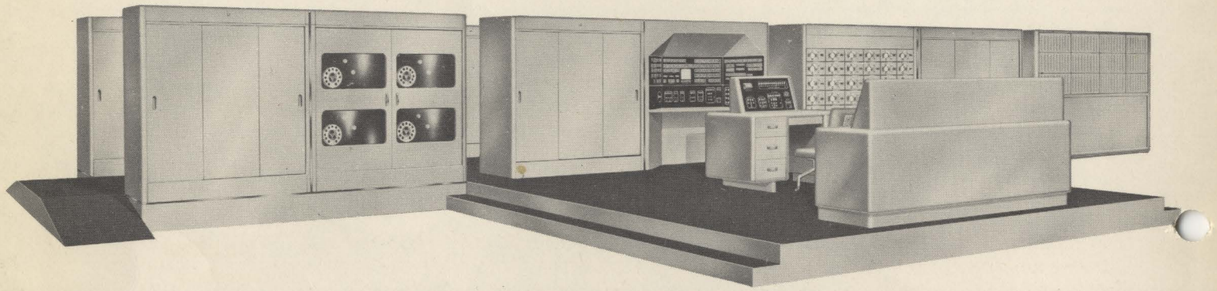
ERA

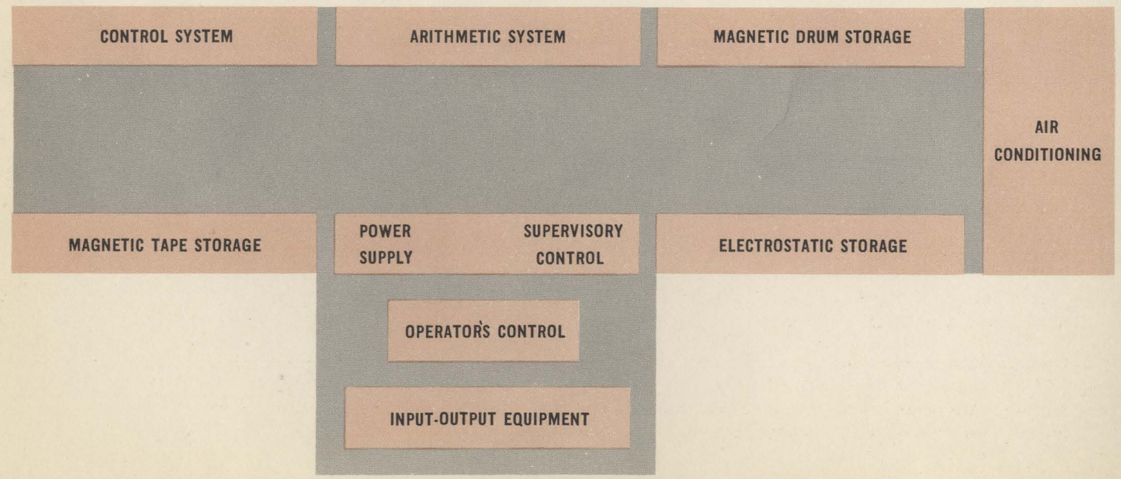
1103

GENERAL PURPOSE COMPUTER SYSTEM

ERA 1103



GENERAL PURPOSE COMPUTER SYSTEM



The ERA 1103 is a general-purpose digital computer system for applications requiring large storage capacity, high operating speed, and great programming versatility. Unusual logical features of the machine facilitate maximum utilization of its very high inherent speed. In addition to performing large scale calculations, the system is adaptable to a wide variety of applications, including simulation and control in real time.

The computer system is fully automatic in that the sequence of operations is determined by programs of internally stored instructions capable of self-modification. For speed, the machine operates in the parallel mode; i.e., all digits of a number are operated upon simultaneously. Internal arithmetic operations are in the binary system. The basic word size is 36 binary digits, or "bits". A word may be an instruction, a number, or an arbitrarily coded quantity.

Storage

The internal memory of the ERA 1103 consists of 16,384 registers of magnetic drum storage and 1024 registers of electrostatic (CRT) storage. Each of these 17,408 registers is individually addressed and directly accessible. Supplementary storage is provided by four magnetic tape units. Information is transmitted to and from the tapes in blocks of a fixed number of words each.

Instruction Structure

A form of "two-address" logic is employed. An instruction word consists of a 6-bit operation code and two 15-bit execution-addresses. The functions of the execution addresses are different for the various types of instructions, but, in general, they specify registers in the memory from which operands are to be obtained or in which results are to be stored.

Number Notation

A "1's-complement" notation is used, in which negative numbers are represented as complements on 2^{36}-1. That is, the negative of a number is formed simply by complementing its digits. In this representation the left-most bit is 0 for positive numbers and 1 for negative numbers. Numbers up to 2^{35}-1 in absolute value may be represented.

Arithmetic

The arithmetic section contains the 36-bit X register, the 36-bit Q register, and the 72-bit accumulator, A. The X register serves as an exchange register or switching center for most of the internal transmissions. It contains the addend, subtrahend, multiplicand, and divisor in the corresponding arithmetic operations. The Q register contains the multiplier, quotient, and logical multiplier in the corresponding operations. Products and sums may be accumulated in A up to its full 72-bit, or "double-precision" capacity. Negative numbers in A are automatically represented as complements on 2^{72}-1. Shifting takes place in both A and Q. The type of shift is circular (end-around) to the left.

Input-Output

The input-output section will accommodate a wide option of terminal equipment; the choice is governed by the requirements at a given installation. Certain equipment is considered basic. This category includes a photoelectric tape reader, a directly connected electric typewriter, and a high-speed tape punch. Optional equipment, which will vary to suit the installation, may include such items as a card reader, a card punch, teletype communication circuits, analog-digital converters from sensing instrumentation, graphic visual displays, and signal circuits to process-actuating mechanisms. The program control and the input-output registers of the ERA 1103 are sufficiently general in their properties to cooperate with this wide diversity of external units.

Programs and data may be initially loaded into the internal memory from punched paper tape via the photoelectric tape reader without the necessity for a programmed input routine. Facilities are also included for optional initial loading via the supplementary storage magnetic tape units.

Communication with the optional external equipment takes place via an 8-bit register (IOA) and a 36-bit register (IOB). These registers, together with the basic input-output registers, are arranged to permit simultaneous use of several units, and to allow computation to proceed while terminal equipment is operating.

GENERAL DESCRIPTION

TERMINOLOGY

The terminology used in discussing the characteristics of the 1103 computing system is explained in the following notes.

Word Length

36 bits (binary digits)

Parallel Access Registers of Arithmetic Section

A 72-bit accumulator (A_{71}, A_{70} ... A_0)
Q 36-bit shifting register (Q_{35}, Q_{34} ... Q_0)
X 36-bit exchange register (X_{35}, X_{34} ... X_0)

A_R right-hand (least significant) 36 bits of A
A_L left-hand (most significant) 36 bits of A

Parallel Access Storage

ES 1,024 words of Electrostatic Storage
MD 16,384 words of Magnetic Drum Storage

Composition of an Instruction Word

Instruction word	36 bits (i_{35}, i_{34} ... i_0)
Operation code	6 bits (i_{35}, i_{34} ... i_{30})
First execution address, u,	15 bits (i_{29}, i_{28} ... i_{15})
Second execution address, v,	15 bits (i_{14}, i_{13} ... i_0)

Allocation of Addresses

ES 00000—01777 (octal)
Q 1---- (octal)
A 2---- (octal)
MD 40000—77777 (octal)

Addresses referring to A and Q may be denoted by a and q respectively.

Sections of Addresses

j one-digit octal number represented by u_{14}, u_{13}, u_{12}.
n four-digit octal number represented by u_{11}, u_{10} ... u_0.
k number of shifts, represented by v_6, v_5 ... v_0.

Contents of Registers

Brackets are used to denote "contents of". Thus:

(u)=36-bit word at address u.
(Q)=36-bit word in Q.
(A)=72-bit word in A.

(A_R)=36-bit word in A_R.
(A_L)=36-bit word in A_L.

Double-length Extensions

D(u)=72-bit word whose right-hand 36 bits are (u) and whose left-hand 36 bits are all alike and equal to the left-most bit of (u).

S(u)=72-bit word whose right-hand 36 bits are (u) and whose left-hand 36 bits are all zero.

D(Q), D(X), S(Q), and S(X) are similarly defined.

L(Q)(u)=72-bit word whose left-hand 36 bits are zeros and each of whose right-hand 36 bits is given by the product of the corresponding bits of (u) and (Q).

L(Q')(v)=72-bit word whose left-hand 36 bits are zeros and each of whose right-hand 36 bits is given by the product of the corresponding bits of (v) and the complement of (Q).

Control Registers

PAK Program Address Counter
UAK U-Address Counter
VAK V-Address Counter
SAR Storage Address Register

Arrangement of Magnetic Drum Storage

4,096 bits on each track
4,096 words on a group of 36 tracks
4 groups of tracks, giving 16,384 words

Variable interlace between the Storage Address Register and the angular location counter permits choice of the angular interval between memory locations having consecutive addresses.

Input-output Registers

IOA An in-out register of 8 bits.
IOB An in-out register of 36 bits.
TWR A typewriter register of 6 bits.
HPR A high-speed punch register of 6 bits.

INSTRUCTION REPERTOIRE

Program Sequence Control

A complete computer operation consists of two parts:

Part One—the execution of the current instruction, CI.

Part Two—the acquisition of the next instruction, NI.

At the start of Part One, the program control registers already contain CI as the result of the second part of the previous operation, and (PAK) is y plus 1, where y is the address from which CI was acquired.

During Part Two, NI is acquired from address held in PAK at the end of Part One, and (PAK) is then increased by one.

Thus, provided that CI does not call for a change in (PAK), NI will be acquired from address y plus 1. In a normal program sequence, successive instructions are obtained from consecutive addresses.

Each range of address allocations (ES,00000-01777;MD, 40000-77777) is treated by PAK, UAK, and VAK as a closed consecutive set.

A departure from the normal sequence is called a "jump", and is achieved by altering (PAK) during Part One of an operation. Instructions that call for a change in (PAK) are called "jump" instructions.

MPuv **MultiPly:** Form in A the 72-bit product of (u) and (v), leaving in Q the multiplier (u).

MAuv **Multiply Add:** Add to (A) the 72-bit product of (u) and (v), leaving in Q the multiplier (u).

DVuv **DiVide:** Divide the 72-bit number (A) by (u), putting the quotient in Q, and leaving in A a non-negative remainder R. Then replace (v) by (Q). The quotient and remainder are defined by: $(A)_i=(u)\bullet(Q)+R$, where $0\leq R<|(u)|$. Here $(A)_i$ denotes the initial contents of A.

SFuv **Scale Factor:** Replace (A) with D(u). Then left circular shift (A) by 36 places. Then continue to shift (A) until $A_{34}\neq A_{35}$. Then replace the right-hand 15 bits of (v) with the number of left circular shifts, k, which would be necessary to return (A) to its original position. If (A) is all ones or zeros, k=37. If u is a, (A) is left unchanged in the first step, instead of being replaced by $D(A_R)$.

RPjnw **RePeat:** This instruction calls for the next instruction, which will be called NIuv, to be executed n times, its "u" and "v" addresses being modified or not according to the value of j. Afterwards the program is continued by the execution of the instruction stored at a fixed ES address F_1. The exact steps carried out are:

Sequenced Instructions

1. Replace the right-hand 15 bits of (F_1) with the address w.
2. Execute NIuv, the next instruction in the program, n times.
3. If j=0, do not change u and v.
   If j=1, add one to v after each execution.
   If j=2, add one to u after each execution.
   If j=3, add one to u and v after each execution.
4. On completing n executions, take (F_1) as the next instruction.
5. If the repeated instruction is a jump instruction, the occurrence of a jump terminates the repetition. In addition, if NIuv is a Threshold Jump or an Equality Jump, and the jump to address v occurs, (Q) is replaced by the quantity j, (n-r), where r is the number of executions that have taken place.

Transmissive Instructions

TPuv **Transmit Positive:** Replace (v) with (u).

TNuv **Transmit Negative:** Replace (v) with the complement of (u).

TMuv **Transmit Magnitude:** Replace (v) with the absolute magnitude of (u).

TUuv **Transmit U-address:** Replace the 15 bits of (v) designated by v_{15} through v_{29}, with the corresponding bits of (u), leaving the remaining 21 bits of (v) undisturbed.

TVuv **Transmit V-address:** Replace the right-hand 15 bits of (v) designated by v_0 through v_{14}, with the corresponding bits of (u), leaving the remaining 21 bits of (v) undisturbed.

ATuv **Add and Transmit:** Add D(u) to (A). Then replace (v) with (A_R).

STuv **Subtract and Transmit:** Subtract D(u) from (A). Then replace (v) with (A_R).

Q-Controlled Instructions

QTuv **Q-controlled Transmit:** Form in A the number L(Q)(u). Then replace (v) by (A_R).

QAuv **Q-controlled Add:** Add to (A) the number L(Q)(u). Then replace (v) by (A_R).

QSuv **Q-controlled Substitute:** Form in A the quantity L(Q)(u) plus L(Q')(v). Then replace (v) with (A_R). The effect is to replace selected bits of (v) with the corresponding bits of (u) in those places corresponding to 1's in Q. The final (v) is the same as the final (A_R).

Replace Instructions

RAuv **Replace Add:** Form in A the sum of D(u) and D(v). Then replace (u) with (A_R).

RSuv **Replace Subtract:** Form in A the difference D(u) minus D(v). Then replace (u) with (A_R).

CCuv **Controlled Complement:** Replace (A_R) with (u) leaving (A_L) undisturbed. Then complement those bits of (A_R) that correspond to ones in (v). Then replace (u) with (A_R).

LAuk **Left shift in A:** Replace (A) with D(u). Then left circular shift (A) by k places. Then replace (u) with (A_R). If u=a, the first step is omitted, so that the initial content of A is shifted.

LQuk **Left Shift in Q:** Replace (Q) with (u). Then left circular shift (Q) by k places. Then replace (u) with (Q).

Split Instructions

SPuk **Split Positive Entry:** Form S(u) in A. Then left circular shift (A) by k places.

SNuk **Split Negative Entry:** Form in A the complement of S(u). Then left circular shift (A) by k places.

SAuk **Split Add:** Add S(u) to (A). Then left circular shift (A) by k places.

SSuk **Split Subtract:** Subtract S(u) from (A). Then left circular shift (A) by k places.

Two-Way Conditional Jump Instructions

SJuv **Sign Jump:** If $A_{71}=1$, take (u) as NI. If $A_{71}=0$, take (v) as NI.

ZJuv **Zero Jump:** If (A) is not zero, take (u) as NI. If (A) is zero, take (v) as NI.

QJuv **Q-Jump:** If $Q_{35}=1$, take (u) as NI. If $Q_{35}=0$, take (v) as NI. Then, in either case, left circular shift (Q) by one place.

One-way Conditional Jump Instructions

IJuv **Index Jump:** Form in A the difference D(u) minus 1. Then if $A_{71}=1$, continue the present sequence of instructions; if $A_{71}=0$, replace (u) with (A_R) and take (v) as NI.

TJuv **Threshold Jump:** If D(u) is greater than (A), take (v) as NI; if not, continue the present sequence. In either case, leave (A) in its initial state.

EJuv **Equality Jump:** If D(u) equals (A), take (v) as NI; if not, continue the present sequence. In either case leave (A) in its initial state.

One-way Unconditional Jump Instructions

MJjv **Manually selective Jump:** If the number j is zero, take (v) as NI. If j is 1, 2 or 3, and the correspondingly numbered MJ selecting switch is set to "jump", take (v) as NI; if this switch is not set to "jump", continue the present sequence.

RJuv **Return Jump:** Let y represent the address from which CI was obtained. Replace the right-hand 15 bits of (u) with the quantity y plus 1. Then take (v) as NI.

Magnetic Tape Storage Instructions

AMjn– **Advance Magnetic tape:** Move the magnetic tape of MT unit j in the forward direction by n blocks. (Note: One block consists of 32 words)

BMjn– **Back Magnetic tape:** Move the magnetic tape of MT unit j in the backward direction by n blocks.

RMjnv **Read Magnetic tape:** Read n blocks from MT unit j (running forward) to 32 n consecutive addresses in ES starting with v.

WMjnv **Write Magnetic tape:** From 32 n consecutive addresses in ES, starting with v, write n blocks on MT unit j (running forward).

Stop Instructions

MSjv **Manually selective Stop:** If j=0, stop computer operation and provide suitable indication. If j=1, 2, or 3 and the correspondingly numbered MS selecting switch is set to "stop", stop computer operation and provide suitable indication. Whether or not a stop occurs, (v) is NI.

FS-- **Final Stop:** Stop computer operation and provide suitable indication.

External Equipment Instructions

EF-v **External Functions:** Select a unit of external equipment and perform the function designated by (v).

ERjv **External Read:** If j=0, replace the right-hand 8 bits of (v) with (IOA); if j=1, replace (v) with (IOB).

EWjv **External Write:** If j=0, replace (IOA) with the right-hand 8 bits of (v); if j=1, replace (IOB) with (v). Cause the previously selected unit to respond to the information in IOA or IOB.

PR-v **PRint:** Replace (TWR) with the right-hand 6 bits of (v). Cause the typewriter to print the character corresponding to the 6-bit code.

PUjv **PUnch:** Replace (HPR) with the right-hand 6 bits of (v). Cause the punch to respond to (HPR). If j=0, omit seventh level hole; if j=1, include seventh level hole.

INSTRUCTION REPERTOIRE

For further information about the application of the ERA 1103 General-Purpose Computer System to your problems, write on your business letterhead to Engineering Research Associates Division of Remington Rand Inc., 315 Fourth Avenue, New York 10, New York.

Engineering Research Associates | DIVISION OF *Remington Rand*

315 FOURTH AVENUE, NEW YORK 10, N. Y.
CANADIAN HEAD OFFICE, TORONTO 1, ONTARIO
SALES AND SERVICE OFFICES IN PRINCIPAL CITIES OF THE WORLD

PRINTED IN U.S.A.—EL—116