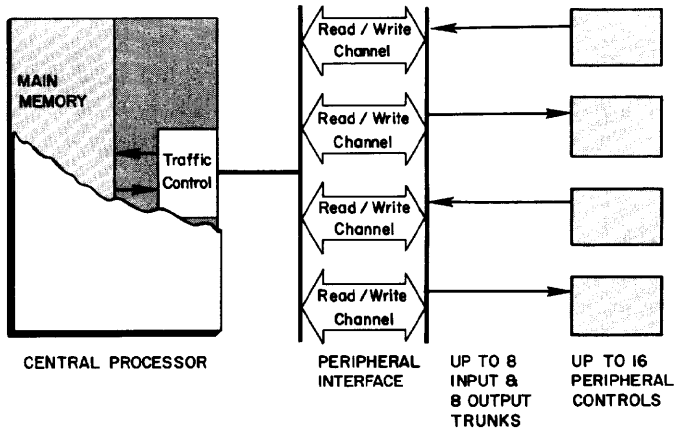


Figure 6. Data Paths Between Central Processor and Peripheral Devices.

3 Peripheral Devices

A significant feature of the Honeywell 200 is the availability of a wide variety of peripheral devices which allow it to utilize numerous input/output media. These devices can be joined in various combinations to form a balanced system which satisfies the input/output requirements of virtually any application. The following units can be connected in any combination of up to eight input and eight output devices:

- Magnetic tape control, to which as many as eight ½ inch tape units can be connected.
- Magnetic tape control, to which as many as four ¾ inch tape units can be connected.
- Card reader-card punch and separate controls for reading and punching.
- High-speed printer and control.
- Random access disc file and control (disc capacities of up to 100 million characters are available).
- Random access drum control, to which as many as eight 2.6-million-character drums can be connected.
- Paper tape reader and control.
- Paper tape punch and control.
- Single-channel and multi-channel communication controls.

Peripheral Controls

Peripheral controls regulate the transfer of data between the central processor and peripheral devices. Specifically, they reconcile the mechanical speeds of the peripheral devices with the electronic speed of the central processor and minimize the interruption of central processor activity due to peripheral data transfers.

All peripheral devices, with the exception of magnetic tape units and random access drum storage units, are connected to their own individual control units. As indicated above, up to eight tape units can be connected to a single tape control, and up to eight magnetic drums can be attached to a single drum control. Any configuration of peripheral controls can be connected to the central processor according to the following rules:

1. Each input trunk can be connected to a separate input control.
2. Each output trunk can be connected to a separate output control.
3. Both an input and an output trunk must be connected to an input/output control (such as a magnetic tape control).

A significant operating feature is the fact that peripheral controls operate independently of the central processor and require memory access only when information transfers are performed. In particular, all data validity checks, such as hole count checks in card reading and punching, are performed by the respective control units and do not involve the central processor in any way.

Card Equipment

Card reading and punching operations are performed by the Model 227 Card Reader-Card Punch. Reading and punching operations are controlled, respectively, by the 207 Card Reader Control and the 208 Card Punch Control. The characteristics of the 227 and its associated controls are presented in Figure 7.

OPERATIONS

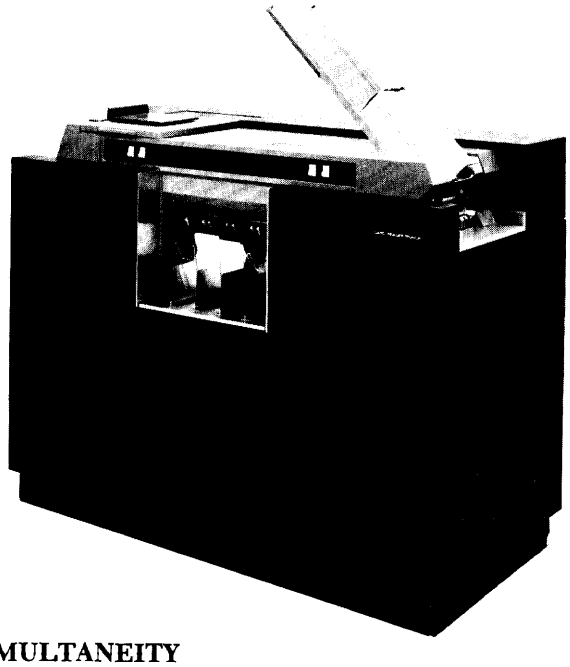
The card reader-punch is capable of reading 800 cards per minute and punching 250 cards per minute.

The device can perform the following operations under program control:

1. Read a card.
2. Punch a card.
3. Reject the last card read or punched into an alternate pocket.

CHARACTER CODES

Both the reader and punch controls translate between Hollerith card codes and H-200 six-bit central processor codes. Either control or both can be equipped to transfer data in the transcription mode. In this mode of operation, each binary 1 corresponds to a punched position and each 0 to an unpunched position on the card being read or punched.



SIMULTANEITY

A single 227 can simultaneously read and punch cards at top speed. This capability is enhanced by the fact that only a relatively small amount of central processor time is required for either reading or punching.

As illustrated in Figure 8, a card can be read during a 75-millisecond interval and punched during a 240-millisecond interval. Central processor activity is suspended in order to transfer data for approximately four milliseconds of a reader cycle and two milliseconds of a punch cycle. Thus, the central processor is free to perform computations and to direct other input/output operations during approximately 95% of a card read cycle and 99% of a card punch cycle.

CHARACTERISTICS	READER SECTION AND 207 CONTROL	PUNCH SECTION AND 208 CONTROL
CARDS	Standard 80-column, 12-row cards.	
PROGRAMMED OPERATIONS	1. Read data from a card and transfer it to specified memory area until entire card is read or input area filled. 2. Reject cards.	1. Punch data from specified memory area into a card. 2. Reject cards.
DATA TRANSFER MODE	Automatic translation between Hollerith card code and 6-bit central processor code is standard; separate transcription-mode reading and punching options available.	
SPEED	800 cards per minute.	250 cards per minute.
TIME BETWEEN CARD CLUTCH POINTS	25 milliseconds.	60 milliseconds.
SIMULTANEITY	Simultaneous read-punch-compute. Reading and punching operations engage the central processor for somewhat less than 4 ms per card and 2 ms per card, respectively. The central processor is free for approx. 95% and 99% of processing intervals shared, respectively, with card read and card punch operations.	
INPUT / OUTPUT AREAS	Any main memory area.	
DATA PROTECTION	Hole-count check at two reading stations; optional illegal card code check.	Hole-count check.
FEED HOPPER/ STACKER CAPACITY	3000 / 1000 cards.	1200 / 1000 cards.
NUMBER OF STACKERS	Two: 1 accept, 1 reject.	Two: 1 accept, 1 reject.
TRUNK REQUIREMENTS	One input trunk.	One output trunk.
MAXIMUM NUMBER PER SYSTEM	Eight.	Eight.

Figure 7. Card Reader-Card Punch Characteristics.

DATA PROTECTION

Card reading is checked by comparing the hole counts obtained at each of two reading stations. Each card can also be checked for the presence of illegal punches. Card punching is similarly checked for hole-count errors. Failure of any of these checks automatically sets an indicator which can be tested by a programmed instruction.

PROCESSING INTERVAL SHARED WITH CARD OPERATION

	Card operation begins	Data transfer begins	Data transfer ends	Card operation ends
Reading	0 ms	21 ms	65 ms	75 ms
Punching	0 ms	42 ms	218 ms	240 ms

CENTRAL PROCESSOR TIME REQUIRED:
Reading — 4 ms (5% of processing interval)
Punching — 2 ms (1% of processing interval)

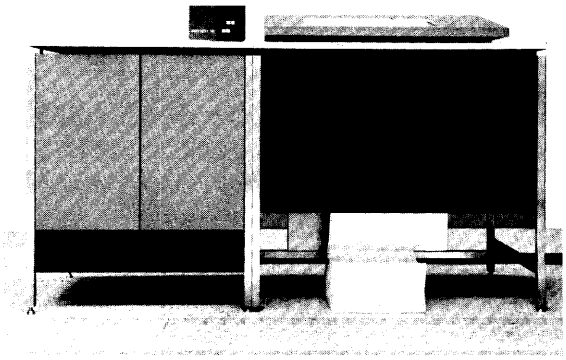
Figure 8. Simultaneity in Card Operations.

Printers

The Model 206 High-Speed Printer and Control prints at the rate of 900 lines per minute. Any one of 56 characters – 10 numeric, 26 alphabetic, and 20 special symbols – can be printed in each of the 120 print positions (132 print positions optionally available). Depending upon the paper stock being used, up to 10 carbon copies can be produced. The characteristics of the printer are summarized in Figure 9.

EDITING

Powerful machine instructions enable the programmer to insert spaces and punctuation in alphabetic and numeric data, to suppress leading zeros in numeric data, and to insert identifying symbols such as dollar signs, asterisks, etc. Data can be printed from any area of the main memory.



OPERATIONS

The high-speed printer can perform the following operations under program control:

1. Print one line and space from 1 to 15 lines (or to the head of the next form).
2. Advance paper without printing.

SIMULTANEITY

As illustrated in Figure 10, a single print operation, *i.e.*, printing a 120-character line and advancing paper one line, can be performed during a 67-millisecond processing interval. Central processor activity is suspended in order to transfer data for approximately 13 milliseconds of this time. Thus, the central processor is free to perform computations and to direct other input/output operations during about 81% of a print cycle.

CHARACTERISTICS	206 PRINTER AND CONTROL
PAPER STOCK	Width: 3½ in. to 22 in., edge-to-edge. Weight: 15-lb., single-part forms to 125-lb., continuous card stock.
PROGRAMMED OPERATIONS	Print and space; space only.
PRINTABLE CHARACTERS	10 numeric, 26 alphabetic, and 20 special symbols.
DATA FORMAT	120 or 132 print positions per line, 10 char./in. Standard vertical spacing is 6 lines/in.; option allows selection of 6 or 8 lines/in.
SPEED	Nine hundred 120-character lines per minute for continuous single-line printing (0.067 sec./line or 108,000 char./min.) Line skip speed is 21 in./sec.
SIMULTANEITY	Printing engages central processor for 13 ms per line. The central processor is free for approximately 81% of a processing interval shared with a printing operation.
OUTPUT AREA	Any main memory area.
REPRODUCTION SYSTEM	Hammer stroke against engraved drum.
DATA PROTECTION	Printer stopped and indicator set if printer not responding correctly to control unit printing signals.
TRUNK REQ'S.	One output trunk.
MAX. NO. PER SYSTEM	Eight.

Figure 9. High-Speed Printer Characteristics.

DATA PROTECTION

Operation of the printer is checked automatically to ensure that it responds correctly to control unit printing signals. If this check fails, an indicator is automatically set which can be tested by programmed instructions.

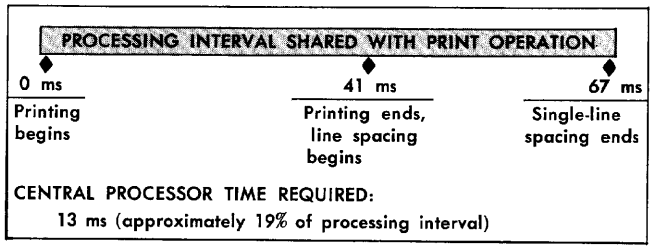


Figure 10. Simultaneity in Print Operations.

Paper Tape Equipment

The Model 209 Paper Tape Reader and Control reads 5-, 6-, 7-, or 8-level tape at the rate of 500 frames per second. The Model 210 Paper Tape Punch and Control punches the same kinds of tape at the rate of 110 frames per second. The characteristics of H-200 paper tape equipment are summarized in Figure 11.

OPERATIONS

The H-200 paper tape equipment can perform the following operations under program control:

1. Read paper tape until a record mark is sensed in memory.
2. Punch paper tape until a record mark is sensed in memory.
3. Rewind or runout tape on the reader.

Tape can also be rewound or runout manually on the reader and runout manually on the punch. Tape stops within the length of a frame at the end of a reading or punching operation, so that the first and last frames in a record can be read reliably.

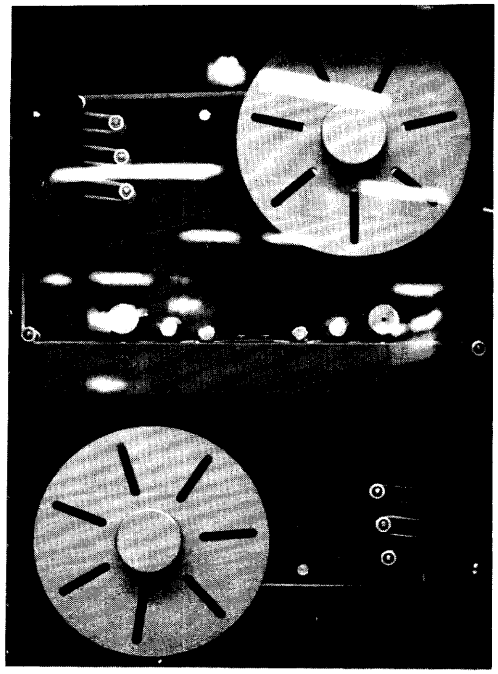
CHARACTERISTICS	209 READER AND CONTROL	210 PUNCH AND CONTROL
TAPE	11/16", 7/8", or 1" chadded tape in reels up to 700 ft. or in strips; tape can be Mylar[1], metallic-coated, or dry or oiled paper.	11/16", 7/8", or 1" nonmetallic tape in reels up to 700 ft. or in strips.
DATA FORMAT	Five- through 8-level codes; 10 frames per inch.	
PROGRAMMED OPERATIONS	Read data from tape (forward or backward) until input area filled; rewind; and runout. Stop distance at end of transfer is less than frame separation.	Punch contents of output area into tape.
SPEED	500 frames (characters) per second.	110 frames (characters) per second.
SIMULTANEITY	Read or punch operation engages central processor for only 2 microseconds for each 5- or 6-level character transferred — 4 microseconds for each 7- or 8-level character; central processor is free to perform other operations during more than 99.9% of a processing interval shared with paper tape data transfer.	
READ / PUNCH MECHANISM	Photoelectric	Die punch.
INPUT / OUTPUT AREAS	Any main memory area; can be any size.	
DATA PROTECTION	Optional check for odd or even parity; otherwise, parity bits read into memory for check by program.	Program-generated frame parity.
TRUNK REQ'S.	One input trunk.	One output trunk.
MAX. NO. PER SYSTEM	Eight	Eight

[1] Registered trademark of E. I. du Pont de Nemours and Company (Inc.)

Figure 11. Paper Tape Reader and Punch Characteristics.

CHARACTER CODES

Any 5- to 8-level paper tape code can be processed by either the reader or the punch. Reading and punching are performed in the transcription mode; *i.e*, a binary 1 in memory corresponds to a punch in the tape, while a 0 corresponds to an unpunched position on tape. The control units can be conditioned by programmed instruction to process either codes of 5 to 6 levels or codes of 7 to 8 levels. This facility offers a decided advantage when processing 5- or 6-level codes, since it minimizes the amount of central processor time required for data transfer (see below).



SIMULTANEITY

Data transfer between the central processor and either the reader or the punch involves the central processor for only two microseconds per 5- or 6-level frame or four microseconds per 7- or 8-level frame. Thus, the central processor is free during more than 99.9% of a paper tape read or punch interval to perform computations and to direct other input/output operations.

DATA PROTECTION

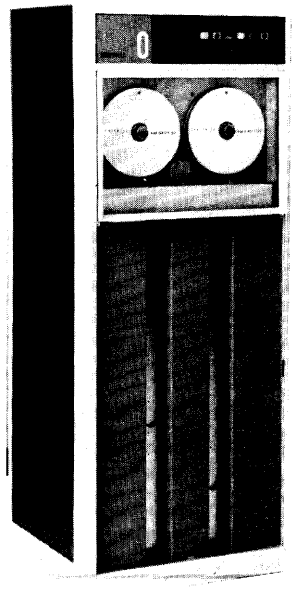
Frame parity can be generated by programmed instruction in preparation for punching. Likewise, frame parity can be checked by the program when reading tape. The reader can also be equipped to check each frame for odd or even parity and to set a program-accessible indicator if this check fails.

Magnetic Tape Units

Two complete series of magnetic tape equipment are offered for use with the Honeywell 200: the 204B series units process one-half inch tape, while the 204A series units process three-quarter inch tape. Both 203B controls for one-half inch tape units and 203A controls for three-quarter inch tape units can be included in the same system. The characteristics of these two series of tape equipment are summarized in Figures 12 and 13.

DATA TRANSFER SPEEDS

Half-inch tapes are processed by 204B series units at read/write speeds varying from 36 to 150 inches per second; these speeds, combined with recording densities of 200 and 556 characters per inch, provide eight possible data transfer rates ranging from 7,200 to 83,300 characters per second. Magnetic tape units in the 204A series offer data transfer rates of 32,000, 64,000, and 88,800 characters per second, based on recording densities of 533 and 740 characters per inch and read/write speeds of 60 and 120 inches per second.



CHARACTERISTICS	MODEL 204B-1,2 TAPE UNITS	MODEL 204B-3,4 TAPE UNITS	MODEL 204B-5 TAPE UNIT	MODEL 204B-6 TAPE UNIT
CONTROL	MODEL 203B-1 TAPE CONTROL		MODEL 203B-2 TAPE CONTROL	
TAPE	Reels of approx. 2400 ft. of 1/2-in. Mylar[1]-base, oxide-coated tape.			
DATA FORMAT	Variable-length records separated by short or 3/4-inch gap. Records consisting of 6-bit characters spaced at 556 or 200 per inch can be read. Normally writes at 556 char/in., but can write at 200 char/in.			
PROGRAMMED OPERATIONS	Read forward, write forward, backspace one record, rewind, rewind and release, and erase. Optional read backward and capability to translate between card images in IBM even-parity tape code and H-200 machine code.			
TRANSPORT	Pneumatic capstans and tape brakes.			
CROSS GAP TIME Short gap 3/4 inch gap	0.45 in. - 12.5 ms 20.8 ms	0.60 in. - 7.5 ms 9.4 ms	0.70 in. - 5.8 ms 6.3 ms	n/a 5.0 ms
READ/WRITE SPEED	36"/sec.	80"/sec.	120"/sec.	150"/sec.
DATA TRANSFER RATE (NOMINAL) 556 char/in. 200 char/in.	 20,000 char/sec. 7,200 char/sec.	 44,400 char/sec. 16,000 char/sec.	 66,700 char/sec. 24,000 char/sec.	 83,300 char/sec. 30,000 char/sec.
REWIND SPEED	108"/sec.	240"/sec.	360"/sec.	360"/sec.
SIMULTANEITY	Simultaneously compute and perform three tape operations: read or backspace-write-rewind-compute. Reading or writing engages central processor for only 2 microseconds per character transferred. Central processor is available for other operations during 83.3 to 98.6% of transfer interval shared with tape unit, depending upon data transfer rate.			
INPUT/OUTPUT AREA	Any main memory area.			
DATA PROTECTION	Write/protect ring and manual protect switch prevent destruction by unintentional write. While writing, TCU generates even or odd frame parity and even channel parity. Checks: Writing — Immediate read back and check of information written. Reading — Frame and channel parity checks. Failure of any check automatically sets a program-accessible indicator.			
TRUNKS	A tape control requires one input trunk and one output trunk.			
MAX. NO. OF UNITS PER SYSTEM	8 tape units per tape control; 8 tape controls per system.			

[1] Registered trademark of E. I. du Pont de Nemours and Company (Inc.)

Figure 12. Characteristics of Half-Inch Tape Units.

CHARACTERISTICS	MODEL 204A-1 ECONOMY TAPE UNIT	MODEL 204A-2 STANDARD TAPE UNIT	MODEL 204A-3 HIGH-DENSITY TAPE UNIT
CONTROL	203A-1 tape control	203A-2 tape control	203A-3 tape control
TAPE	Reels of approx. 2400 ft. of ¾-in. Mylar[1]-base, oxide-coated tape.		
DATA FORMAT	Variable-length records separated by 2/3-inch gaps; eight 6-bit characters per six frames.		
PROGRAMMED OPERATIONS	Read forward, write forward, backspace one record, rewind, rewind and release, and read, regenerating channel.		
TRANSPORT	Pneumatic capstans and tape brakes; read-write head retracts on rewind.		
CROSS GAP TIME	11.0 ms	5.5 ms	5.5 ms
RECORDING DENSITY (NOMINAL)	400 frames/inch 533 char/inch	400 frames/inch 533 char/inch	556 frames/inch 740 char/inch
READ/WRITE SPEED	60"/sec.	120"/sec.	120"/sec.
DATA TRANSFER RATE (NOMINAL)	32,000 char/sec.	64,000 char/sec.	88,800 char/sec.
REWIND SPEED	180"/sec.	360"/sec.	360"/sec.
SIMULTANEITY	Read or write-rewind-compute. Reading or writing engages central processor for only 2 microseconds per character transferred. Central processor is available for other operations during 82.2 to 93.6% of transfer interval shared with tape unit, depending upon data transfer rate.		
INPUT/OUTPUT AREA	Any main memory area.		
DATA PROTECTION	Write/protect ring and manual protect switch prevent destruction by unintentional write. While writing, TCU generates frame and channel parity bits. Checks: Writing — Check for presence of write-head power and absence of write-protect condition. Reading — Check frame and channel parity. Failure of any check automatically sets a program-accessible indicator.		
TRUNKS	A tape control requires one input trunk and one output trunk.		
MAX. NO. OF UNITS PER SYSTEM	4 tape units per tape control; 8 tape controls per system.		

[1] Registered trademark of E. I. du Pont de Nemours and Company (Inc.)

Figure 13. Characteristics of Three-Quarter Inch Tape Units.

CROSS-GAP TIMES

In Figures 12 and 13, magnetic tape units are characterized in terms of "cross-gap" rather than "start/stop" time. This points to a unique advantage provided in Honeywell tape units. When a tape read or write operation is completed, the tape unit begins a deceleration interval which is coincident with the creation of part of the inter-record gap on tape. However, it is not necessary for the unit to stop before beginning to execute a new read or write operation. If such an operation is begun at any time during the deceleration interval, the unit merely accelerates, completes the inter-record gap, and begins the next operation.

It is apparent that a decided speed advantage may be gained by initiating a new read or write operation early in the deceleration period when the tape is still travelling at high speed. The gap is created at full speed and cross-gap time is minimized.

TAPE PROCESSING SIMULTANEITY

A powerful feature of the Honeywell 200 is its tape processing simultaneity. All tape units are capable of transferring data simultaneously with other central processor operations. The half-inch tape drives, in particular, are capable of reading and writing simultaneously with other central processor operations.

The ability to perform tape operations simultaneously is further enhanced by the fact that the central processor is involved in a tape read or write operation during only two microseconds per character transferred. Thus, the majority of a tape processing interval is available to the central processor to perform computations or direct other peripheral operations. As indicated in Figures 12 and 13, the proportion of available central processor time during a data transfer interval shared with a tape read or write operation ranges from 82.2% to 98.6%, depending upon the data transfer rate of the tape unit being used. A typical tape processing interval is shown in Figure 14.

TAPE COMPATIBILITY

All half-inch tape drives and controls can be equipped to process tapes which have been written by IBM 727 or 729 series tape units or to write tapes to be read by these units. This capability includes end-of-file mark recognition, the ability to record at a density of 200 characters per inch, and the ability to translate between card images in IBM even-parity tape code and H-200 machine code.

DATA PROTECTION

The design of all H-200 tape units incorporates the vacuum techniques which have earned Honeywell drives an outstanding reputation for error-free operation. The vacuum techniques are used to mount, drive, and stop the tape so as to avoid any danger of damage; the reading surface of the tape has physical contact only with the read/write head.

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

All information written on a half-inch tape unit is immediately read and checked. During a write operation, a parity bit is generated for each frame and another is generated for each data channel. These bits accompany the record on tape. Frame and channel parity are checked while reading. Failure of any of these checks automatically sets an indicator which can be tested by a programmed instruction. The three-quarter inch tape units have the further ability to regenerate any tape channel on the basis of the parity established by the other channels and the frame parity bits.

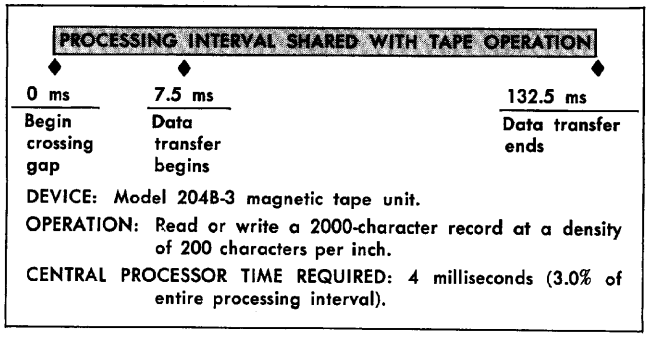


Figure 14. Simultaneity in Typical Tape Operation.

Random Access Drum File and Control

Up to eight drum storage units can be connected to the Model 270 Random Access Drum Control. Each drum provides storage for 2,621,441 characters, allowing a total capacity of approximately 21 million characters. The characteristics of the drum file and control are presented in Figure 15.

CHARACTERISTICS	MODEL 270 RANDOM ACCESS DRUM AND CONTROL
DATA CAPACITY	2,621,440 characters/drum; up to eight drums per control.
DATA FORMAT	Record length varies with size of main memory I/O area; data stored as 6-bit characters.
PROGRAMMED OPERATIONS	Search and write; search and read.
DRUM ACCESSING FRAMEWORK	Drum divided into 8 zones of 64 tracks. Each track subdivided lengthwise into 40 sectors. Drum address specifies drum, zone, track, and sector.
ACCESSING METHOD	Drum rotates at 1200 rpm under 512 read-write heads (one per track).
SPEED	Avg. access time 25 milliseconds; transfer rate avgs. 102,000 char/sec.
SIMULTANEITY	Read or write engages central processor for only 2 microseconds per character transferred. The central processor is free to compute or initiate other peripheral operations during approx. 79.6% of data transfer interval shared with drum.
INPUT/OUTPUT AREA	Any main memory area.
CHECKING	Character parity automatically generated on write and checked on read. Discrepancy causes indicator to be set.

Figure 15. Characteristics of 270 Drum File and Control Unit.

OPERATIONS

The random access drum file is capable of performing the following operations under program control:

1. Locate a specified storage area on a drum and write data until a record mark is sensed in memory.
2. Locate a specified storage area on a drum and read data until a record mark is sensed in memory.

TIMING

The drum file rotates at a constant speed of 1200 rpm. Therefore, any record can be located in no more than 50 milliseconds and an average of 25 milliseconds. Data is transferred between the central pro-

cessor and the drum at an average rate of 102,000 characters per second.

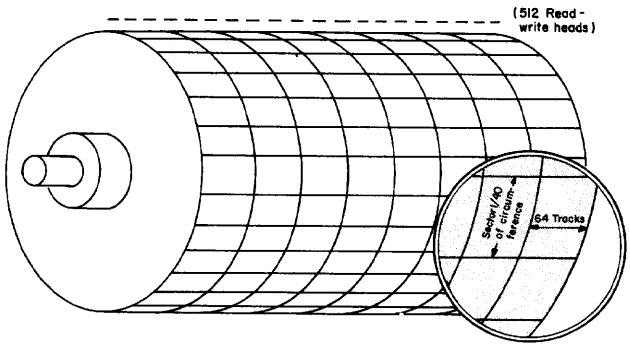


Figure 16. Layout of Drum File Surface.

SIMULTANEITY

Even though the rate of data transfer between the drum and the central processor is very high, the central processor is available to perform other operations during the majority of a processing interval shared by a drum operation. Because only two microseconds of central processor time is required to transfer each character, the central processor can compute or initiate other peripheral operations during approximately 79.4% of the data transfer interval.

DATA PROTECTION

The drum control automatically generates a parity bit for each character to be written; these bits accompany the record on the drum. An automatic character parity check is performed while reading; any discrepancy causes a program-accessible indicator to be set.

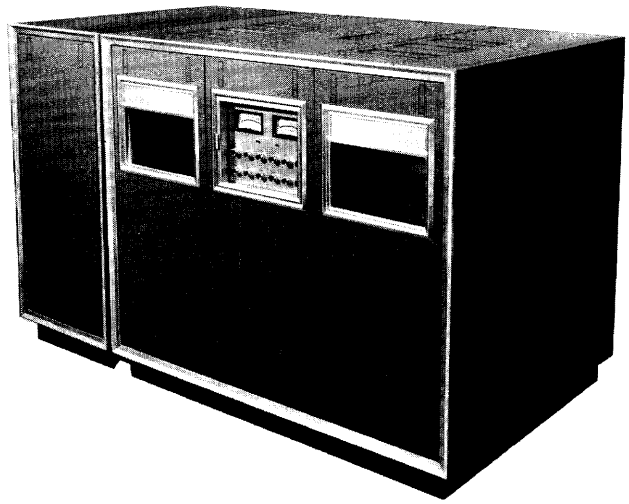
Random Access Disc File and Control

Series 260 Random Access Disc Files having storage capacities up to 100,663,296 characters may be included in a Honeywell 200 system. The characteristics of the various disc file models are presented in Figure 17.

OPERATIONS

The random access disc file can perform the following operations under program control:

1. Locate a specified storage area on a disc and write data until a record mark is sensed in memory.
2. Locate a specified storage area on a disc and read data until a record mark is sensed in memory.



CHARACTERISTICS	SUMMARY
DATA CAPACITY	One to 24 discs, each having a capacity of 4,194,304 characters: Model No. / Capacity (characters) / Model No. / Capacity (characters) 260-1 / 4,194,304 / 260-6 / 25,165,824 260-2 / 8,388,608 / 260-7 / 50,331,648 260-3 / 12,582,912 / 260-8 / 75,497,472 260-4 / 16,777,216 / 260-9 / 100,663,296 260-5 / 20,971,520
DATA FORMAT	Record length varies with size of main memory I/O area; data stored as 6-bit characters.
PROGRAMMED OPERATIONS	Search and read; search and write.
DISC ACCESSING FRAMEWORK	Disc face divided into 6 concentric zones which are each subdivided into 128 concentric tracks. Partial radii divide zones into a total of 128 sectors. Disc address specifies disc, face, sector, and track.
ACCESSING METHOD	Discs rotate at 900 rpm under 6 movable read-write heads (one per zone).
SPEED	Avg. access time 118-138 milliseconds; transfer rate varies with zone from 23,550 c/sec. to 64,300 c/sec.
SIMULTANEITY	Read or write engages central processor for only 2 microseconds per character transferred; on the average, central processor is free to compute or initiate other peripheral operations during approx. 89.6% of disc data transfer interval.
INPUT/OUTPUT AREA	Any main memory area.
CHECKING	Address-verification check on disc search. Character parity generated on write and checked on read.

Figure 17. Characteristics of 260 Series Disc File and Control Units.

TIMING

The access time required to locate any storage area on a disc face is the sum of the head positioning time and the disc latency time. Since all discs rotate at 900 rpm, the disc latency averages one half rotation or 33 milliseconds. The time required to position all six read/write heads per disc face averages from 85 to 105 milliseconds. Therefore, the average access time is 118 to 138 milliseconds. The data transfer rate between the disc file and the main memory varies between 23,550 and 64,300 characters per second, depending upon the disc zone referenced (Figure 18).

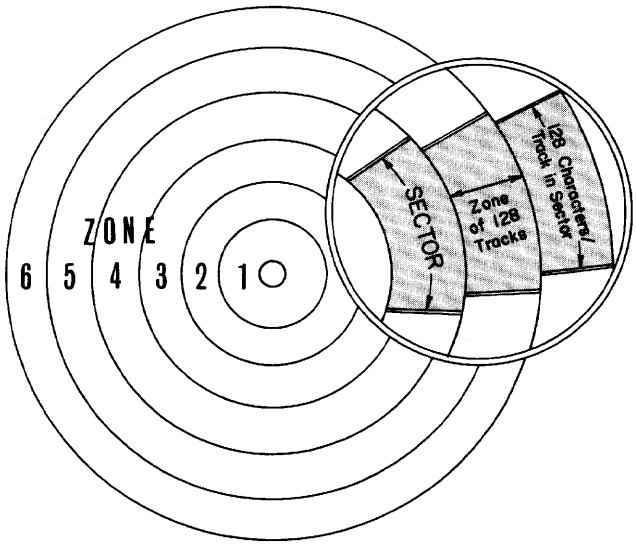


Figure 18. Detail of Disc File Layout.

SIMULTANEITY

Even though the rate of data transfer between the disc file and the central processor is very high, the central processor is available to perform other operations during the majority of a processing interval shared by a disc operation. Because only two microseconds of central processor time are required to transfer each character, the central processor can compute or direct other peripheral operations during an average of 90% of the data transfer interval.

DATA PROTECTION

The disc control automatically generates a parity bit for each character to be written; these bits accompany the record on the disc. An automatic character parity check is performed while reading; the detection of a discrepancy causes an indicator to be set.

Data Communication Controls

Two different types of communication controls are available to enable the Honeywell 200 to receive and transmit data over toll and leased communication lines: the 281 Single-Channel Communication Control and the 284 Multi-Channel Communication Control. The characteristics of the various models of controls are presented in Figure 19.

The 281 controls the transmission and reception of messages in 5- to 8-level codes at rates of up to 5,100 characters per second. It is a half-duplex control; *i.e.,* messages are both transmitted and received, but not simultaneously. Additional 281 controls can be added to a system in order to provide full duplex or multiple-channel operation.

The 284 is a multi-channel device which can control the transmission and reception of messages over as many as 64 communication lines. A Model 285 Communication Adapter Unit (CAU) is required as an interface between the 284 and each channel being used. Data can be transferred at rates up to 300 characters per second in a single line. Even when allowance is made for the normal input/output programming accompanying such an operation, a total character rate (all channels) of up to approximately 2500 characters per second can be maintained.

OPERATIONS

A communication control can perform the following operations under program control:

1. Transmit a character from memory.
2. Transmit an entire message from memory (type 281 only).
3. Accept a character from a remote device and store in memory.
4. Accept an entire message from a remote device and store in memory (type 281 only).

The system can be equipped with a hardware interrupt capability which allows other completely unrelated applications to proceed automatically at full

CHARACTERISTICS	TYPE 281 SINGLE-CHANNEL COMMUNICATION CONTROL		TYPE 284 MULTI-CHANNEL COMMUNICATION CONTROL
DATA FORMAT	5- through 8-level character codes.		
OPERATION MODE	Single-channel, half duplex; sends or receives by single character or by whole message.		Messages processed character-by-character; various models accommodate half-duplex channels as follows: 284-1, 16; 284-2, 32; 284-3, 48; and 284-4, 64.
DATA TRANSFER SPEED	Up to 5,100 characters/sec.		Up to 300 characters/sec. in a single channel.
COMPATIBLE NETWORKS	Network	Data Modem	Control Unit Model
	TELEX or TWX prime	Bell 100 series DATA-SET	281-1; or 284 with 285-1 Communication Adapter Unit.
		Bell 200 series DATA-PHONE	281-2; or 284 with 285-2 Communication Adapter Unit.
	Voice-grade lines	Bell 402C DATA-PHONE (send only)	281-3; or 284 with 285-3 Communication Adapter Unit.
		Bell 402D DATA-PHONE (rec. only)	281-4; or 284 with 285-4 Communication Adapter Unit.
CENTRAL PROCESSOR INTERRUPTION DURING DATA TRANSFER	Two microseconds per character in 5- or 6-level code; 4 microseconds for 7- or 8-level code.		Four microseconds per character.
COMMUNICATION WITH CONTROLLING PROGRAM	Program can test for presence of incoming data or readiness of control to transmit. Optional hardware program interrupt performs same function automatically.		Controlling program is interrupted automatically when message is received.
DATA PROTECTION	Sending: Where applicable, program can initiate automatic message receipting system. Receiving: Indicator set if transmission lapses for longer than specified time. Optional character parity check available.		
TRUNKS	One input trunk and one output trunk per communication control.		
MAX. NO. PER SYSTEM	Varies with anticipated total character rate in all channels.		

Figure 19. Characteristics of Data Communication Controls.

speed in the intervals between receipt and transmission of messages.

NETWORKS AND TERMINAL DEVICES

Both the 281 and the 284 can control communications over teleprinter and voice-grade lines between the H-200 and a wide variety of terminal devices. Figure 20 illustrates a few of the communication networks that can be established with these controls.

SIMULTANEITY

Data transfer between the central processor and a 281 communication control engages the central processor for two microseconds for each character in 5- or 6-level code or four microseconds for each character in 7- or 8-level code. Four microseconds of central processor time is required for every character transferred to or from a 284 control. For example, if an H-200 communication control is connected via a 201B DATA-PHONE to a voice-grade line carrying 8-bit characters at a rate of 250 per second, the central processor is free to perform other operations during 99.9% of message transfer time.

DATA PROTECTION

Data being transmitted and received by an H-200 communication control is protected by three different methods: checks for transmission lapses, an optional character parity check, and a semi-automatic message-receipting system. Failure of a transmission or parity check automatically sets an indicator which can be tested by a programmed instruction. When desired, a transmitting control can interrogate the status of the receiving control to insure that the previous message was correctly received.

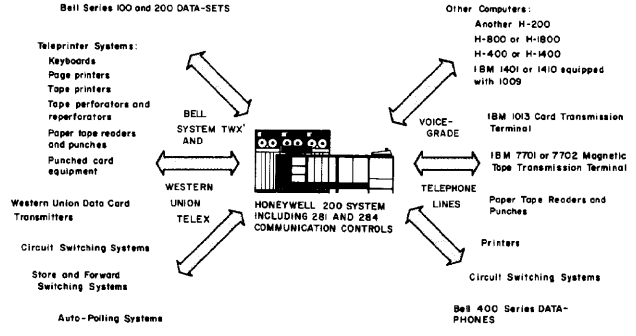


Figure 20. Sample of Possible H-200 Data Communication Connections

4 Programming Aids

A complete set of software is offered with the Honeywell 200 which enhances the effectiveness of the system by mating the basic data processing power of the hardware with ease of programming and simplicity of operation. Three separate systems are scheduled: the Easycoder Assembly and Operating System, a COBOL (COmmon Business-Oriented Language) compiler, and Automath, a scientific-language compiler.

The Easycoder System

The Easycoder Assembly and Operating System is designed to assist the users of Honeywell 200 Systems in all phases of their data processing efforts. It greatly simplifies programming and speeds other operations required for efficient operation. The major elements of the Easycoder system are:

1. The Easycoder symbolic language.
2. The Assembly Program, which accepts programs written in Easycoder language, translates them into machine language, and provides a listing of each program with appended diagnostic information. Program loading and dump routines are generated according to the programmer's specifications.
3. Sort-generating programs.
4. A tape-handling routine, THOR, which provides tape positioning and data editing/updating capabilities.
5. TABSIM, a routine which simulates on the H-200 the operation of tabulating equipment.
6. Magnetic tape input and output routines.
7. A report-generating program.
8. A set of utility routines for handling other common processing problems.

Most elements in the Easycoder system are implemented for the minimal configuration consisting of a central processor with 2048 characters of memory and a card reader-card punch. However, certain functions are not available in this basic configuration and are so indicated in the following discussion.

EASYCODER PROGRAMMING STATEMENTS

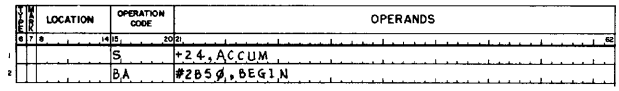
The Easycoder Assembly Language includes easily remembered operation codes which are abbreviations for the names of programming statements that the Assembly Program can interpret. For example, the mnemonic SI is the operation code for the Set Item Mark statement, which is the direct counterpart of an H-200 machine instruction. The mnemonic ORG, on the other hand, is the operation code for an Origin statement, which is not assembled but which directs the Assembly Program in allocating memory space to the succeeding statements.

A typical Easycoder statement may consist of a tag, a mnemonic operation code, one or more operand addresses (or literal operands), and one or more variant characters. Addresses may be either absolute decimal numbers or symbolic tags. The normal placement of these elements on a coding form is shown here.

TYPER	LOCATION	OPERATION CODE	OPERANDS
	COMPAR	BCE	GOTOIF, SAMEAS, 23

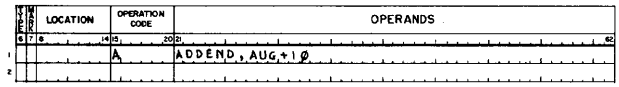
The BCE (Branch if Character Equal) instruction shown in this example can be referenced elsewhere in the program by the symbolic tag COMPAR. (The location field could be blank or it could contain an absolute address so as to assign a specific location to the machine-language equivalent.) The Assembly Program produces a machine-language instruction which will cause a program branch to location GOTOIF if location SAMEAS contains the octal value 23, as indicated by the variant character.

An Easycoder statement may also specify the value of an operand directly by means of a literal[1]. In this example, the assembled instruction will cause the



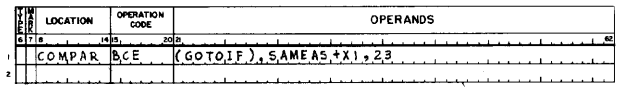
decimal literal 24 to be subtracted from the contents of the field tagged ACCUM. A statement may also contain a binary or octal literal.

Symbolic addresses may be expressed relative to a tag defined elsewhere in the program. For example, the following statement is assembled as an instruction



which will add the contents of the field tagged ADDEND to the field 10 locations beyond the location tagged AUG.

Both indexed and indirect addressing (see page 10) may be specified in Easycoder language. For example, assume that the BCE statement above is modified as follows:



In this case, the variant character is compared to the contents of the location whose address is formed by adding the address of the data tagged SAMEAS to the value stored in index register 1 (X1). If the characters are identical, the program branches, not to location GOTOIF, but to the location whose address is stored at location GOTOIF (indirect addressing).

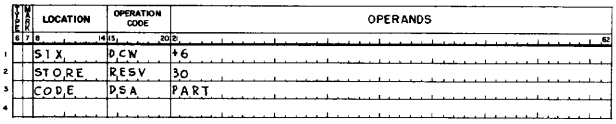
DATA FORMATTING STATEMENTS

As indicated above, there are two different types of Easycoder programming statements. The operations obtainable by using those statements which have machine-language counterparts are described in Section 5. (The components of machine instructions are described on page 10.) The statements which do not have machine-language equivalents are data formatting statements and assembly control statements.

Data formatting statements direct the Assembly Program to perform the following operations:

1. Store a specified constant in a field whose location is indicated by an absolute or symbolic address.
2. Store a specified address as a constant in a field whose location is indicated by an absolute or symbolic address.
3. Set aside a specified memory area whose location is indicated by an absolute or symbolic address.
4. Set aside one or more specified memory areas and punctuate them as indicated for fields, items, and records. In addition, indexing of all references to these fields, items, and records can be specified. (This function is not implemented for the minimum equipment configuration.)

Typical Easycoder data formatting statements are shown below. The first statement directs the Assembly Program to store the decimal value +6 as a constant in a location to be addressed by the tag SIX.



The second statement reserves an area of 30 locations which can be referenced by the tag STORE. The third statement directs the Assembly Program to store the absolute address assigned to the tag PART in a field which can be referenced by the tag CODE.

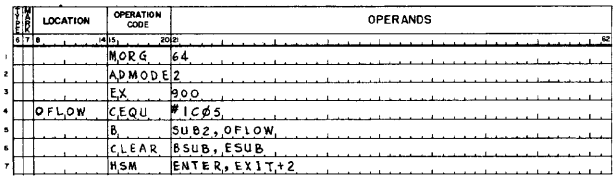
ASSEMBLY CONTROL STATEMENTS

Easycoder assembly control statements direct the Assembly Program to perform the following operations:

1. Specify the beginning and end of a program.
2. Load a block of coding at a particular address.
3. Provide for a temporary interruption of object program loading in order to execute a block of object program instructions.
4. Assemble addresses in either 2-character or 3-character form.
5. Equate a tag with either an absolute address or another tag.
6. Equate an octal constant with a tag which represents a variant character.
7. Generate a dump routine to accompany the object program, which will obtain a printed listing of the contents of any portion of main memory.
8. Generate a routine to accompany the object program, which will clear specified portions of memory to zeros before the object program is loaded.

[1] Literals are not implemented for the minimum equipment configuration.

Typical Easycoder assembly control statements appear as follows. The first statement indicates that



	LOCATION	OPERATION CODE	OPERANDS
1		MORG	64
2		ADMODE	2
3		EX	900
4	OFLOW	CEQU	#1Ø85
5		B	SUB2,OFLOW
6		CLEAR	BSUB,ESUB
7		HSM	ENTER,EXIT+2

memory allocation for subsequent statements should begin with the next location whose address is a multiple of 64. The second statement directs the Assembly Program to assemble the address portions of all subsequent statements as two-character addresses. The third statement causes the Assembly Program to provide for a temporary interruption of object program loading and for a transfer of program control to location 900.

The fourth and fifth statements illustrate how a symbolic tag can be used in place of a variant character. The CEQU statement directs the Assembly Program to equate the tag OFLOW to the octal value 5. The fifth line contains a Branch statement which causes the program to branch to the location tagged SUB2 if the condition specified by the variant character tagged OFLOW is present (condition 5 represents arithmetic overflow).

The CLEAR statement on the sixth coding line specifies that the area from location BSUB to location ESUB is to be cleared to zeros before the object program is loaded. The HSM statement directs the Assembly Program to generate a memory dump routine capable of listing the contents of the memory area from location ENTER to location EXIT+2.

EASYCODER ASSEMBLY PROGRAM

The Easycoder Assembly Program translates the symbolic source program (written on the Easycoder coding form and subsequently punched into a source program card deck) into machine-language entries. Mnemonic operation codes are replaced by their machine equivalents. Absolute addresses are assigned to all symbolic tags which appear in the location field and replace all other references to those tags. All operations indicated by data formatting statements and assembly control statements are performed. In addition to an executable machine-language object program, the Assembly Program also produces a printed listing containing the symbolic source program and the corresponding object program entries.

SORT-GENERATING PROGRAMS

Easycoder sort-generating programs produce routines, based on parameters specified by the programmer, which will sort data on as many as seven keys. Fixed-item-length files of up to one full reel may be sorted; items may be blocked one or more per record, and records can be as large as 800 characters.

The generated sort routines use the Polyphase sorting technique developed by Honeywell. This technique permits efficient sorting with as few as three magnetic tape units; however, larger equipment configurations can be used to advantage. The sort times shown in the table below exemplify the power of Easycoder-generated routines. The times are based on an H-200 system containing a 4,096-character memory and 64,000-character-per-second tape units. Item size is 80 characters and items are blocked four per record.

No. of Items	No. of Tape Units	Time in Minutes
10,000	3	8
10,000	4	5
40,000	3	33
40,000	4	25

Easycoder sort generators are easy to use: parameters specifying the nature of the data to be sorted and the equipment to be used are punched into a single card which is input to the generating program at object time.

THOR (Tape-Handling Option Routine)

THOR is a general-purpose tape-handling and correction routine controlled by input parameters read from punched cards or entered from the H-200 control panel. Under the direction of these parameters, THOR positions magnetic tape and copies, corrects, and edits the data contained thereon. Specifically, THOR can be instructed to initiate any of the following operations:

1. Move tape forward or backward by a specified number of records.
2. Rewind the tapes on one or more tape units.
3. Locate the item on tape containing specified information in a particular field.
4. Copy data from one tape to another, updating it as specified.
5. Compare data on two different tapes and print records containing differences.
6. Edit data for printing.

TABSIM (Tabulating Equipment Simulator)

TABSIM prepares printed reports from input consisting of a deck of punched cards (or a tape file of card images). As its name indicates, TABSIM simulates the operations of standard tabulating equipment.

The basic input to TABSIM is a deck containing data or "detail" cards and control cards. The control cards contain instructions for the processing of the data cards. The output is a printed or punched report in the format requested by the input control cards. In general terms, the output report represents the data on the detail cards edited and processed arithmetically. More specifically, TABSIM can perform any of the following operations:

1. Print a listing of all input cards. Edit the data to include dollar signs, commas, etc. Rearrange the data on the input cards (*e.g.*, place the contents of column 80 in print position 7).
2. Distinguish between different types of input cards, and print each group according to the control card specifications for that group.
3. Add, subtract, multiply, and divide detail data and totals derived from detail data. For instance, fields within a group can be added to produce totals; these totals can then be added to totals for other groups, etc., to produce a final total at the end of the report.
4. Crossfoot detail lines and print results.
5. Print heading lines.
6. Print or punch summary information.
7. Perform all control functions associated with these operations (*e.g.*, skipping to total lines, spacing after printing, etc.).

Both printing and summary punching can be accomplished during the same TABSIM run or either can be accomplished separately. Output can be listed on either pre-printed or blank forms.

TAPE INPUT AND OUTPUT ROUTINES

A tape input/output package is available with the Honeywell 200 to control the blocking and unblocking of data stored on magnetic tape and to process conventional file identification and control information. The routines in this package open and close tape files, get or write items in a file, and force tape swaps.

REPORT-GENERATING PROGRAM[1]

An Easycoder program which generates routines to create reports according to user specifications is under development. In order to use the report generator, the programmer prepares a set of parameters defining control fields and report lines. These parameters are input to the report generator at object time. Routines are generated which automatically edit data and produce reports in the manner specified by the parameters.

[1] The report-generating function is not implemented for the minimum equipment configuration.

Honeywell 200 COBOL Compiler System

COBOL comprises a language which is a standardized, business-oriented subset of English, and a processing system called a compiler. The programmer describes a solution to a business problem in COBOL language, and then the processing system generates machine-language instructions capable of performing the operations described by the programmer's statements.

The projected Honeywell 200 COBOL is a syntax-directed, systems-integrated compiler which can operate in a system consisting of 16,384 characters of memory, six magnetic tape units, a card reader (or paper tape reader), and an on-line printer. The compiler is designed to be modularly expandable and is self-adapting to memories larger than the minimum.

The H-200 COBOL compiler possesses several significant operating features: the operating system is job-oriented, it includes built-in facilities for program testing, it permits parallel execution of card-to-tape and tape-to-printer operations, and it is self-monitored.

Automath 200 — Scientific-Language Compiler System

Automath consists of a scientific language (FORTRAN II) and a processing system. The solution to a scientific problem is described in Automath language, and then the processing system, called a compiler, generates a machine-language program to perform the operations constituting the problem solution.

The timetable for H-200 software includes an Automath compiler which will operate in three passes, the final output of which is an object program either on punched cards or, for load-and-go operation, on magnetic tape.

The Automath 200 compiler will operate in a system configuration comprising 16,384 characters of memory, four magnetic tape units, a card reader, and a card punch. Additional memory and tape units can be used to advantage.

5 Instructions

The Honeywell 200 repertoire includes approximately 32 very powerful instructions. These instructions, which are variable in length, are of five types: arithmetic, logic, control, editing, and input/output.

Instruction Descriptions

Each H-200 instruction is described below in terms of four parameters: name, operation code, format, and execution time. The following format is used to describe all instructions.

Instruction Name **Mnemonic Op Code**

Easycoder Instruction Format: Execution Time (microseconds)

Description of instruction execution.

The operation initiated by each format of an instruction is described separately, and an execution time is given which is based on a typical situation involving two-character direct addressing and five-character data fields, except where otherwise indicated. (The instruction summary table in Section 6 contains detailed timing information for each instruction.)

CONVENTIONS

The following symbols are used in the instruction descriptions to convey the meanings indicated.

SYMBOL	MEANING
A and B	Addresses of main memory locations.
(A) and (B)	The contents of the fields indicated by A and B.
(R_a)	The contents of the A-address register.
(R_b)	The contents of the B-address register.

The phrases "field indicated by A" and "field at A" are synonymous and refer to the data field whose rightmost limit is location A and whose leftmost limit is marked by the next word-marked character to the left of location A. The expression "instruction indicated by B" refers to the instruction whose operation code is stored in location B.

Arithmetic Instructions

Six arithmetic instructions perform both decimal and binary arithmetic. Four basic instructions, decimal add, decimal subtract, binary add, and binary subtract, can be supplemented by two optional instructions, zero and add, and zero and subtract.

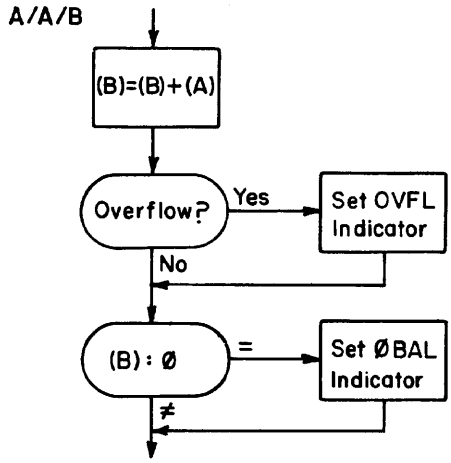
Decimal arithmetic instructions treat their operands as signed numeric data. Normal algebraic sign control is in effect during the execution of these instructions. Any zone bit configuration other than 10 in the rightmost character of a decimal field causes the field value to be interpreted as positive; 10 indicates a negative field value. Decimal arithmetic results are delivered with normalized sign bits; positive and negative results are indicated, respectively, by the standard sign bit configurations 01 and 10.

An indicator is set at the completion of each decimal arithmetic operation to indicate the presence or absence of a zero result. A different indicator is set if overflow is sensed. The status of either of these indicators can be tested by a subsequent programmed instruction.

In binary arithmetic operations, the operands are treated as unsigned binary numbers; overflow and zero balance are disregarded.

Instruction Format	Execution Time	
Decimal Add **A**		
	(A)<(B)	(A)≥(B)
A/A/B: (A) is added algebraically to (B), and the result is stored in the field at B.	44	64

Instruction Format	Execution Time (A)<(B)	Execution Time (A)≥(B)
A/A: (A) is added algebraically to itself; the sum is stored in the field at A.	40	40
A: The data in the field specified by (R_a) is added algebraically to the data in the field specified by (R_b) and the sum is stored in the field specified by (R_b).	36	56



Decimal Add Operation

Decimal Subtract S

	Execution Time (A)>(B)	Execution Time (A)≤(B)
S/A/B: (A) is subtracted algebraically from (B), and the result is stored in the field at B.	44	64
S/A: (A) is subtracted algebraically from itself, and the result is stored in the field at A.	40	40
S: The data in the field specified by (R_a) is subtracted algebraically from the data in the field specified by (R_b), and the sum is stored in the field specified by (R_b).	36	56

Binary Add BA

Instruction Format	Execution Time
BA/A/B: (A) is added to (B), and the sum is stored in the field at B.	42
BA/A: (A) is added to itself; the sum is stored in the field at A.	38
BA: The data in the field specified by (R_a) is added to the data in the field specified by (R_b) and the sum is stored in the field specified by (R_b).	34

Binary Subtract BS

BS/A/B: The twos complement of (A) is added to (B), and the result is stored in the field at B.	42
BS/A: The twos complement of (A) is added to (A), and the result is stored in the field at A.	38
BS: The twos complement of the value in the field specified by (R_a) is added to the value in the field specified by (R_b), and the result is stored in the field specified by (R_b).	34

Zero and Add ZA

ZA/A/B: (B) is cleared to zero, and (A) is moved to field at B. All zone bits in (B), other than those in the sign position, which are normalized, are cleared to zero.	44
ZA/A: All zone bits in the field at A, other than those in the sign position, which are normalized, are cleared to zero.	38
ZA: The field specified by (R_b) is cleared to zero, and then the data in the field specified by (R_a) is moved to the field specified by (R_b). All zone bits other than those in the sign position, which are normalized, are cleared to zero.	36

Zero and Subtract ZS

ZS/A/B: (B) is cleared to zero, and (A) is moved to the field at B. All zone bits in (B), other than those in the sign position, are cleared to zero; the sign is reversed.	44
ZS/A: All zone bits in the field at A, other than those in the sign position, are cleared to zero; the sign is reversed.	38
ZS: The field specified by (R_b), is cleared to zero, and then the data in the field specified by (R_a) is moved to the field specified by (R_b). All zone bits other than those in the sign position are cleared to zero; the sign is reversed.	36

Logic Instructions

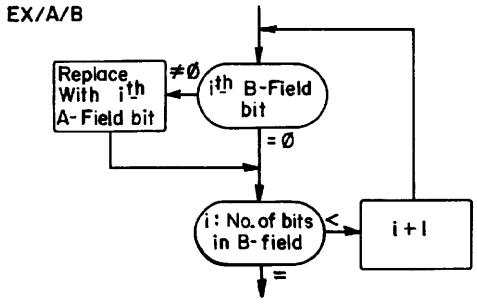
Seven instructions are included in this category. Extract, Half Add, and Substitute manipulate data on an individual bit basis, combining bits from two different fields according to rules based on AND and OR logic. Each of the remaining instructions causes a program branch to be performed, contingent upon the existence of a precisely defined condition.

Extract EX

EX/A/B: Each 1-bit in the field at B is replaced with the corresponding bit from (A); all zeros in (B) are undisturbed. 42

EX/A: Each 1-bit in the field specified by (R_b) is replaced with the corresponding bit from (A); all zeros in the field specified by (R_b) are undisturbed. 38

EX: Each 1-bit in the field specified by (R_b) is replaced with the corresponding bit from the field specified by (R_a); all zeros in the field specified by (R_b) are undisturbed. 34



Extract Operation

Half Add HA

HA/A/B: The binary fields at (A) and (B) are added without carry, and the result is stored in the field at B. 42

HA/A: The binary field at (A) is added without carry to the binary value in the field specified by (R_b); the result is stored in the field specified by (R_b). 38

HA: The binary value in the field specified by (R_a) is added without carry to the binary value in the field specified by (R_b); the result is stored in the field specified by (R_b). 34

Compare C

C/A/B: (B) is compared bit-by-bit with an equal number of characters in the field indicated by A; indicators are set to show the result of the comparison. 34

C/A: The value in the field specified by (R_b) is compared bit-by-bit with an equal number of characters in the field at A; indicators are set to show the result of the comparison. 30

C: The value in the field specified by (R_b) is compared bit-by-bit with an equal number of characters in the field specified by (R_a); indicators are set to show the result of the comparison. 26

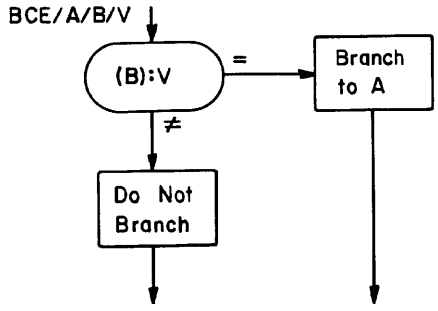
Substitute SST

SST/A/B/V: Each bit in the character at B which corresponds to a 1-bit in V is replaced by the corresponding bit in the character at A; other B-bits remain undisturbed. 20

SST: Each bit in the character specified by (R_b) which corresponds to a 1-bit in the variant character last specified in an instruction is replaced by the corresponding bit in the character specified by (R_a); other bits in the character specified by (R_b) remain undisturbed. 10

Branch if Character Equal BCE

BCE/A/B/V: A program branch to the instruction at A occurs if the character at B 20



Branch if Character Equal Operation

is the same as V; otherwise, the program continues in sequence.

BCE: A program branch to the instruction specified by (R_a) occurs if the character at (R_b) is the same as the variant character last specified by an instruction; otherwise, the program continues in sequence. 10

Branch B

B/A: The contents of the instruction address register are stored in the B-address register, and a program branch to the instruction at A is performed. 10

B/A/V: If the indicator specified by V is set, the contents of the instruction address register are stored in the B-address register and a program branch to the instruction at A is performed; otherwise, the program continues in sequence. 12

VARIANT	INDICATOR	VARIANT	INDICATOR
00	B < A (low compare)	11	Sense Sw 1 ON
01	B ≠ A	12	Sense Sw 2 ON
02	B > A (high compare)	13	Sense Sw 3 ON
03	B = A	14	Sense Sw 4 ON
04	Zero balance	15	Unconditional
05	Overflow		

Correspondence between Branch Instruction Variants and Indicators.

Branch on Character Condition BCC

BCC/A/B/V: If the character at B contains the type of punctuation or sign indicated by V, the contents of the instruction address register are stored in the B-address register, and a program branch to the instruction at A is performed; otherwise, the program continues in sequence. 20

BCC: If the location specified by (R_b) contains the type of punctuation or sign indicated by the variant character last specified by an instruction, the contents of the instruction address register are stored in the B-address register, and a program branch to the instruction at A is performed; otherwise, the program continues in sequence. 10

Control Instructions

The instructions in this category are used to manipulate data within the main memory and the control memory, to prepare main memory storage areas for the processing of data fields, and to control the sequential selection and interpretation of instructions in the stored program.

Set Word Mark SW

SW/A/B: Word marks are set in the locations specified by A and B; the data in these locations are undisturbed. 16

SW/A: A word mark is set in A; the data in this location is undisturbed. 12

SW: Word marks are set in the locations specified by (R_a) and (R_b); the data in these locations are undisturbed. 8

Set Item Mark SI

SI/A/B: Item marks are set in A and B; the data in these locations are undisturbed. 16

SI/A: An item mark is set in A; the data in this location is undisturbed. 12

SI: Item marks are set in the locations specified by (R_a) and (R_b); the data in these locations are undisturbed. 8

Clear Word Mark CW

CW/A/B: Locations A and B are cleared of word marks; the data at these addresses are undisturbed. 16

CW/A: The word mark at location A is cleared; the data at this address is undisturbed. 12

CW: Locations (R_a) and (R_b) are cleared of word marks; the data at these addresses are undisturbed. 8

VARIANT	BRANCH CONDITION
00	Item mark
01	Word mark
02	Negative sign
03	High-order bits contain zeros
04	Positive sign
05	High-order bits are 11
06	No punctuation mark
07	Word mark or item mark

Branch Conditions Indicated by BCC Variants.

Clear Item Mark — CI

CI/A/B: Locations A and B are cleared of item marks; the data at these addresses are undisturbed. 16

CI/A: The item mark in location A is cleared; the data in this location is undisturbed. 12

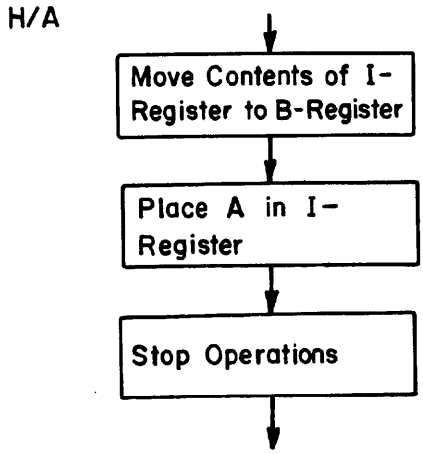
CI: Locations (R_a) and (R_b) are cleared of item marks; the data in these locations are undisturbed. 8

Halt — H

H: The machine is halted unconditionally. The instruction address register contains the address of the instruction following the halt. 6

H/A: The contents of instruction address register 1 are transferred to the B-address register, and A is placed in instruction register 1; then the machine is halted. 10

H/A/B: The machine is halted unconditionally. A and B are stored in the address registers as halt identification symbols. 14



Halt and Branch Operation

No Operation — NOP

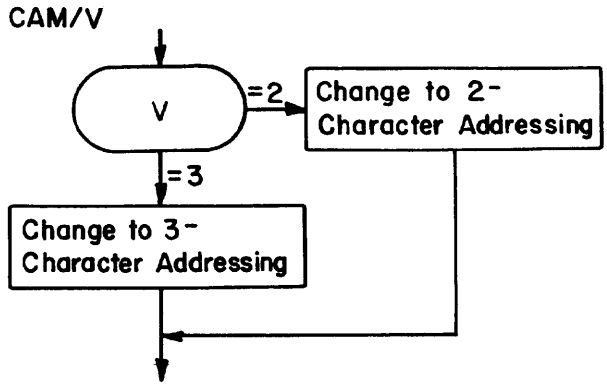
NOP: Instruction address register 1 is incremented, but no other operation is performed. 6

Change Sequencing Mode — CSM

CSM: The contents of the two instruction address registers are interchanged, and a program branch to the address now in instruction address register 1, (which controls program sequence) is performed. 8

Change Addressing Mode — CAM

CAM/V: The addresses in all following instructions until the next CAM instruction are interpreted, in accordance with V, as either two- or three-character addresses. 8



Change Addressing Mode Operation

Resume Normal Mode — RNM

RNM: The contents of instruction address register 1 and the interrupt register are interchanged, and a program branch to the new contents of instruction address register 1 (which controls program sequence) is performed. 8

Move Characters to Word Mark — MCW

MCW/A/B: Data and item marks are moved from the field indicated by A to the field indicated by B. 32

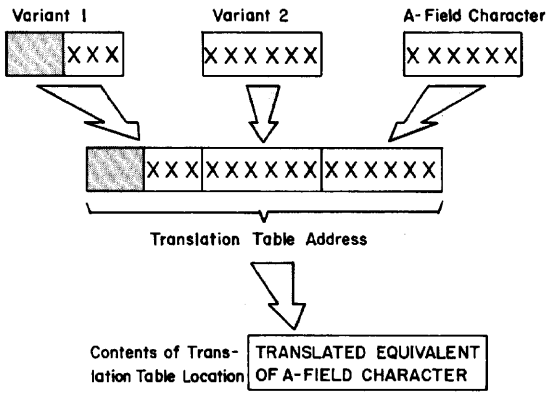
MCW/A: Data and item marks are moved from the field indicated by A to the field indicated by (R_b). 28

MCW: The data and item marks in the field indicated by (R_a) are moved to the field indicated by (R_b). 24

Move and Translate — MAT

MAT/A/B/V_1/V_2: The characters in the field indicated by A are successively translated according to the contents of a stored 46

translation table; their equivalents are placed in corresponding successive locations in the field indicated by B. The translation table address (binary) of an equivalent character is formed by placing together the binary equivalents of V_1, V_2, and the character to be translated, in that order.



Formation of Translation Table Address of Equivalent

This instruction finds particular application in the solution of code conversion problems. For example, it can be used to convert easily and efficiently between central processor codes and peripheral media codes having different configurations.

Extended Move — EXM

EXM/A/B/V: A specified amount of the data and/or the punctuation in the field indicated by A are moved to the field indicated by B. Different variant characters can be specified to control: 34

1. whether or not punctuation marks are moved;
2. the type(s) of punctuation moved;
3. whether or not data is moved;
4. the direction in which the source field is to be scanned; and
5. how much of the source field is to be scanned.

Load Characters to A-Field Mark — LCA

LCA/A/B: The data and punctuation in the field indicated by A are moved to the field indicated by B. 32

LCA/A: The data and punctuation in the field indicated by A are moved to the field indicated by (R_b). 28

LCA: The data and punctuation in the field indicated by (R_a) are moved to the field indicated by (R_b). 24

Store Control Registers — SCR

SCR/A/V: The value in the control register indicated by V is stored in the field indicated by A. V can be specified to indicate any of 13 control registers. 16

Variant Character	Control Register	Variant Character	Control Register
67	A-Address register	12	RWC2 — Starting location counter
70	B-address register		
77	Instruction address register 1	03	RWC3 — Current location counter
64	Instruction address register 2	13	RWC3 — Starting location counter
01	RWC1 — Current location counter	05	RWC1 ' — Current location counter
11	RWC1 — Starting location counter	15	RWC1 ' — Starting location counter
02	RWC2 — Current location counter	66	Interrupt register

Correspondence Between LCR/SCR Variant Characters and Control Registers

Load Control Registers — LCR

LCR/A/V: The value in the field indicated by A is moved to the control register indicated by V. V can be specified to indicate any of 13 control registers; if the instruction address register is specified, a program branch to the instruction at A is performed. 18

Edit Instructions

Move Characters and Suppress Zeros — MCS

MCS/A/B: The data in the field indicated by A is moved to the field indicated by B. Then, zeros and commas to the left of high-order numeric characters in the B field are suppressed. 52

Instruction Format | Execution Time

MCS/A: Zeros and commas to the left of the high-order numeric characters in the field indicated by A are suppressed. — 48

Move Characters and Edit — MCE

MCE/A/B: The contents of the field indicated by A are moved into replaceable character positions in the edit control word in the field indicated by B, and then the contents of the B field are edited to suppress unneeded credit and minus symbols, zeros, and commas, and to insert blanks, asterisks, and dollar signs where necessary. — 76 *

Input/Output Instructions

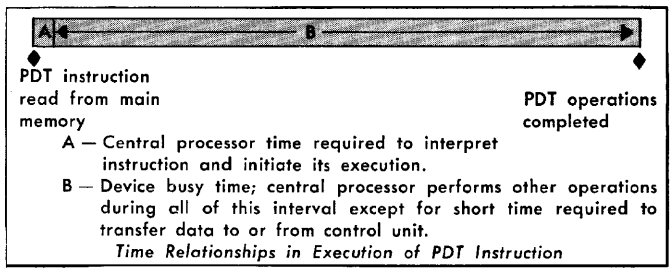
Effective control over data transfers between the central processor and peripheral units and over the peripheral units themselves is maintained by the use of two basic instructions: peripheral data transfer (PDT) and peripheral control and branch (PCB). The PDT instruction is used to initiate data transfer operations and certain other related operations, such as backspace magnetic tape and erase magnetic tape.

The PCB instruction performs two distinct functions: (1) it initiates strictly mechanical operations such as magnetic tape rewinds and card rejections; and (2) it causes a program branch to be performed contingent upon the setting of peripheral condition indicators. The latter facility allows programmed tests for such peripheral conditions as read or write errors, busy peripheral devices or control units, and magnetic tape unit at end of tape.

Peripheral Data Transfer — PDT

PDT/A/C_1/C_2/. . . . . /C_n: Data is transferred between the field indicated by A and the control unit and peripheral device indicated by control characters C_2 and C_3. Data transfer is in the direction (read or write) indicated by C_2 and over the read/write channel indicated by C_1. The number of control characters (n in the instruction format) varies with the device being addressed.

A write operation is terminated when all of the data in the output field has been transferred. A read operation is terminated when the memory input area has been filled or when a standard unit of information, such as a record or a card, has been read.



Time Relationships in Execution of PDT Instruction

The timing of a PDT instruction involves three considerations: (1) the time required by the central processor to interpret the instruction; (2) the time required by the central processor to transfer data to or from the control unit involved; and (3) the amount of time the peripheral device is busy. The time required to interpret the average (six-character) PDT

FUNCTION	UNIT OF DATA	CENTRAL PROCESSOR DATA TRANSFER TIME	DEVICE BUSY
Punch Card	80-col card	1.946 ms	240 ms
Read Card	80-col card	3.866 ms	75 ms
Punch Paper Tape	500-char record (6-level code)	2 ms	4.6 sec.
Read Paper Tape		2 ms	1 sec.
Write/Read/ Backspace Magnetic Tape	500-char record	2 ms 2 ms	13.3 ms (1) 11.2 ms (2)
Erase Magnetic Tape (½-in. units only)		none	Approx. 29.2 ms
	This operation erases the data from approximately 3½ inches of magnetic tape. No data is transferred.		
Skip Write Magnetic Tape (¾-in units only)	This operation moves tape forward until an optical window is sensed. It is used to move automatically from the beginning of tape label record to the point on tape where data is written.		
Search Drum and Read/ Write	500-char record	2 ms	30 ms avg.
Search Disc and Read/ Write	500-char record	2 ms	127.5 ms avg.
Print & Space	120-char record	13 ms	67 ms (3)
Space Paper on Printer	none	none	Approx. 49 ms for 5 lines
Transfer Data to/from Communication Control	100-char message (6-level char code)	0.02 ms	Varies with speed of remote device

(1) This time assumes a 204B-5 half-inch tape unit operating at a density of 556 char/in. and using a short gap; data transfer rate is 66,700 char/sec.

(2) This time assumes a 204A-3 tape unit (¾-in. tape) which transfers data at 88,800 char/sec.

(3) This time includes 20 ms required for post-print spacing of one line; only 7.2 ms are required for each additional line spaced.

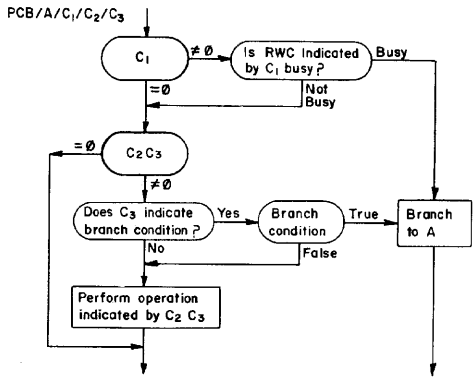
* This time assumes that A and B indicate six- and seven-character fields, respectively, that zero suppression scans four characters, and that a $ is floated left 2 places into the high-order character position.

instruction is 16 microseconds. The table on the previous page outlines the functions obtainable with the PDT instruction and provides representative central processor data transfer times and peripheral device busy times for each operation.

Peripheral Control and Branch PCB

PCB/A/C_1/. . . ./C_n: If C_1 specifies a read/write channel, branch to the instruction at A if the channel is busy. If the channel is not busy, or if no channel is specified, perform whatever operation(s) are specified by control character(s) following C_1. These operations may include performing a program branch contingent upon a peripheral condition, instructing a specified control unit to perform a particular operation, or setting a control unit to a particular mode of operation. The specified operations which can be initiated by the PCB instruction are outlined below.

1. Program branch operations – PCB instruction configurations are available to test for each of the conditions in the list below. If the condition exists, a program branch to the instruction at A is performed.
   - Read/write channel busy
   - Read/write channel busy
   - Device busy or inoperable
   - Illegal punch in last card read
   - Hole-count error in last card read or punched
   - Print error
   - Magnetic tape control unit read or write logic busy
   - Magnetic tape unit positioned at beginning or end of tape
   - Error in reading or writing magnetic tape
   - Parity error while reading paper tape
   - Parity or loss of transmission error detected by communication control
   - Communication control has received specified answer-back signal
   - Disc control unit detects read or write error
   - Drum control unit detects read or write error
2. Setting of control unit operation mode: PCB instruction configurations are available which direct the central processor to condition a particular control unit for operating in a specific mode. Such conditioning remains in effect until reset by a subsequent PCB instruction. The appropriate control units can be conditioned by



Peripheral Control and Branch Operation

use of the PCB instruction to operate as follows:
   - Read or punch cards in Hollerith code
   - Read or punch cards in direct transcription mode
   - Reject cards read or punched which contain hole-count errors
   - Generate busy signal if cards read or punched contain hole-count errors
   - Reject cards read which contain illegal punches
   - Check characters read from paper tape for odd or even parity, as specified
   - Read paper tape in a forward or reverse direction, as specified
   - Read or punch 5- or 6-level paper tape code
   - Read or punch 7- or 8-level paper tape code
3. Single-occurrence peripheral device operations: The PCB instruction is also used to direct a single performance of one of the following operations by a particular peripheral device:
   - Rewind magnetic tape
   - Rewind magnetic tape and release unit
   - Rewind paper tape (reader)
   - Runout paper tape (reader)
   - Communication control, start 300 ms answer-back timer
   - Communication control, send specified answer-back signal and start 300 ms timer

Central processor time required for executing a PCB instruction varies with the length of the particular instruction under discussion. Instruction length varies in turn with the type of operation specified, with A-address length, and with whether or not performance of the instruction includes a program branch. In any case, very little central processor time is required, the range being from 10 to 18 microseconds.

6 Tables

Instruction Formats and Timing

Each Honeywell 200 instruction is described in the following table in terms of its operation code, formats, and timing formulas. In addition, reference is made in each case to the page where the operations initiated by the instruction are described. The instructions are organized in the table by alphabetical order of mnemonic op code.

The formulas given in the table provide execution times in memory cycles. Such a time may be converted to microseconds by multiplying it by two.

Equivalent expressions for symbols used in the table are as follows:

SYMBOL	MEANING
A	Address of A-operand field.
B	Address of B-operand field.
V	Variant characters.
N_i	Number of characters in the instruction.
N_a	Number of characters in the field indicated by A.
N_b	The number of characters in the A- or B-operand field, whichever is shorter.

Arithmetic, Logic, Editing, and Control Instructions

MNEMONIC OP CODE	INSTRUCTION NAME	INSTRUCTION FORMAT(S)	TIMING FORMULA(S) (memory cycles)	DESCRIPTION PAGE
A	Decimal Add	A/A/B A/A A	$\frac{(A)\leq(B)}{N_i+2+N_w+2N_b}$ $\frac{(A)>(B)}{N_i+2+N_w+4N_b}$ N_i+2+3N_a $3+N_w+2N_b$ $3+N_w+4N_b$	27
B	Branch	B/A B/A/V	N_i+2	30
BA	Binary Add	BA/A/B BA/A BA	$N_i+1+N_w+2N_b$ N_i+1+3N_a $2+N_w+2N_b$	28
BCC	Branch on Character Condition	BCC/A/B/V BCC	N_i+4	30
* BCE	Branch if Character Equal	BCE/A/B/V BCE	N_i+4	29
BS	Binary Subtract	BS/A/B BS/A BS	$N_i+1+N_w+2N_b$ N_i+1+3N_a $2+N_w+2N_b$	28
C	Compare	C/A/B C/A C	$N_i+2+N_w+N_b$	29
* CAM	Change Addressing Mode	CAM/V	4	31
CI	Clear Item Mark	CI/A/B CI/A CI	N_i+3	31
* CSM	Change Sequencing Mode	CSM	4	31

* Optional advanced programming instruction.

MNEMONIC OP CODE	INSTRUCTION NAME	INSTRUCTION FORMAT(S)	TIMING FORMULA(S) (memory cycles)	DESCRIPTION PAGE
CW	Clear Word Mark	CW/A/B CW/A CW	N_i+3	30
EX	Extract	EX/A/B EX/A EX	N_i+1+3N_w	29
* EXM	Extended Move	EXM/A/B/V	N_i+1+2N_a	32
H	Halt	H H/A H/A/B	N_i+2	31
HA	Half Add	HA/A/B HA/A HA	N_i+1+3N_w	29
LCA	Load Characters to Word Mark	LCA/A/B LCA/A LCA	N_i+1+2N_a	32
* LCR	Load Control Registers	LCR/A/V	N_i+5	32
MAT	Move and Translate	MAT/A/B/V_1/V_2	N_i+1+3N_t (1)	31
* MCE	Move Characters and Edit	MCE/A/B	$N_i+1+N_a+2N_b+2X+2Y$ (2)	33
* MCS	Move Characters and Suppress Zeros	MCS/A/B	N_i+1+4N_a	32
MCW	Move Characters to Word Mark	MCW/A/B MCW/A MCW	N_i+1+2N_w	31
NOP	No Operation	NOP	3	31
* RNM	Resume Normal Mode	RNM	4	31
S	Decimal Subtract	S/A/B S/A S	$\frac{(A)\geq(B)}{N_i+2+N_w+2N_b}$ $\frac{(A)<(B)}{N_i+2+N_w+4N_b}$ N_i+2+3N_a $3+N_w+2N_b$ $3+N_w+4N_b$	28
SCR	Store Control Registers	SCR/A/V	2-char. addresses: N_i+4 3-char. addresses: N_i+5	32
SI	Set Item Mark	SI/A/B SI/A SI	N_i+3	30
SST	Substitute	SST/A/B/V SST	N_i+4	29
SW	Set Word Mark	SW/A/B SW/A SW	N_i+3	30
* ZA	Zero and Add	ZA/A/B ZA/A ZA	$N_i+2+N_w+2N_b$ N_i+1+3N_a $3+N_w+2N_b$	28
* ZS	Zero and Subtract	ZS/A/B ZS/A ZS	$N_i+2+N_w+2N_b$ N_i+1+3N_a $3+N_w+2N_b$	28

* Optional advanced programming instruction.

(1) N_t: total no. of characters transferred from translation table to field specified by B.

(2) X: no. characters scanned during zero suppression; Y: no. characters scanned during dollar sign insertion.

Input/Output Instructions

MNEMONIC OP CODE	INSTRUCTION NAME	INSTRUCTION FORMAT	TIMING FORMULA(S) (memory cycles)	DESCRIPTION PAGE
PCB	Peripheral Control and Branch	PCB/A/C_1/ . . . /C_n (1)	No branch: N_i+1 Branch: N_i+2	34
PDT	Peripheral Data Transfer	PDT/A/C_1/C_2/ . . . /C_n (1)	N_i+2 (time required to interpret instruction)	33

PDT Function	Central Processor Time Required for Data Transfer (2)	Time Device Busy (3)
Read Card	1933	75
Punch Card	973	240
Read Paper Tape	1 per 5- or 6-level char.	2 per frame
Punch Paper Tape	2 per 7- or 8-level char.	9.1 per frame
Read Magnetic Tape Write Magnetic Tape	1 per char.	$\frac{T_{cg}+N_c}{R_t}$ (4)
Search Drum and Read Search Drum and Write	1 per char.	Avg.:25+0.01 per 6-bit char.
Search Disc and Read Search Disc and Write	1 per char.	Avg.:118+0.019 per 6-bit char.
Print and Space	13 per 120-char. line	67 ms per 120-char. line
Transfer Data to or from Communication Control	1 per 5- or 6-level char. 2 per 7- or 8-level char.	Varies with speed of remote device.

(1) No. of control characters depends upon type of device being addressed.
(2) Central processor times are given in memory cycles (2 microseconds each) except where other units are indicated.
(3) Device times are given in milliseconds.
(4) T_{cg}: cross-gap time of unit being used (see page 17); N_c : no. of characters in record transferred; R_t : transfer rate of unit being used (see page 17).

Correspondence Between H-200 Central Processor, Card, and Printer Codes

Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer	Key Punch	Card Code	Central Processor Code	Octal	High Speed Printer
0	0	000000	00	0	—	X, 0 or X (1)	100000	40	—
1	1	000001	01	1	J	X, 1	100001	41	J
2	2	000010	02	2	K	X, 2	100010	42	K
3	3	000011	03	3	L	X, 3	100011	43	L
4	4	000100	04	4	M	X, 4	100100	44	M
5	5	000101	05	5	N	X, 5	100101	45	N
6	6	000110	06	6	O	X, 6	100110	46	O
7	7	000111	07	7	P	X, 7	100111	47	P
8	8	001000	10	8	Q	X, 8	101000	50	Q
9	9	001001	11	9	R	X, 9	101001	51	R
	8, 2	001010	12	'		X, 8, 2	101010	52	#
#	8, 3	001011	13	=	$	X, 8, 3	101011	53	$
@	8, 4	001100	14	:	*	X, 8, 4	101100	54	*
Space	Blank	001101	15	Blank		X, 8, 5	101101	55	"
	8, 6	001110	16	Blank (2)		X, 8, 6	101110	56	Blank (2)
	8, 7	001111	17	&		X or X, 0 (1)	101111	57	Blank (2)
&	R, 0 or R (1)	010000	20	+		8, 5	110000	60	Blank (2)
A	R, 1	010001	21	A	/	0, 1	110001	61	/
B	R, 2	010010	22	B	S	0, 2	110010	62	S
C	R, 3	010011	23	C	T	0, 3	110011	63	T
D	R, 4	010100	24	D	U	0, 4	110100	64	U
E	R, 5	010101	25	E	V	0, 5	110101	65	V
F	R, 6	010110	26	F	W	0, 6	110110	66	W
G	R, 7	010111	27	G	X	0, 7	110111	67	X
H	R, 8	011000	30	H	Y	0, 8	111000	70	Y
I	R, 9	011001	31	I	Z	0, 9	111001	71	Z
	R, 8, 2	011010	32	;		0, 8, 2	111010	72	@
•	R, 8, 3	011011	33	.	,	0, 8, 3	111011	73	,
□	R, 8, 4	011100	34	)	%	0, 8, 4	111100	74	(
	R, 8, 5	011101	35	%		0, 8, 5	111101	75	C_R
	R, 8, 6	011110	36	□		0, 8, 6	111110	76	Blank (2)
	R or R, 0 (1)	011111	37	Blank (2)		0, 8, 7	111111	77	Blank (2)

(1) This card code-central processor code equivalency available as an option.
(2) Symbol which will be printed by otherwise non-standard printer bit configuration.

HONEYWELL
ELECTRONIC
DATA
PROCESSING
WELLESLEY HILLS 81,
MASSACHUSETTS