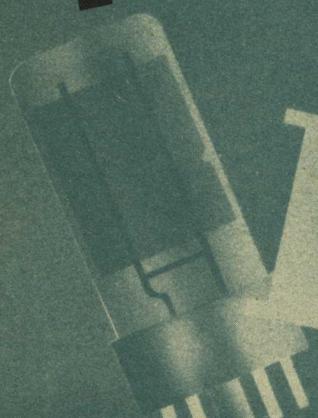
Another
NORTHROP
DEVELOPMENT

MADDIDA*

TRADE MARK



The New DESK-SIDE

Digital Differential Analyzer

PATENTS APPLIED FOR

NORTHROP

* Pronounced "MAD - DĪ - DA"

MADDIDA

DIGITAL DIFFERENTIAL ANALYZER

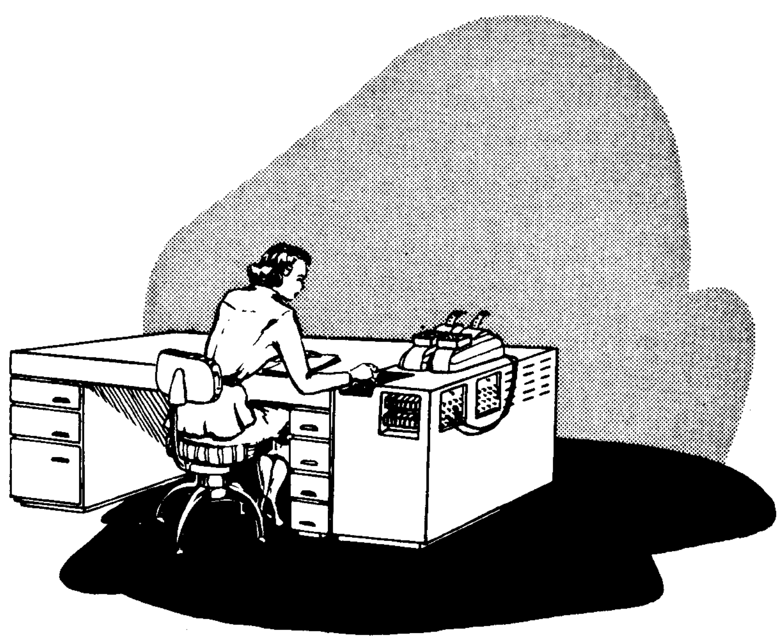
DEVELOPED BY

NORTHROP AIRCRAFT, INC.
Hawthorne California

DECEMBER, 1950

BROCHURE NO. 38

what is maddida ? . . . . . .



WHAT IS MADDIDA?

MADDIDA (rhymes with "mad Ida") is a digital differential analyzer employing a magnetic drum type of memory. It derives its name from the capitalized letters in its description as a MAgnetic Drum DIgital Differential Analyzer. MADDIDA was developed by Northrop Aircraft Inc., and is being released for general use because of its tremendous value in many different fields of science and industry.

Because MADDIDA combines large mathematical capacity, phenomenal accuracy, extreme ruggedness and reliability, and amazingly small size and compactness at a cost that places it very low in the range of moderately priced electronic computers, it makes accessible to academic institutions, engineering firms, and industry at large, a mathematical tool of tremendous usefulness.

With the advent of the new era of electronic computers, old problems are being reviewed and new problems developed that formerly were incapable of accurate solution through sheer lack of time and manpower. Now, a new mathematical approach to the solution of engineering and research problems is gradually replacing the old methods of approximation of trial and error in many fields, but the prohibitive cost of the conventional high speed electronic computers make them unavailable to any except Government agencies. Occasionally those groups with legitimate problems of high priority and importance are able to obtain some computing time on these high priced and extensive facilities, but the available time is always limited and the period of waiting for access to the machines often extends into months.

MADDIDA now places within economic reach of almost everyone in need of such a tool, a differential analyzer type computer that can serve not only for the computation and solution of many mathematical problems but also as a tool for teaching students this new field, and as a tool capable of performing control functions as well as computations. The ease and simplicity of using this machine will undoubtedly bring to light many other uses for this small size but large capacity computer.

Being a differential analyzer, MADDIDA may be used as a computer, as a trainer, as a control, or as a design instrument.

As a computer, it will furnish the solutions to ordinary differential equations or sets thereof, either linear or non-linear, with available and optional accuracies far in excess of any differential analyzers now existing.

As a control, the input can consist of yes or no information or continuous information such as the observation of incremental changes of instrument readings. The output can be used to start and stop motors at prescribed intervals, turn switches on or off in accordance with computed results, and perform any control operable by a yes-or-no, on-or-off type control, actuating them in a pattern obtained as a result of computations performed by the machine on input data.

As a training instrument, it is economically feasible for any academic institution to maintain one of these portable but versatile computers as a means of training and familiarizing students with techniques in the new and rapidly advancing field of computers.

The versatility of the logical design permits the use of this instrument even as a means of checking the design of computing equipment.

WHO MIGHT USE MADDIDA?

COLLEGES AND UNIVERSITIES

In the departments of mathematics, physics, and other sciences as a differential analyzer type of computer. In the department of mathematics, electronics, or computer design as a laboratory training tool for educating students in the design and operation of electronic computers. In the postgraduate groups as a tool in the development and solution of new problems in research.

ENGINEERING

In engineering firms and the engineering departments of large industrial firms for the solution of complex problems in engineering design such as stresses in large structures, vibration and windload calculations, heat transfer problems, etc. Particularly useful in the preliminary tentative analysis of new theories reducible to differential and integral equations.

AVIATION INDUSTRY

For the solution of complex aerodynamic problems such as flight trajectories of missiles; aeroelastic problems of wing and body design; flutter and vibration problems; control problems; jet, rocket, or other propulsive device type of design problems; stress and structure analysis, etc.

INDUSTRY

For the solution of industrial control problems such as production control, quality control, direct continuous control of chemical or physical processes and general research such as heat or fluid flow and its control, etc.

WHAT MADDIDA 44A DOES

MADDIDA is an example of the application of a new design technique in computers developed by Northrop Aircraft, Inc. While the computer is a giant in mathematical capacity, its physical size and number of components are extremely small. This reduction in components is important not only from the standpoint of initial cost and upkeep, but is most important from the standpoint of reliability. Northrop experience in operating preliminary models has shown that these computers will operate for months at a time without error or breakdown. This type of performance has been hitherto unheard of for large scale automatic computers.

Mathematically, MADDIDA will solve the same type of problem as will an analog differential analyzer, but with much greater accuracy. In general, the computer will solve any ordinary differential equation of any order or degree, linear or non-linear, or any simultaneous set of such equations. It will also solve integral equations, transcendental algebraic equations and simultaneous sets of such equations. Some partial differential equations can be handled with special techniques. There is no restriction as to the complexity of equations solvable other than that imposed by the finite number of integrators available in the computer.

One advantage of this machine is that in setting up a problem to be solved, differential equations need not first be reduced to difference equations as is the case with other types of digital computers. The differential equations may be taken as they stand and coded for the machine in a very rapid and straight-forward manner.

The fundamental operation in MADDIDA is numerical integration, although its 44 integrators are capable of performing other operations such as addition, comparison, multiplication, etc.

During computation, information is transmitted between integrators in the form of incremental changes in variables. Each integrator has a dx input through which it receives incremental changes in an independent variable x; a dy input through which it receives changes in a dependent variable y; a dz output through which it delivers incremental changes in a new variable z, which bears the relation:

$$z = \int y\, dx$$

The source of dx may be the output of any one of 42 other integrators, or its own output, or any one of 12 empirical input channels. The source of dy may consist of the algebraic sum of from one to seven of the above sources in any combination. The dy input may also be omitted when using the integrator as a constant multiplier. In cases where the equation is such that dx must also consist of the sum of several channels, an additional integrator is used, coded as a simple adder.

Computation within the computer is done in the binary number system, since this leads to more compact and reliable circuitry. This means that initial conditions must be typed into the computer as binary or octal numbers; however, output devices are available which will tabulate the results of MADDIDA computations directly in the decimal system.

The problem of scaling (positioning the decimal point) is much the same as that for an analog differential analyzer, and straight-forward systems have already been developed. Control of the scale in MADDIDA is provided through the ability to arbitrarily designate the number of significant digits to be carried in any individual integrator. In most problems scaling can be conveniently handled in powers of two, making it unnecessary to use constant multipliers to change the scale.

In addition to the operations of integration and addition already mentioned, an integrator may be coded as a servo. This is useful in inverting operations and it is sometimes necessary to achieve proper scaling. Such a digital servo may be set up either as a proportional device with a predictable lag or it may be coded to operate as a tight, on-off servo where the servo error can be kept down to less than one part in a million.

Integrators may also be coded to exercise decision; for example, one might be set to stop all computation when a variable passes through zero or exceeds certain limits, or it might be set to add or drop terms from an equation when one variable exceeds another.

There are no plug boards, nor are there any physical interconnections to make in setting up a problem on MADDIDA. The desired interconnections between integrators are easily expressed as a binary code, and this code is typed into the computer along with initial conditions. The actual interconnection of integrators is done fully automatically in the machine's electronic circuits.

While many functions can be generated within the computer by solving auxiliary differential equations, it is at times convenient to insert purely arbitrary or empirical data into the computation. For this reason 12 empirical input channels are provided in the machine. These input channels may be fed from various input devices. One such arrangement is now being made available as a graph follower with which graphical data is semi-automatically placed on perforated tape. A number of tapes may then be simultaneously fed into the computer through these input channels while computation proceeds. The independent variable (or speed of tape advance) is under control of the computer. Step functions may also be inserted at pre-set intervals with such apparatus.

Also provided in MADDIDA are 12 output channels. Read-out equipment operated from these channels, now available is: A tabulating printer which will accumulate and print results decimally, and an automatic plotter to present results graphically. There are many other devices which can be developed that will operate from these channels*

These input and output channels facilitate the use of MADDIDA as a control device. For example, the input channels might take information directly from instruments, and the output channels might directly operate switches, valves, or motors.

There are many advantages of a small digital computer such as MADDIDA over analog equipment. For one, analog equipment in general is only as accurate as its components, and the magnitude of errors in a computation is often very difficult to determine. In a digital machine the error can be deduced mathematically, since it is not a function of mechanical or electrical components. For another, the on-off character of digital computer circuits eliminates drift problems and

*These and other input and output devices are developed and furnished by Benson-Lehner Corp., West Los Angeles 25, California.

makes them easy to service. In most cases the location of a fault in the machine, such as, for example, a bad vacuum tube, can be deduced mathematically by the running of test problems.

Occasionally, an extremely complex problem may be encountered for which a 44 integrator machine is not sufficient, and in this case two or more MADDIDAS may be intercoupled. Small interconnection synchronizing circuits are available which make it unnecessary to synchronize the memory drums of coupled machines.

While MADDIDAS can be constructed with more than 44 integrators, it is considered less economical to operate a single very large machine than a group of 44 integrator machines which can be coupled and uncoupled at will.

GENERAL DESCRIPTION MADDIDA 44A

THE DIFFERENTIAL ANALYZER PRINCIPLE

In order to understand the underlying principles of MADDIDA, it is first necessary to consider the general process of the differential analyzer type of computer. For the sake of explanation, we will describe first the mechanical analog type of differential analyzer. Referring to Figure 1, a disc is driven at a rate $\frac{dx}{dt}$. The angular position of this first disc represents the instantaneous value of an independent variable x. The disc is geared by friction to a second disc which then turns at a rate $\frac{dz}{dt}$. The relation, or gear ratio, between the two shafts is obviously proportional to the distance from the point of disc contact to the center of the first disc, shown as y in Figure 1. There will also be some constant k involved, which is dependent on the relative sizes of the two discs.

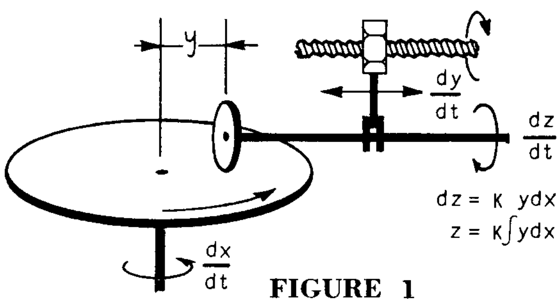


FIGURE 1

First, let us assume y is a constant. If we consider the angular position of the second shaft as representing the instantaneous value of an output variable z, then, neglecting initial displacements, we will always have the relation, $z = kyx$. If, however, during the process of rotation, the screw feed in Figure 1 is turned at a rate $\frac{dy}{dt}$, causing y to vary; and further, if this rate $\frac{dy}{dt}$ is in some manner externally geared to the first disc so that the relation is independent of time, we will have at any instant the relation $z = k\int y\, dx$.

In the case of the electrical analog computer, integration may be performed by passing a current into a capacitor. Since the voltage e across the capacitance c is the integral of the current i into the capacitor, then $e = 1/c \int idt$. The similarity of this equation to the one above is evident with the substitution of 1/c for k. For differentiation, this type of machine measures the voltage across an inductor since this is equal to the derivative of the current through the inductor ($e = Ldi/dt$). Some of these computers find it more convenient to use special feedback amplifiers to perform addition or processes of integration and differentiation.

APPLICATION OF THE DIFFERENTIAL ANALYZER PRINCIPLE

An example of the solution of a simple equation by differential analyzer means is pictured in Figure 2. Input labeled dx corresponds to

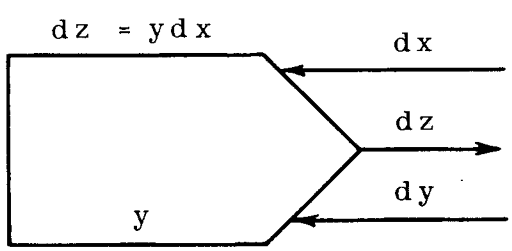


FIGURE 2

the first shaft in Figure 1; output labeled dz corresponds to the output shaft; and the input labeled dy corresponds to the screw feed. We may consider the basic equation of a single integrator to be $dz = y\,dx$. In Figure 3a, we have connected the dz output up to the dy input, thereby causing the relation, $dz = dy$. Substituting in these two equations shows that we have set up an integrator to solve the differential equation $\frac{dy}{y} = dx$. If we assume an initial value of x and place a corresponding initial value of y on the screw feed, as x is increased or decreased the screw feed will maintain the relation $y = e^x$.

More complex equations can be solved by using a number of integrators. A few of the more commonly used connections are shown in Figures 3b, c, d. In fact, there is no theoretical limit on order, degree, or linearity of equations that may be solved in this manner. Simultaneous ordinary differential equations present no difficulty and some partial differential equations can be handled by special techniques.

If provisions are made for the computer to halt when a certain variable becomes zero, then the roots of highly complex algebraic equations can be quickly located. Simultaneous algebraic equations can also be solved provided sufficient integrators or gears are available to handle all the coefficients involved.

THE DIGITAL DIFFERENTIAL ANALYZER

The principal disadvantage of the analog machines just described is that their accuracy is no greater than the accuracy of their components. In the mechanical analog differential analyzer, there is a limit to the accuracy of the machined discs and this limit becomes narrower as wear sets in. In the electrical type, there is a similar limit on the stability of electrical components. Most electrical types also have the additional difficulties: (a) that variables tend to drift over a period of time and (b) that one is forced to use time as their independent variable in most cases, and this is a definite restriction on the type of problem that can be solved.

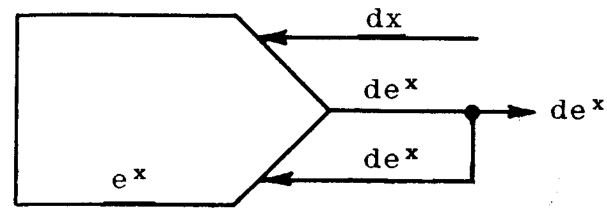


Figure 3 a

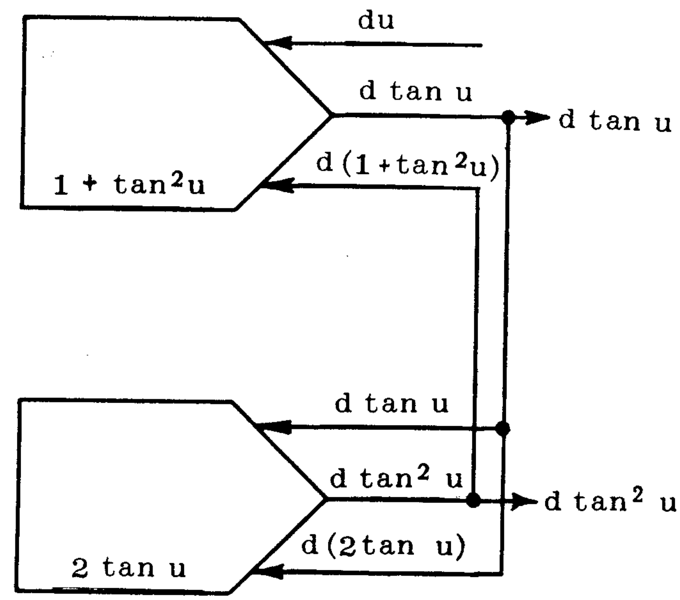


Figure 3 c

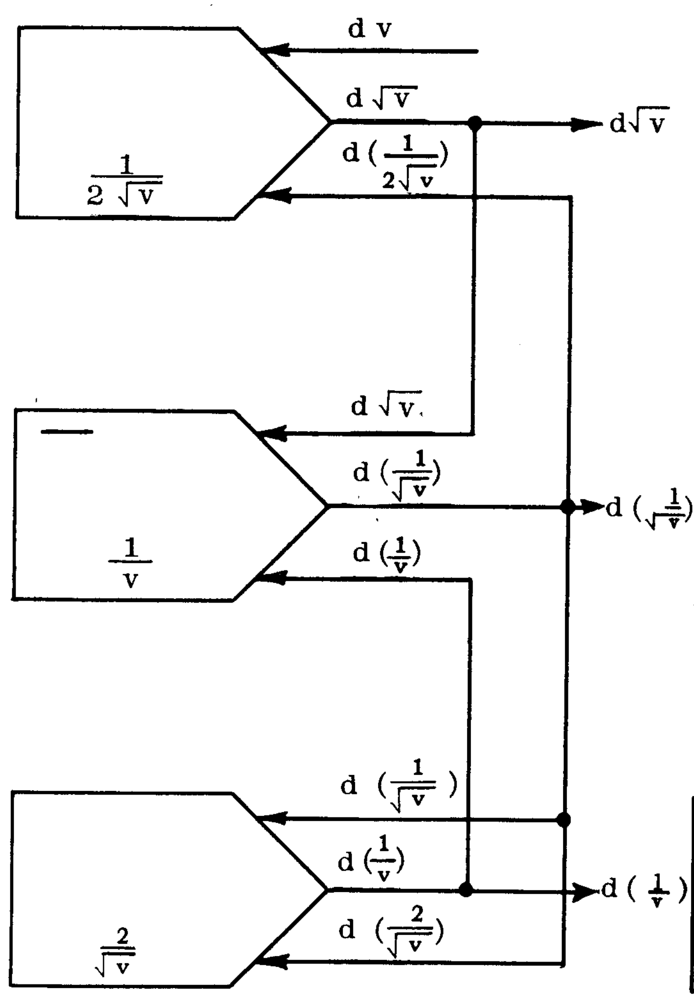


Figure 3 b

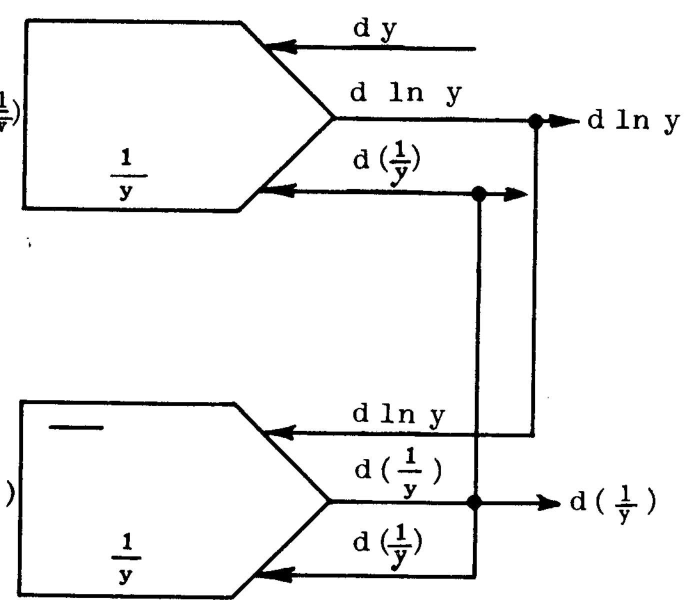


Figure 3 d

FIGURE 3 SERIES

The MADDIDA removes these disadvantages by accomplishing the same results through the transferring of numbers. Since there is no theoretical limit on the number of digits that may compose a number, the accuracy and stability of values in a digital computer may be as high as is desired. To understand the basic principle of the digital integrator, we will picture two numerical registers as shown in Figure 4. The lower register is

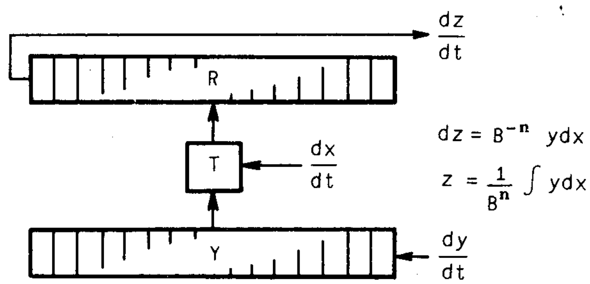


FIGURE 4

accumulating the variable y. It is being fed by uniform pulses at a rate $\frac{dy}{dt}$ and the number y is the instantaneous sum of the pulses received. As integration progresses, the entire number y is being repeatedly added into the number in the upper register R at a rate $\frac{dx}{dt}$. Neither the Y register nor the R register is cleared or reset in the process. The result is that the R register periodically overflows at a rate $\frac{dz}{dt}$. It can be seen that the relation between $\frac{dz}{dt}$ and $\frac{dx}{dt}$ is dependent on the magnitude of y and on a constant which represents the length of the registers.

Again if $\frac{dy}{dt}$ is externally dependent on $\frac{dx}{dt}$, we may eliminate time and consider that at any instant $z = k\int y\,dx$, where x is the total number of transfers that have been made, y is the total number of pulses accumulated in the Y register at any instant and z is the total number of overflows which have come out of the R register. The constant k can be further defined by $k = \frac{1}{B^n}$, where n is the length of the registers in number of digits and B is the radix of the number system used. For example, if we are speaking of decimal registers, B would be 10; since MADDIDA computes in the binary system, B in this computer is two. The number of digits n can be arbitrarily chosen between one and 29.

SPEED OF COMPUTATION

The rate at which MADDIDA 44A picks up and adds binary digits is approximately 100 kilocycles or 100,000 digits per second. Since we have 44 integrators of 32 pulse spaces each, the total recirculation cycle is completed about 70 times per second. This means that a particular variable fed from a single point has a maximum rate of change of 70 units per second, or if it were fed, for example, from seven points, a maximum rate of change of about 500 units per second. Considering this last case, if we were interested in accuracy of the order of one part in 1,000, a variable could go through its entire range in two seconds. If we were interested in one part in 10,000, the same could take 20 seconds, etc. Continuous input and output channels operate up to 70 per second although higher rates of input or output might be obtained by paralleling channels.

Although this computing speed is not as great as that of some of the other large scale digital computers, there are many factors which offset this difference. For one, the fact that MADDIDA does not have to proceed through long cycles of step by step instructions in order to integrate brings up its over-all computing speed. For another, the simplicity of coding and filling MADDIDA vastly reduces coding time as compared to other digital computers. The increase in reliability and reduction in size and complexity gained by the choice of frequencies used makes MADDIDA a practical and efficient tool for automatic computation.

THE TRANSFER OF INFORMATION

For passing information from one integrator to another, a unique system has been developed by Northrop Aircraft which results in the compressed circuitry of MADDIDA. It is easily recognized that pulses sent between integrators must transmit negative information as well as positive. MADDIDA handles this in the following manner: If a steady stream of pulses (or "ones") is fed into the Y register of an integrator the register increases its contents at a maximum rate. If no pulses at all are being fed into the register (equivalent to a stream of zeros), the register subtracts or decreases its contents at a maximum rate. If alternate pulses and spaces (alternate ones and zeros) are fed into the integrator, the register oscillates about a fixed value. For an example of typical operation, a slowly increasing variable would be fed by alternate ones and zeros with occasional extra "ones" thrown in.

So far it is seen how the Y register interprets this information; to understand how the dx transfer system operates and how an integrator puts out its output in this same system, one must understand the handling of the sign of a number within the integrator. Referring again to Figure 4, we will reserve the first space on the left of the Y register for indication of whether y is positive or negative. If y is positive, a one will appear in this space. Y then appears as a binary number starting in the second space from the left and extending as many digits as desired. The termination of y is marked with an additional "one" or "start" pulse which is annexed to the right of the y number. This permits a register to operate to any desired number of significant digits.

Let us first assume y is zero. The only thing present in the Y register will then be a "one" in the sign position. The first time y is added into R this "one" will appear in the first space on the left of the R register. The second transfer that is made will clear R and cause it to overflow. Thus with a y equal to zero, the integrator will put out alternate ones and zeros as one would expect in this system.

As y increases positively, the R register will occasionally overflow twice in a row; when y gets to a maximum, the R register will overflow every time. When y goes negative a zero appears in the sign position and the value of y appears as a complement, that is to say, with ones replacing zeros, etc. As y approaches a maximum in the negative direction, the Y register approaches a condition of all zeros, the result being that the R register never overflows, and the integrator puts out steady zeros.

The dx input is handled in a very simple manner. If the input signal is one, it adds y into R as has been explained, but if the input signal is zero (no pulse) it subtracts y from R.

This completes the explanation of how an integrator handles the sign of its variables and its inputs and outputs with the exception of the fact that an additional pulse space is provided in MADDIDA 44A for sign reversal of an integrator's output. When this space is filled, the output of an integrator is complemented.

In order to explain how information gets from one integrator into another we will set up a simplified version of the computer's Z line. We will consider a string of integrators each taking 32 pulse times to pass through the read or record circuit. We will also assume we have a short line on the drum wherein recorded information remains on the drum for only 31 pulse times and is then read off and recorded again at the same spot where it was originally recorded. As the first integrator goes by, its output, consisting of pulse or no pulse, will be transferred into this Z line and recorded on the drum at that instant. 32 pulse times later the second integrator will be ready to deposit its output on the Z line. Since the Z line represents a delay of only 31 pulses, the first integrator's output will only have come off the line, gone back on the line, and moved over into the second position. This causes the second integrator's output to be recorded on the Z line directly behind the first integrator's output. The third integrator coming 32 pulse times later will, of course, place its output on the Z line after those of integrators 1 and 2, etc. After 31 integrators have gone by the Z line will contain the most recent outputs of all of them. Since the Z line completely recycles during the time that any integrator is passing through the recirculation circuit, it makes available the outputs of all the integrators to the integrator which is passing through.

Obviously the Z line of MADDIDA 44A is more complex than the one described since there are 44 integrators and 12 input lines to be recorded on it. However, the above explanation demonstrates the precessing plan on which it operates.

To pick up the dy inputs for an integrator two counters are used known as the Σdy counters. These counters watch the two address channels L_1 and L_2. The dy address of a given integrator consists of the information as to what must be picked up and added to form the dy input for that integrator. This dy address is placed in the form of zero to seven pulses in the L channels and this entire address is offset one integrator space so that it comes through the recirculation circuits just ahead of the integrator to which it applies.

There are two read heads on MADDIDA 44A's Z line. One Σdy counter simultaneously watches the L_2 channel and the first of the Z heads. The other Σdy counter watches the L_1 channel and the second Z head. When one of these counters observes an address pulse in the L line it promptly picks up whatever information is going through the corresponding Z head at that particular instant. This information will be a one or a zero that has come from an input channel or from another integrator. If the Z line contained a one the Σdy counter will add one to its contents and if the Z line contained a zero at that instant the Σdy counter will substract one from its contents. After scanning both of the L channels over the interval the algebraically combined contents of the Σdy counters are transferred to a stepping circuit. This leaves the Σdy counters free to pick up the address pulses for the next integrator. In the meantime the stepping circuit waits for the start pulse of the integrator now coming into view and on receipt of this pulse it adds its contents into the Y channel of that integrator.

For example we may consider Figure 8. This shows a portion of the initial coding of MADDIDA'S memory. Consider integrator No. 26. The initial y is zero and the initial coding consists only of a pulse in P_{31} indicating y is positive, and the start pulse in P_{15} indicating that the least significant digit of y appears in P_{16}. The dy address of integrator 26 appears in the block labeled 25. As explained above, this offsetting permits the scanning of integrator 26's dy address during the previous interval of time when integrator 25 is in view. The only pulse appearing in the dy section of the L lines is the one in the spot numbered 30. This indicates, simply, that for its dy input integrator 26 uses the output of integrator 30.

In this particular problem it was desired that integrator 26 use for its dx address the output of integrator 31. It will be noted in the dy address section of integrator 26 (occurring in block 25) that the position numbered 31 occurs in the L_1 channel at P_{19}. Therefore, in the dx address section of integrator 26 (occurring in the upper left-hand corner of block 24) the binary number 19 was written in and afterward a 1 to signify the L_1 channel rather than L_2 channel.

More generally, the dx address consists of the binary notation of where the particular input information is found in the dy address section for the same integrator. The dx address is stepped into the dx counter which counts off the pulse times and causes the dx register to observe the Z line at the proper instant to pick up the dx input for the integrator.

GENERAL LAYOUT

So far in our description, we have visioned registers physically laid out as such. Actually, binary addition or subtraction can be performed one column at a time and it is therefore not necessary to have entire registers present in the electronic circuit of the machine at one time. In MADDIDA the registers of integrators are nothing more than spaces on the magnetic drum. Numbers from these registers are read off one column at a time into the adding and filling circuits and the altered numbers are replaced on the drum without delay. The integrators are serviced serially in this manner, all 44 integrators being serviced in each revolution of the drum. Actually they are serviced at a somewhat higher rate than the drum turns because they are not replaced on the drum in the same spot from which they are removed.

The memory drum uses six channels. One of these is a permanently recorded clock channel which keeps the machine in synchronization. Although the drum may be driven by a synchronous motor, variations in line frequency do not cause the computer to slip out of synchronization. Two channels hold the Y and R information, corresponding to the two registers already discussed. Two more channels which we refer to as L_1 and L_2 contain hook-up information and specify what problem is being solved. A sixth channel referred to as the Z line is used for intercommunication between integrators and transmits this information in accordance with the data contained in the L channels.

All information is placed in the memory initially by means of a binary typewriter on the computer's control panel. Suitable circuitry allows the information to be typed in as slow or as fast as the operator wishes and his filling can be viewed on the oscilloscope as it enters the memory. Mistakes in filling can be easily corrected without "starting over."

Figure 5 shows a simplified functional diagram of MADDIDA 44A. Not all of the blocks shown exist as entities within the machine, but are added here for the sake of clarity.

Starting with the six channels of the memory, the permanent clock channel (C) is used for the general synchronization of the entire computer. The Y channel information takes four routes. One takes it to the start flip-flop (S) which goes on after the start pulse occurs in the Y channel of an integrator, and goes off again after P_{31}. In order to recopy the start pulse, the Y channel is recorded directly back into the memory while the start flip-flop is off.

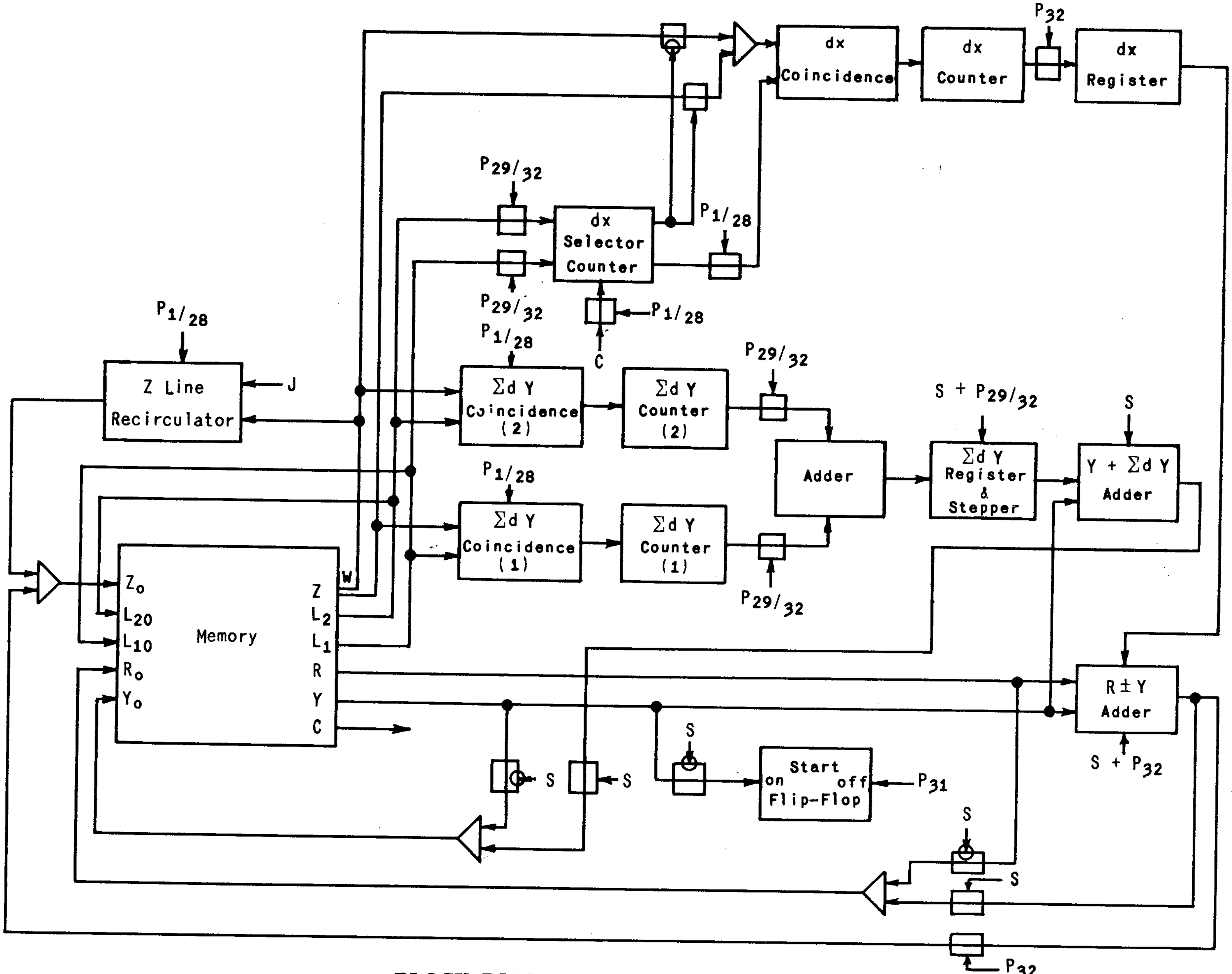
Another route takes Y after the start pulse into the R $\pm$ Y adder where it is added to or substracted from R depending on the condition of the dx register. After the start pulse Y also goes into the Y + Σdy where it is brought up to date and then recorded back into the Y channel. The R channel information goes into the R $\pm$ Y adder where it is increased or decreased by Y depending on the dx register and then recorded back into the R channel. The output from the R $\pm$ Y adder at P_{32} is the output of the integrator and is recorded directly into the Z channel where the latest outputs of all the integrators as well as the empirical inputs (J) are maintained by the Z line recirculator. R is also recorded while the start is off in order to recopy the output sign reversal pulse, if any, at P_{32} in the R channel.

There are two coincidence circuits which observe the two L channels and the two Z heads. Whenever an address pulse comes out of an L channel the current condition of the corresponding Z head is transmitted into one of the two Σdy counters. This process occurs from P_1 through P_{28}, and during the next four pulse times the contents of these two counters are summed into the Σdy register and stepper through the adder. This clears the counters to pick up the Σdy for the next address. The total Σdy increment is held in the Σdy register and stepper until the start pulse. Following the start pulse it steps into the Y + Σdy adder. The L channels also feed into the dx selector counter. This counter picks up the binary number dx address listed in the L channels from P_{29} through P_{32}. The counter then counts downward from that number during P_1 through P_{28}, and when it reaches one, the condition of the proper Z head is gated through a coincidence circuit into the dx counter. In this manner the dx input can be selected from any desired source. At P_{32} this information is gated into the dx register, thus clearing the dx counter to pick up the next dx.

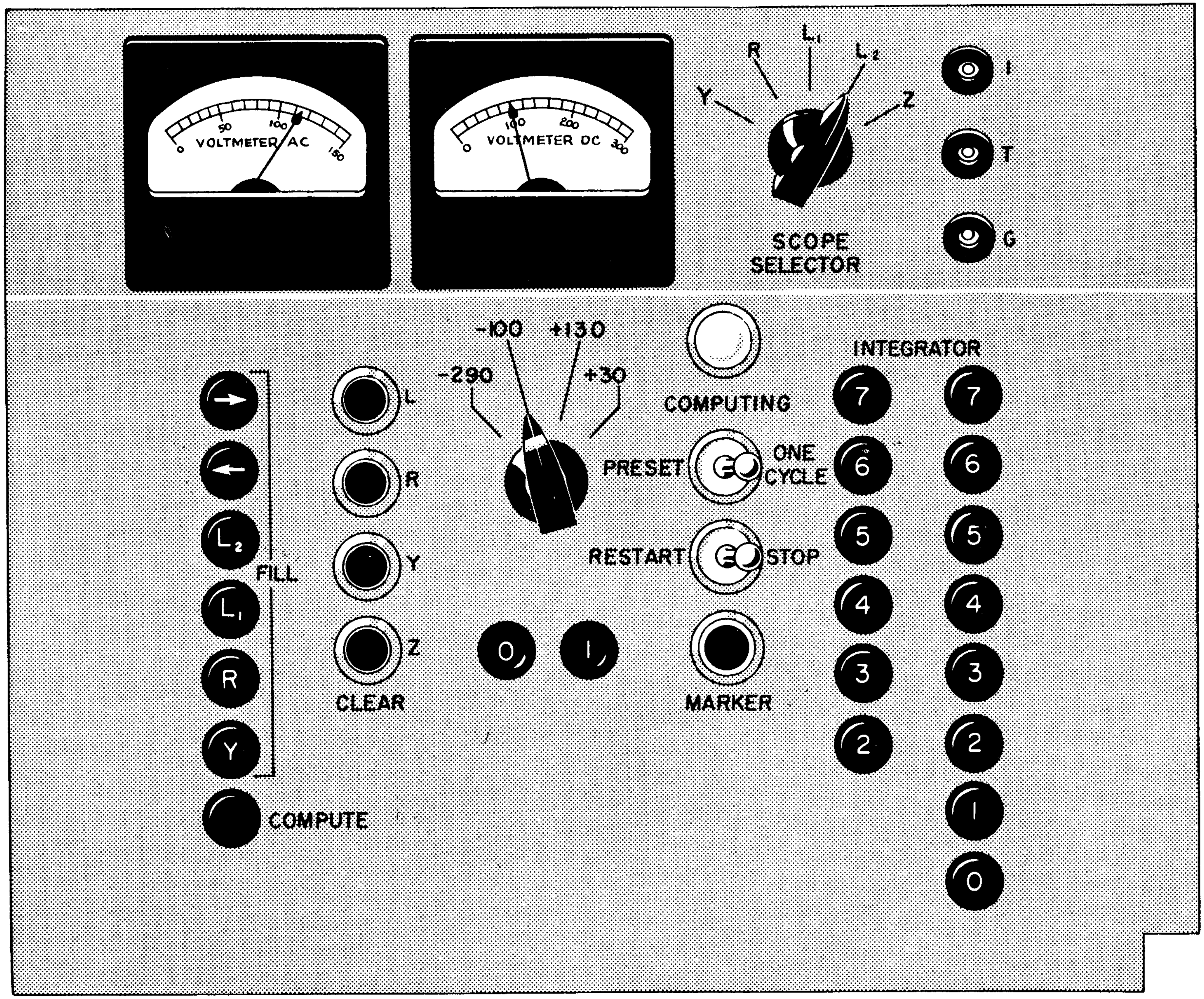
OPERATION OF MADDIDA 44A

Figure 6 shows the control panel of the MADDIDA 44A computer. A meter is provided for checking the AC line voltage to make sure that it is within proper limits. A second meter in conjunction with a four-step selector switch allows one to check each of the four direct current voltages in the computer's power supply. These two instruments are merely for convenience since internal protecting relays automatically cut off the computer when these voltages are for any reason incorrect.

At the upper right of the control panel are the connections for the oscilloscope. An adjoining five-point selector switch allows one to view Y, R, L_1, L_2 and Z channels respectively. The particular integrator whose chan-



BLOCK DIAGRAM OF MADDIDA 44A
FIGURE 5



MADDIDA 44A CONTROL PANEL
FIGURE 6

nels the operator wishes to observe is selected on the double bank of push buttons on the right side of the control panel. Using the oscilloscope in this manner directly observes the memory contents and serves as a guide and a double check on the filling of the machine, as well as an observation of the computation as it proceeds.

At the left side of the panel are seven push buttons for various phases of operation. At the right of this bank are four "clear" buttons which clear out memory lines as labeled. In the center of the panel are the "zero" and "one" buttons which comprise the binary typewriter on which information is given to the machine. To the right of these keys is a marker button which will be explained below.

Typical operation might be as follows: After checking voltages to make sure machine is in order, the four "clear" buttons are pushed to clear the memory. The particular integrator to be coded is selected with the integrator buttons, selecting the first digit on the first column and the second digit on the second column. The oscilloscope selector switch is placed on Y and the "fill Y" button is depressed. The empty Y channel of this particular integrator will now appear on the oscilloscope. The marker button is now depressed and the "one" or "zero" key tapped. This throws a marker on the oscilloscope, which serves to keep one's place as he fills in initial conditions. The operator can now type in the initial binary number on the "zero" and "one" keys. The marker will automatically step along as he types.

If during the insertion of this number the operator makes a mistake, he may depress the "shift left" button. Tapping either of the typewriter keys will now cause the marker to back up. He may then again depress the "fill Y" button and go ahead with the corrected number. The corrected information is written over the incorrect without the need of erasing. If, after filling the entire number, the operator notices he has made an error in the first few digits, he may tap the marker button again. This will flip the marker back to the start of the number and save repeated "shift left" operations. He may now place in the correct digits. If at this point he wanted again to go to the right of the number without changing the digits in between, he may depress the "shift right" button. In this condition tapping either of the typewriter keys will advance the marker to the right without altering the information in the channel.

The operator may now fill the R channel in the same manner by the use of the "fill R" button and he may type in the appropriate hook-up information into the L_1 and L_2 channels. After filling and coding all the integrators to be used, he will depress the "compute" button and tap either of the typewriter keys. Computation then is in progress. A warning light is on while the machine is computing and during this time the operator must not press any of the buttons to the left of the typewriter keys. If he wishes to manually halt the computer, he may do so by tapping the marker button. He can now put the machine in a different phase of operation to alter address information, etc.

A toggle switch labeled "restart" is provided. On one setting of this switch the computer, if not manually halted, will compute for whatever interval it has been set, and will then stop and print its results. On the other position of this switch the computer will not only stop and print, but will automatically restart and go through the next interval. The restart may be thrown off during computation, causing the computer to stop at the end of the current interval.

Above this switch is a second toggle for single operation. When this switch is placed on "one cycle," each time a typewriter key is tapped the computer goes once around the 44 integrators and stops. This mode of operation is useful for inching the computation along and observing the trends of the variables before placing the machine into automatic computation.

The operator is free to observe any of the integrators while the machine is in computation simply by depressing the proper integrator selector buttons at the right of the panel and placing the oscilloscope selector switches on the channel which he wishes to observe.

CODING

To illustrate the simplicity of coding a problem for MADDIDA 44A, let us consider the following equation with its initial condition. (The independent variable dx of this problem is assumed to be time, but it could just as easily be any other function.):

$$\frac{d^2y}{dx^2} = -\frac{dy}{dx} + y^2 + \sin y + A \qquad A = -\frac{1}{4}$$

$$x_o = 0 \quad y_o = 0 \quad \left(\frac{dy}{dx}\right)_o = \frac{1}{8} \quad \left(\frac{d^2y}{dx^2}\right)_o = -\frac{3}{8}$$

First, we sketch the equation in form of a block diagram as illustrated in Figure 7. In this diagram each block represents an integrator. The independent variable for each

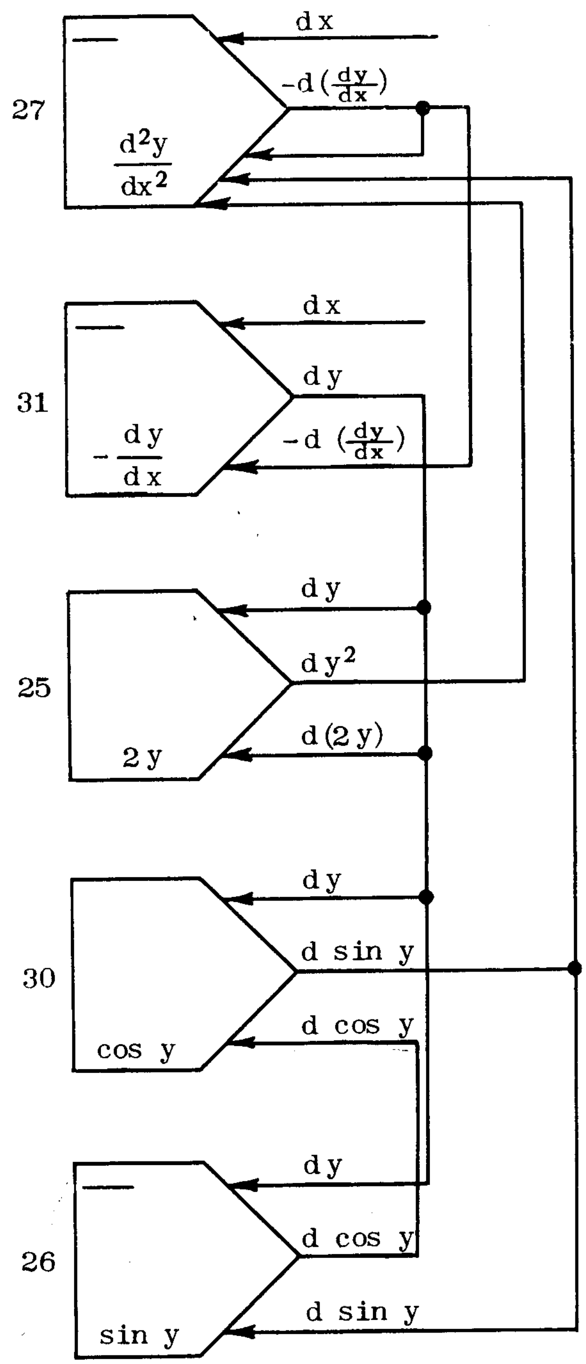


FIGURE 7

integrator is shown on top of the block, the dependent variable on the bottom, and the output in the center. The value represented by the Y-number is written inside the block and a minus in the upper left-hand corner represents a sign reversal (which is obtained by placing a pulse in P_{32} of the R channel of the particular integrator).

Next, we assign arbitrary integrator numbers and now we are ready to fill the coding chart (see Figure 8).

Let us explain only the coding of integrator No. 27 as this illustrates completely the procedure. From Figure 7 we note that the dy of that integrator consists of the outputs of integrators 27, 30 and 25. On the coding chart we will find the dy address for integrator 27 in the 26 block and that integrators No. 27, 30 and 25 are found in L_2 P_{21}, L_1 P_{20} and L_1 P_{21} respectively. Consequently, the only thing required to pick up these integrators as a dy input is to place pulses in these spots.

The independent variable for this integrator is time. We locate time in the dy block for this integrator in L_1 P_{22}. Remembering this, we go to the dx block of integrator 27 (located in block 25) and place in position L_1 P_{30} a pulse if it was L_1 or no pulse if it was L_2. In the other five positions of the dx block we place the P-number where we located the independent variable in the dy block (in this case, 22) in binary form (10110).

Now we are ready to scale our problem. In general the scale of a variable is the ratio between the number seen in an integrator and the true number in real units. We have found it convenient to express this number in the form 2^{S_Y}, and then refer to S_Y itself as the scale.

As the numbers stand in an integrator the binary point is considered to lie between P_{30} and P_{31}. This enables us to read the numbers in the octal number system by grouping the binary digits in groups of three starting with P_{30} and reading to the right. The one exception to this rule is integrator 24 which is the integrator that is reserved for issuing stop and print signals. In integrator 24 only the binary point must be considered to lie between P_{27} and P_{28}.

We will also speak of the scale of increments passing into and out of integrators. For example, if the scale of the output of the integrator is S_{dz}, this means that it will take $2^{S_{dz}}$ pulses to represent a change of one real unit in the variable z. The scales within an integrator are related by the following equations:

$$S_Y = S_{dz} - S_{dx}$$

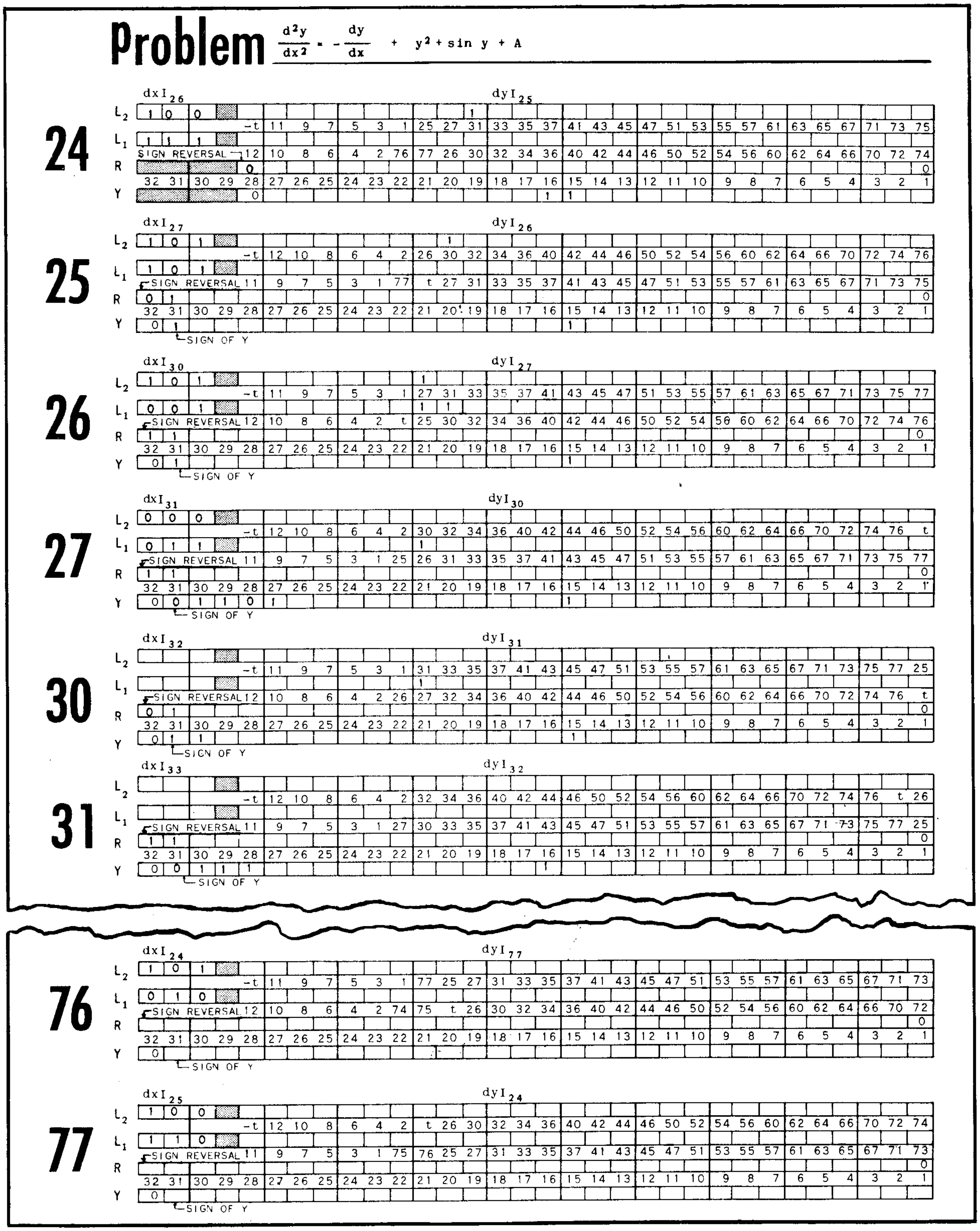
$$\text{Start Pulse Location} = 30 + S_Y - S_{dy}.$$

In the latter equation 30 must be replaced by 27 when dealing with integrator 24 only.

Stating these equations for every integrator will give an easy set of scaling equations as there is a high degree of freedom and also, in most cases, there will be only very few actual equations, as the same scale factor will by necessity appear in very many integrators which are directly interconnected.

In setting up these scaling equations it is required to know the approximate maximum value of the y-number so that S_y can be picked small enough to keep Y from exceeding the capacity of the integrator.

Figure 9 illustrates the complete scaling setup for our problem and gives all the equations which have to be solved. Integrator No. 24 has been set up so that the machine will stop for print after an increment of $\Delta x = 1/8$.



CODING CHART
FIGURE 8

Integrator	S_{dx}		S_{dy}		S_{dz}		S_y Max.	S_y	Start Pulse Location
24	S_1	15				3		-12	15 (or less)
25	S_3	15	S_3-1	14	S_2	14	-1	-1	15
26	S_3	15	S_2	14	S_2	14	-1	-1	15
27	S_1	15	S_2	14	S_2	14	-1	-1	15
30	S_3	15	S_2	14	S_2	14	-1	-1	15
31	S_1	15	S_2	14	S_3	15	0	0	16
t					S_1	15			

Scaling equations: $S_2 - S_3 \leq -1$

$S_2 - S_1 \leq -1$

$S_3 - S_1 \leq 0$

Picking $S_1 = 15$ gives: $S_2 = 14$

$S_3 = 15$

For integrator 24:

$$\frac{1}{2^{S_{dz}}} = \frac{1}{8} \quad \therefore \quad S_{dz} = 3$$

y - number = 1. $S_y = S_{dz} - S_{dx} = -12$

FIGURE 9