Programming Design Features of the GAMMA 60 Computer

P. DREYFUS

The GAMMA 60 is a complete, closed-loop electronic system designed to analyze, process, and produce information. The GAMMA 60 can at the same time read, write, punch, print, calculate, sort, compare, and make decisions. (See Fig. 1.)

In order to do so, the GAMMA 60 is equipped with:

1. Input equipment (card readers, paper tape readers, magnetic tape units).
2. Data processing units (calculators, comparators, transcoders).
3. Storage media (high-speed memory, magnetic drums, magnetic tape units).
4. Output equipment (card punches, paper tape punches, line printers, magnetic tape units).
5. Control elements which allocate instructions and data (program distributor, data distributor).

The control elements and the high-speed memory make up the central unit of the GAMMA 60.

The input-output units, auxiliary storage units, and data processing units are functional elements connected to the central unit.

The fundamental principle in the design of the GAMMA 60 has been the assignment to the functional elements of a completely closed loop mode of operation within the generally autonomous framework of the GAMMA 60. A functional element attains complete autonomy once it has been furnished with basic data (from cards, punched tape, magnetic tape) and with a medium on which to record the information it produces (blank cards, paper tape, magnetic tape). Fig. 2 illustrates a block diagram of the GAMMA 60.

The compatibility of the functional elements is made possible by their ability to:

1. Request instructions.
2. Request data transfers.
3. Execute the specialized tasks for which they have been designed completely. Independent of the other components of the GAMMA 60, once they have received operating instructions and data.

This possibility which permits the simultaneous operation of all the functional elements is the basis of full utilization of the GAMMA 60.

Within the generally autonomous framework of the GAMMA 60, made possible by the automatic scheduling of the control element, it is possible to: 1. execute simultaneously data processing functions on items at different stages of evolution in the same problem, and 2. execute simultaneously the data processing functions of several different problems (simultaneous parallel operation).

This makes it possible to obtain considerable improvement in over-all execution times by operating each of the elements at optimum rates.

Information in the GAMMA 60

Information in the GAMMA 60 is always in coded form. Information may be numeric or alpha numeric. Characters and numbers can be stored in the memory of the GAMMA 60 in the code of an external medium (punched card code, punched paper tape code).

However, information can only be completely processed after having been translated into the internal codes of the GAMMA 60 by the transcoder. The internal codes of the GAMMA 60 are composed as follows:

Numeric information retains its decimal structure, but each decimal digit is represented by a combination of four binary digits (bits), a binary digit being represented by either the presence (1) or absence (0) of a stable electric or magnetic state.

The ten decimal digits are represented in the GAMMA 60 as shown in Table I. This representation of numbers is called the binary coded decimal representation.

Fig. 1. A GAMMA 60 installation

This model is equipped with (clockwise from the top right) main unit and high-speed memory, power supplies, 2 magnetic drum units, 16 magnetic tape units and their control unit, 4 printers, 2 card readers, and 1 card reader punch.
In the GAMMA 60 the number 247 would be written as:

001000111

Alphanumeric information retains its usual structure, but each character (alphabetic or numeric) is represented by six bits.

The 64 possible combinations make it possible to represent certain special characters in addition to the 26 letters of the alphabet and 10 decimal digits.

Letters, numbers and other characters are defined in Table II.

A character is represented by the group of binary bits found in the same line and in the same column.

W is represented in the machine 010111
8 is represented in the machine 101000
w = is represented in the machine 110000

The unit of information transfer in the GAMMA 60 is a group of 24 bits called a catena (the Latin word for chain).

Thus, 1 catena may contain the following:

24 bits, representing data in binary form
6 decimal digits
4 alphanumeric characters

Memory capacity and the length of information transfers are expressed in catena.

### Description and Operation

The GAMMA 60, as has been seen, is composed of the central unit to which are connected a variable number of functional elements. The number and type of elements connected is a function of the particular application.

#### CENTRAL UNIT

The central unit contains the program distributor, the data distributor and the high-speed memory (HSM). The central unit serves as:

- A dispatcher (program distributor and data distributor) which:
  1. Distributes instructions.
  2. Schedules data transfers.
  3. Co-ordinates the activity of the various elements.

- A turn table (high-speed memory) through which all information being transferred to or from the elements must pass. The latter do not possess buffers but simply register of limited capacity in which information is stored only during operational cycles of the elements.

#### THE HIGH-SPEED MEMORY

The high-speed memory, serving as buffer memory between the various elements, may contain at any instant:

1. Program information (instructions) which are the commands to the elements.
2. Quantitative information or data which are to be processed by the elements.

The high-speed memory and the elements are linked by two information channels:

1. Distributor bus over which is transferred all information from the HSM to the elements.
2. Collector bus over which is transferred all information from the elements to the HSM.

#### THE PROGRAM DISTRIBUTOR

This unit is designed to:

1. Receive requests for instructions issued by the elements to obtain instructions for themselves or for another element.
2. Choose at a given time from among the requesting elements, one of them, by means of priority circuits, the order of priority having been determined in advance.
3. Read out the instructions destined for the elements as they are selected and interpret them. This interpretation function is very important and will be discussed in the chapter on programming.
4. Give to the various elements the commands and means to begin operational cycles (their type of operation to be carried out, addresses of data to be processed).

The program, which is a list of instructions intelligible to the different elements of the GAMMA 60, is stored in the auxiliary and large capacity memories (magnetic drum, magnetic tape, etc.) and brought as successive blocks into the

### Table II. Character Representation in the GAMMA 60

<table>
<thead>
<tr>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>01</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>+</td>
<td>-</td>
<td>(</td>
<td>)</td>
<td>x</td>
</tr>
<tr>
<td>11</td>
<td>=</td>
<td>&gt;</td>
<td>&lt;</td>
<td>A</td>
<td>&amp;</td>
<td>$</td>
<td>%</td>
<td>£</td>
<td>$</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Dreyfus—Programming Design Features of the GAMMA 60 Computer*
THE DATA DISTRIBUTOR

This unit is designed to:

1. Receive requests for data transfers issued by the various elements to obtain as necessary, data transfers to or from the high-speed memory.
2. Select, in order to effect a data transfer, a given receiver or emitter element and the source or destination zone in the high-speed memory. The choice between conflicting requests is resolved by the priority circuits, the order of priority having been determined in advance.

THE FUNCTIONAL ELEMENTS

Each functional element contains the following:

1. Registers in which are stored the information to be processed.
2. A local built-in program which carries out the specialized functions of the element.
3. Auxiliary registers to store data addresses and instructions.

Every element is capable of:

1. Sending requests for instructions to the program distributor via the instruction request circuits, one line for each element.
2. Sending requests for data transfers from high-speed memory to the data distributor via the data request circuits, one line for each element.
3. Receiving instructions from the program distributor over the unique instruction bus to which every element is connected.
4. Receiving data (input elements excepted) via the unique distributor bus to which all the elements concerned are connected.
5. Sending data (output elements excepted) to the high-speed memory via the unique collector bus to which all the elements concerned are connected.

THE OPERATION OF AN ELEMENT

The operation of every element follows this procedure:

1. Requests instructions from the program distributor.
2. Consideration of the request (depending on the position [rank] of the element in the priority chain and on the number of requests for instructions from other elements waiting to be selected).
3. Request from the program distributor to the data distributor for a transfer from the high-speed memory of instructions relative to the element under consideration.
4. Consideration of this request by the data distributor (depending on the position of the program distributor in the priority chain and on the number of requests for data transfers waiting to be fulfilled) and execution of the transfer.
5. Analysis of the instruction by the program distributor and transfer to the selected element of the instructions and addresses of data to be processed.
6. Requests from the element to the data distributor for a transfer from the HSM of the data to be processed.
7. Consideration of this request and, when the priority considerations are fulfilled, the execution of the data transfer.
8. Closed loop operation of the element and eventual producing of a result.
9. Requests to the data distributor for a transfer of the results to the HSM.
10. Consideration of this request by the data distributor and, when the priority conditions are fulfilled, execution of transfer.
11. Return to operation 1.

For each of the elements Magnetic Drum, Arithmetic Calculator, Printer, etc., the procedure is the same with a few exceptions.

For example steps 9 and 10 are nonexistent during the operation of the printer or of a HSM to magnetic drum data transfer: steps 6 and 7 are nonexistent during the operation of a card reader or a magnetic drum to HSM data transfer.

Operation 8 is generally much longer than the sum of all other operations. This fact demonstrates how a single program distributor and a single data distributor are sufficient to monitor simultaneous operation of a large number of elements.

Technical Characteristics of the GAMMA 60

MEMORIES

The GAMMA 60 possesses a set of memory devices for information storage, the functional characteristics of which are closely related to two technological considerations. These are capacity and access times.

Of the types of memory available those with the shortest access times are called high-speed memories. On the other hand storage of information in high-speed memories is expensive and therefore must be temporary.

Growing out of requirements diametrically opposed to the preceding, magnetic tape storage offers the advantage of large capacity but has long access times. This type of memory, which is relatively inexpensive to manufacture, makes it possible to store voluminous information which is organized in the same sequence in which the data are to be processed.

Magnetic drum storage occupies an intermediate position with respect to capacity and access times.

HIGH-SPEED MEMORY

Physically, the high-speed memory is a static memory of saturated magnetic cores. Two stable states of magnetization of the cores can be obtained. These two states are made to correspond to the symbols 0 and 1.

Technologically, the capacity of the
The magnetic drum turns at a constant speed of approximately 3,000 rpm. The capacity of each track is 200 catenae. Thus a magnetic drum has a total capacity of 25,600 catenae, 153,600 decimal digits, or 102,400 alphanumeric characters. The magnetic drum is an addressable memory, each catena on the drum having an address in the range from 00000 to 25,599.

The access time to an address on the drum depends on the position of that address with respect to the read and write head at the instant the contents of that address are called for. Thus the access time varies from 0 to 20 milliseconds. The magnetic drum is characterized by:
1. Maximum access time: 20 milliseconds.
2. Average access time: 10 milliseconds.

The read or write time for one catena is 100 microseconds (\(\mu\)sec). Data transfers to and from the magnetic drum are completely variable in length.

MAGNETIC TAPE

Magnetic tape provides the GAMMA 60 with a very large capacity memory. The tape itself consists of a plastic (Mylar) tape which is coated on one side with a layer of iron oxide. Magnetic tapes are handled by tape units which consist of:
1. Devices for unreeling and rereeling the tape.
2. A write head.
3. A read head which also serves during a write operation to check the information being written on tape.

TAPE CHARACTERISTICS
Length: 1,100 meters (3,600 feet).
Width: 12.7 millimeters (1/2 inch).
Packing Density: (200 per inch).

DATA PROCESSING ELEMENTS

ARITHMETIC CALCULATOR
Number representation
The arithmetic calculator operates on numbers in a form similar to logarithms. Such numbers may be either normalized...
floating decimal point or have a programmed (fixed) decimal point.

Normalized form

Any signed decimal number can be represented by the product of a number between 0.1 and 1.0, a positive or negative power of 10, and the algebraic sign. For example:

\[-3242.3 = -0.32423 \times 10^1\]

\[+0.0034 = +0.34 \times 10^{-2}\]

A number can thus be completely specified by its algebraic sign, a power of 10 which represents its order of magnitude and a mantissa which lies between 0.1 and 1.

So as not to have to manipulate negative exponents, a bias of 40 is added systematically. In addition, only the digits of the mantissa to the right of the decimal point need be specified. The two numbers in the previous example are written:

<table>
<thead>
<tr>
<th>Sign</th>
<th>Exponent</th>
<th>Mantissa</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>44</td>
<td>3242300000</td>
</tr>
<tr>
<td>+</td>
<td>38</td>
<td>3400000000</td>
</tr>
</tbody>
</table>

Actually numbers to be processed by the arithmetic calculator are stored in two catenae in the GAMMA 60.

The first binary position of the first catena of the number contains the sign: + sign = 1, - sign = 0. The seven following positions contain the biased exponent which may vary from 00 to 79. This represents a true variation of the exponent from -40 to +39.

The 40 remaining binary positions of the two catenae permit the representation of ten significant decimal digits in binary coded decimal form.

Programmed Decimal Form

In accounting applications it is particularly useful to be able to write all figures placing the decimal point in a known position. In order to do so, the programmed decimal form is used in the GAMMA 60.

To represent numbers in this form the position of the decimal point between the ten significant figures of a number is specified.

Writing a number in programmed decimal form with the decimal programmed at 6 means that the decimal point is to be taken as being six places to the left of the least significant figure. In this example, the integral part of the number contains at most four digits. The fractional part contains at most six digits.

The position of the programmed decimal point is specified in the part of the number in its machine coded form reserved for the exponent. The position of the programmed decimal point may vary from 0 to 10. The variation of the position of the programmed decimal point is actually expressed as a number between 40 and 50. When no decimal places are to be retained (programmed decimal at 0), the exponent is actually 50. When no figures to the left of the decimal point are to be retained (programmed decimal 10), the exponent is actually 40.

The number \(-3242.3\) with programmed decimal 6 is represented in the GAMMA 60:

\[0 4 4 3 2 4 2 3 0 0 0 0 0 0\]

and with programmed decimal 0:

\[0 5 0 0 0 0 0 0 0 0 3 2 4 2\]

and with programmed decimal 10:

\[0 4 0 3 0 0 0 0 0 0 0 0 0\]

Operation times for operands of 10 significant figures expressed in floating or programmed decimal form are the following:

- Addition: 100 \(\mu\)sec.
- Subtraction: 100 \(\mu\)sec.
- Comparison: 100 \(\mu\)sec.
- Round off: 100 \(\mu\)sec.
- Multiplication: 300 \(\mu\)sec.
- Division: 600 \(\mu\)sec.

Double Precision Arithmetic

Normally the arithmetic calculator handles operands of ten significant decimal digits. However, in certain scientific applications more precision is required. There exist programs for the GAMMA 60 which make it possible to perform calculations on operands containing 11 to 19 significant figures. This type of arithmetic is called double precision arithmetic.

The Logical Calculator

The logical calculator is the element of the GAMMA 60 designed to perform arithmetic operations on binary operands and to carry out logical operations.

Generalized Comparator

The Generalized Comparator makes it possible to carry out the following operations in the GAMMA 60.

- One way comparisons. Two variable-length operands are compared catena by catena, starting with the left most catena of each operand. The comparison process stops automatically as soon as a difference is detected. Thus both numeric and alphanumeric data may be compared.
- Execution time is 44 \(\mu\)sec per catena.

Input-Output Equipment

Input Elements

The following are the devices which make it possible to bring information in the GAMMA 60:

1. 80-Column Card Readers. Each card reader consists of three sets of read brushes. The first two sets make it possible to read and check-read a card. If the check fails the third set of brushes makes it possible to re-read the card. Each reader has two receiving bins.

Reading speeds: 300 cards per minute 400 characters per second.

Any number of card readers may be connected to the GAMMA 60.

2. Punched paper tape readers. The punched paper tape readers can read all 5- and 8-channel paper tape codes.

Each reader has only one sensing element with a speed of 200 characters per second.

Any number of paper tape readers may be connected to the GAMMA 60.
OUTPUT ELEMENTS

1. Printers. A printer consists of a rotating-type cylinder of 120 printing positions, each printing position consisting of 60 characters distributed around the circumference of the cylinder, thus forming 120 "type wheels," (see Fig. 5). The edge punched paper is driven past the cylinder by means of sprocket wheels. When a character moves past the printing point a small hammer behind the paper is actuated pressing the paper against the cylinder. Fast paper feeds and line spacing are program controlled. Operating speed is 300 lines per minute. Any number of printers may be added to the GAMMA 60.

2. 80-Column Card Reader-Punch. A card punch is always equipped with read brushes. A card reader can exist without card punch. A single track serves both units.

Each card reader-punch is equipped with three sets of read brushes. The third set is used to read back and check the information punched. Card handling speed is 300 cards per minute. Any number of card reader-punches may be connected to the GAMMA 60.

3. Paper tape punches. It is possible to punch 5 and 8 channel tapes on the GAMMA 60. Punching speed is 25 characters per second. Any number of paper tape punches may be connected to the GAMMA 60.

Programming the GAMMA 60

Programming the GAMMA 60 consists in writing in symbolic form a list of all the operations to be carried out on one or several items of information for the particular problem at hand.

A program consists of a combination of several elementary instructions called canonical instructions.

Since information transfers in the computer are discontinuous, the length of an elementary instruction has been assigned as 1 catena; that is the length of the unit transfer in the machine.

Canonical Instructions

When an operation is to be executed by an element of the machine the following specifications must be supplied:

1. Which element is called for?
2. The high-speed memory addresses of the operands and/or the address at which the result is to be placed, stored.
3. The operation to be carried out.

As the program distributor reads a program, it decodes instructions which are stored in successive memory locations. A useful possibility is to be able to jump either conditionally or unconditionally to an instruction not stored in the same sequence of consecutive addresses.

From the preceding, one can see the necessity for four types of canonical instructions:

1. The C-instruction which designates the element called for.
2. The A-instruction which designates the addresses of operands in the HSM.
3. The D-instruction which designates the operation to be performed.
4. The B-instruction or branch which modifies the usual order of instruction execution.

The Program

A single canonical instruction is in general not adequate to define an elementary operation in the GAMMA 60. A sequence of canonical instructions which completely specify an elementary operation constitute a "complete instruction."

A sequence of complete instructions, all relating to the same element, constitute an elementary program sequence.

A sequence of elementary program sequences which are related to the same program function constitute a subroutine.

Some types of program functions are:

1. Subroutines which perform calculation of functions such as the elementary mathematical functions sine, cosine, exponential, logarithmic etc.
2. Subroutines which perform certain functions...input, output, high-speed memory sort routines.
3. Special subroutines which are due to the particular idiosyncrasies of the problem.

The first two types of subroutines are known as general-purpose subroutines and can be grouped into libraries. Stored in large capacity memory, they are available to the GAMMA 60 at all times.

A routine program consists of an ensemble of special and general-purpose subroutines to solve a particular problem, which are assembled through the use of a master program.

As actually carried out, a segment of a master program, perhaps with parallel sequences, is brought into the high-speed memory. Various subroutines are then brought into HSM as they are needed and jumps back and forth can be executed between the subroutines and the master program as necessary.

Generally to facilitate the coding of problems for the GAMMA 60, a symbolic instruction code is used. The symbolic instructions are then translated by the computer into canonical instruction.

There are two types of symbolic instructions. The first type is a symbolic representation of one canonical instruction. The second type is the symbolic representation of a sequence of canonical instructions. (Such instructions are processed by the master program and become program jumps to special and general-purpose subroutines.)

The symbolic code has been designed to facilitate as much as possible the work of the programmer. The instruction vocabulary is simple, and mnemonic abbreviations have been used. For example, the instruction which calls for comparison of two program sequences is written in symbolic code COMP. Operation of the arithmetic calculator is programmed ARIT.

As an example, to execute a high-speed memory-to-magnetic drum transfer of 12 catena starting with HSM address 108 to magnetic drum addresses starting with 17,400, the program is written:

```
DRUM L 108
DSA 17,400
TMD 12
```

DRUM 1 selects magnetic drum No. 1. L loads the address 108 in the current address register of the Magnetic Drum. DSA specifies the magnetic drum address where the information is to be stored. TMD is an operation code which specifies the operation and the number of catena to be transferred.

When this program is brought into the GAMMA 60 it is translated from its symbolic form into an internal code which is intelligible to the program distributor. An important characteristic of the GAMMA 60 is the use of relative addressing in writing subroutines.

As in the case of the design of the symbolic code to facilitate programming, the numbering of symbolic instructions has been designed in such a way that the programmer does not have to be concerned, when he writes his program, with the addresses in high-speed memory from which his program is to operate. When written by the programmer, the instructions of a subroutine are simply numbered from 0 to n. It is the master program which causes the GAMMA 60 to carry out the transformation of relative addresses into absolute addresses once the actual high-speed memory allocation is decided upon.

Additional programming flexibility is provided in the GAMMA 60. It is possible to write instructions which specify, not
the actual address of an operand, but the address at which the address of the operand can be found. This operation is called substituted or indirect addressing.

Checking in the GAMMA 60

Internal checking in the GAMMA 60 is extended to all data processing functions, from the input of data to the output of results.

These distinct problems must be solved.

1. Internal data transfers must be checked.
2. Data processing operations must be checked.
3. Data input and output operations must be checked.

INTERNAL DATA TRANSFER CHECKS

In internal code in the GAMMA 60 associated with each catena is a check digit. This check digit is the remainder of the division by 7 of the catena considered as a pure binary number of 24 bits. Three binary bits are required to represent the check digit. Every time a catena is transferred the checked digit is recalculated and is compared with check digit accompanying the catena. The check digits must be equal; if not, a faulty transfer has occurred. This type of check provides remarkable security. (Note: It is well to specify that a catena in the high-speed memory, or in any register in the GAMMA 60, necessitates the memorization of 27 binary bits and not 24 as has been assumed up to now. Each block of the HSM actually contains 27 core planes and not 24.)

CHECKING OF DATA PROCESSING OPERATIONS

The check of data processing operations is based on the fact that the same operations are carried out on the data and the check digits separately, then the following rule is applied. The check digit of the result of an operation must be identical with the result of the same operation carried out on the check digits of the operands. More briefly stated; the check digits of the result must equal the result of the check digits. If this is not true then the operation is in error.

CHECKING INPUT AND OUTPUT

With the exception of punched paper tape on which each character is punched with a parity check bit which makes it possible to check the information punched as it is being read, information on external media (punched cards, printed characters) not possessing check digits must be checked during read-in on printout by a different procedure from that previously described.

MAGNETIC TAPE UNITS

When reading magnetic tapes, each catena stored with its corresponding check digit is checked for validity. When writing on magnetic tapes, a read head provides simultaneous reading thus allowing immediate checking. If an error is detected, automatic rewrite occurs; if the drop-out cannot be corrected, the corresponding block on the tape is inhibited to further use. This allows use of imperfect tapes.

CARD READERS

Each card is read by two sets of brushes and the results of both reads are compared. In case of disparity, a third set of brushes makes it possible to re-read the card. The information from the third read should validate one of the first two reads. If the information read by the third brush still does not match, the card is rejected into a special bin and an error indicated.

CARD PUNCHES

On the card track after the punch there is a set of read brushes which are used to check-read the card just punched. The check is performed by comparing the information read with that just punched.

PRINTERS

The motion of each print hammer trips an electrical pulse, which by comparison with the position on the print cylinder makes it possible to identify the character printed. This "echo" is compared with the information that was to be printed.

ERROR ROUTINES

All checks are performed automatically and locally and are of no concern to the programmer.

Once an error is detected, it causes the execution of an error routine. The machine pinpoints the error and tries to correct it.

If the error is due to the malfunction of an element, then the computer cannot remedy the situation. The computer is stopped (partially or completely) with the type of error and source indicated on the console of the GAMMA 60. Rapid trouble shooting and maintenance can then be performed.

CONCLUSIONS

The GAMMA 60 is a universal computer, the power, the flexibility, and the speed of which make it an indispensable management tool.

The possibility of connecting to the GAMMA 60 functional elements in proportions and dimensions tailored to the application make it possible to assemble, at any time, a computer suited to the exact needs of the user.

The possibility of executing several problems simultaneously gives the GAMMA 60 great flexibility and power (see Figs. 6–9).

In addition to these features, the sturdy design of the hardware of the GAMMA 60 and the extended use of transistor circuits make it a remarkably trouble free and reliable machine.

Discussion

Question: When will the GAMMA 60 be available? How many machines are already ordered?

Mr. Dreyfus: The prototype will begin operation in June 1959 and will be complete and in full operation in October. The first two deliveries are scheduled to take place before the end of 1959.

As of today we have seven firm orders for systems in Europe.

Question: When will the GAMMA 60 be available in the United States?

Mr. Dreyfus: Taking into account reservations and commitments, deliveries here are about 2 years off.

L. L. Auerbach (Auerbach Electronics Company): What provisions are available for real-time input and output in the GAMMA 60?

Mr. Dreyfus: Real-time computation can occur both in data processing and industrial control applications, not to speak of military and engineering real-time problems. It is easy to handle these problems on the GAMMA 60, for at any time, and whatever the machine is actually busy on, one may pick up a different processing routine which will time share the machine with the other programs on hand. This routine has access to all the processing power of the systems, searching files, computing, making decisions, editing, and obviously, input and output. One such application would be a permanent inventory control where the main file need not be on fast, but limited-capacity, random access media but on ordinary magnetic tape, this being up-dated permanently with a maximum of 15 minutes between keyed input and actual posting with eventual response.

Another type of real-time computation will occur in control processes. In this case, information is coming in at random through proper input devices. The information handling process will be initiated by the input device without any delay even if other processes have been started almost contemporaneously and are not yet completed. The machine then simulates a multiple processing system although re-
The GE-100 Data Processing System

R. H. HAGOPIAN  H. L. HEROLD  J. LEVINTHAL
J. WEIZENBAUM

A T THE 1955 Eastern Joint Computer Conference, members of the Stanford Research Institute described a developmental prototype of a business data processor for accomplishing bank checking account bookkeeping. This paper describes the electronic data processing system evolved in the course of converting the developmental prototype into a production system capable of meeting the requirements of both satisfactory operation for banking and manufacturability.

The banking problem may be defined as receiving handwritten checks from a customer, converting the information on the checks to posted entries in a ledger, and periodically sending a copy of the ledger entries to the customer as a statement. The checks must also be sorted and filed so that they may be included with the customer's statement (see Fig. 1).

It is essential that the system be able to process the daily volume of entries in no more than 24 hours so that processing may be kept current. This must be done in spite of business fluctuations or machine down time.

A further time limitation is the determination within a fixed time (imposed by local clearing house regulations) of checks that cannot be honored for reasons such as insufficient funds or stop-payment actions since bad checks must be returned rapidly enough to find the cashier and permit the cashing bank to both recover the funds it has expended in cashing the check and to prevent loss of interest on these funds during this period. This deadline also must be met in spite of mutilated checks, checks without identifying account numbers, operator error, or equipment malfunction.

The system design, including operating procedures, programs, and equipment, has been governed to a great extent by the necessity of meeting these two time requirements while working with prime documents that are essentially identical to those now in use. The system's components include input equipment, processing equipment, and output equipment (see Fig. 2). The most important input equipment is the document handler, which reads and sequences the entries. Additional input devices are the Flexowriter and the photoreader for name and address changes. The central processor performs the computing and data processing for the system, and the output is provided by the high-speed printer. Magnetic tape units provide high-speed auxiliary storage.

Document Handler

The development of a paper handling system which sorts the items and converts the transaction data into computer language represents the most important technological breakthrough in the design of a system which will meet bank deadlines. Because this system has the unique feature of utilizing the source document itself as the means of direct input to the central processor, the document handler unit may be properly regarded as the key to the General Electric Company's GE-100 system. The major components of the document handler are a correlaton character reader, 12-pocket document sequencer, and sequencer control.

In the bank checking account application, all of the necessary transaction data are printed in magnetic ink in a single line along the bottom of each check (see Fig. 3). When a customer writes a check which has been preprinted with the customer's account number, it is ultimately returned to the bank and the dollar amount is encoded with magnetic ink during the proofing operation when the bank verifies the dollar total of checks received. Bundles of checks are then placed in the feeder station of the document sequencer and are automatically fed past the magnetic read head of the character reader. The transaction data are read electronically and transmitted to the central processor by means of the sequencer control unit, which converts the single character output to the modified binary-coded decimal notation used in this system. The checks are automatically distributed to sequencer pockets as designated by the central processor on the basis of validity, readability, and other information on the item.

After the first pass of the bank checks, during which the data are captured by the processor for later processing, the character reader and document sequencer subsystem may be placed in off-line operation for automatic fine sorting of the checks by account number, transaction code, or transit routing code. Pocket decisions are made by the sequencer control during the off-line operation.

The automatic processing of checks requires that the printed information must be read reliably from folded, crumpled, and soiled documents. The system must function even when the printed information is overstamped or overwritten with ordinary ink or pencil, and it must operate satisfactorily regardless of the color or tinted pantograph on the checks. Finally, any one of the conventional printing processes may be used to print the checks.

The GE-100 character reader is a completely transistorized unit capable of reading 14 characters (ten digits and four special symbols). A block diagram of the character reader is shown in Fig. 4. The printed characters are first magnetized to saturation in order to erase any previous magnetic history. As they pass the reading head, they generate unique voltage waveforms which can be identified with each of the characters.

The waveforms are amplified by a low noise preamplifier and amplifier and then fed into a lumped constant tapered delay line. The delay is sufficient to encompass the longest waveform generated by the widest character. From the delay line onward, the circuitry divides into 14 parallel recognition channels and control functions. The stored waveform is sampled at a number of points along the delay line.