THE TRW-230 STORED LOGIC MULTIPLE PURPOSE COMPUTER SYSTEM

AN INTRODUCTION TO THE TRW-230 COMPUTER SYSTEM

TRW COMPUTER DIVISION

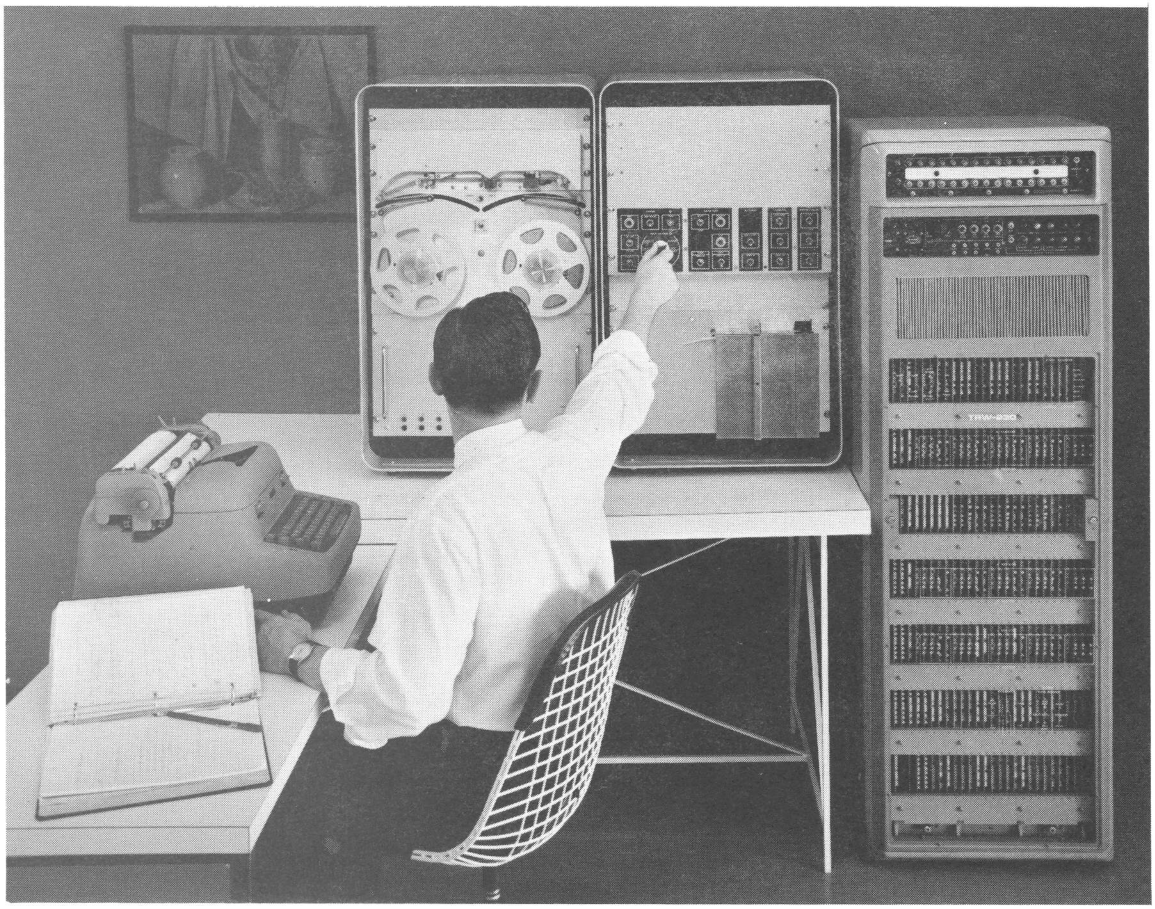
THE TRW-230 STORED LOGIC MULTIPLE PURPOSE COMPUTER SYSTEM

AN INTRODUCTION TO THE TRW-230 COMPUTER SYSTEM

TRW COMPUTER DIVISION
THOMPSON RAMO WOOLDRIDGE INC.
8433 Fallbrook Avenue • Canoga Park, California

M 230-3U3 REV. JULY 63

CONTENTS



The TRW-230 Computer With Basic Input-Output Equipment

1 A "STORED LOGIC" MULTIPLE PURPOSE COMPUTER

The TRW-230 is a binary, stored program digital computer, designed to operate with extreme reliability, and to be efficient at a great many different tasks—including those involving real-time control and on-line data processing. It is a solid-state, parallel, random access core memory machine, with powerful and extensive logical capabilities, that combines many attractive qualities with an unusually low price.

It offers such performance characteristics as:

6-microsecond memory cycle time
8,192-word basic main frame memory, expandable to 32,768 words
15-bit input channel and 15-bit output channel with 5 control lines
15-bit parallel information transfers
11 real-time interrupt lines

together with such physical features as:

Exceptionally high shock and vibration resistance
Wide ambient temperature range (32° to 122°F)
Compactness (59 by 20 by 16 inches) and transportability

and such operational conveniences as:

Simple installation
Wide range of peripheral equipment
Easy-to-operate control and maintenance panels

all at low cost and with prompt delivery.

Chief reason for this combination of qualities at the low cost is the use of an advanced design technique: Stored Logic.

In the Stored Logic technique, much of the control logic that is normally wired into a computer is instead stored as part of the program in core memory. Stored Logic affects both hardware and software—hardware, in that it eliminates the need for many circuit components, thereby increasing reliability and reducing size, weight, and cost of the computer; software, in that it enables the programmer to specify whatever logical organization (instruction repertoire, order structure, multiple-word length efficiency) best fits the processing problem at hand. Stored Logic provides an exceptionally large number of instruction combinations that can be used in either machine or symbolic language, and it gives the TRW-230 so many application capabilities that it becomes in effect a multiple-purpose rather than a general or special purpose, computer.

As used in the TRW-230, the essence of Stored Logic consists of a sophisticated minimum of hardware functioning through three levels of instructions. These are:

1. MICROCOMMANDS. The TRW-230 has six registers, plus an adder and an address counter, between which it can cause many intricate shifts and transfers of words. Every three microseconds, on clock pulse, a number of such actions, or microcommands, occurs simultaneously. Although this is the only type of instruction wired into the machine, no one need actually write single microcommands. The lowest level of written instructions is that of a combination of microcommands, the logand.

2. LOGANDS. "Logand," a contraction of "logical command," refers to TRW-230 machine-language instructions. A logand, which is above the level of microcommands and below that of conventional computer instructions, occupies one cell in memory and consists of three fields, reserved respectively for (1) a primary command, (2) an addressing option, and (3) either an address, a secondary command, parameter, or condition. It is according to the contents of these fields that the logand activates appropriate combinations of microcommands.

3. LOGRAMS. Written logands are grouped into symbolic macro-instructions called lograms. (The word "logram" is a contraction of "logical program.") Lograms cover both relatively simple operations such as multiplication, and more complex ones such as extraction of square root or rectangular-to-polar coordinate transformation. Lograms thus correspond to the subroutines as well as to the wired instructions of other computers. But because they are tailor-made from individually selected logands, which in turn comprise the most economical microcommands, they involve none of the wasteful hardware operations and duplicate instructions associated with task changes in other computers of comparable size.

Thus generic portions of instructions are assembled into programming blocks of increasing complexity, and generally only the highest level of these need concern the user, since TRW supplies a logram library. The programmer merely writes calling sequences for the existing lograms, much as he would write symbolic instructions for a conventional machine.

It is thus largely because of its Stored Logic feature that the TRW-230 offers more performance, versatility, and user convenience than could ordinarily be expected of an inexpensive computer. And to allow the fullest use of these capabilities, particularly in real-time applications, the TRW-230 Computer System contains a varied and highly flexible array of asynchronously functioning input-output equipment.

Available with each TRW-230 system are spare parts and tools; maintenance, installation checkout, and programming services; and maintenance and operation courses. Free programming instruction classes are held several times a year at TRW's Canoga Park site.

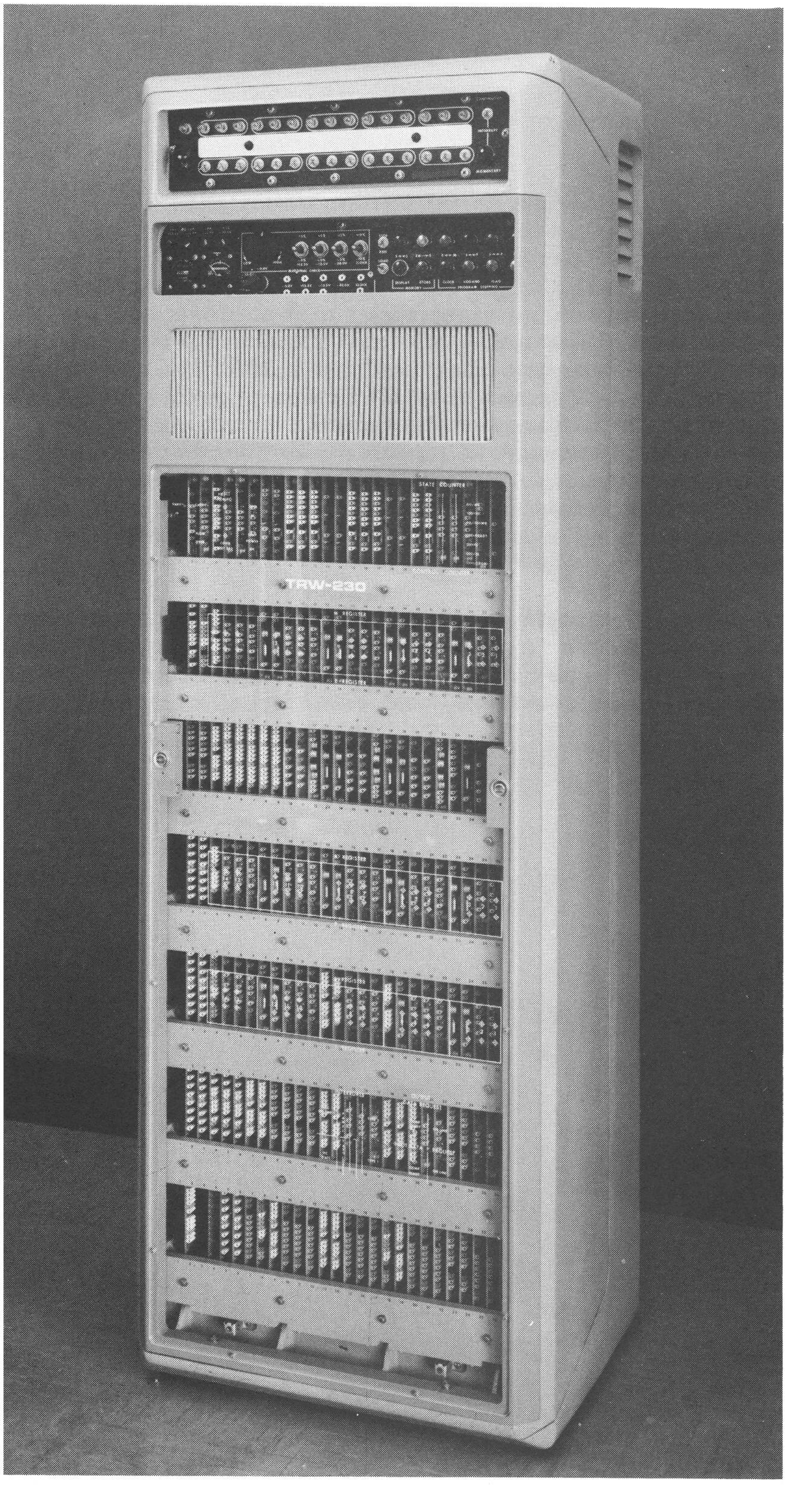


Figure 1. The TRW-230 Computer

2 COMPUTER HARDWARE

The TRW-230 Digital Computer was designed and packaged to facilitate operation and maintenance, and to insure reliability under prolonged adverse environmental conditions. The computer is housed in a rugged two-piece aluminum casting, and its design provides for shock, vibration, drip-proofing, and operation in ambient temperatures of 0° to $122^{\circ}F$. All circuitry is solid-state. Basic module for computer circuitry is the etched circuit card, held rigidly by precision side-guides and hold-down bars. Power requirements are less than 600 watts of 118 volts ac, at 60 or 400 cycles.

In the interests of operating and maintenance ease, all computer elements are accessible from the front, all active circuits have color-coded test points, and all flip-flops are provided with indicator lights. Air filters have been placed near eye level, and cable access is convenient.

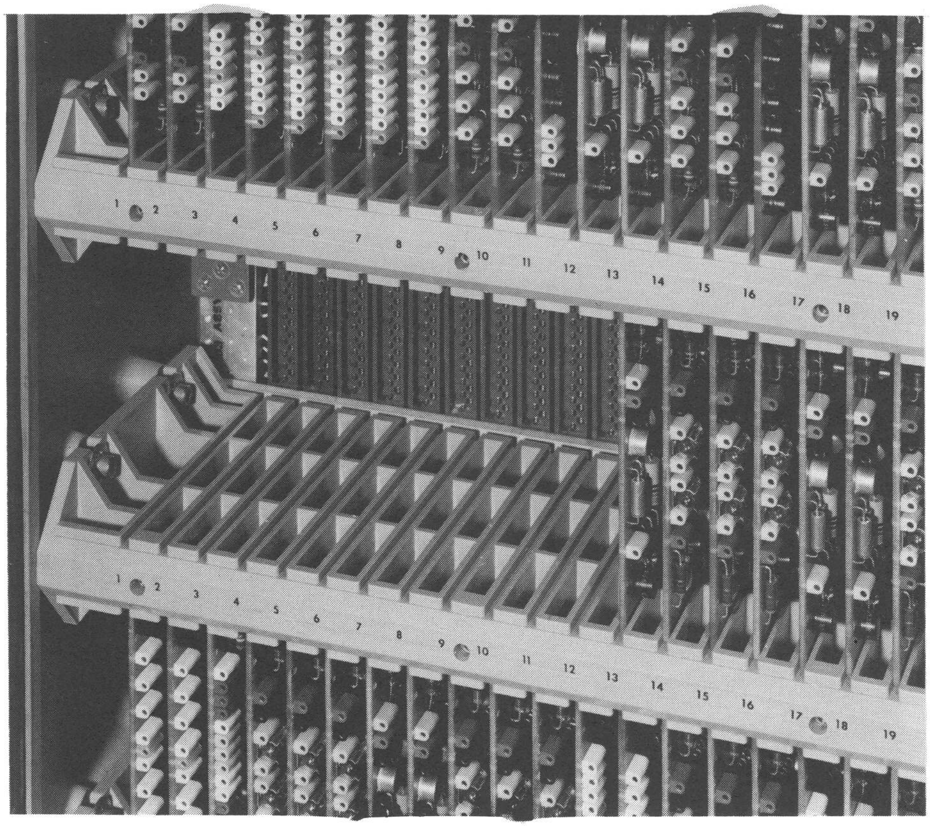


Figure 2. Circuit Cards with Test Points and Indicator Lights.

CONTROL PANELS

The maintenance and control panels (Figure 3) are particularly simple and convenient to operate.

The top control panel contains the following features: 15 neon indicator lights display each binary position in the A register, and a toggle switch beneath each light permits the operator to set the corresponding digit position to a ONE (up) or a ZERO (down). A frosted plastic strip below the lights enables him to write down identification for particular bit groupings temporarily. The "On-Off" switch controls AC power for all power supplies and fans. The "Momentary" interrupt button and "Continuous" interrupt switch make it possible to interrupt the program being executed and have the computer branch to whatever dump or other routine has been stored in the appropriate memory locations. The "Momentary" interrupt button will cause this action to occur whenever the button is pressed; the "Continuous" interrupt switch, once set, will cause it to occur after the execution of each logand not contained in the interrupt routine.

The maintenance panel, just below the control panel, contains twelve buttons and two toggle switches on the right side to provide operator and maintenance engineer with direct control over the contents of all computer registers. The left side contains the test inputs and outputs and the hardware control features needed for maintenance and diagnosis of equipment.

The controls on the right side of the maintenance panel combined with the control panel permit manual loading of all registers and memory cells.

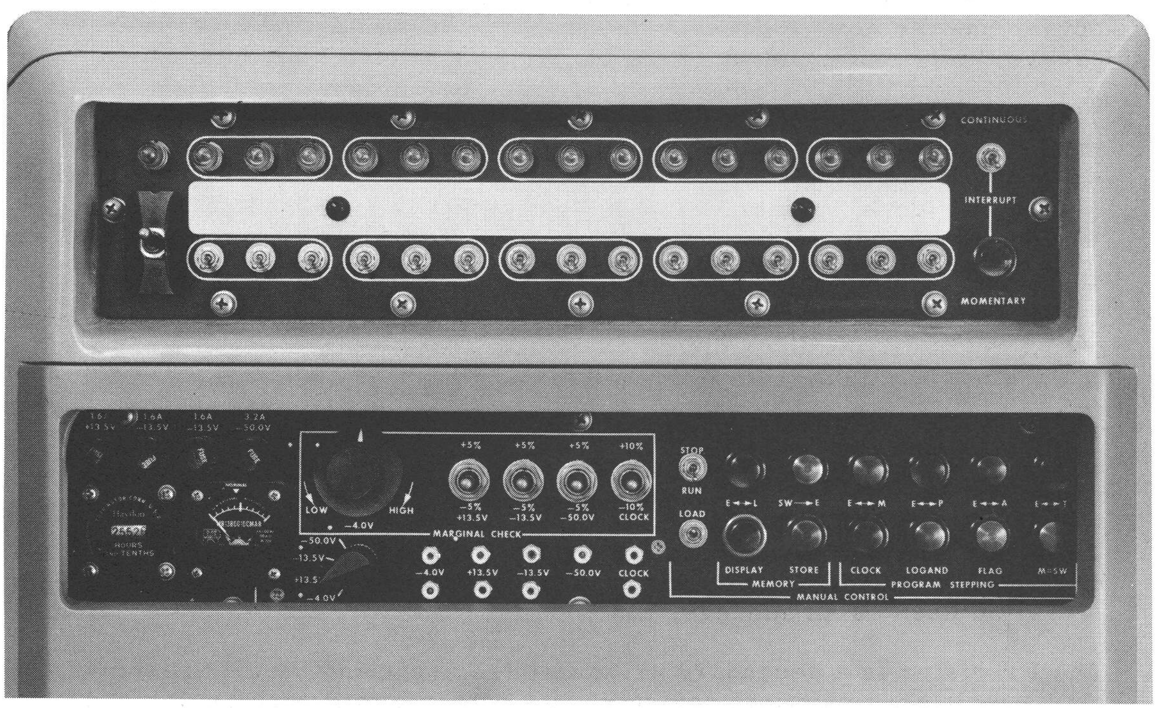


Figure 3. Control and Maintenance Panels

WORD SIZE AND MEMORY

The TRW-230 is a binary parallel word machine, using two's complement arithmetic. Each word contains fifteen bits and can thus contain any unsigned number between 0 and 32,767. Since its internal code is binary, the machine can be programmed as if a word contained virtually any type of information. For example, a word may be considered to contain a basic machine instruction (logand), the binary-coded equivalent of two alphanumeric characters, a signed numeric quantity between -16,384 and +16,383, a memory address, or an arbitrarily-coded bit pattern representing the status of physical control devices.

When the computer is programmed in symbolic language (with lograms), TRW-230 Stored Logic permits efficient use of any word length in multiples of 15 bits.

The computer's basic internal memory capacity consists of an 8,192 15-bit word magnetic core unit, which can be expanded to a maximum of 32,768 words. (Additional core memory units come in 8K, 16K, and 24K sizes, externally packaged.) Read and restore take 3 microseconds each, thus resulting in a 6-microsecond memory cycle.

LOGICAL ORGANIZATION

The TRW-230's logical organization is shown in Figure 4. The six 15-bit registers have deliberately been given noncommittal names because few of them have distinctive functions. (Most TRW-230 instructions do not include memory addresses, but rather addressing options that reference the preset contents of the M, A, or P register as an address.)

The M register controls all memory accesses except those made through the L register to scratchpad addresses 0-63. Before A or P are used as an address, their contents are exchanged with M, and after use M is always restored. In this way M is used to access operands, and also serves as the instruction counter. For the latter purpose M is always incremented at least once during each instruction.

The E register is the memory exchange register, and is also an input-output register. It holds operands during iterative instructions and does a few other odd jobs from time to time.

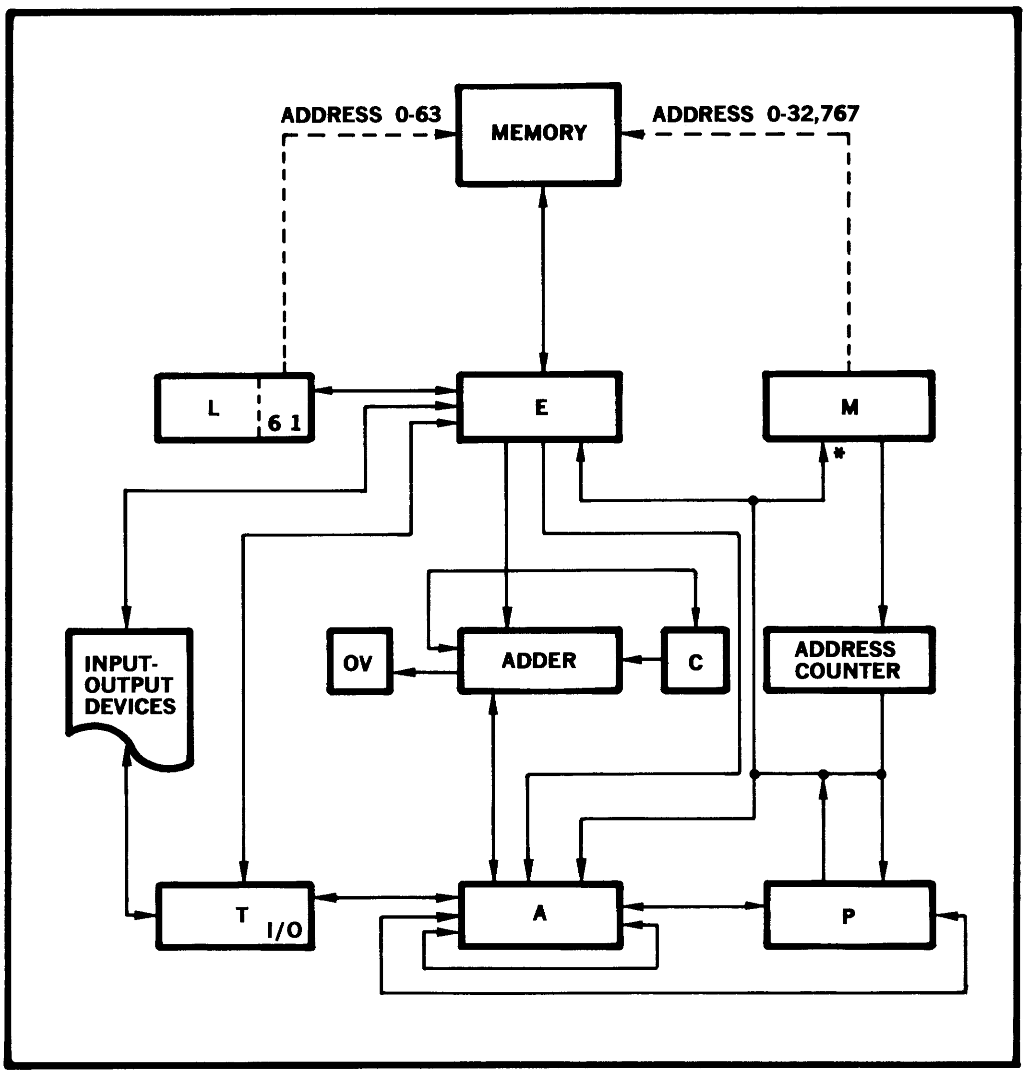
The L register holds instructions during execution, holds operands temporarily between instructions on occasion, and may be used to address the 64 words of scratchpad memory.

The A register is the principal arithmetic register and resembles a conventional accumulator under most circumstances. The contents of A may be used as an address, also.

The Pregister is a secondary arithmetic register and usually resembles a multiplier-quotient register. Its contents may also be used as an address, and frequently P is employed to control a program sequence in the interpretive mode.

The T register is chiefly used as an output buffer for slow-speed devices, but when not needed for this purpose, it can be used to store intermediate operands from A or E.

The other registers are a parallel full-adder, a parallel half-adder, an overflow indicator and a carry flip-flop. All transfers from the M register pass through the address counter, where the previous contents of M may be incremented by unity. The programmer controls this function except when the instruction address is incremented.



* ONE-WAY PATH

Figure 4. Logical Organization

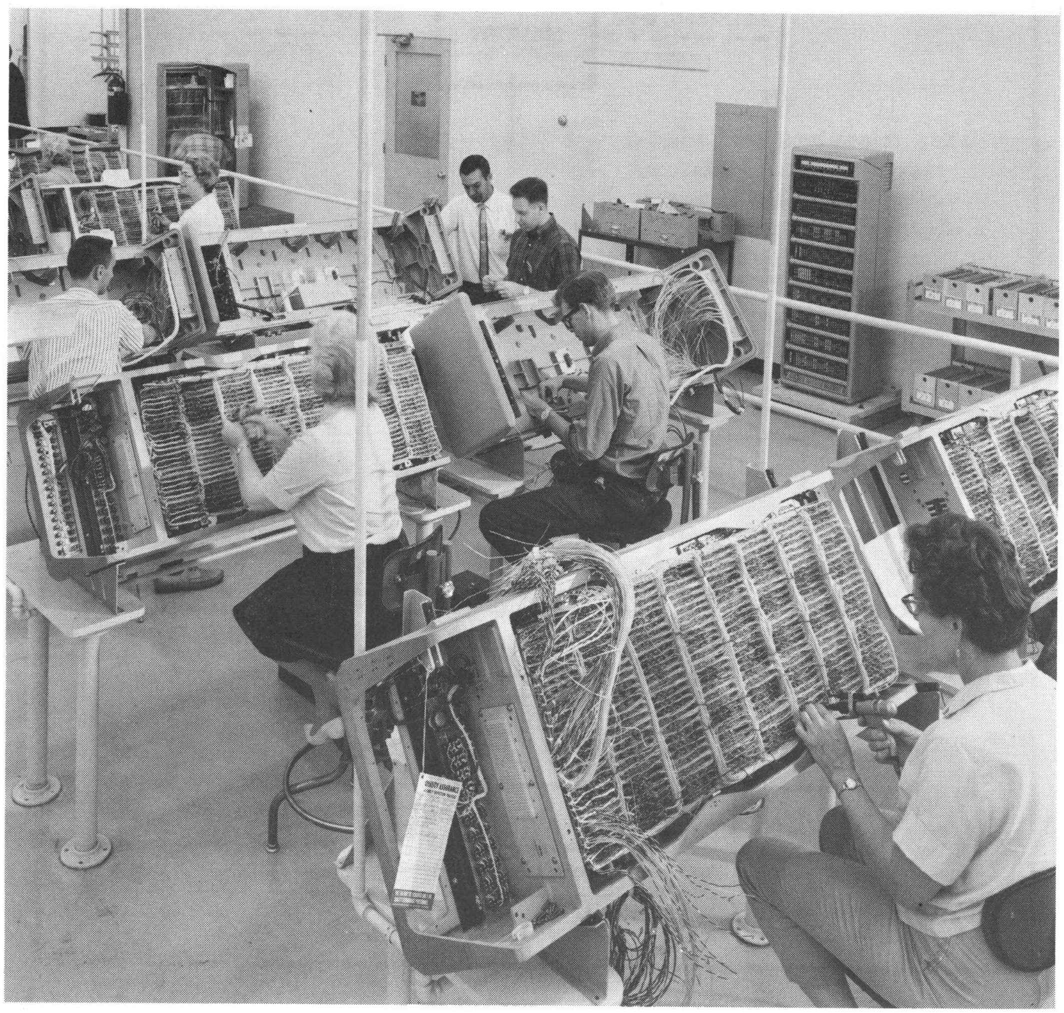


Figure 5. TRW-230 Compters Being Assembled

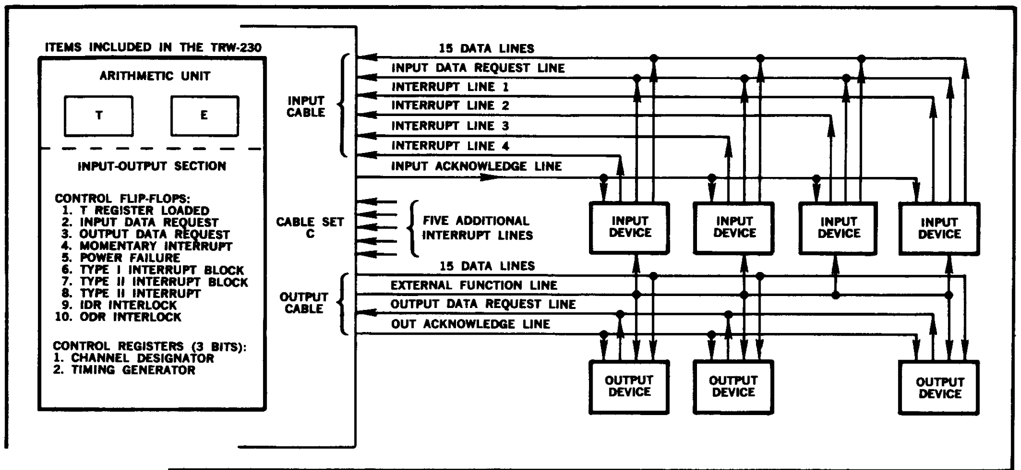


Figure 6. Input-Output Connections

All communications between the computer and external devices, including the highly efficient transmission of interrupt signals according to given priorities, are controlled by the computer's internal input-output section.

The input-output section communicates with the external devices through Cable Set C, which contains 15-bit parallel input and output data lines. Cable Set C transfers 15 bits of data and is used for standard peripheral devices such as typewriter, paper-tape reader, paper-tape punch, teletype, and so on.

The cable set also contains control lines as well as data lines. The output cable contains an Output Data Request Line, an Output Acknowledge Line, and an External Function Line; the input cable, an Input Data Request Line, an Input Acknowledge Line, and nine interrupt lines. These nine plus the two lines that generate interrupts for input and output data requests make a total of eleven external interrupt lines available to the programmer.

Cable set C provides buffered output through the T register.

Special input-output logands facilitate selection of a cable, selection of one of a number of devices on the cable, and effecting of transfers in either word or block form.

Interrupts

The interrupt system controlled by the input-output section offers a number of special advantages:

1. It enables the computer to interleave internal data processing, at its own high speed, with data transfers to or from slower-speed peripheral devices at their maximum communicable rates—thus providing

asynchronous I/O and eliminating computer waiting time.

2. It is hardware-implemented and therefore eliminates the programming overhead that would be required by the testing for programmed interrupts.

3. It consists of several types of interrupts, each with its own service-routine address, and therefore can be used with a system of priorities — again, without additional programming.

4. It is highly flexible and expandable, permitting use not only of varying sets of external devices, but of varying roles for each device (because the reason for interrupting is transmitted to the computer along with the interrupt).

The different types of interrupt are:

Type I: An interrupt caused by a peripheral device whenever it is ready to read (or write) the next word of an output (or input) message. It is generated over an input or output data request line by a signal from a device that has already been connected to the data lines by computer action. This arrangement, by eliminating computer waiting time, permits fast response to the randomly reading or writing device.

Type II. A more conventional type of interrupt, caused usually by a peripheral device signalling to the computer to be connected to an input or output data line. The device generates the interrupt over any of nine external interrupt lines. The Type II interrupt can also be generated by either of two interrupt switches on the computer control panel over two internal interrupt lines, and by programmed interrupts.

System priorities for the interrupts are:

1. Type I - Input Data Request from a transmitting device

2. Type I - Output Data Request from a receiving device

3. Type II - Any of the 9 interrupt lines, the 2 interrupt switches, or a programmed interrupt

Both Type I and Type II interrupts inhibit other interrupts of the same type. Also, they are inhibited by certain input-output and control logands.

The interrupts automatically generate an Interrupt Logand that transfers control to unique, programmed subroutines.

All types of interrupts may be generated by computer programming, primarily for the purpose of inhibiting other interrupts of the same type, or for testing.

Transfer Rates

Data transfer rates for the TRW-230 are as follows:

Cable	Minimum Interval (μsec)	Maximum Rate (15-bit words/sec	Maximum Rate (bits/sec)
C "IN"	30	33,333	499,995
C "OUT"	33	30,303	454,545

Power-Failure Control

In the event of power failure, a power-failure control circuit generates a signal that prevents operations from ceasing haphazardly. Its effect is to first let the computer complete the current read-write cycle, thus permitting core memory to be restored, and then to halt computer operations.

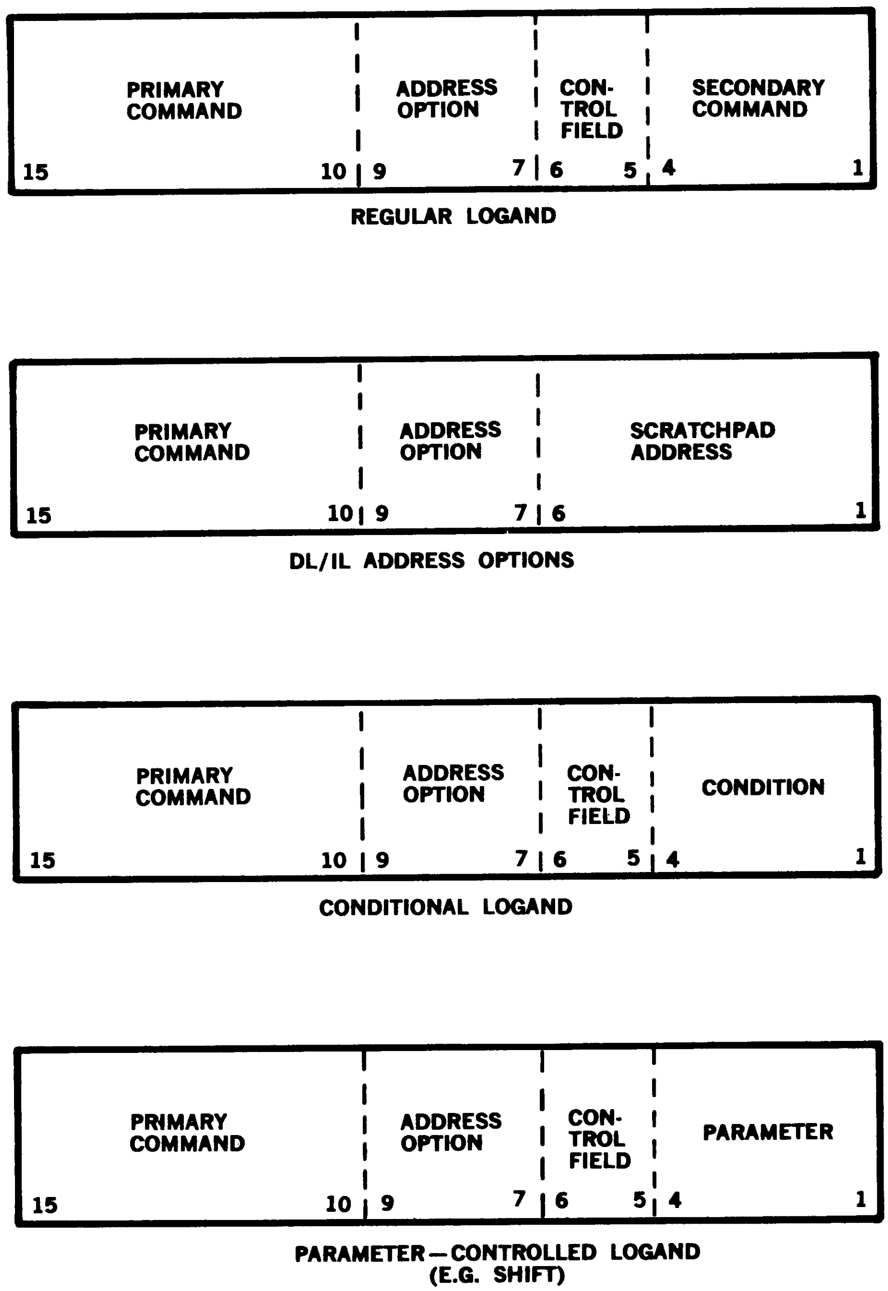


Figure 7. Logand Format

3 PROGRAMMING

MACHINE-LANGUAGE PROGRAMMING

Although many users will be more concerned with the lograms that constitute the TRW-230's symbolic language than the logands that comprise its written machine language, the latter will help them to understand logand structure and its relation to computer organization.

The TRW-230 has 82 microcommands with 8500 unique (logand) combinations. Fortunately, the logand structure is such that even those working exclusively with logands need not recall each as an entity, but can use fairly simple rules to construct the logand they require. Inherent in the tremendous variety available, though, is unparalleled flexibility. Incidental functions can be performed along with primary functions, all in a single instruction (logand) cycle. For example: "read memory, perform a complement for, addition, and an address modification" could be a single instruction; "read memory, extract a word with a mask, modify memory, retain extracted bits, perform an addition or complementation, and also an address modification," could be another single instruction.

Logand formats are shown in Figure 7.

The most common type is the regular logand, which appears at the top of the figure. It contains a 6-bit field for the primary command, a 3-bit field for the addressing option, a 2-bit control field, and a 4-bit field for the secondary command.

There are 32 regular primary commands (such as Load, Store, Add, etc.) 16 of which may also be used as secondary commands. They control register transfers and operations like add and the logical functions. In this format there are six addressing options, direct (D) or indirect (I) from A, P, or M. The control option determines whether or not the address used should be incremented by unity and whether or not a memory access is made.

The regular command list, when used with the DL and IL address options (second format), is analogous to the conventional single-address instruction format, with the address contained in the instruction word.

The special logands (third and fourth formats) include the conditional control functions, shifting, multiplication, division, input-output, and so forth, including table search and match functions.

Examples of special logands that manipulate multiword blocks are:

Block Transfer. Transfers a block of data from one location to another within memory.

Block Input-Output. Transfers a block of data between memory and an external device. (The length of the transfer is specified by the programmer.)

Table Search. Conditionally compares a computer word sequentially with the words of a block. The program branches when the condition is met or the end of the sequence is reached. This logand is very powerful for sorting and merging operations.

Match. Conditionally compares two sequences of computer words. The program branches when the condition is met. This logand is especially useful for comparing a set of results against upper or lower limits or for search for a multiword key.

Sort. Compares a sequence of words in a block with a word in memory. Also compares each word in the sequence with the succeeding word in the sequence.

These special logands, and the method of their control, make the TRW-230 faster and easier in complete data processing tasks than many more expensive machines.

Logand execution times are as follows:

Regular logands* Conditional branch and skip logands Single-word, input-output logands	12μsec or 18μsec
Unconditional shift logands	12μsec plus 3μsec times number of shifts or counts
Interrupt logands Terminate interrupt logands	18μsec
Special logands involving more than one data word (block transfer or I/O, match, table search, sort, etc.)	18μsec plus 12μsec times number of word elements

A list of internal machine-language commands appears at the end of this section.

* 12μsec for D-, 18μsec for I-addressing options.

Examples of regular logands are:

LA/DL/36	(load into A the contents of cell 36)
SP/IL/7	(store the contents of P in the cell at the address given in cell 7)
LT/DP/C/NØ	(load into T the operand from the address found in P; control field normal; no operation as secondary command)

Since the M-register acts as a logand-address counter, the next logand in sequence is always taken from the location given by the M-register. Normally, consecutive logands are thus taken from locations in numerical sequence. The loading of the M-register, as it pertains to the logand sequence, can be accomplished with selected primary commands and address options to effect a transfer of control to any desired location in memory. This feature, along with the automatic incrementing of the contents of the P-register under address-option control, is the key to the TRW-230's comparatively simple and rapid linkage ability. Logands lead into logands in this manner, and sequences of logands lead into other sequences of logands.

SYMBOLIC-LANGUAGE PROGRAMMING

The user's programmer usually prepares an operational program simply by writing calling sequences for lograms, as if he were writing symbolic instructions for another machine. He is not concerned with the sequencing of logands, either within a logram or from one logram to the next, since this is taken care of by the self-incrementing pseudo-address counter. The TRW SYMBOLIC ASSEMBLER assembles the program by translating the calling sequences into machine language, then selecting the required lograms from a logram library. The Assembler outputs the translated program and, if desired, a program loader. Switch settings on the operator's console select the output medium to be used: paper tape, magnetic tape, or cards. The assembled program and lograms are in relocatable format so they can be positioned any place in memory prior to program operation.

The division of the stored program into two parts—the job program consisting of calling sequences and data addresses, and the logram package consisting of collections of logands—makes the TRW-230 Stored Logic system interpretive. In a conventional interpretive system, however, an executive routine is interposed between the instructions written by the programmer and the subroutines that actually execute them. It functions to preserve the sequence of interpretive operations, to locate data and storage locations for the interpretive subroutines, and to perform the many transfers of control typically necessary. The logram on the other hand, combines both executive and interpretive elements in a single and highly efficient subroutine by utilizing the stored logic flexbility inherent in the TRW-230. Thus TRW-230 interpretive language programming imposes no serious housekeeping penalties.

A program employing TRW-INTERPRETER language may be demonstrated by the following example:

Evaluate the expression

$$\text{VALUE} = \frac{\sqrt{\text{QUICK} + \text{GAMMA}^2}}{\text{SMALL}}$$

where QUICK, GAMMA, and SMALL are names of floating point values stored in computer memory.

The program may be written as shown below.

	Interpretive Instruction		Meaning
	ORG	1000	Set origin "BEGIN" to location 1000
BEGIN	LP/DM/C/NØ		Standard logands for entering Interpretive Mode
	PZE	*+2	
	LP/DP/C/NØ		
	PZE	LDF	Load Floating Point Accumulator
		GAMMA	"FPA" with C(contents of) (GAMMA)
	PZE	MPF	C(FPA) x C(GAMMA)
		GAMMA	Replace C(FPA)
	PZE	ADF	C(FPA) + C(QUICK)
		QUICK	Replace C(FPA)
	PZE	SRF	Sq. root of C(FPA) replace C(FPA)
	PZE	DVF	C(FPA) ÷ C(SMALL)
		SMALL	Replace C(FPA)
	PZE	STF	C(FPA) are stored into
		VALUE	memory location VALUE
	PZE	*+1	Exit from Interpretive Mode
	BR/DM/F/UN		Flag stop HALT

COMPILER LANGUAGE PROGRAMMING

The TRW FORTRAN COMPILER makes it possible to write programs using a higher level language than either machine or interpretive language. Programs written for the compiler take the form of English language statements that approximate mathematical notation. These statements, for the most part, specify only the method of solution; the task of translating them into step-by-step computer operations is performed by the compiler. Non-compiler programming, by contrast, requires that the programmer decide the method and also devise the detailed sequence of operations for effecting the solution.

With TRW FORTRAN it is possible to express problems in an easily learned format without having to understand all the details of computer programming and operation. A FORTRAN program for evaluating the expression in the preceding example

$$\text{VALUE} = \frac{\sqrt{\text{QUICK} + \text{GAMMA}^2}}{\text{SMALL}}$$

consists of a simple statement to the compiler as follows:

```
VALUE = SQRTF (QUICK + GAMMA **2)/SMALL
```

Of course, more complex operations require lengthier programs. Even so, such programs are much easier to write and shorter than their ordinary symbolic language counterparts which may require the coding of hundreds of separate operations.

The following example shows a FORTRAN program for performing a quite complicated task. As explained in the program's initial "comment" statements, the program calculates a product matrix from matrices X and Y, both being 5 by 5 arrays. It employs a subroutine "PRODM" which evaluates the product matrix according to the expression

$$C_{ij} = \sum_{k=1}^{\ell} A_{ik} B_{kj}$$

The statements for the program and the subroutine are followed by an explanation of each statement's function.

<u>MAIN PROGRAM</u>

	Statement Name		Statement
(1)	C		PROGRAM TO CALCULATE THE MATRIX PRODUCT,
(2)	C		Z, OF X(5x5), AND Y(5x5) USING FORTRAN
(3)	C		SUBROUTINE PRODM
(4)			DIMENSION X(10, 10), Y(10, 10), Z(10, 10)
(5)			READ INPUT TAPE 5, 20, ((X(I, K), I=1, 5), K=1, 5),
(6)		1	((Y(K, J), K=1, 5), J=1, 5)
(7)	20		FORMAT (1X, 5E 14. 6)
(8)			CALL PRODM (X, Y, Z, 5, 5, 5)

MAIN PROGRAM (Continued)

	Statement Name	Statement
(9)		WRITE OUTPUT TAPE 6, 20, ((Z(I, J), I=1, 5), J=1, 5)
(10)		CALL EXIT
(11)		END

SUBROUTINE

	Statement Name	Statement
(12)	C	MATRIX MULTIPLY SUBROUTINE PRODM
(13)		SUBROUTINE PRODM (A, B, C, N, L, M)
(14)		DIMENSION A(10, 10), B(10, 10), C(10, 10)
(15)		DO 102I = 1, N
(16)		DO 102J = 1, M
(17)		SUM = 0
(18)		DO 201K = 1, L
(19)		SUM = A(I, K) * B(K, J) + SUM
(20)	201	CONTINUE
(21)		C(I, J) = SUM
(22)	102	CONTINUE
(23)		RETURN
(24)		END

DEFINITIONS

Main Program

(1), (2), (3) Comments explaining purpose of program.

(4) Dimension statement defining storage requirements of matrix arrays X, Y, and Z.

(5) I/O statement for reading data from tape 5 according to format specified by statement 20. The subscripted array name X, with indexing parameters I and K, will cause reading of 25 values which fill the matrix array X in sequence by row major, column minor.

(6) Continuation of above I/O statement that causes the input of 25 values to fill matrix array Y sequentially by row major, column minor.

(7) Format statement defining form in which input data is presented and form to which each value will be converted. The data will be decimal numbers associated with power-of-ten exponents, and will be of the form ±XX.XXXXXX E±XX. The numbers will be binarized and stored as floating point words.

(8) A call statement calling subroutine PRODM which requires an argument list of 6 parameters. The parameters must be ordered within the list in the same sequence specified by the second statement (heading statement - item 13) in subroutine PRODM.

(9) I/O statement instructing computer to transmit output data to tape 6 according to format specified by statement 20. The subscripted array name Z with indexing parameters I and J will cause the 25 elements of matrix Z to be printed by row.

(10) Call statement that calls the EXIT subroutine to effect an exit from the program.

(11) End statement informing compiler that the last statement in the program has been read.

Subroutine

(12) Comment describing purpose of subroutine PRODM.

(13) Heading statement that defines the name of the subroutine and lists the parameters to be used by it.

(14) Dimension statement defining the storage requirements of matrix arrays A, B, C.

(15) A DO statement indicating that the following block of statements are to be executed repetitively. A block of such statements is called a DO-loop. All statements within it, except for the opening DO statement, constitute the range of the loop. The last statement of the loop is defined as statement 102. The number of iterations is defined by I=1, N. The first time the loop is executed I is set to 1 and is increased by 1 for each successive pass through the loop until I=N. At that time the loop is completed and control passes to the statement following statement 102. This DO statement specifies an outer loop.

(16) This DO statement indicates an intermediate loop beginning with the next statement and terminating at statement 201. This loop is repeated until the value J is equal to M, at which time control is returned to the outer loop.

(17) An arithmetic statement that sets the floating point value SUM to zero.

(18) This DO statement defines an inner loop beginning with the following statement and terminating at statement 201. This loop is repeated until the value K is equal to L, at which time control is passed back to the intermediate loop.

(19) An arithmetic statement that forms the elements in the product matrix C. The elements are formed by multiplying the elements in the ith row of matrix A by the elements in the jth column of matrix B and adding the products.

(20) The continue statement is a dummy used as the final statement in the range of the intermediate DO-loop.

(21) Arithmetic statement that stores each element in the product matrix C as it is formed.

(22) Continue statement used as the final statement of the outer DO-loop.

(23) Return statement is used as the termination of a subroutine or function subprogram, and returns control to the calling program.

(24) The end statement informs the compiler that the final statement in the routine has been read.

LIST OF MACHINE-LANGUAGE COMMANDS

1. Regular Primary Commands

Load Type

NØ	No operation
LA	Load A
LP	Load P
LT	Load T
LM	Load M
RA	Replace A
RP	Replace P
RT	Replace T
RM	Replace M
AP	Exchange A and P
AT	Exchange A and T
ZE	Clear E
XA	Extract to A
MA	Merge to A
XE	Extract to E
ME	Merge to E
DX	Double Extract
AS	Add Single Word
AL	Add Low-Order Word
AI	Add Intermediate Word
AM	Add High-Order Word
CS	Complement Set
CC	Complement Clear
CH	Complement Hold

Store Type

SE	Store E
SA	Store A
SP	Store P
ST	Store T
HA	Hold A
HP	Hold P
HT	Hold T
HM	Hold M

2. Secondary Commands

Load Type

NØ	No operation
LA	Load A
LP	Load P
LT	Load T
LM	Load M
AP	Exchange A and P
AT	Exchange A and T
XA	Extract to A
MA	Merge to A
AS	Add Single Word
AL	Add Low-Order Word
AI	Add Intermediate Word
AM	Add High-Order Word
CS	Complement Set
CC	Complement Clear
CH	Complement Hold

3. Special Primary Commands

Shift

SØ	Shift Open	The contents of the A register (or A and P registers combined) are shifted left or right (open-ended shift).
SC	Shift Closed	Same as Shift Open, except that no bits are lost (a circular shift).
NR	Numeric Right	Same as Shift Open, except that sign bit of the A register is propagated.
MP	Multiply Positive	Two numbers of like signs are multiplied together and a third quantity is added to the product (actually a multiply and add).
MS	Multiply Signed	Same as Multiply Positive, except that the two operands are of unlike signs.
DV	Divide	Forms the quotient of two positive operands.
FL	Float Left	Shifts the contents of the A register (or of the A and P registers combined) left until the number is normalized.
RC	Repeat Count	The contents of the A, P or M register are incremented by a specified amount.

Conditional

BR	Branch	Branches control if one of 15 specified conditions is met.
SK	Skip	A logand in sequence may be skipped if one of 15 specified conditions is met.

Data Processing

TB	Table Search	Each word of a table is compared with a specified search word until one of 4 conditions of comparison is found.
MV	Move	A block of consecutive words is moved to a specified location. The move can be terminated by one of 6 conditions.
MH	Match	Corresponding words of two blocks are compared until one of 4 conditions is met.
SR	Sort	Each word of a table is compared with the succeeding word. Each word is also compared with a specified memory word until one of 4 conditions is met.

BO	Block Output	A block of information is sent to a peripheral device.
BI	Block Input	A block of information is fed to the computer from some peripheral device.

Input-Output

CF	Control Function	Selects one of 7 combinations of input-output channels. Performs other miscellaneous I/O functions.
EF	External Function	Connects a peripheral device to the previously selected channel.
WO	Word Output	Single-word transmission to a peripheral device.
WI	Word Input	Single-word transmission from a peripheral device.

Interrupt

IT	Interrupt	Causes an unconditional branch to a closed interrupt routine.
TM	Terminate	A unique method of returning to the main program after execution of an interrupt subroutine.

4 PERIPHERAL EQUIPMENT

Associated with the TRW-230 Digital Computer is a wide variety of input-output equipment, described below under the headings of Standard Peripheral Equipment Group, Magnetic Tape System, and Individual Devices. The first two headings cover equipment groups, each with its own controller for communication between the computer and the group's units; the third covers various devices that communicate with the computer either directly or through controllers attached to single units.

STANDARD PERIPHERAL EQUIPMENT GROUP

The Standard Peripheral Equipment Group consists of four input-output devices and an unusually flexible distributing exchange, or controller. The units of this group are:

TRW-240 Controller
TRW-251 Paper Tape Reader and Reeler
TRW-261 Paper Tape Punch
TRW-285 Input-Output Typewriter
TRW-286 Send-Receive Set (teletype)

The TRW-240 enhances overall system capability by acting as a common controller in setting up multiple connections between the devices of this group and the computer, and by enabling them to operate off-line. Multiple connections can be made in many different modes, or combinations, which are summarized below. (A particular mode is selected by a combination of computer control and switches on the TRW-240 operator panel.)

Input Modes

Data being entered on punched tape or cards off-line, for subsequent input to the computer, can be duplicated simultaneously as both teletype and printed typewriter output.

Input data being read from the typewriter keyboard by the computer on-line, can be duplicated simultaneously on punched tape or cards. (Off-line data being entered on the typewriter for printed output can be duplicated simultaneously on punched tape and/or cards.)

Teletype input, off-line, can produce punched tape; on-line, it can undergo code conversion while being read into the computer.

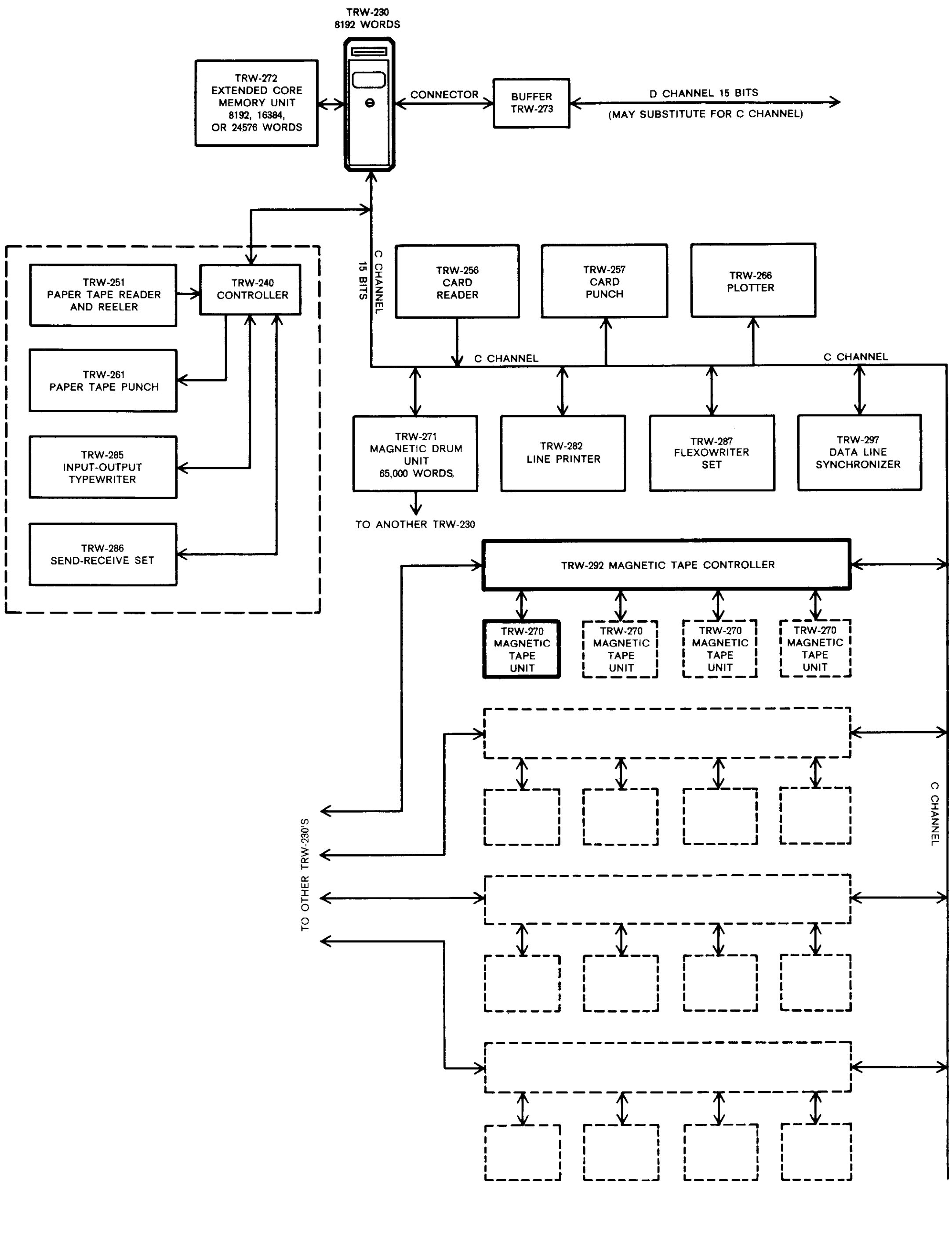


Figure 8. TRW-230 and Peripheral Equipment

Output Modes

Computer output data being transcribed by the typewriter can be duplicated simultaneously on punched tape or cards.

Computer output data being transmitted for teletype can be duplicated simultaneously as typewriter copy and punched tape or cards.

MAGNETIC TAPE SYSTEM

The Magnetic Tape System is designed for extreme reliability, in keeping with the high reliability and rugged packaging of the TRW-230. It consists of:

TRW-292 Magnetic Tape Controller
TRW-270 Magnetic Tape Units

The TRW-292 Magnetic Tape Controller controls, by program selection, up to four TRW-270 Magnetic Tape Units. As many as four controllers, each with four tape units may be connected to the TRW-230. This tape system operates in alphanumeric or binary mode. In the alphanumeric mode, each 15-bit computer word contains two 6-bit characters, written sequentially on the tape. In the binary mode, each 15-bit word is divided into three segments, also written sequentially. Each tape contains 7 tracks (6 for data, 1 for parity), with alternative packing densities of either 200 or 556 characters per inch—resulting in transfer rates of 15,000 and 41,700 characters/sec respectively. The density desired is selected manually by use of a "Hi/Lo" switch on the TRW-270 control panel. Tapes generated are compatible with the IBM 729 tape format.

INDIVIDUAL DEVICES

The individual devices are:

TRW-272 Extended Memory Units
TRW-271 Magnetic Drum Unit
TRW-273 Buffer Unit
TRW-282 Line Printer
TRW-287 Flexowriter
TRW-256 Card Reader
TRW-257 Card Reader/Punch
TRW-266 X-Y Plotter
TRW-297 Data Line Synchronizer
TRW-288 Send-Receive Set

TRW-272 Extended Memory Units. For systems requiring more than the basic 8K-word storage, additional core memory units are available in increments of 8, 192 words to raise total capacity to 32, 768 words.

TRW-271 Magnetic Drum Unit. The magnetic drum unit provides storage for 65,536 15-bit words plus parity. It contains 256 tracks of storage, each track containing 256 words. Reading or writing from the drum is by block transfer only; that is, the first word of a block to be transferred must be the first word of a track. Minimum access time to the first word to be transferred is 200 microseconds; average time is 8.5 milliseconds, with transfer rates thereafter averaging 15,000 words per second. Up to 4 drum units may be connected to the computer.

TRW-273 Buffer Unit. Essentially an intermediary between the computer and fast peripheral devices, the Buffer Unit takes over many of the input-output request and acknowledge functions that would otherwise be handled by the computer. It accomplishes the actual data transfers to and from the computer by means of a cycle stealing technique, and in so doing bypasses the computer's input-output section. The unit extends the computer's asynchronous I/O capability, and provides capacity for attaching additional peripheral devices or another TRW-230 computer via D-channel input and output cables.

TRW-282 Line Printer. Prints output from the computer at 300 lines per minute, 120 columns per line. Characters printed may be alphabet, numerals, spaces, and 27 mathematical and special symbols.

TRW-287 Flexowriter Set. Communicates directly with computer at the rate of 10 characters per second. Unit consists of a Flexowriter and controller and may be used to input data from punched paper tape or the keyboard; it will accept output information from the computer to produce a paper tape or printed report. The unit may be disconnected from computer and used for tape punching and listing.

TRW-256 Card Reader. Reads up to 200 80-column punched cards per minute, from a 500 card hopper. Consists of a controller and reader unit capable of connecting, disconnecting, and of six card reading options under program control.

TRW-257 Card Reader/Punch. Combination input and output unit using 80-column cards. Reads at 200 cards per minute, punches at 100 cards per minute.

TRW-266 X-Y Plotter. A digital, incremental plotter for producing point-to-point plots or continuous graphs under computer control. The unit consists of controller and digital recorder. It accepts 8-1/2 x 11 or 17 x 11 inch sheets of chart paper, or 12 inch rolls 120 feet long.

TRW-297 Data Line Synchronizer. A buffer and controller unit to permit digital data communication between computer and telephone or teletype line. Word length and information transfer rates can be varied to meet almost any requirement.

TRW-288 Send-Receive Set. Unit consists of a teletype and controller to enable direct use of the teletype as an input-output device to the computer. Designed for operation without a TRW-240 Controller.

5 APPLICATIONS

It is in the nature of digital computers to have a broad range of applications. In the TRW-230, stored logic programming flexibility, ruggedness, and extreme reliability inevitably broaden this range. A few TRW-230 areas of application are listed below:

Engineering and Sciences

Hydraulic systems analysis
Chemical engineering
Electronic circuit design
Thermodynamics
Stress analysis
Optical components design
Civil engineering
Metallurgy
Meteorology
Physics
Aerodynamics
Astronomy

Data Processing

Flight test and aerial survey data reduction
Oceanographic data reduction
Personnel record processing
Inventory control
Forecasting
Budgeting
PERT Systems
Production planning

Special Purpose and Real-Time Systems

Satellite or peripheral processing for a central computer
Replacement for special purpose analog mechanisms
Analog data scaling, calibration and linearization
Display systems driving
Servo loop control
Power plant control
Control systems
Telemetry data reduction
Buffering
Simulation systems and training devices
Ground support and test systems for automatic checkout
Communications control systems

SUMMARY OF TRW-230 CHARACTERISTICS

PHYSICAL

Components
Solid State

Size
16 inches deep, 20 inches wide, 59 inches high

Weight
530 pounds

Environmental Tolerance
Shock and vibration resistant. Will operate at temperatures from 32° to 122°F, at relative humidity up to 95%.

Power
118 volt, 60 or 400 cycle, 600 watts, single phase

Site Preparation
None required

ORGANIZATION

Operation
Parallel by 15-bit word elements
333 kc clock rate

Word Length
Variable in multiples of 15 bits; i.e., 15, 30, 45, etc.

Order Structure
Variable by Stored Logic to be single, double, triple, or any address structure

Instruction Code
82 microcommands with 8500 meaningful combinations (logands)

Macro-instruction code adaptable to the problem by appropriate selection of lograms

Memory

8,192 word elements, random-access core, expandable to 32,768 words maximum

6 microsecond read-restore memory cycle

INPUT-OUTPUT

Transfers

Input-output devices communicate with computer via C-channel with 15-bit parallel input and output cables, plus control lines for input request and acknowledge, output request and acknowledge, and device function control. I/O rate — 33,333 words per second input, 30,303 words per second output. Optional input-output buffer unit available with D-channel lines identical to C-channel for higher speed input-output, extension of asynchronous I/O capability, and multiple-computer arrangements.

Interrupt

Multi-level automatic interrupts via 11 interrupt lines

18 microseconds reaction time

AVAILABLE PROGRAMMING AIDS

System Support Catalog

Master reference catalog for all programming aids, applications information, and TRW and user contributed programs.

TRW FORTRAN COMPILER

A compiler system for translating programs written in the form of FORTRAN language statements into the machine language level of the TRW-230. The system is operational on computers having either paper tape or magnetic tape input-output. The compiler enables people with no programming experience to write computer programs.

TRW INTERPRETER

A system which accepts and operates programs written in a simplified, yet very powerful programming language. The system interprets symbolic instructions, macro-instructions, and pseudo-instructions. It allows for the use of symbolic addresses and data and generates error alarms when illegal operations are attempted. Ideal for use in scientific and engineering open shop operations, it also facilitates experimentation and program development in more extensive computing facilities.

TRW SYMBOLIC ASSEMBLER

Produces operational programs from symbolic language coding. It accepts regular instructions, pseudo-instructions and logram calling sequences. Symbolic addressing can be used with the system, and error alarms are generated when illegal operations are attempted.

SUBROUTINE LIBRARY

Contains over 200 macro-instructions for performing the most commonly used computing operations. These routines are incorporated into a program by giving the routine's name and the operand address or parameters it requires. The library is added to continually and presently contains routines for: trigonometric functions, data conversion, arithmetic operations, solution of equations, logical operations, matrix manipulation, input-output, curve fitting and data smoothing. Most operations are in single and double precision fixed point and double precision floating point. The routines minimize programming effort and reduce the space required for storing programs in memory. The execution times of some typical routines or "lograms" are listed at the end of this section.

FLOATING POINT SYSTEM

General purpose, floating point system with 15-bit characteristic and 30-bit mantissa. Typical instructions in the system are: load accumulator, load absolute value, store accumulator, store zero, add, subtract, multiply, divide, compare equal, compare less than, float, unfloat.

INPUT-OUTPUT UTILITY ROUTINES

For program preparation, checkout, entering program parameters, and loading the program. The basic set for a minimum complement of I/O equipment includes: assembly format loader, keyboard input program, teletype unit, memory dump on typewriter, program dump on tape. Additional routines are available for other peripheral devices.

DIAGNOSTIC ROUTINES

Programs for testing computer operations. Allow the user to distinguish equipment malfunction from program errors to reduce unnecessary maintenance calls. Includes tests for bootstrap loader, instruction logic, core memory, Paper Tape Reader, Paper Tape Punch, Typewriter, Magnetic Tape System, Plotter, Line Printer.

MONITOR, ASSEMBLER, AND SIMULATOR SYSTEM (MASS)

Program for assembling and simulating operations of a TRW-230 program on an IBM 7090 computer.

TYPICAL INSTRUCTION TIMES

Operation times include memory access.

Basic Logands:	Add	12 μsec
	Multiply	57 μsec
	Divide	57 μsec

Table Search: 18 μsec setup
12 μsec per 15-bit word element

Match: 12 or 18 μsec

Block Transfer: 18 μsec setup

Lograms	Single Length (15-bit) μsec	Double Length (30-bit) μsec
Compare "Equal"	108	144
Compare "Less Than"	108	150
Compare "Greater Than"	108	150
Binary to B C D (One 2-bit and Seven 4-bit characters)	-	983
BCD to Binary (One 2-bit and Seven 4-bit characters)	-	945
Square Root	606	1753
Sine $\theta(-2\pi<\theta<2\pi)$	951	5085
Cosine $\theta(-2\pi<\theta<2\pi)$	873	5181
Arcsin	2086	5623
Arctan	730	1068
Polar to Rectangular (r, θ to x, y)	1708	-
Rectangular to Polar (x, y to r, θ)	3171	-

TRW COMPUTER DIVISION OFFICES

For information or consultation on your computing problems, contact the TRW-230 Regional Sales Manager at the location nearest you:

ATLANTA
3272 Peachtree Road, N.E.
Atlanta 5, Georgia
Tel.: 233-3292

BOSTON
114 Waltham Street
Lexington 73, Massachusetts
Tel.: VO 2-6370
TWX: 617-862-3219

CHICAGO
400 N. Michigan Avenue
Wrigley Bldg., Suite 516
Chicago 11, Illinois
Tel.: 222-1940

CLEVELAND
1070 East 152nd Street
Cleveland 10, Ohio
Tel.: 383-6090
TWX: 383-2005

HOUSTON
4740 Ingersoll Street
Houston 27, Texas
Tel.: 713-666-2303

HUNTSVILLE
3313 Memorial Parkway, S.W.
Huntsville, Alabama
Tel.: 881-1512
TWX: 205-881-3531

LOS ANGELES
8433 Fallbrook Avenue
Canoga Park, California
Tel.: 346-6000
TWX: 213-348-2361

NEW YORK
1071 Main Street
Paterson, New Jersey
Tel.: LA 5-0770
TWX: 201-278-2417

ROME, N.Y.
1333 E. Dominick Avenue
Rome, New York
Tel.: FF7-6100
TWX: 315-337-7570

WASHINGTON, D.C.
4933 Auburn Avenue
Bethesda, Maryland
Tel.: 654-9025
TWX: 301-365-8883
or
905 - 16th Street, N.W., Suite 506
Washington 6, D.C.
Tel.: 783-5188
TWX: 202-965-0494

THOMPSON RAMO WOOLDRIDGE, INC.
TRW COMPUTER DIVISION
CANOGA PARK, CALIFORNIA

Effective: 15 July 1963

TRW-230 COMPUTER PURCHASE AND LEASE PRICE SCHEDULE

Model Number	Equipment or Item Description	Purchase Price	Basic Monthly Lease Price
TRW-230	Digital Computer Main Frame with 8, 192 Word Memory, Includes Voltage Regulator	$78, 500	$2, 050
TRW-240	I/O Controller for the TRW-251, 261, 285, & 286	14, 750	375
TRW-251	Paper Tape Reader/Reeler; 400 Characters/ Second	5, 025	130
TRW-256	Card Reader, 200 cards per minute	12, 500	375
TRW-257	Card Input/Output Unit; 200 input/100 output (reader, punch, controller)	36, 000	940
TRW-261	Paper Tape Punch; 60 Characters/Second	1, 625	40
TRW-266	X-Y Plotter	*	*
TRW-270	Magnetic Tape Unit	23, 000	610
TRW-272/1	Extended Core Memory Unit with 8, 192 Word Memory, Includes Voltage Regulagor	42, 500	1, 215
TRW-272/2	Extended Core Memory Unit with 16, 384 Word Memory	58, 000	1, 650
TRW-272/3	Extended Core Memory Unit with 24, 576 Word Memory	74, 000	2, 100
TRW-273/1	Buffer Channel	22, 950	640
TRW-282	Medium Speed Line Printer, 300 Lines/ Minute	30, 000	900
TRW-285	Input/Output Typewriter	3, 300	85
TRW-286	Send/Receive Set with Teletype Unit for use with TRW-240	3, 900	120
TRW-287	Flexowriter and Controller	14, 000	445
TRW-287/1	Flexowriter Controller Only	9, 700	300
TRW-288	Serial Send/Receive Set with Teletype Unit for use without TRW-240	3, 250	80
TRW-292	Magnetic Tape Controller, Includes Voltage Regulator	22, 500	595
TRW-296	Motor Alternator	1, 300	40
TRW-297/1	Dataphone Adapter	*	*

*Prices Available Upon Request.

Notes:
1) All prices F.O.B. Canoga Park, California, and exclude Federal, State or local taxes.
2) Purchase prices for computer and/or systems include installation within the Domestic U.S.; programming, operation and maintenance Manuals.
3) All equipments are covered by one-year warranty on workmanship and material.
4) A separate schedule is available for maintenance prices on purchased equipment.

THOMPSON RAMO WOOLDRIDGE, INC.
TRW COMPUTER DIVISION
CANOGA PARK, CALIFORNIA

Effective: 15 July 1963

GENERAL LEASE TERMS

1. Basic Monthly Rates include installation within the Domestic U.S., programming and operation Manuals, and access to TRW, Inc. library of programs and subroutines.

2. Basic Monthly Rates include service and parts to maintain the equipment in good working order.

3. Basic Monthly Rates entitle lessor to "on-call" service between the hours of 8:00 AM and 5:00 PM, Mondays through Fridays, excluding holidays. For maintenance service outside these hours, lessor will be charged at a rate of \$12.00 per hour portal-to-portal for all calls within a 100 mile radius of Canoga Park, California or a 50 mile radius of Washington, D.C.

   In cases where a system is installed outside these areas, the lessor will be billed for travel and subsistence related to service calls.

4. Minimum lease term for prices quoted herein: Twelve (12) months.

5. The lessor may, at any time during the rental period following acceptance of the equipment, exercise an option to purchase any or all of the leased equipments at the prices and terms prevailing at the time of such purchase. Credits against the purchase from rentals previously paid under the same lease agreement will be allowed on the basis of the following schedule, but may not exceed seventy percent (70%) of the price prevailing at the time the option is exercised:

   75% of rentals paid, if option is exercised during first year following acceptance, thereafter 60% of actual rental will be applied as purchase credit.

6. Basic Monthly Rates entitle lessor to two-hundred (200) hours of equipment usage per calendar month. The equipment may be used at any time during the month. For equipment usage in excess of two hundred hours, there will be an additional charge computed as follows: Forty percent (40%) of one two-hundredth (1/200) of the Basic Monthly Rate for each additional hour of equipment use.

7. Basic Monthly Rates are F.O.B. Canoga Park, California.

8. Basic Monthly Rates exclude any Federal, State or Local Taxes which may apply.