This paper examines the machine utilization and job turnaround problems of a large computer center by analyzing the information handling and queuing problems occurring between jobs.

☐ *A system designed to overcome these difficulties is described and the results of simulating the system are reported. The system design includes the interconnection of input-output computers with large scale processors by means of commonly shared disk files.*

☐ *Although this paper deals with a study which is not yet completed, the techniques developed and the results obtained to date are of general interest.*

A multiprocessing approach to a large computer system

by F. R. Baldwin, W. B. Gibson and C. B. Poland

This paper is an interim report on a study of computer room operation in a large computational center. The study is being conducted by Lockheed and IBM personnel at the Lockheed Missiles and Space Company (Sunnyvale, California). In order to understand the motivation of the specific study objectives stated below, a brief description of the Lockheed Missiles' computational procedures are given.

study background

The Information Processing Center at Lockheed Missiles, under the direction of Mr. E. K. Fisher, is among the largest in industry. The Center's IBM® machines of direct concern in this study are three 7090's supported by nine 1401's, several of which are exclusively used as peripheral input-output devices. Two of the 7090's are four-channel, 20-tape systems, while the third is a two-channel, 16-tape system. Serving this complex are nearly 200 closed shop programmers. Since their knowledge of programming is comprehensive, a wide range of systems is utilized including a number of special systems developed at the Center. In addition, an open shop services an approved list of 800 scientists and engineers who supply 39% of the total work load. The computing and data processing applications range from intricate scientific and engineering computation to data reduction work and include a heavy load of commercial data processing involving the sorting, file maintenance, and other record-keeping necessary to support a 25,000-employee aerospace facility.

Involved in computer operations, exclusive of keypunchers and programmers, are more than 100 individuals. Theirs is the

task of dispatching, scheduling, job transporting, maintenance of a tape library of nearly 10,000 reels, and simply operating the Center three shifts a day, seven days a week.

Several interrelated problems (common to large installations and tending to increase in significance with the size of a center) were recognized. The high ratio of job setup and tear down time to job execution time was noted along with the associated problem of machine operators spending a large amount of time on routine tasks such as labelling and transporting tapes, etc. It was observed that idle CPU time accrued even on machines backlogged with work. Perhaps the most significant problem recognized was the magnitude of the job turnaround time (the elapsed period from the submitting of a program to the return of the results). For nearly all large installations this period, disregarding priorities, extends from a minimum of six or eight hours to well over twenty or even thirty hours. With the general prospect of having only one computer run per day for each job, many managers find it necessary to assign multiple projects to their programmers in order to keep the programmer busy and to insure project completion within schedules. It would seem reasonable, however, that a person could work far more efficiently if he were handling only 1 or 2 projects at a time rather than 4 or 5. The final problem singled out for attention was that of expediting priority work. The importance of this problem is self-evident.

study objectives

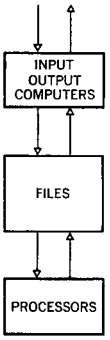
With the above problems in mind, the following objectives were formulated:

1 *To increase the productivity of equipment*—the idea being to keep the processors processing and at the same time keeping the I/O devices reading and printing until all work is exhausted.
2 *To increase the productivity of operators*—by obviating or mechanizing routine tasks such as physical job transportation within the machine room, tape label preparation, scheduling stacks of tapes, etc.
3 *To reduce turnaround time*—thus increasing the efficiency of the programming effort (and of those dependent upon this effort).
4 *To solve the priority problem*—preferably without adding to the machine costs.

system selection

Having formulated objectives, several considerations determined the direction of the study. For example, in connection with programming systems, it was realized that, although improvement in compilers and the system monitor would contribute, concerted efforts had been made in this area over the past several years. It was also known that product improvements were not apt to be sufficient. For example, in another study it was calculated (assuming no procedural, routine, or I/O changes) that for the typical case, increasing the 7090 CPU speed by a factor of fifty would increase the throughput by a factor of only two. Thus it was decided to examine the system performance during intervals between jobs with particular attention to the associated information queuing and handling problems. On the basis of this examination it was decided that a new system configuration design

Figure 1



was needed to minimize the latter problems.

A design problem of this nature is essentially one of selection and evaluation, with the evaluation being conducted by means of simulating its performance with the aid of a computer. Even though the best possible judgment is exercised in the selection of cases for evaluation, there is obviously no guarantee of reaching the design objectives. In the event that a system is found that meets the design requirements, there is no assurance that this is the optimal such system. Despite the shortcomings of this procedure it is continually proving effective in evolving improved systems.

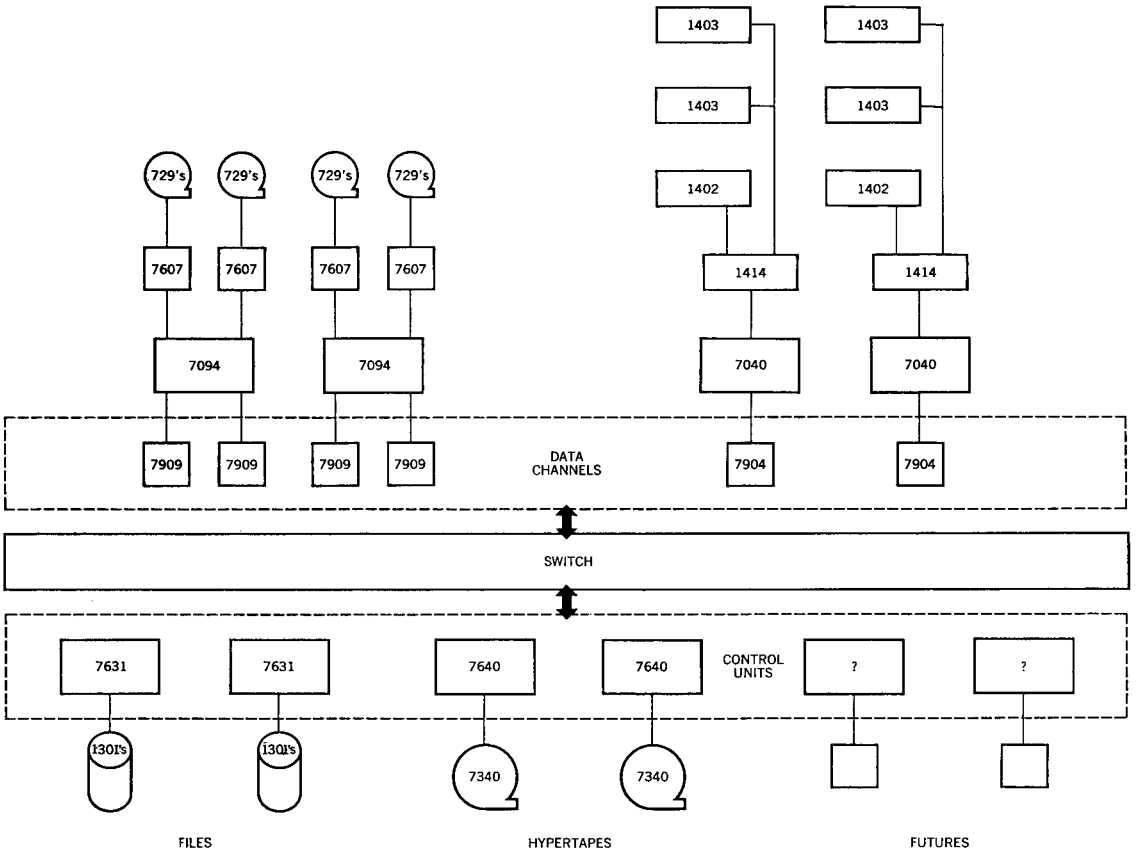
Of the various possible systems configuration designs, it was finally decided to select the configuration functionally described in Figure 1. (Of course, for a system of this complexity the simulation procedure is extensive and it was hoped that this initial selection would prove effective in meeting the study objectives.)

the system concept

The system conceptually speaking is very simple. The concept is to interconnect input-output computers with large scale processors by means of commonly shared disk files. For this study the input-output computers are IBM 7040's and the large scale processors are IBM 7090's or 7094's.

Figure 2 shows the system configuration and logical relationship of a switch that has been introduced to permit the files to

Figure 2 The system configuration



be shared by the input-output and large scale computers. In fact, the switch permits the connection of any control unit and any channel capable of communicating with the device. The data channels are on one side of the switch and on the other side there are three types of units: (a) 7631 File Control Units. Each FCU can have a variable number of file modules attached, and each module can have a unique assignment such as scratch area, Fortran area, etc. Unused module space can be used to house production programs, etc. (b) 7640 Hypertape Control Units. Each hyper control handles up to as many as 20 hypertapes. (c) 7750 Programmed Transmission Control and similar units. All processors can handle Tele-processing® equipment at any time. As new communications equipment comes along the only requirement is that its interface handle the switch.

the system analysis

The analysis of the system was divided into four parts as follows:

1 *engineering analysis* There were two considerations: (a) the engineering feasibility of the switch and (b) the system size permissible in terms of cable restrictions and maintenance requirements.
2 *evaluation of the system in terms of study objectives*
3 *analysis of the operating system* The system at hand will obviously require an operating system. This part of the analysis is yet to be considered in detail (e.g., its precise specification, an accurate estimate of the time required to program it, the requirements relative to the standardization of existing systems which it would operate, etc.). It will suffice for the purposes of this paper to assume that its main purpose is that of scheduler.
4 *economic evaluation* Approximate estimates were made at the beginning of the study to ascertain a high probability of economic feasibility. However, detailed analysis has not been completed since the other parts of the study must also be completed in order to supply certain necessary data.

The rest of this paper deals with Parts 1 and 2 (which to this point, the study has been primarily devoted).

engineering analysis

The switch has been studied in sufficient detail to determine engineering and cost feasibility. The switch is actually a set of switches and the failure of one will not affect the others. It is capable of handling at least eight 7909's, four 7904's, twelve 7631's, and four 7640's. The cost is completely dependent upon the number of switching points. The switch is 1 × 20 maximum (1 data channel can be switched to up to 20 control units or 1 control unit to any of 20 data channels). It is program switchable and the time required to switch is on the order of a few hundred microseconds. As mentioned, if one switch should fail, no other switch, channel, or control unit is affected. There is no direct connection between the core storage units of any two computers. Also note that 729's are on the opposite side of the 7090/94's from the switch and they are not switchable. The probable 7040 arrangement is also illustrated, namely, two 1403's and a 1402 on

each 1414 of which there is one per 7040. Also shown is the place where future Tele-processing gear can be added to the system. Of course, the attachment of two 1403's will require minor engineering modification.

With regard to the size of a system which could be arranged without exceeding prescribed cable limits and without violating necessary maintenance clearances, one feasible configuration is shown in Figure 3. As large as it is, it is still possible to add more 729's, more 7909's, more hypertapes and more files. Of the four 7090/94's, two are two-channel, 14-tape computers and the other two are four-channel, 20-tape computers. The area immediately to the left of the 7040's is reserved for future Tele-processing equipment.

The figure clearly indicates the system's growth potential. It is possible to have a maximum of twenty data channels and twenty control units. Increasing the system size, will in no way affect the operating system except to give it more facilities with which to work.

There is another important observation about the system. The engineers engaged in this study have suggested that it should be possible to take immediate advantage of technological improvements that will occur in readers, printers, files and Tele-processing equipment since it is expected that the older units in this system can be replaced by the newer units as they are developed. This possibility is maximized in this system because of the level at which the switching to interconnect units occurs. This system was designed with this consideration in mind.

qualitative system analysis

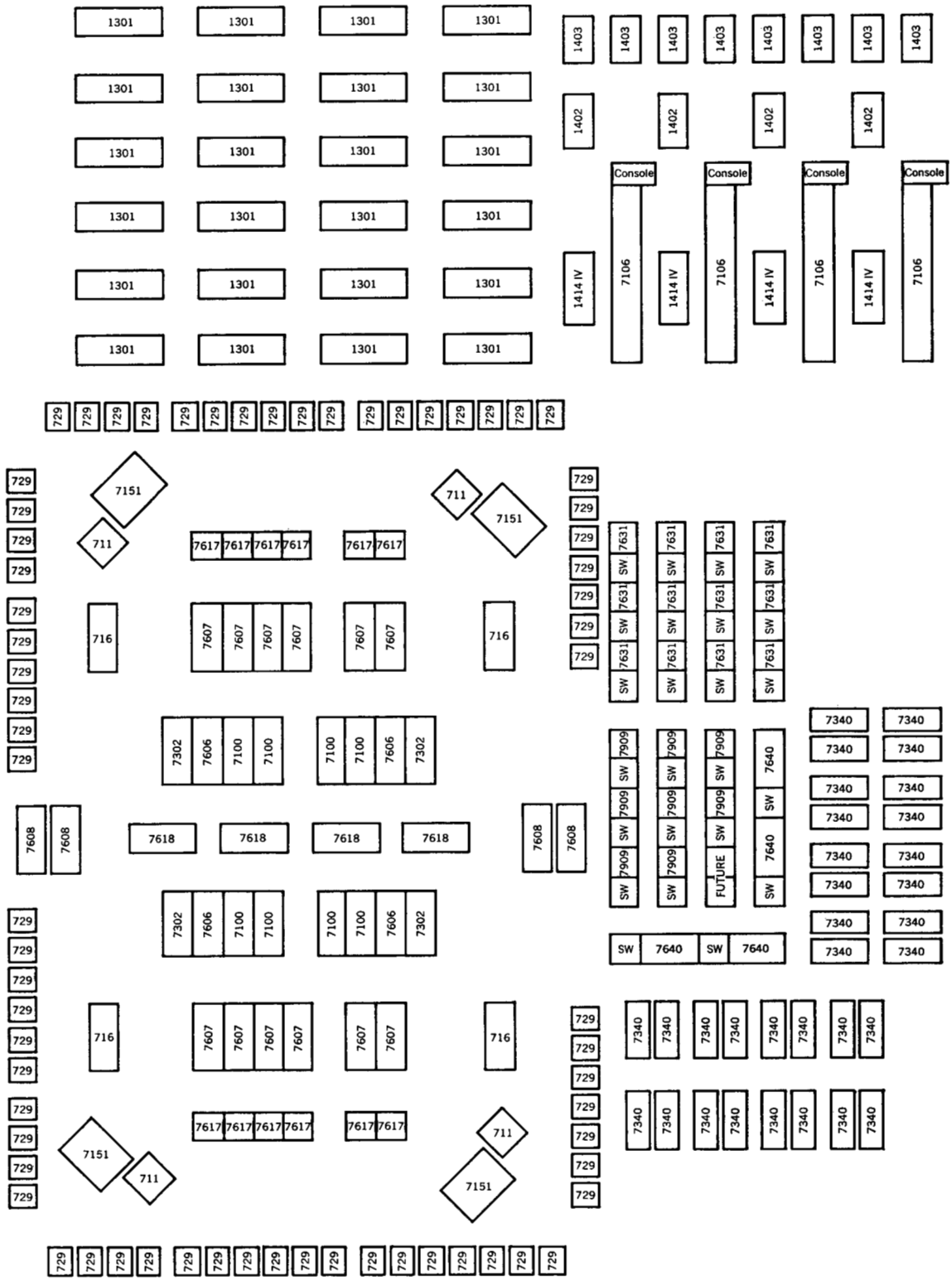
We turn now to a general description and qualitative analysis of the system during which it will become evident that study objectives 2 and 4 (relating, respectively, to the productivity of machine operators and to the priority problem) have been met. After completion of this section the principal part of the study involving the simulation of the system will be presented to validate the observations of the present section and to provide some quantitative measurements of systems performance relating to study objectives 1 and 3.

Whereas today off-line equipment for card-to-tape and tape-to-printer operations is used, the new system features I/O units on-line and off-line at the same time. Information enters the file via a card reader on an I/O computer and since the processor also has access to the file, the information may now go into the processor without interruption or manual intervention. After processing is finished, the output data is placed on the file which is then switched to the I/O computer.

The files are allowed to act as input buffers to the processors, but not in the sense of during-job buffers. Input can be stacked for the processors as the bank of card readers brings jobs into the file. The output proceeds in reverse (without tying up valuable processor time) by having the reports stacked in the files until printers and punches are available.

In other words, the system functions as would a row of card readers and a row of printers served by a 2-way conveyor belt

Figure 3 One feasible configuration



bringing jobs to and removing jobs from the computer room. All the printers are capable of printing information from any file. All processors have access to information on any file and can store results in these files. As a processor completes a job, it returns to the files for whatever is required next. Simultaneously, operators are taking card decks off the conveyor belt and placing them in the emptiest card reader. Other operators are removing printed reports and are being instructed by the system of necessary forms changes. Operators on the 7090/94's are being instructed by the operating system when to mount tapes for those jobs which will still require them—many jobs won't, since programming system, scratch area, printer output, and card I/O will be file-oriented.

Thus, multiprocessing exists with respect to the I/O and central computers. Full speed operation of all equipment is attained and there is no reader, processor, or printer delay as long as there is work to be done.

Consider the priority problem. As jobs develop in the input area of the files, certain ones should be given priority. Input is now stored on a random access media. The more important jobs are selected and executed ahead of less important jobs. It is an automatic priority system. Note that there is no delay to either priority or non-priority work due to batching at the input station by type of run, and no operator intervention is required in priority handling.

The processors schedule themselves. At the point where a 7090/94 has completed a job, it goes to the files for the scheduling program and for job and facility descriptions. From these it dynamically schedules the next job for itself and updates the job and facility descriptions. The role of scheduler is played by the 7090/94 needing work; and this function rotates among the processors. The time involved to perform this between-job scheduling is estimated to be less than 1 second.

If we assume a number of jobs in the file input buffer area, at the point when a 7090/94 has completed its work it will try to execute a high priority job first. Its decision is dependent only upon the availability of facilities. The order of the job entry to the input buffer is immaterial.

With programming systems stored in the files, if for example the Commercial Translator is needed, it can be obtained and work begun immediately. The programming systems are available randomly and the sequence of jobs no longer depends on the type of language in which they were written. As new programming languages are developed, they can easily be placed in the system and, of course, the access time will be the same.

Most stacked output in present monitor systems cannot be dynamically expedited. Once jobs are executed in a given order, generally that is the necessary order for output. However, in this system, jobs in the output buffer may be rescheduled. Thus a priority job is scheduled coming into the system and rescheduled going out.

Further, the requirements for forms are known to the system

through the job description, so output scheduling is done to minimize forms changes.

On the processor level, no 7090/94's will be idle if work is available. The computers proceed immediately from one job to another, pausing only an instant for scheduling. There will be occasions where necessary tape conflicts exist, but these are minimized because the operating system can (a) schedule onto multiple processors and (b) pad in tapeless (non-conflicting tape) jobs while operators are selecting required tapes and freeing-up previously used drives. To repeat, more "tapeless" jobs will now exist because the compiler, loader, and scratch functions have become file oriented.

One of the features of the system is the ease of pinpointing component bottlenecks in the system. For example, if card readers are having difficulty keeping up with jobs on the "conveyor belt," then more card reader capacity can be added. In a similar manner, when jobs begin to congest the input buffer, then the processing capacity can be increased. Finally, if it is the output load which is causing delays in the system, correction can be made by the modular addition of printers.

At any convenient time interval the operating system can report its maximum buffer size. We have continual knowledge of the peaks in the file buffers and thus are alerted to the need for more reader, processor, or printer capacity. In the present day systems, acquisition of such knowledge represents an additional load and set of procedures which must receive the attention of the operating staff.

In the new system many past difficulties have been obviated. (a) Batching no longer exists. Each job is independent of all others, is placed immediately in the emptiest card reader, and after reaching a disk is an immediate candidate for processing. (b) Card-to-tape, tape-to-printer operations and associated tape handling have disappeared. The transporting is done at electronic speed. Typically, the only times an operator will see a Fortran job will be on placement of the deck in a 1402 and on removal of the output from a 1403. (c) Scheduling is now accomplished automatically by the operating system at electronic speeds. Even for a very large complex this will require less than an hour a week for all computers.

In the new system, production decks may enter the 7090/94's without the necessity of card-to-tape or selection from tape library operations previously required since production program storage is now internal. Furthermore, we have millisecond access to these programs.

Relative to program development, debugging will be facilitated by the new system since the last iteration may be stored internally with only the correction being entered through the 1402.

Although the system doesn't include a remote debugging facility, the system design is consistent with this requirement.

Since the system is file-oriented the usual benefits will accrue. For example, files may be used (a) to store intermediate results, and as a scratch area instead of using magnetic tape; (b) as an

extension of memory (e.g., a place for exception routines during an accounting application to allow greater use of core for actual data) again replacing magnetic tape; (c) to store the basic company data—once introduced to the files this data is accessible by all machines and is available for processing and inquiry. It is also open to communications equipment. The Tele-processing control equipment may be placed next to the hypertape and file control units and when a communications link is needed it is switched on-line to a processor. No reorganization of the total system is required.

In other studies to date, several systems concessions have appeared necessary. Each processor must have access to its own set of programming systems (and other vital basic information). Hence, there is duplication of some file data for each processor. One 7909 channel normally has access to a single control unit, be it files or hypertape. This limits the function of that data channel. It is identified in function by the content of its modules. Identical 7090/94's down to the last module are preferred in order to enable each machine to process any application. If this is not practical then procedural restrictions need to be imposed. One further point, no machine can switch control units among its own channels.

the simulator and its use

To properly analyze the performance of the system it was necessary to simulate its operation. It was also decided that the best results would be obtained by simulating the flow of Lockheed's own work load (through the system operating as described above). To do this there were two major undertakings: (a) the design and coding of the simulator, and (b) the collection of information about LMSC's jobs required as input to the simulator.

In testing the soundness of the system several scheduling rules were attempted before a successful means of scheduling was found. Priority rules had to be tested. The interactions of various components (for shared units) had to be determined. With all the logic working, the typical items of interest became (a) the rate of job movement through the computer room, (b) the level of equipment utilization, and (c) the appropriate balance of equipment.

Four items make up the input to the simulator:

1 A thorough *description of all the jobs* which the 7090/94's are to process.
2 A complete *description of the machines* to be simulated. This includes a functional assignment for some of the components.
3 The *procedures* to be followed. Both those applicable to any operating system and those applicable to the operation of this unique system (of 7090/94's, 1301's, etc.).
4 The *necessary routines* such as housekeeping, timekeeping, utility routines, association of equipment, etc.

This generally describes the simulator and its input. The results the simulator produces include all the original information plus the simulated data. Three different programs are used to take output tapes and produce the necessary reports for analysis.

The work description used as a basis in this study consisted of some 1900 jobs resulting from a 7-day sample made at Lockheed. Existing accounting information would have been insufficient to permit simulation. For each job submitted during the 7-day period, the programmer was asked to fill out a form with the questions:

> How many lines printed? What kind of forms? How many cards in the job? BCD or binary cards or both? Kind of priority code? Times it was logged in, out? Time it reached a 7090? How many tapes did it use? What system was used? Execute or compile, or both? How many instructions? How long does it run?

Organizing, editing, and placing the data on tape in usable form was an extensive task.

In the simulator each unit is either assigned as a fixed unit (e.g., a 729) or as a switchable unit (e.g., a 1301). Furthermore, units like files can be designated as having shared functions (such as storage of all programming systems and production programs on a single 1301) or having a unique function (such as output buffer area only). To vary the configuration, a deck (currently read on-line) which describes and associates all equipment is modified. Depending on the change, the time required to manually do this is from several minutes to approximately an hour.

Several operational procedures, assumed to be incorporated in the system simulated, may be of general interest here. First of all is the matter of priorities. Every time a 7090/94 finishes a job, all jobs are examined to find the most important one to be done next. If there are several of equal importance, the one which entered the system first is chosen. If, at execution, this choice would cause idleness (e.g., tape set up required) this job remains scheduled for this computer and a search is made for the next job which can be done in accordance with the same set of priority rules.

A rigorous advance scheduler was not developed for the 7090/94's in the simulated system. Determination of the next job for a machine occurs at the time it is ready to do it. The scheduling is sequential rather than time-dependent. Secondly, tapes, if required, are selected 24 minutes in advance of job execution in the simulator. This allows sufficient time to select them from the library and transport them to the 7090/94. However, the 24 minutes is entirely arbitrary and can easily be changed. During this period the scheduler attempts to pad in jobs whose facility requirements are non-conflicting. Also, arbitrary times of 3 minutes each for tape setup and for tear down are used. In addition, there is a 10-minute lead time used from *log in* to the 1402 for certain simulations. Further, execution procedures are entirely different for each type of run.

We simulate for each major programming system the time required to bring it from the disk, simulate its own core overlays or passes, etc. Between each job a complete new access of a

programming system is made since batching does not exist in this system.

Another procedure of interest is the assumption that any job must be able to run on any processor. To add a 7080 to the system, for example, would require a modification of the simulator. It is also assumed that jobs can be read by any 1402 and that they are always placed in the emptiest card reader. The simulator will schedule output onto the most appropriate 1403. The operating system also requires a complete description of all machines and the work active in the system. These are simulated as being housed in a file.

The simulator measures several items: (a) job turnaround time; (b) job delay time in the system; (c) the input and output buffer sizes; (d) the system's throughput—the time required to do a given amount of work; (e) the utilization of equipment—total amount of time the 7090/94's, the 1402's and other major components are busy; (f) interference times for tapes and files (in most cases these turned out to be quite insignificant).

The simulator operates at about 20 times real time and normally 2 to 3 jobs are simulated per minute. The amount of card reading and printing for a job has a significant effect on its simulation time. The simulator is written in FAP (about 7000 instructions) and currently runs under a Lockheed monitor.

some simulation results

For a system of four 7090's it was found that a single 1301 module could not handle the printer traffic without excessive interference (the simulator assumes that output is randomly distributed throughout the file). However, two output modules handled the load with only 0.08 hours of interference during 49 hours of simulated 7090 operation. The maximum output buffer capacity requirement was approximately 80% of one module. One 1301 module handled both the traffic and capacity requirements for input buffering. References to the programming systems and to the system-information file did not produce measurable system interference.

To date 14 different simulations have been made. In various combinations the configurations have ranged from three 7090's and three 7040's to four 7090's and four 7040's, from ten file control units to twelve, from a low of 24 file modules to a high of 36. Runs were made with one 1402 and both one and two 1403's per 7040. On some runs input and output buffer areas were assigned to the same file control unit; on others this function was split to separate control units. Usually four modules of scratch area were put under each scratch control unit. The idea in assigning the scratch function was to guarantee each 7090 two such channels should they be needed.

For the seven days of work simulated, several running rules were tested. In one situation jobs are made available to be simulated according to their exact time of arrival at the dispatch desk. In another case all jobs to be simulated for the day were assumed available at the start of the day. A third case involved the simulation of all jobs logged in on one day as opposed to those executed on one day.

The numbers at the tops of the columns in Figure 4a are the total 7090 hours by day spent in job execution and set up at Lockheed. On Friday for example, nearly 66 hours were logged to these categories. Also shown is the amount of time the simulated processors operated. Thus the areas at the tops of the columns show the savings in the total 7090 hour requirements associated with changing to the new system and the corresponding reduction in setup time. This improvement is exclusive of that which one would also expect to receive by inclusion of IOCS, etc. Such improvements are not reflected in the areas at the tops of the columns. For the week, the savings in 7090 time used by the new system was over 37 hours, an 11% reduction.

Figure 4a 7090 throughput compared to present system

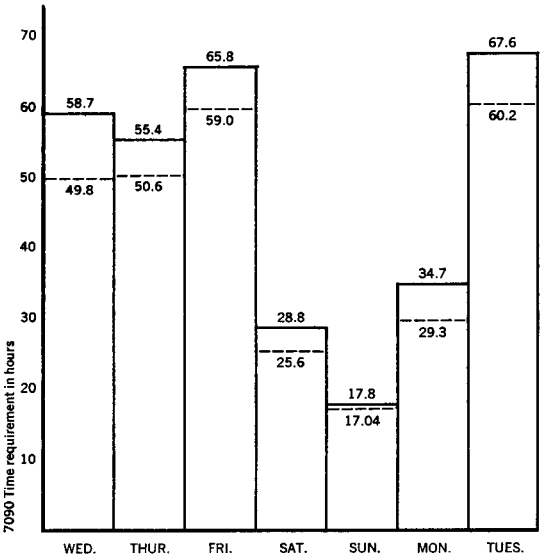


Figure 4b Turnaround time with four 7090's, four 7040's compared to present system

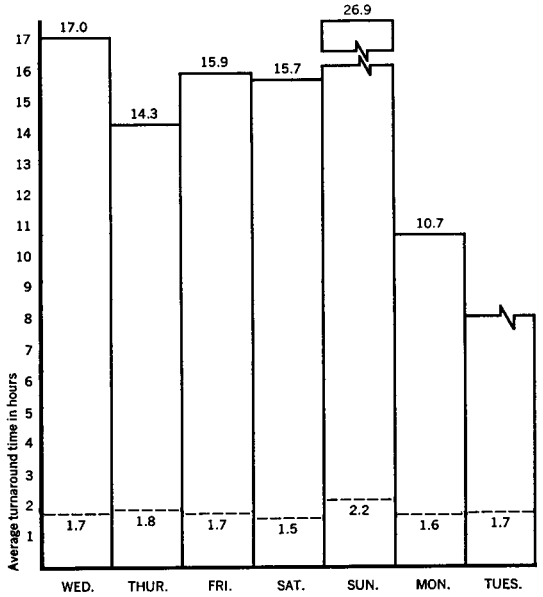
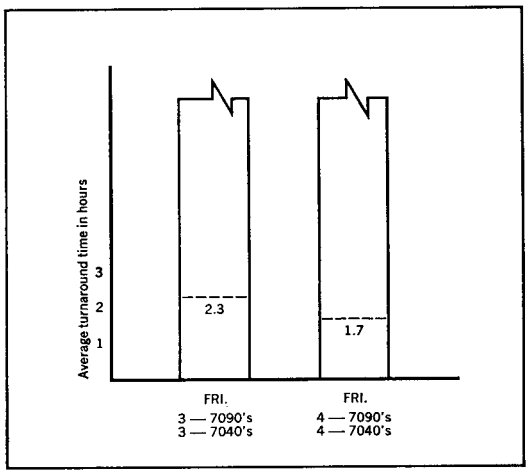


Figure 4c Effect of system size on turnaround time



The effect on throughput efficiency of changing from four 7090's to three was found to be very slight.

In Figure 4b, the magnitude of turnaround improvement achieved is shown by comparison of Lockheed's present operation with a simulation of the new system under an identical work load. The figures are slightly optimistic because they do not include the time required to manually remove forms from the 1403 and separate the reports for distribution.

At the top of each column is the average turnaround actually experienced by Lockheed during each sampled day. On Wednesday the average job including priority, non-priority, production, and check-out took 17 hours from the time it was logged in until it was logged out. The simulated system average turnaround results using four 7090's and four 7040's on the same days, are shown by the lower entries in each column. For Wednesday this time was 1.7 hours, or ten times faster than currently experienced. The smallest improvement was by a factor of nearly 7; the best, by a factor of 12. The average turnaround improvement factor for the week is between 8 and 10, and the average turnaround time is less than 2 hours.

Figure 4c illustrates the effect on turnaround when there are only three 7090's and three 7040's, instead of four of each. It takes about 1/2 hour longer, on the average, to get a job completed.

This concludes the interim report. Obviously, until the programming system analysis is completed, no conclusions can be made as to the feasibility of the particular system considered in this paper. However, it is felt that the techniques presented in this paper will be useful for numbers of other important systems engineering problems.

ACKNOWLEDGMENT

The fundamental role of Lockheed Missiles' Information Processing Center staff in the study has already been indicated. In addition, the authors also wish to express their appreciation to the IBM Data Systems Division product engineers for their assistance.