# 4. System Design of the SEAC and DYSEAC

A. L. Leiner, W. A. Notz, J. L. Smith, and A. Weinberger

## 1. INTRODUCTION

In the course of developing comprehensive system plans for the SEAC and DYSEAC, certain standard methods and procedures were evolved for producing a large-scale system design. These standard procedures, including first the development of system specifications, then the development of functional plans, and finally the development of wiring plans, are described in this paper. Some of the problems encountered in formulating the specifications and system plans are also discussed [1][1].

The flow of development generally followed in creating such large-scale computers is charted in figure 4.1. As indicated, two sets of factors (which can be considered as the initial boundary conditions of the system-design problem) affect the choice of system features for a machine: first, the set of factors related to the intended use of the machine, and second, those related to the type of components or "building blocks" with which the machine is to be constructed. Because these two sets of factors are basically unrelated to each other, they often present contradictory requirements. For example, a proposed machine feature may appear ideal when evaluated solely in terms of the intended use of the machine but may entail an unacceptable engineering risk when evaluated in terms of component reliability and cost. The necessity for effecting compromises and avoiding conflicts of this kind between the rival claims of operational effectiveness and engineering reliability and economy strongly influenced the system designs of the SEAC and DYSEAC.

As is also indicated in figure 4.1, the principal machine components whose properties profoundly influenced the system design of these computers were the internal memory units, the external communication units, and the internal switching and small-scale storage circuitry. The internal memory units included an acoustic delay-line memory and an electrostatic Williams' tube memory. The external communication units included such devices as mechanical keyboard-printers, magnetic recording units, special cathode ray tube display devices, input converters for translating analog information to digital form, and digitally-actuated output mechanisms. The internal high-speed switching and storage circuitry included the following fundamental digital elements for controlling pulse signals: (1) The *and-gate*, with or without an inhibition input, and the *or-gate* were the fundamental elements utilized for combining or switching pulse signals. Groups of these gates are assembled with an amplifying tube and pulse transformer to make a *pulse repeater*. This pulse repeater carries out the logical switching functions of the gates included in it and also amplifies and restandardizes the signals going through it. (2) The so-called *dynamic flip-flop* was the fundamental bistable device utilized for providing one-bit storage. This device is composed of a pulse repeater and a *delay line* connected to form a closed loop around which a single pulse can be kept circulating repeatedly with a recirculation period of exactly one pulse-repetition cycle.

These elements were used uniformly throughout the SEAC and DYSEAC both for word generation and for central functional control purposes. No other basic elements were used in the internal system. In figure 4.2, items 1 through 6 illustrate these fundamental elements and the symbols adopted for representing them. Table 3 explains their mode of operation.

From these basic elements, small composite units were developed for carrying out typical simple processing operations according to the rules of binary arithmetic. For example, comparators, counters, decoders, complementers, adders, storage and shifting registers were developed. Such units are illustrated in figure 4.2, items 7 through 16. Using these small composite units, larger subsystems were then organized for carrying out more complex arithmetic and control operations such as the arithmetic operations of multiplication and division, or the control operation of selecting

[1] Figures in brackets indicate the literature references on page 92.

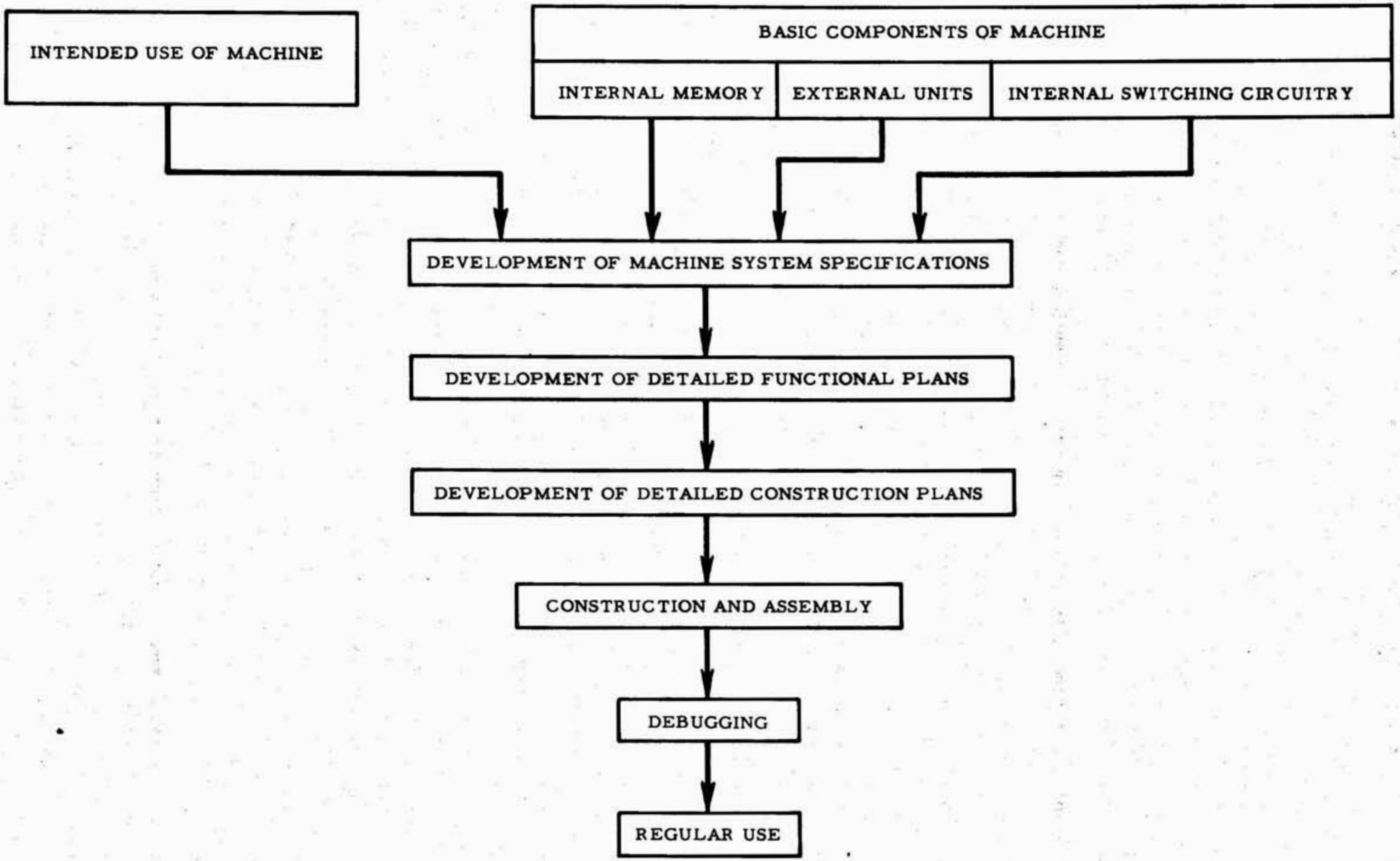

INTENDED USE OF MACHINE
BASIC COMPONENTS OF MACHINE
INTERNAL MEMORY
EXTERNAL UNITS
INTERNAL SWITCHING CIRCUITRY
DEVELOPMENT OF MACHINE SYSTEM SPECIFICATIONS
DEVELOPMENT OF DETAILED FUNCTIONAL PLANS
DEVELOPMENT OF DETAILED CONSTRUCTION PLANS
CONSTRUCTION AND ASSEMBLY
DEBUGGING
REGULAR USE

FIGURE 4.1. *Computer development flow chart.*

a word from a designated memory location. Table 4 lists some typical operations for which such subsystems were developed.

These composite units and subsystems provided a set of basic techniques by means of which computations could be performed on digital data and complex procedures could be employed for integrating large masses of unorganized information. Once devised, they served as a storehouse of building blocks and organization schemes from which more comprehensive full-scale systems could be developed. In this way, they provided the means for fashioning automatic supervisory control facilities capable of directing large families of external devices carrying out complex tasks.

## 2. DEVELOPMENT OF SYSTEM SPECIFICATIONS

In developing system specifications for the SEAC and DYSEAC, an effort was made to specify a balanced system in which each component part was organized to do only what it needed to do and no more. Such a system usually contains the fewest possible parts and consequently is more economical to construct, debug, and maintain. As the characteristics of the principal memory, switching, and external communication units to be incorporated into the system were widely varied, the problem of achieving an effective balance between these units arose. A major boundary condition to the problem was imposed by the engineering decision to use a mercury acoustic delay-line memory for high-speed storage. The access speed characteristic of this type of memory governed the choice of computing speeds for the switching units and input-output speeds for the external communication units. More specifically, a purely serial arithmetic unit was chosen instead of (for example) a serial-parallel or fully parallel unit because in the acoustic memory the recirculation period of 384 $\mu$sec for an 8-word recirculating tank increases the time required to read a word into or out of the memory by seven-sixteenths of this period, on the average, which is 168 $\mu$sec. For the four references to the memory required in most SEAC and DYSEAC arithmetic operations, this lengthens the time needed to execute an operation by 672 $\mu$sec. As the actual basic computing time required to carry out the four sequential steps of a complete addition operation, using simple and efficient serial techniques, is only 192 $\mu$sec, obviously not much over-all gain in speed would be achieved by reducing the arithmetic computing time unless a corresponding reduction could be effected in the memory access time. Table 1 shows the average times for execution of the various types of arithmetic operations performed by SEAC and DYSEAC and the portion of these times occupied by memory access waits.

TABLE 1. Average rates for certain SEAC and DYSEAC operations

| Operation | Performance rate in SEAC | Performance rate in DYSEAC | Percentage of time occupied by memory access waits |
|---|---|---|---|
| | *msec* | *msec* | % |
| Addition, subtraction, logical transfer---- | 0.9 | 0.9 | 78 |
| Multiplication, division------------------ | 3.0 | 3.0 | 23 |
| Comparison-------------------------------- | 0.7 | 0.7 | 72 |
| Shift (half-word shift)------------------- | ------------------ | 1.1 | 61 |
| Justify (half-word shift)----------------- | ------------------ | 2.0 | 33 |
| Summation (per word, for 100 words)-------- | ------------------ | 0.06 | 12 |

From this table it will be noted that addition times and multiplication times are the same for DYSEAC as for SEAC—the speed ratio being about 3:1 for executing these operations. If, as is the case in many computational problems, additions occur about three times as often as multiplications, the speed-up achieved in the over-all execution time of a problem by striking out a given percentage of the operations in the program is the same regardless of whether the operations eliminated are additions or multiplications or a mixture of both. In this rough sense, the speed ratios may be said to be balanced. Actually, the occurrence of multiplication in solving a problem on DYSEAC will be rather less than with SEAC because of the newly added high-speed Shift and Justify operations by which many procedures requiring one complete multiplication time with SEAC can be done much more rapidly with DYSEAC.

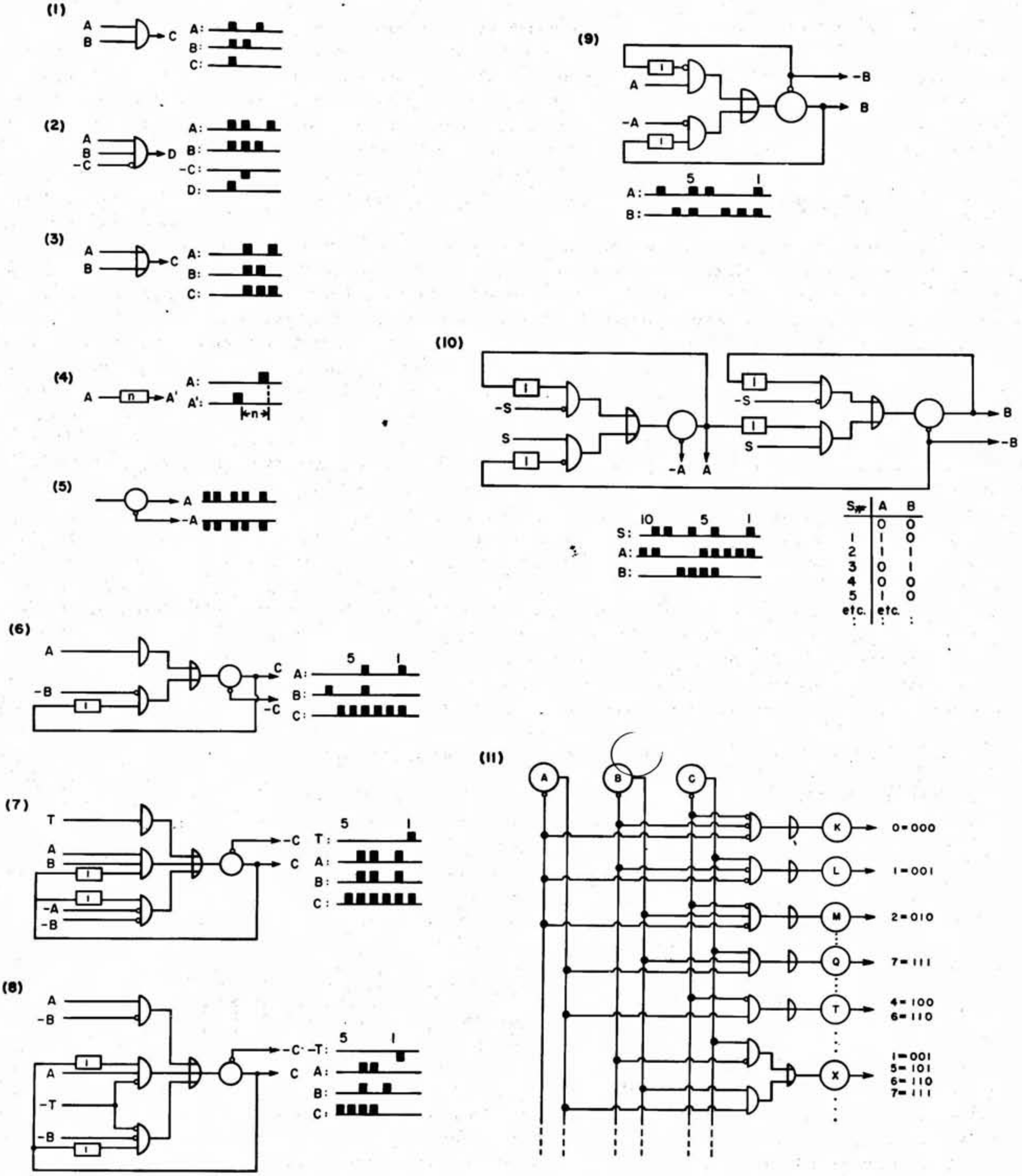


FIGURE 4.2. *Fundamental elements for computer systems.*
(See table 3 for explanation)

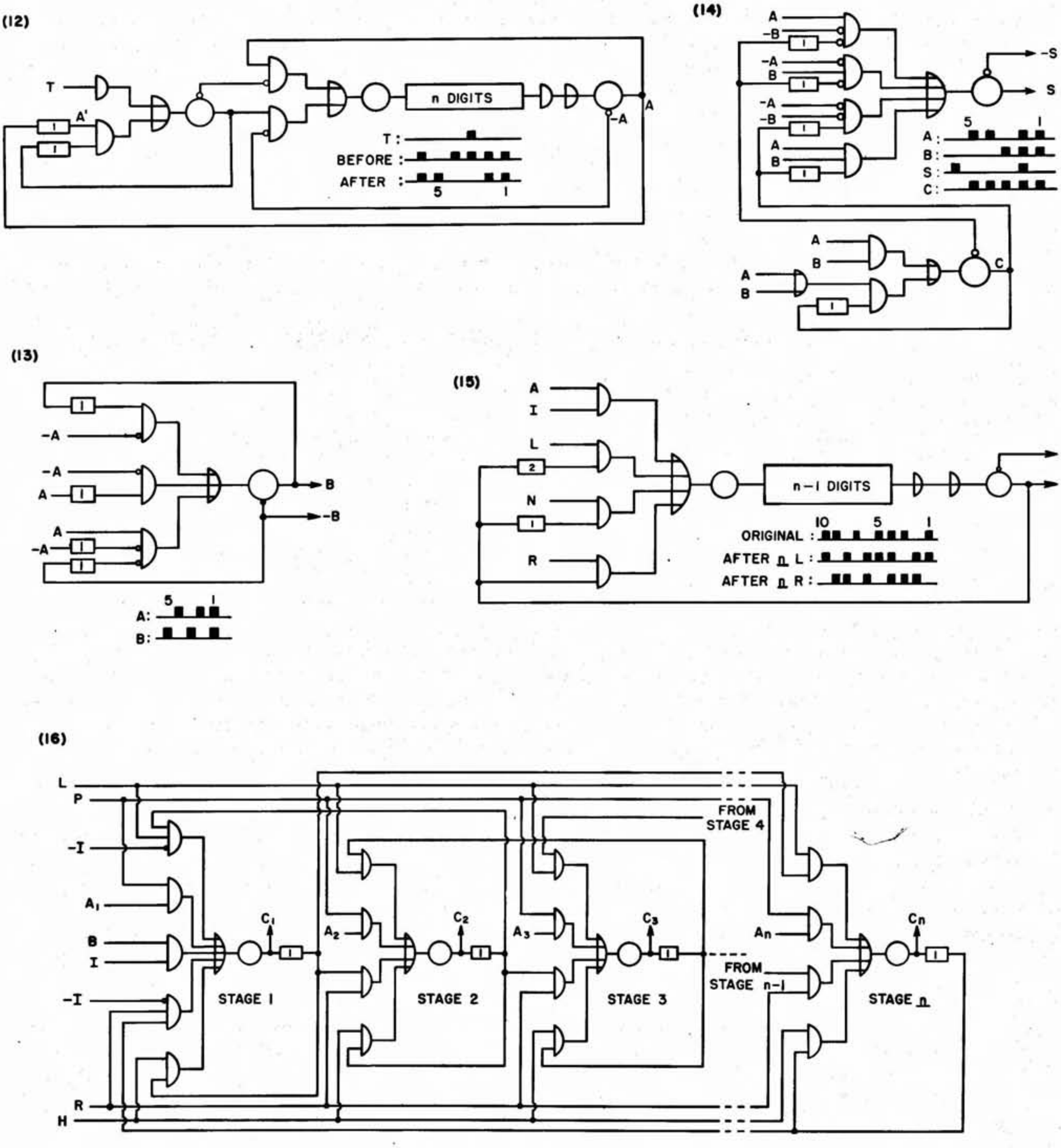


FIGURE 4.2. *Fundamental elements for computer systems*—Continued

(See table 3 for explanation)

Turning now to the balance between computing rates and input-output rates, table 2 shows the time required to transfer a word between the high-speed memory and an external unit operating at the pulse-repetition rates associated with conventional magnetic recording units. As the time required to execute an average instruction is of the order of a msec, it takes four to six times as long to transfer a given instruction word from a typical external magnetic unit into the high-speed memory as it takes to execute it after it has been stored within the machine. Ordinarily, associated with each instruction there are from one to four words of input-output, namely, at least the instruction word itself and possibly any or all of the operands and the result. Therefore, the total input-output time required to transmit these data in and out of the machine would not ordinarily be greater than roughly two dozen instruction-execution times. If each instruction is executed about this many times after it has been inserted into the machine, then the internal processing and external transfers will, in the long run, occupy roughly the same amount of time. In the case of DYSEAC, where computing and input-output transfers can proceed concurrently, the over-all time taken to solve a problem will be equal to the time required for the longer of these two processes.

TABLE 2. Rates for input-output operations with magnetic recording units, averaged over 512 words

| Pulse rate of external wire or tape unit | Time to transfer one word between external unit and DYSEAC memory |
|---|---|
| *kc* | *msec* |
| 4 | 12.8 |
| 8 | 6.8 |
| 12 | 4.8 |
| 16 | 3.8 |

It appears, therefore, that for a high-speed memory of the present acoustic delay line type, little benefit would accrue from increasing these input-output rates unless the problem being solved involves extensive hunting along the wire or tape in search of particular words. If the nature of the application requires such hunting procedures, then a better-balanced system could be obtained by increasing the over-all input-output rate in direct proportion to the average number of words which must be searched through in order to find a desired word.

Other major problems were encountered in formulating system specifications for the SEAC. On the one hand these specifications had to be kept to a minimum so that the machine could be completed at the earliest possible date for interim use; and on the other hand the specifications had to include provision for expanding this interim machine, both internally and externally, by the annexation of advanced high-speed internal memory equipment and external input-output equipment as they became available. Neither the developmental time-scale nor the operating properties of these new components could be precisely defined at the time the specifications had to be drawn up. Furthermore, the future uses of the machine could not be precisely defined either.

Fortunately, at the time the development of the DYSEAC was undertaken the situation was quite different. Even though the range of application and method of use intended for the DYSEAC are broader and more varied than for SEAC, it was possible to define them more precisely at the start of the program. Consequently the DYSEAC system could be organized more effectively than the SEAC; and though both machines contain the same relative proportions of equipment for carrying out communication, control, and processing functions (see fig. 4.3) the DYSEAC is able to provide considerably more powerful operating features using the same amount of equipment. These operating features and their relation to the intended use of the machine are described in the DYSEAC article.

After the problems related to the intended uses of the machines and to the operating characteristics of the principal component parts had been defined and evaluated, and before the final system specifications were formulated, several other analytical studies were carried out. For example, coding studies were conducted to determine optimum internal programming specifications, i.e., what the format and content of the number and instruction words should be, what the program-sequencing procedures should be, what arithmetic and control operations should be included in the system, and other related questions. As a result of such studies, the approximate number of instructions and

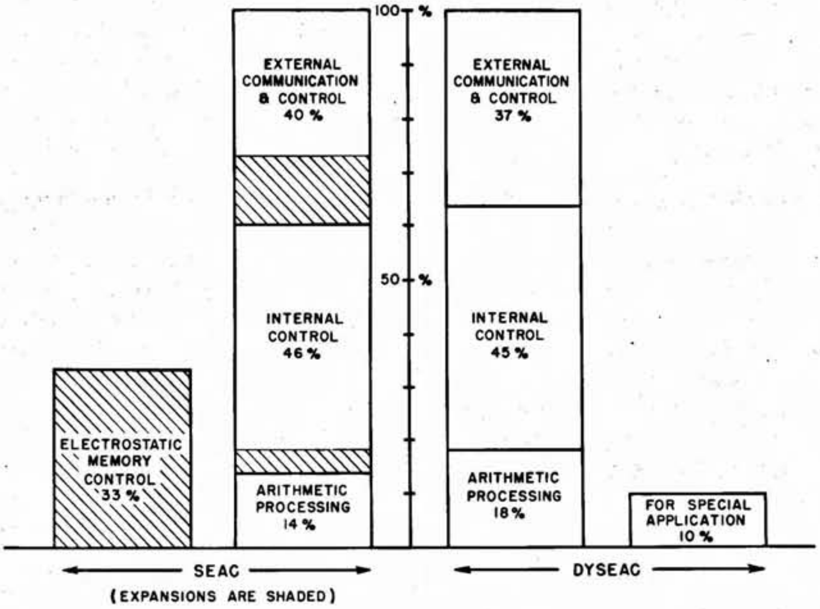


FIGURE 4.3. *Proportion of equipment for performing major functions in SEAC and DYSEAC systems.*

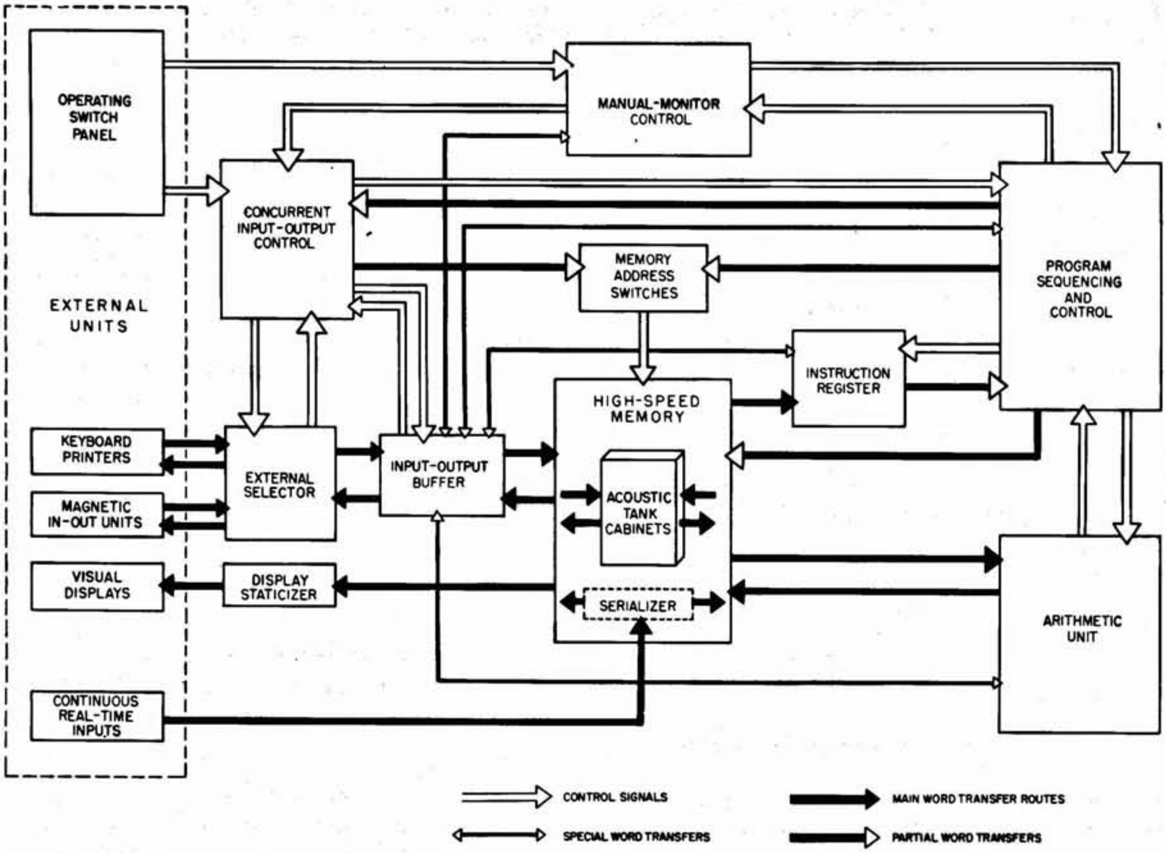


FIGURE 4.4. *Block diagram of DYSEAC.*

memory cells needed for solving various problems can be established, as well as the frequency of access to various classes of words, over-all solution times, etc. These data are in turn useful for carrying out further analyses aimed at determining the relationship between the problem-solving capacity of the system and the equipment cost. For example, by relating the unit cost and access time of various types of storage equipment with the relative frequency of use of the various classes of words involved in the program, the most economical choice of proportions for each type of storage device needed to perform the problem at a given over-all speed can be determined.

It should be noted that a strong interdependence exists between decisions made as a result ot these various considerations. In designing SEAC, for example, the decision to provide for the incorporation of an additional parallel memory as well as the initial serial memory exerted a profound effect on the design of two apparently unrelated parts of the system, namely, the input-output buffering system and the program-sequencing system. The parallel memory was coupled to the otherwise serial machine by means of a staticizing shift register capable of communicating with the parallel memory in a simultaneous broadside fashion from all of its register cells and with the serial units in a step-by-step serial fashion through a single register cell. The flexible serial-parallel conversion abilities of this device together with its ability to operate over a wide range of shift speeds led to its adoption as the input-output pick-up register for the system [2]. On the other hand, the DYSEAC, which does not require this sort of facility for staticizing information, uses the more economical circulating electrical delay-line storage technique for its pick-up register. This register, although purely serial in operation, can be loaded with up to 45 binary digits delivered from an external input device via 45 distinct channels in a broadside fashion.

A second effect of the decision to provide for a parallel memory on SEAC relates to the program-sequencing system. Initially, the memory capacity of the machine was less than 1,024 words, and therefore address numbers of 10 bits each were adequate. Hence, an instruction word containing four addresses of 10 bits each could be used. When it became necessary to provide for expansion of the initial memory capacity up to as much as 4,096 words, extra space was needed in the instruction word to represent numbers in excess of 1,024. To make room for the longer address code designation, therefore, without extending the length of the instruction word, a three-address instruction word of 12 bits per address was adopted as an optional alternative to the four-address instruction word. Along with this three-address designation, a consecutive-number scheme for sequencing instructions was used. This scheme in turn facilitated the adoption of a floating relative-address scheme for program sequencing [3]. These examples illustrate how design decisions affecting a single unit of the system were propagated and made themselves felt throughout the remainder of the system.

## 3. DEVELOPMENT OF DETAILED FUNCTIONAL PLANS

The previous decisions defining the over-all system specifications for the computer to a great extent implicitly determined the general nature of the over-all blocks or major units in the system as well as their control inter-relationships and principal information-transfer routes. At this stage of development, the system is usually represented by the familiar block diagram composed of simple labelled boxes interconnected by means of directional lines (see fig. 4.4).

The next step in the evolution of the system design was to break each of these large blocks into smaller blocks, according to the functions suggested by the general operating specifications. For example, an arithmetic unit might be broken into boxes representing an adder and some storage registers whose number and characteristics depended on the operating specifications. A control unit might be broken into boxes representing subunits whose functions are, for example, the selection, acquisition, and distribution of information from the memory to other parts of the computer, or the issuing of control signals to the arithmetic unit that direct it through certain motions that are required in order to perform a given arithmetic operation. At this stage of development, definite rules regarding the cause-effect relationships among all the basic units became fixed. Also, the time delays imposed by the information processing, that is, the speeds at which signals can make their way through the various blocks, had to be estimated and upper and lower limits specified.

Because the system being designed was centrally synchronous, over-all timing considerations now came to the fore. To use a rough analogy, all the units and subunits had to be so designed that they could be geared together and perfectly meshed. The next step, therefore, was to investigate and specify definitively the exact timing inter-relationships which must exist among all of the major units in order to insure that transfer of data and control signals among them can take place in proper sequence. This step is necessary in order to force the *time* sequence of the passage of signals from one unit to the next to correspond to the desired *cause-effect* sequence through which the units are functionally related to each other. This process is sometimes referred to as "closing the timing loops." For example, with a synchronous machine using a circulating delay line memory, one of the major timing loops to be closed is the so-called memory-to-arithmetic-unit-to-memory loop. Closing this loop means establishing that a pulse read out of a memory location into the arithmetic unit can pass through the arithmetic unit and arrive back at the memory in time to be rewritten into the memory at exactly the proper instant. The proper instant for this rewriting is determined by the requirement that, when the pulse is read out of the memory at a later time, it will emerge from the memory at an instant which is precisely an integral number of word-transfer cycles after the instant at which it emerged on the previous occasion.

Once the major timing loops were closed, it became possible to establish a fixed over-all timing schedule for the system as a whole. That is to say, a definitive pulse-time "time-table" was adopted for most of the major units which communicate with each other and for the over-all timing-pulse generators whose functions are to furnish signals commonly used by all the other units. From this point on, every remaining unit and subunit in the system had to be so designed that it would be capable of performing its functions entirely within the confines of the fixed time schedules adopted for the over-all timing signals.

It is advantageous to establish these fixed over-all time schedules at an early stage of the design program, because this facilitates dividing up the detailed work on the various major units among a team of several designers. So long as each designer adheres to the established timing "boundary conditions," the detailed design of each unit can be carried ahead separately with less fear of drastic reaction from developments in the design of other units. This sharing of design responsibility is feasible, however, only if the timing schedules are fixed at the outset and left unchanged.

In carrying ahead the design of each unit and subunit, the next step was to convert the logical cause-effect relationships which must exist within and among the various subunits into space-time relationships. The cause-effect relationships may be expressed by well-known algorithms for the case of an arithmetic unit, while for a control unit the description of the relationships may take the form of specially devised tables of procedural rules and process flow diagrams. Whatever their form, these cause-effect relationships may be reduced to stylized logical statements; for this purpose the notation of Boolean algebra often provides a convenient shorthand. In order to convert the cause-effect relationships to space-time relationships it was necessary, first, to select fundamental building blocks capable of executing the required functional relationships on incoming pulse signals, and, second, to determine the proper time at which these signals should be transmitted between the individual input and outputs of each building block.

This work was carried out with the aid of *process block diagrams*, an example of which is shown in figure 4.5. These diagrams consist mainly of simple inter-connected rectangular blocks labelled with descriptive titles designating the general logical relationships that they bear toward each other. Each block represents a specific type of subunit drawn from a list of semistandardized prototype units designed to perform various common functions. Typical examples of such units are the comparators, counters, adders, registers, etc., described in the Introduction. From this collection of prototypes, a functional unit specially suited to any particular situation could usually be devised merely by making simple modifications of the standard prototype model.

The manner in which each building block fitted into a unit to perform its system function was further specified by prescribing the times at which it was to be turned on or off by each connected clock. At first, the entire unit was laid out in blocks with timing signals indicated only approximately. As the design process proceeded, the necessary timing specifications became more precisely definable, and the specific timing pulses and exact delay-line lengths needed in order to insure proper time meshing of all the gating stages could be specified.

The next step, which overlapped the previous one to some extent, was to prepare so-called *functional diagrams*, using the symbols illustrated in items 1 through 5 of figure 4.2. Although basically only five varieties are used, these are sufficient to determine implicitly every physical component of the machine. The internal details of each block on the process block diagrams were worked

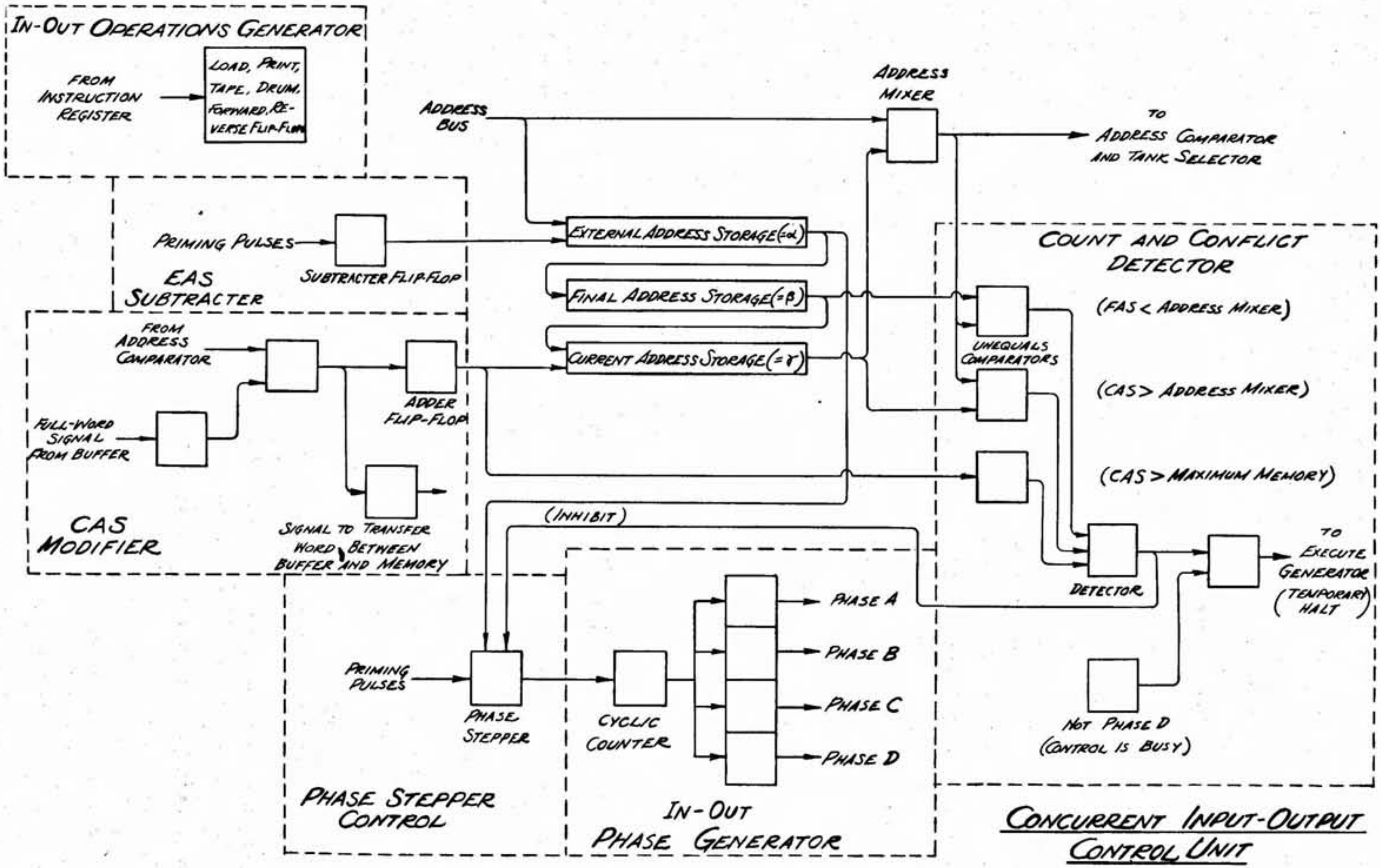

IN-OUT OPERATIONS GENERATOR
FROM INSTRUCTION REGISTER
LOAD, PRINT, TAPE, DRUM, FORWARD, RE-VERSE FLIP-FLOPS
EAS SUBTRACTER
PRIMING PULSES
SUBTRACTER FLIP-FLOP
CAS MODIFIER
FROM ADDRESS COMPARATOR
FULL-WORD SIGNAL FROM BUFFER
ADDER FLIP-FLOP
SIGNAL TO TRANSFER WORD BETWEEN BUFFER AND MEMORY
ADDRESS BUS
EXTERNAL ADDRESS STORAGE(=α)
FINAL ADDRESS STORAGE(=β)
CURRENT ADDRESS STORAGE(=γ)
ADDRESS MIXER
TO ADDRESS COMPARATOR AND TANK SELECTOR
COUNT AND CONFLICT DETECTOR
(FAS < ADDRESS MIXER)
UNEQUALS COMPARATORS
(CAS > ADDRESS MIXER)
(CAS > MAXIMUM MEMORY)
DETECTOR
TO EXECUTE GENERATOR (TEMPORARY HALT)
NOT PHASE D (CONTROL IS BUSY)
(INHIBIT)
PRIMING PULSES
PHASE STEPPER
CYCLIC COUNTER
PHASE A
PHASE B
PHASE C
PHASE D
PHASE STEPPER CONTROL
IN-OUT PHASE GENERATOR
CONCURRENT INPUT-OUTPUT CONTROL UNIT

FIGURE 4.5. *Example of a process block diagram.*

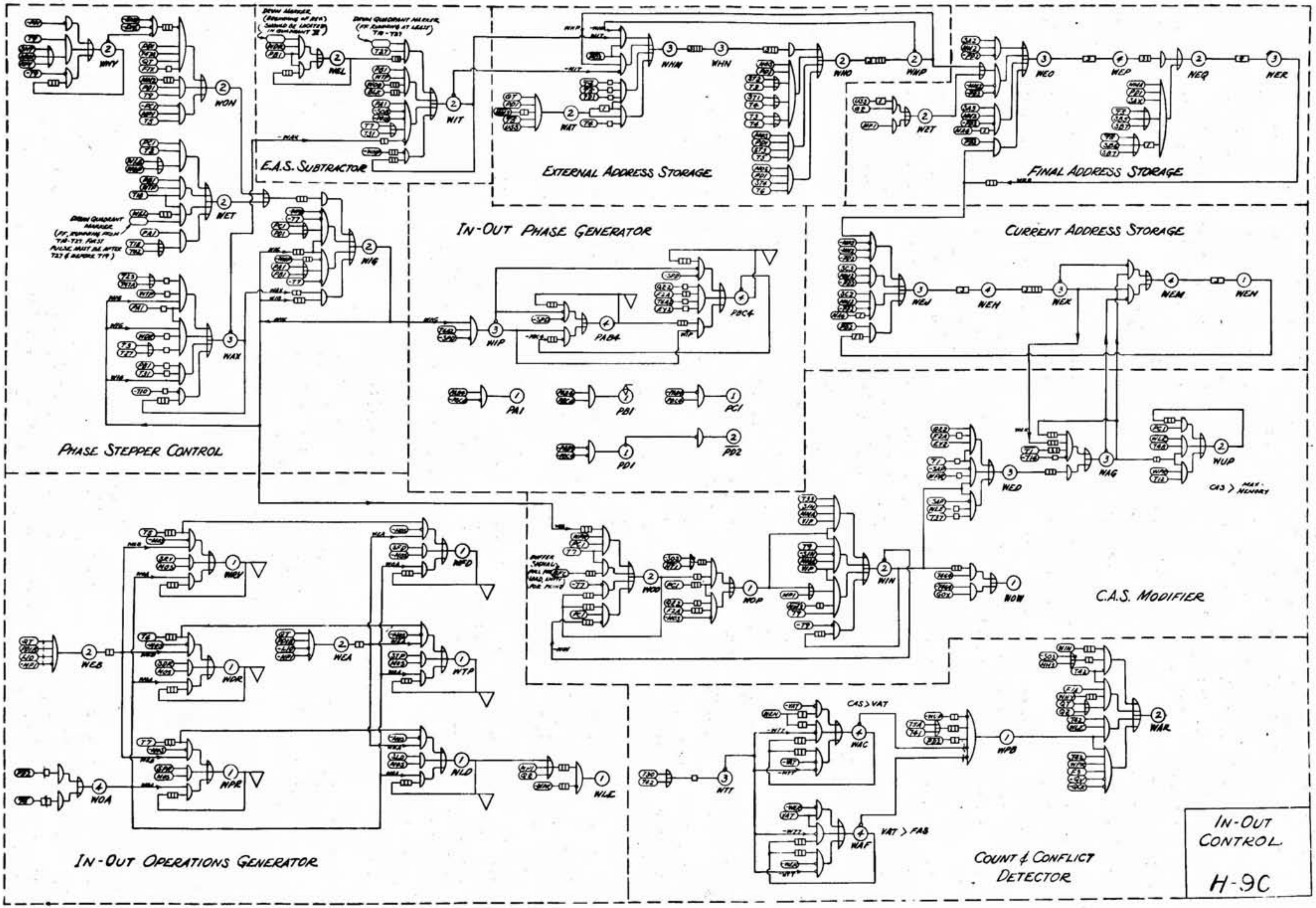


FIGURE 4.6. Example of a functional diagram.

out in terms of these symbols and recorded on the functional diagrams (see fig. 4.6). As noted above, this procedure consisted largely of expanding, modifying, and combining these fundamental elements so as to fit them to the special case at hand. The manipulative techniques of Boolean algebra were occasionally found helpful at this point for reducing gating configurations to standard form or for checking the equivalence of alternative circuit arrangements. Each tube stage was assigned a unique designation (usually a three-letter symbol), which serves to identify the signals that the stage emits. As the stages became meshed together, numbers specifying the entry and exit times of the signals for each gate, delay line, and tube were recorded. In designing SEAC, tolerances on these quantities were calculated nominally to the nearest 0.01 $\mu$sec. With DYSEAC, however, because all components were standardized, it was sufficiently precise to record the timing on the functional diagram only to the nearest nominal quarter of a microsecond.

In the process of preparing the functional design, an effort was made to minimize the number of components utilized, particularly with respect to tubes, next with respect to delay lines, and finally with respect to diodes. The choice of the most economical and efficient arrangement for the functional elements depended quite strongly, however, on such factors as the size and content of the physical packages into which the electronic components are grouped and on their electronic operating specifications. Factors which affected the choice of optimum forms on the functional layouts were, for example, (1) the maximum permissible stray capacitance that is introduced as a result of extended lead lengths, and (2) the limitations on the maximum load that can be drawn from a tube. Functional layouts, therefore, even though they depict no electronic components explicitly, still must be designed with careful regard for the specific physical components by which they are later to be realized.

Another type of interaction between successive stages of the development program may occur after the functional design is partially or even fully completed. At this stage the designer may find that the operating capabilities of the machine can be usefully expanded by making minor design rearrangements which team up the same facilities in new combinations. In this way, additional operating features can often be provided at little or no additional equipment cost. This feedback from functional design to system specifications can produce significant over-all improvements when close coordination is maintained between the two activities.

## 4. TRANSLATION TO DETAILED WIRING PLANS

After the functional plans had been completed, the evolution of the design proceeded toward the preparation of the detailed working plans from which the actual electric wiring of components could be carried out. The form that these wiring plans took for each machine depended strongly on the physical packaging methods employed for holding the electronic components on the chassis, and on the fabrication procedures followed in wiring up the soldered connections. Although extensive use was made of plug-in component packages in both SEAC and DYSEAC, the nature of the packages and the fabrication techniques used in both cases were so dissimilar that the types of wiring plans prepared for the two machines had to be radically different.

In the SEAC, because fabrication and wiring were carried out by skilled technicians who were capable of reading circuit diagrams, it was possible to use conventional wiring diagrams containing the usual electronic symbols for tubes, transformers, delay lines, resistors, and germanium diodes. In this machine, transformers and electric delay lines up to 2 $\mu$sec were packaged in individual plug-in units, but diodes were packaged in groups, electrically isolated from each other and usable in logically unrelated circuits. Over 20 distinct types of these diode packages were used, each package containing generally from 4 to 7 diodes connected in various standard configurations. After the circuit diagram for each chassis was completed, therefore, the diodes indicated on it had to be put through a so-called "clusterizing" process, in which groups of diodes on the diagram were classified according to the different types of standard configurations to which they corresponded. Each such group was then surrounded by boundary lines and the enclosed area, designated according to its type, was assigned a socket location on the chassis. This clusterized circuit diagram was used by the wiring technicians in wiring up the chassis.

In the DYSEAC, however, a completely different form of component packaging was employed [4]. (Each DYSEAC etched-plate tube package contains not only a tube-transformer unit but also preassembled and-gates and or-gates which can be associated with the tube. Also included are supplementary free components by means of which these standard preassembled gates can be expanded or additional gates incorporated.) Furthermore, a considerable portion of the actual chassis wiring work was carried out by relatively inexperienced technicians who were not expected to cope with the type of circuit diagrams used for SEAC. As a result, sets of working plans were drawn up for DYSEAC which explicitly listed the precise socket and pin identification numbers of each of the approximately 10,000 pairs of socket pins in the machine between which soldered connections were to be made. In order to produce these, a method was employed which involved the preparation of three distinct types of working plans, namely, *packaging diagrams*, *chassis charts*, and *wiring tables*.

The packaging diagram was prepared by using the functional diagram as a guide. For each tube stage appearing on the functional diagram, a rectangular package symbol was entered on the packaging diagram (see fig. 4.7). The next step was to transcribe to the *inside* of the package symbol all of the gating appearing on the functional diagram which fell within the category of standard built-in types preassembled inside the package. Following this, the nonstandard gating structures which were not of the preassembled types (that is, types which required the use of supplementary free diodes or other components) were transcribed from the functional diagram and entered *outside* the package symbols. Finally, components and connections required for circuitry reasons and not appearing on the functional diagram were entered. Thus the information on the packaging diagram is more complete than on the functional diagram in that it represents all the connections which must be made on the chassis sockets.

By means of the packaging diagram it was possible to determine the total number of packages that were required for all the gating stages and delay lines. After this was done, each package was assigned to a specific socket position on the chassis; a certain small percentage of sockets were left unoccupied to serve as spares. All assignments were recorded on a chassis chart, which is a pictorial representation of an actual chassis (see fig. 4.8). The assigning of package positions was done with a view toward minimizing both the total number and length of intra-chassis wires between the sockets and the length of interchassis patch-cord.

As the next step, the available components in each package were assigned to perform the required internal functions for that particular package. The pin numbers associated with each component as it was assigned were written on the packaging diagram in such a manner that every functional signal input or signal output terminal on the package is explicitly identified as being connected so a particular pin.

A careful inventory of the components used in each and every package was kept on the chassis chart as each component was assigned. By means of this inventory the designer could know at all times throughout the process exactly how many components of every type were still available near any region of the chassis. At the end of the process of assigning components, there was usually a slight surplus of components of all types left on the chassis. If an overdraft of components occurred, however, a spare socket could be utilized to provide the needed components. Out of a total of about 21,000 diodes used in DYSEAC etched circuit packages, approximately 22 percent are surplus; of the 3,800 delay-line segments (0.25 $\mu$sec), about 10 percent are unused.

Next, the signal distribution between packages on the chassis was indicated by writing on the packaging diagram the socket number and pin number of the signal sources which were to be connected to the components that were previously assigned as the destinations of these signals. After this had been done for the complete chassis, the interchassis signal distribution was indicated. This wiring between chassis is done by patch-cords which connect to pins of special plugs that are mounted on bridges above the various chassis. A package input or output that communicates directly with some other chassis was labelled on the packaging diagram with its local bridge socket number and pin number. On the chassis chart the notation appended to the bridge socket and pin was the unique functional symbol (usually three letters) identifying the particular tube package on the packaging drawing.

The preparation of the packaging diagrams and the chassis charts were preliminaries to the last step which was the ultimate goal, viz., the preparation of a set of wiring tables which enabled technicians to wire up the chassis (see fig. 4.9). The wiring tables consist essentially of columned pages whose entries are numbered serially corresponding to the 60 pins on a socket. Each column corresponds to a particular socket and is labelled accordingly. For example, a column headed 68A

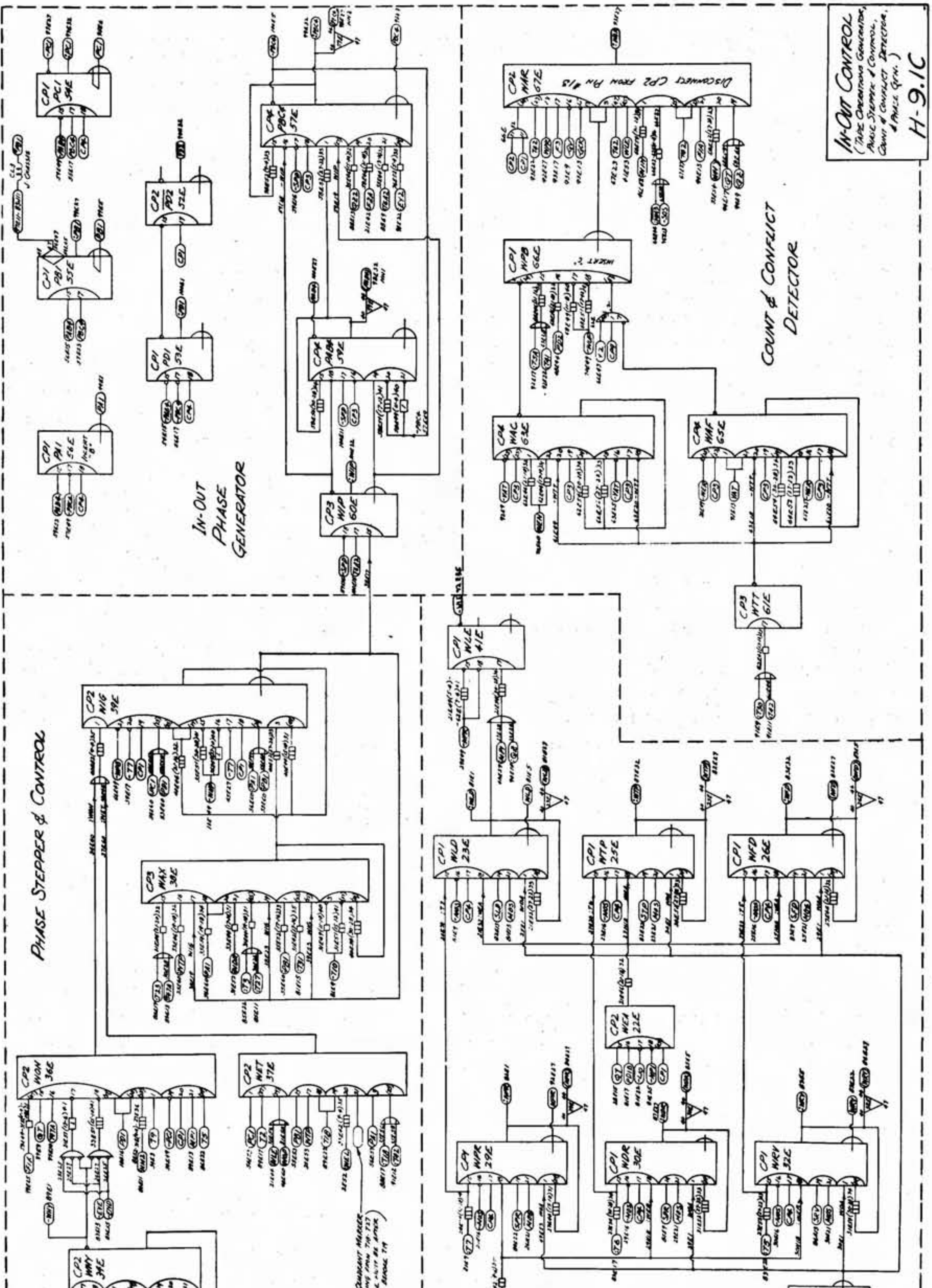

IN-OUT CONTROL
(TAPE OPERATIONS GENERATOR, PHASE STEPPER & CONTROL, COUNT & CONFLICT DETECTOR, & PHASE GEN.)
H-9.1C
PHASE STEPPER & CONTROL
IN-OUT PHASE GENERATOR
COUNT & CONFLICT DETECTOR
IN-OUT OPERATIONS GENERATOR
DISCONNECT CP2 FROM PN #13

FIGURE 4.7. *Example of a packaging diagram.*

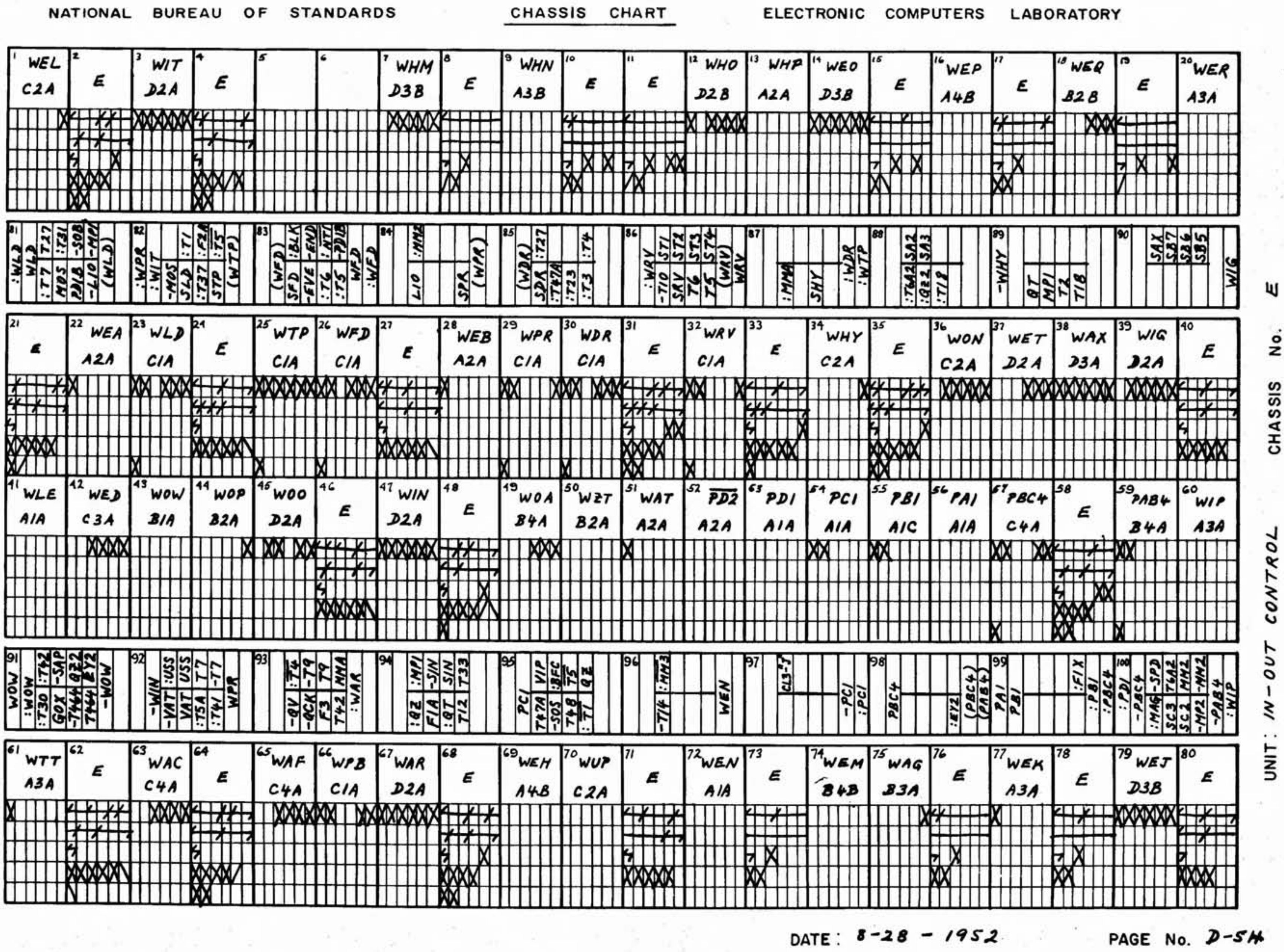
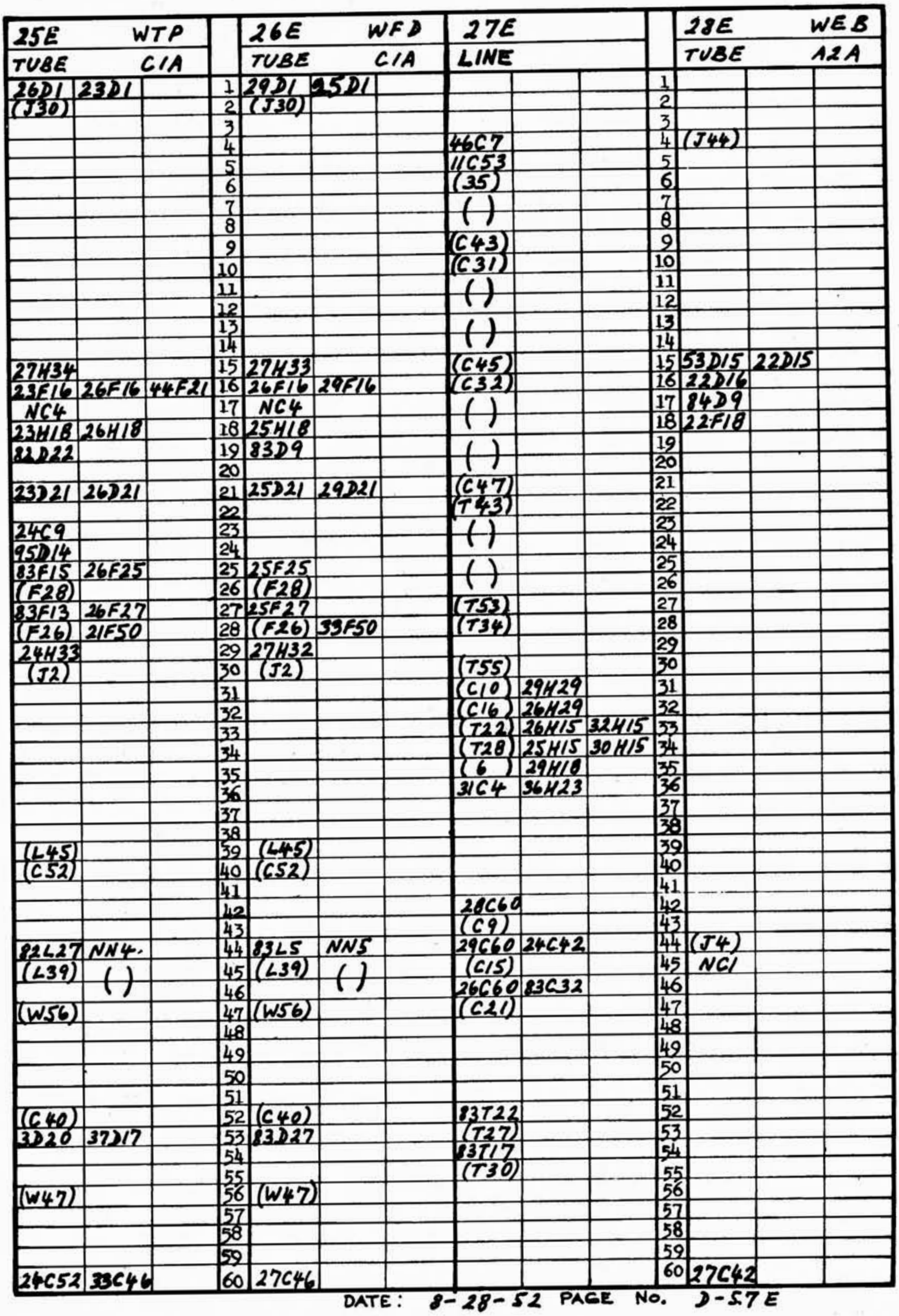


FIGURE 4.8. *Example of a chassis chart.*



| 25E WTP | | | | 26E WFD | | | 27E | | | | 28E WEB | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TUBE C/A | | | | TUBE C/A | | | LINE | | | | TUBE A2A | | |
| 26D1 | 23D1 | | 1 | 29D1 | 25D1 | | | | | 1 | | | |
| (J30) | | | 2 | (J30) | | | | | | 2 | | | |
| | | | 3 | | | | | | | 3 | | | |
| | | | 4 | | | | 46C7 | | | 4 | (J44) | | |
| | | | 5 | | | | 11C53 | | | 5 | | | |
| | | | 6 | | | | (35) | | | 6 | | | |
| | | | 7 | | | | ( ) | | | 7 | | | |
| | | | 8 | | | | | | | 8 | | | |
| | | | 9 | | | | (C43) | | | 9 | | | |
| | | | 10 | | | | (C31) | | | 10 | | | |
| | | | 11 | | | | ( ) | | | 11 | | | |
| | | | 12 | | | | | | | 12 | | | |
| | | | 13 | | | | ( ) | | | 13 | | | |
| | | | 14 | | | | | | | 14 | | | |
| 27H34 | | | 15 | 27H33 | | | (C45) | | | 15 | 53D15 | 22D15 | |
| 23F16 | 26F16 | 44F21 | 16 | 26F16 | 29F16 | | (C32) | | | 16 | 22D16 | | |
| NC4 | | | 17 | NC4 | | | ( ) | | | 17 | 84D9 | | |
| 23H18 | 26H18 | | 18 | 25H18 | | | | | | 18 | 22F18 | | |
| 82D22 | | | 19 | 83D9 | | | ( ) | | | 19 | | | |
| | | | 20 | | | | | | | 20 | | | |
| 23D21 | 26D21 | | 21 | 25D21 | 29D21 | | (C47) | | | 21 | | | |
| | | | 22 | | | | (T43) | | | 22 | | | |
| 24C9 | | | 23 | | | | ( ) | | | 23 | | | |
| 95D14 | | | 24 | | | | | | | 24 | | | |
| 83F15 | 26F25 | | 25 | 25F25 | | | ( ) | | | 25 | | | |
| (F28) | | | 26 | (F28) | | | | | | 26 | | | |
| 83F13 | 26F27 | | 27 | 25F27 | | | (T53) | | | 27 | | | |
| (F26) | 21F50 | | 28 | (F26) | 33F50 | | (T34) | | | 28 | | | |
| 24H33 | | | 29 | 27H32 | | | | | | 29 | | | |
| (J2) | | | 30 | (J2) | | | (T55) | | | 30 | | | |
| | | | 31 | | | | (C10) | 29H29 | | 31 | | | |
| | | | 32 | | | | (C16) | 26H29 | | 32 | | | |
| | | | 33 | | | | (T22) | 26H15 | 32H15 | 33 | | | |
| | | | 34 | | | | (T28) | 25H15 | 30H15 | 34 | | | |
| | | | 35 | | | | ( 6 ) | 29H18 | | 35 | | | |
| | | | 36 | | | | 31C4 | 36H23 | | 36 | | | |
| | | | 37 | | | | | | | 37 | | | |
| | | | 38 | | | | | | | 38 | | | |
| (L45) | | | 39 | (L45) | | | | | | 39 | | | |
| (C52) | | | 40 | (C52) | | | | | | 40 | | | |
| | | | 41 | | | | | | | 41 | | | |
| | | | 42 | | | | 28C60 | | | 42 | | | |
| | | | 43 | | | | (C9) | | | 43 | | | |
| 82L27 | NN4. | | 44 | 83L5 | NN5 | | 29C60 | 24C42 | | 44 | (J4) | | |
| (L39) | ( ) | | 45 | (L39) | ( ) | | (C15) | | | 45 | NC1 | | |
| | | | 46 | | | | 26C60 | 83C32 | | 46 | | | |
| (W56) | | | 47 | (W56) | | | (C21) | | | 47 | | | |
| | | | 48 | | | | | | | 48 | | | |
| | | | 49 | | | | | | | 49 | | | |
| | | | 50 | | | | | | | 50 | | | |
| | | | 51 | | | | | | | 51 | | | |
| (C40) | | | 52 | (C40) | | | 83T22 | | | 52 | | | |
| 3D20 | 37D17 | | 53 | 83D27 | | | (T27) | | | 53 | | | |
| | | | 54 | | | | 83T17 | | | 54 | | | |
| | | | 55 | | | | (T30) | | | 55 | | | |
| (W47) | | | 56 | (W47) | | | | | | 56 | | | |
| | | | 57 | | | | | | | 57 | | | |
| | | | 58 | | | | | | | 58 | | | |
| | | | 59 | | | | | | | 59 | | | |
| 24C52 | 33C46 | | 60 | 27C46 | | | | | | 60 | 27C42 | | |

DATE: 8-28-52 PAGE No. D-57E

FIGURE 4.9. *Example of a wiring table.*

means the information entered in that column pertains to the 68th socket in chassis A. The package type is also indicated in the heading. Each item in the column pertains to a particular pin on that socket, and the entry made for each item specifies some other pin to which it is to be connected. The entries are in the form number-letter-number, specifying respectively the socket serial number, the type of signal (such as direct positive output, negative output, delayed signal), and the pin position in the socket. To help guard against errors a double entry system was employed, whereby each length of wire was represented by an entry for both of the pins that it connects. If a pin was to be connected to more than one other pin, the appropriate number of entries were made for each item.

Technicians receiving copies of the wiring tables did the wiring on all the chassis, following the wiring tables implicitly. The power wiring and other standardized wiring was done according to standard instructions appropriate to whichever of the five possible package types was specified in each column heading. The total wiring table record for the connections between the etched circuit packages of DYSEAC occupies about 270 pages (8 by 10 in.) and specifies approximately 20,000 soldered pin connections.

## 5. CONCLUSION

In the development of a large-scale computer, system-design work occupies the middle ground between electronic engineering and operations analysis of the intended application. The final system design must be practicable to construct with available building blocks and current fabrication techniques, and must also be capable of satisfying the ultimate user's needs., In working out design problems concerned with reconciling these two requirements, standard methods of attack which would be generally or universally applicable probably are not obtainable. On the other hand, design problems concerned with the development of functional and construction plans, rather than with the development of system specifications, probably would provide a fruitful field for research into generally applicable procedural methods. In particular, problems which might benefit from such research are those concerned with (1) analyzing and synthesizing the complex logical relationships existing in large-scale systems and (2) translating such relationships into explicit wiring plans. With respect to the first class of problem, it should be noted that although methods of mathematical logic such as Boolean algebra have been found useful and effective in handling certain types of system design problems, the scope of their utility to date has been strictly limited to problems of comparatively minor importance in the system design work at this laboratory. With respect to the second class of problem, a very promising possibility appears to lie in the idea of using digital computers themselves to carry out the processes involved in compiling wiring plans. These detailed and tedious processes of recording, classifying, rearranging, keeping inventories, and the like, are in fact feasible tasks for present-day machines. Furthermore, in the future the tasks of the computer might not need to be limited to the production of just the wiring tables prescribing the locations of the various parts of soldering lugs between which connections are to be made. With the development of methods for forming wiring interconnections out of prefabricated standard interchangeable parts, and with the increasing use of mechanized assembly techniques in the manufacture of electronic equipment, it is possible that the computer may one day also be utilized in fabricating the actual equipment. That is, as the final wiring data could be produced within the computer, possibly it could also produce mechanical patterns (e.g., punched paper cards or tape) suitable for governing the selection and positioning of the prefabricated parts out of which the interpin connectors might be formed. In this way, the somewhat dramatic concept of setting a computer to build a computer might well be brought nearer to realization.

TABLE 3. Explanation of figure 4.2

Individual pulses occurring successively in a pulse train are shown pictorially in the order *right to left*. A pulse represents a 1-digit; the absence of a pulse represents a 0-digit.

| Item on figure | Name | Mode of operation |
|---|---|---|
| 1------ | And-gate----------- | A pulse appears at C if and only if pulses occur on A and B simultaneously. More than two inputs may be used. |
| 2------ | And-gate with inhibition input. | A pulse appears at D if and only if pulses occur on A and B simultaneously and no pulse occurs on C. |

TABLE 3. Explanation of figure 4.2—Continued

| Item on figure | Name | Mode of operation |
|---|---|---|
| 3 | Or-gate | A pulse appears at C if a pulse occurs on either A or B, or both. More than two inputs may be used. |
| 4 | Delay-line | A pulse occurring at A appears at A′ after the span of time *n* microseconds. |
| 5 | Amplifying tube and two-pole transformer output. | The output at A consists of positive-going pulses; at –A, negative-going pulses. There is always a one-to-one correspondence between A and –A. |
| 6 | Basic flip-flop | The flip-flop is turned *on* (i.e., emits continuous pulses at C) beginning with the first pulse of A, and is turned *off* (emits no pulses) beginning with the first pulse of B which is not accompanied by a pulse of A. |
| 7 | Equality comparator | T is a priming pulse occurring before the first pulses on A and B that are to be compared. After the last pulses of the trains A and B have appeared, C emits a pulse if and only if pulse trains A and B are identical. |
| 8 | Inequality comparator. | T is a clearing pulse that occurs at or before the first pulses on A and B that are to be compared. After the last pulses of A and B have appeared, C emits a pulse if and only if the binary number represented by A is greater than that represented by B. The digits of A and B occur in order of increasing significance. |
| 9 | Binary counter | B emits continuous pulses beginning when an odd number of pulses have occurred on A, but B emits no pulses when an even number of pulses have occurred on A. |
| 10 | Two-stage cyclic counter. | Stepping pulses occur on S. With both stages off, the first S turns on A. If A is on and B is off, the next S turns on B. If A and B are on, the next S turns off A. If A is off and B is on, the next S turns off B, etc. |
| 11 | Matrix decoder | Let the pulses (or no pulses) emitted by A, B, and C at any instant represent a 3-binary-digit number, ABC. Examples of decoding are: M emits a pulse if and only if ABC = 2; T emits a pulse if ABC = 4 or 6. Such matrices are readily expandable so as to decode larger numbers of digits and in a wider variety of ways. |
| 12 | Add-1 counter | T is a priming pulse whose timing determines the power of 2 to be added to the n-binary-digit number contained in the delay-line register. Starting at the time of T, all digits of the original pulse train are inverted (1 becomes 0, 0 becomes 1) up to and including the first 0. The digits occur in order of increasing significance. If A′ is from –A instead of A, the device subtracts 1. |
| 13 | Two's complementer. | All digits of A up to and including the first 1 emerge from B unchanged, but all succeeding digits of A after the first 1 are inverted. Digits occur in the order of increasing significance. |
| 14 | Adder | A and B carry pulse trains that represent two binary numbers to be added. The output S is the pulse train that represents the sum of A and B. The carry digits are at C. All digits occur in order of increasing significance. |
| 15 | Recirculating delay-line register. | This illustrates how an n-binary-digit number contained in such a register may be shifted to the left or right by precession. N, L, R, and I are mutually exclusive control pulses, and in the absence of all others, N is present. N is *normal recirculation*, L is *precess left*, R is *precess right*, and I is *insert*. After a continuous train of *n* L- or R-pulses, the contents have been shifted one pulse position to the left or right with respect to their original positions. Apart from mere shifting, such a precessing register may be used as a pick-up register to collect digits presented on input A at a synchronous but irregular rate, in normal or reverse order, and to arrange them in normal order for distribution as a pulse train. This is done by causing a single I-pulse corresponding in time to each A-digit to be substituted for the first pulse of a train of L-pulses or for the last pulse of a train of R-pulses. |

TABLE 3. Explanation of figure 4.2—Continued

| Item on figure | Name | Mode of operation |
|---|---|---|
| 16------ | Staticizer--------- | This illustrates a different type of pickup register, and it also shows a method of converting from parallel to serial mode of transfer and vice versa. I is serial-insert control pulses for source B. P is parallel-insert control pulses for sources $A_1$, $A_2$, $A_3$, . . . , $A_n$, H is *hold* pulses, R is *shift right*, and L is *shift left*. I, P, H, R, and L are mutually exclusive except that I is always accompanied by R or L. In the absence of all others, H is present. A single R-pulse causes the content of each flip-flop to be transferred to the one immediately to the right, and a single L-pulse, to the left. For normal order of B-digits, a single R-pulse is used with each I-pulse, while for reverse order of B-digits a single L-pulse is used instead. A continuous train of R-pulses allows an information pulse train to be inserted or removed from the register in normal serial order. A single P-pulse inserts digits from $A_1$, $A_2$, $A_3$, . . . , $A_n$ simultaneously. The parallel outputs are $C_1$, $C_2$, $C_3$, . . . , $C_n$. |

TABLE 4. Examples of typical basic operations in computing systems

| Types of operations | Procedures which must be developed for carrying them out |
|---|---|
| 1. *Arithmetic*: | |
| Addition and subtraction. | Basic addition-subtraction procedures including negative number representation, complementing practices, sign determination of sums, overflow recognition. |
| Multiplication and division. | Basic multiplication-division procedures including the shifting of numbers in a register, the flow of information between various registers, time sequencing and control of steps required in performing such operations. |
| Compound operations-- | Specialized procedures, formed from combinations of basic procedures, which are needed to achieve higher speeds of operation for special purposes, e.g., extra-rapid accumulation achieved by the DYSEAC Summation operation at up to nine times the normal DYSEAC Addition rate. |
| 2. *Control:* | |
| Central control operations. | Basic procedures for achieving centralized control functions. (In SEAC and DYSEAC, most control functions are performed by means of suitable combinations of the outputs of only four basic units: a timing generator that provides cyclic recurrent signals for marking off elapsed time within each serial word-cycle, a four-state phase generator whose outputs correspond to each of the four phases of each operation, a phase-stepping unit that advances the phase generator and marks the initial cycle of each phase, and an operations generator that specifies the type of operation being performed.) |
| Memory-address selection. | Procedures for achieving rapid memory access by means of space-selection switches using matrix decoders for digital address codes, time-selection using counter-timers, and space-voltage analog selection using special digital-to-analog converters. |
| 3. External transfer operations. | Procedures for transferring digital information between internal memory of a computer and low-speed non-synchronous input-output devices [2]. |
| 4. External control operation. | Procedures for providing flexible external control over the internal system and for achieving versatile joint control by both internal and external parts of the system acting jointly [5, 6]. |
| 5. Program sequencing------ | Procedures for sequencing the individual instructions in a program and providing for branching in its execution [3]. |

## 6. REFERENCES

[1] A. L. Leiner, W. A. Notz, J. L. Smith, and A. Weinberger, System design of the SEAC and DYSEAC Trans. Inst. Radio Engrs., vol. EC-3, No. 2 (June 1954).

[2] A. L. Leiner, Buffering between input-output and the computer, Review of Input and Output Equipment Used in Computing Systems, Joint AIEE-IRE-ACM Computer Conference, S-53 (March 1953).

[3] A. L. Leiner, Provision for expansion in the SEAC, Mathematical Tables and Other Aids to Computation **36**, 233-237 (October 1951).

[4] R. D. Elbourn and R. P. Witt, Dynamic circuit techniques used in SEAC and DYSEAC, Trans. Inst. Radio Engrs., vol. EC-2, No. 1 (March 1953); Proc. Inst. Radio Engrs. [10] **41** (1953).

[5] A. L. Leiner and S. N. Alexander, System organization of the DYSEAC, Trans. Inst. Radio Engrs., vol. EC-3, No. 1 (March 1954).

[6] A. L. Leiner, System specifications for the DYSEAC, J. Assoc. Computing Machinery **1**, No. 2 (April 1954).