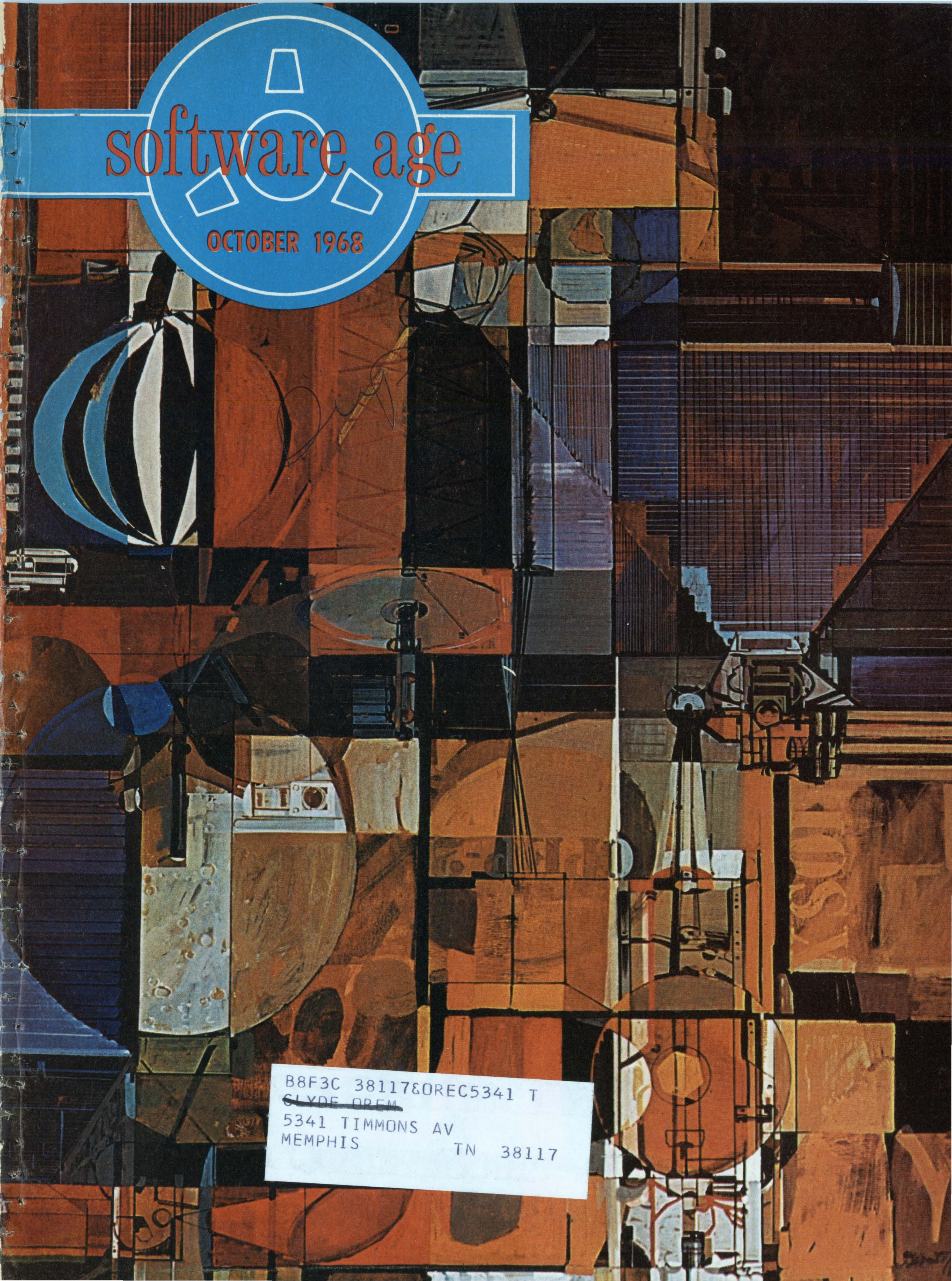
software age

OCTOBER 1968



B8F3C 38117&OREC5341 T
~~CLYDE OREM~~
5341 TIMMONS AV
MEMPHIS TN 38117

Can there be this kind of excitement in programming?

Try Xerox and see

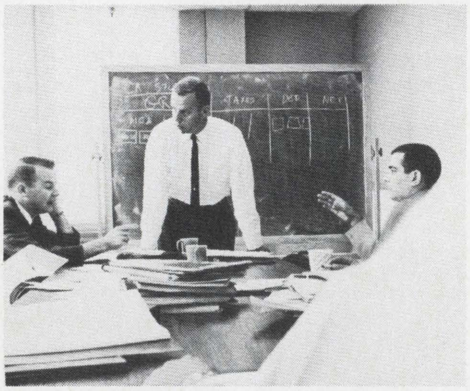
You hear the rapids before you see them—a distant, almost subsonic roar . . . feel the stream pick up speed . . . then a rush through a smooth, narrow chute—and you're in white water! The dangerous thrust of foam-masked rocks, the sweeping power of fast, broken water and the nimble response of the light canoe to your stroke—this is wild, wet excitement of white water canoeing*.

We think the good life includes excitement, off the job and on. And that data processing, as a career, has potential for intellectual excitement offered by few other professions. But not every programming environment can provide it.

We can at Xerox. In breadth and depth.

At Xerox, EDP is a vital ingredient in almost every phase of our operations . . . and becoming more so. Because we're an organization built on a foundation of innovation. And service.

The way we see it, both depend upon data. Accurate data. On the past. The present. To project the future. All along the spectrum from marketing and engineering through manufacturing, service and field support.



One of the advanced systems you might work on at Xerox is EBS . . . Equipment Billing System. Here, Bud Seiph, EBS Manager, leads discussion of new systems approaches.

The kind of programming jobs this creates can work very hard for your future, as well as ours. To start with, it can involve you deeply in many phases of our operations. You learn sales as well as systems analysis. Or distribution as well as disc-oriented software techniques. Too, you'll get exposure to a variety of sophisticated systems, like our integrated Production Planning and Inventory Control System (PPICS) that oversees everything from inventory and bills of material to scheduling and workload distribution. Or a long-range strategic competitive planning model.

All this exposure does more than make you a better programmer. It can be the groundwork for several lines of advancement. As a programming specialist. A systems analyst. A functional specialist in one of the areas you've become acquainted with . . . and into management.

If you've a Bachelor's degree and/or experience with IBM 360 series systems using COBOL, the Univac 1108 using FORTRAN or COBOL, or IBM 7000 series using COBOL or AUTOCODER, we'd like to show you the kind of excitement that goes hand-in-hand with programming at Xerox. These openings are in Rochester, New York. Please send your resume to M. H. Hartigan, Dept. MZ-69-K1, Xerox Corporation, P.O. Box 1995, Rochester, N.Y. 14603.

XEROX

An Equal Opportunity Employer (m-f)

*Rochester is less than a 2-hour drive from some of the finest canoeing in the northeast.

software age

OCTOBER, 1968 Vol. 2—No. 9

CONTENTS **PAGE**

Publisher — H. L. Rothra
Associate Publisher — David W. French
Advertising Manager — Norman Jacobs
Production Manager — Betsy Pavkovich
Director of Circulation — Howard Rogers
Circulation Manager — Judith Arnopolin

Circulation of this issue more than 110,000

COVER ART—John Desatoff, TRW Systems

SOFTWARE AGE is published monthly
by PRESS–TECH, Inc.
(312) 869–1244

1020 Church Street, Evanston, Illinois 60201. Subscription free to qualified readers. Others, $10/yr. Individual copies, $1. Foreign subscriptions, $10/yr.

Main Sales Office: Norman Jacobs (Adv. Mgr.) or Norman Brodsky, 1020 Church Street, *Evanston, Illinois* 60201—Telephone (312) 869–1244. Gerald Green, 60 E. 42nd, *New York, N. Y.*—Telephone (212) 697–5356. Richard Faust, 9800 S. Sepulveda Blvd., *Los Angeles, California* 90045—Telephone (213) 776–0100. *Boston, Massachusetts* 02021—Telephone (617) 542–1466. Richard D. Clemmer, 4900 Red Fox Drive, Annandale, Virginia 22003 *(Washington, D. C.)*—Telephone (202) 461–9792. Richard Faust, *Palo Alto, California*—Telephone (415) 327–8340.

CONTROLLED CIRCULATION POSTAGE PAID
AT MADISON, WISCONSIN

What would you do if your top programmer were activated tomorrow?

Without instant documentation for your programs you'd be in trouble. That's where our Quick Draw comes in. Quick Draw is a programming tool that uses the power of your own computer to produce documentation for your programs. It produces flow charts, format listings, and cross references to data names and paragraph names. And it does it faster than your programmer could hope to. So fast, in fact, you save up to 30% of present programming costs. Quick Draw was developed especially for use with COBOL, FORTRAN, BAL, as well as other assembly languages. And it's applicable for most computers. Interested in having a reserve force of your own? Just send the coupon. No obligation of course.

NCR

THE NATIONAL CASH REGISTER COMPANY, DAYTON, 45409 ®

Tell me more about Quick Draw.

Mail to:
NCR — Quick Draw
Box 111 — Walnut Street Station
Dayton, Ohio 45412

Name ____ Title ____
Company ____
Street ____
City ____
State ____ Zip ____

For more information, circle No. 66 on the Reader Service Card

new products

A major new **proprietary software package** that will enable managers of medium-sized data processing installations to improve day-by-day scheduling, planning and control of personnel and machine time was announced by Brandon Applied Systems, Inc. The package, the Resource Management System (RMS), consists of two subsystems: one for operations control, the other for project control.

RMS is designed to be compiled and run on a minimum 65K IBM 360/30, under DOS. The system is written in COBOL E, with certain routines in ALP where technical factors preclude COBOL. Equipment scheduling can be done for any machine or combination of machines; each is considered as a work station, with qualitative and quantitative attributes. Multi-processing environments are treated as multiple work stations. Interface with OS-360 can be provided for input data and scheduling purposes.

For more information, circle No. 52 on the Reader Service Card

* * *

A recently-introduced family of **tapes for 60-microinch wavelength recording** assure more accurate and dependable data reproduction than any broadband tape now available, according to Memorex Corp.

Bias readjustments and re-equalization of recorders to maintain normal plus and minus db tolerances are unnecessary with Memorex Series 63 tapes. The Series 63 tapes provide a high degree of reel-to-reel uniformity even at wavelengths as short as 60-microinches (2#MHz at 120#ips).

The new tapes are offered in a choice of coating thicknesses to meet all requirements from predetection broadband recording to the full spectrum of instrumentation recording. Series 63 tapes have complete operational compatibility with today's advanced, broadband recorders.

For more information, circle No. 53 on the Reader Service Card

* * *

The Graphic Systems Division of Computer Industries Inc. has announced the development of a new **digital plotting system operating** from punched paper tape. The new system will be marketed as the PTD (Punched Paper Tape Delta Incremental Plotting System). Among the applications of the new PTD are the plotting of engineering variables from time-shared computers at a remote terminal, verification of tapes for numerical control machine tools.

Special features of the PTD include multiple-step programming, which allows up to 127 incremental steps in either of two two-dimensional directions from a single input command. This type of programming allows the plotter to be driven from paper tape supplied by a teletype, and is ideal for remote-terminal graphics, permitting data to be transmitted in delta format, affording significant savings in computer and transmission time. The system employs either a low-cost 12-inch plotter or a more flexible 30-inch plotter.

For more information, circle No. 54 on the Reader Service Card

* * *

An engineer, scientist or any one who uses complicated mathematics can now carry a **lightweight computer/calculator keyboard** to any location, dial into a central computing station from a standard telephone and perform his calculations. System has been introduced by the Mathatronics Division of Barry Wright Corporation, and marketed by the company's Wright Line Division.

The new system uses an acoustical coupler which sends and receives tone signals through a standard telephone receiver.

To operate the system dial the number of the central computing station, place the telephone receiver in a special nest on the keyboard unit and enter problems.

The system also has a number memory that will store up to eight constants, and a formula memory that allows it to learn and automatically work complex formulas.

If desired, the keyboard units may be directly wired to the central station, or can even be operated by means of a two-way radio hook up.

For more information, circle No. 55 on the Reader Service Card

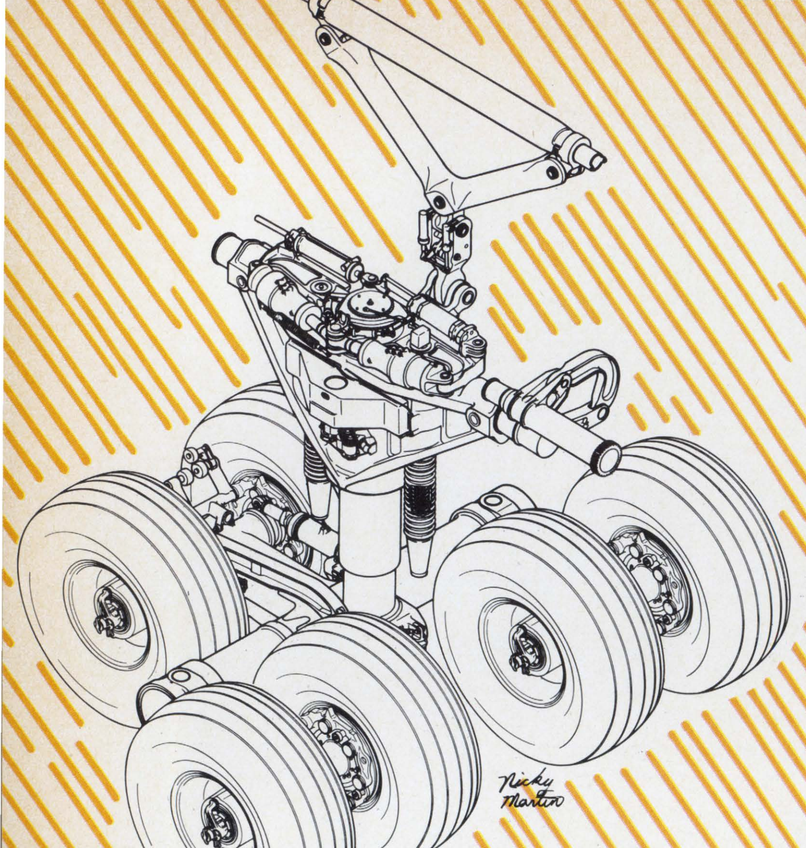
* * *

Scientific Data Systems has developed a **time sharing software system** for its Sigma 5 computer, thus adding simultaneous time sharing to the Sigma 5's batch processing

(Continued on page 30)

Co

of the C



Norman S. Currey

Aircraft Design Engineer Specialist, Advanced Design, Lockheed–Georgia Co.

■ The C-5 landing gear was probably the first landing gear developed by a form of computer-aided design. The sophistication of present-day techniques had not been established at the time of the C-5 study phase, but the various parameters and processes were quite similar.

Prior to the C-5 era, the selection of an airplane's landing gear configuration was a fairly elementary process. There would be two shock struts for the main gear and one for the nose gear. The strut loads were statically determinate, and it was a simple matter to determine tire loads. These loads depended upon how many tires were used, and this was to a great extent a matter of space availability in the landing gear housing. Except in rare cases, there were always one, two or four tires per main gear shock strut, and these were arranged in simple, conventional patterns. Such parameters as flotation (the ability of an airplane to land on a given runway a specified number of times before runway break-up), maintainability, reliability, bump capability and kneeling capability were rarely if ever considered as contractual requirements.

Then came the C-5, and to understand the necessity for sophisticated landing gear selection, it is necessary to review some of the design and operational philosophy of that airplane.

The C-5 is the largest aircraft in the world, with a normal design gross weight of 728,000 pounds. It is about twice as heavy as present heavy transport aircraft, and yet it must be able to operate on the same runways as those aircraft. At a somewhat lighter weight it must be able to operate on bare soil or matted runways, and these surfaces by definition could have a roughness equivalent to 3-inch high step bumps or severe waviness.

A fleet of C-5's can transport a fully-equipped infantry division anywhere in the world, utilizing air strips at which there are few maintenance facilities and a minimum of ground support equipment for loading and unloading the aircraft.

These requirements necessitated a high flotation landing gear to prevent conventional runway break-up at maximum weight, and to permit soft runway operation. The gear had to be designed to traverse substantial airfield roughness such that airframe structural effects were minimized. Maintenance facilities being minimal at remote air strips, the gear had to require little maintenance and had to have high reliability, and in order to eliminate the need for elaborate docks, the gear had to incorporate a device to lower the fuselage close to the ground—this is referred to as a kneeling capability. It was considered that the C-5 would have to land in high cross-wind conditions, so it was required that the gear be able to rotate left and right of neutral to permit such landings. In this case, the landing gear travels straight down the runway while the fuselage remains vectored into wind

mputer Aided Design

-5A Main Landing Gear

by as much as 20 degrees to the runway centerline.

Lockheed guaranteed these capabilities as a part of the contract, and also guaranteed the airplane weight. So despite the fact that the landing gear had to incorporate far more features than any previous gear, it also had to have minimum weight—a combination which provided an unparalleled engineering challenge to the designers of the C-5.

Parametric Study

In determining an optimum C-5 airlifter configuration, three gross weight variants were considered by Lockheed-Georgia for early sub-optimization of the landing gear. These variants had gross weights of 550,000, 650,000 and 750,000 pounds respectively. A parametric study was conducted to define the optimal landing gear for any of these configurations. Study results were fed back and became part of an over-all design and size determination analysis.

During the course of the study, some 660 individual main landing gear configurations were evaluated for each of the three gross weights under consideration. These configurations included three different types of installation and six different bogie shapes. Figure 1 shows the installation types—wing-mounted, underfloor-mounted, and side fuselage-mounted gears. The wing-mounted gear transmits loads directly into the wing structure, permitting a simpler and more efficient fuselage structural design. Under some circumstances, suitably-faired wing pods may enhance aerodynamic characteristics of the wing by improving over-all cross section area distribution.

The underfloor mounting permits a minimum number of wheels in tandem and may obviate the need for main gear steering, resulting in a simpler and more reliable design.

The side fuselage-mounted gear has the advantage of simpler pod fairings and gear door installation in addition to having less effect on underfloor structural design.

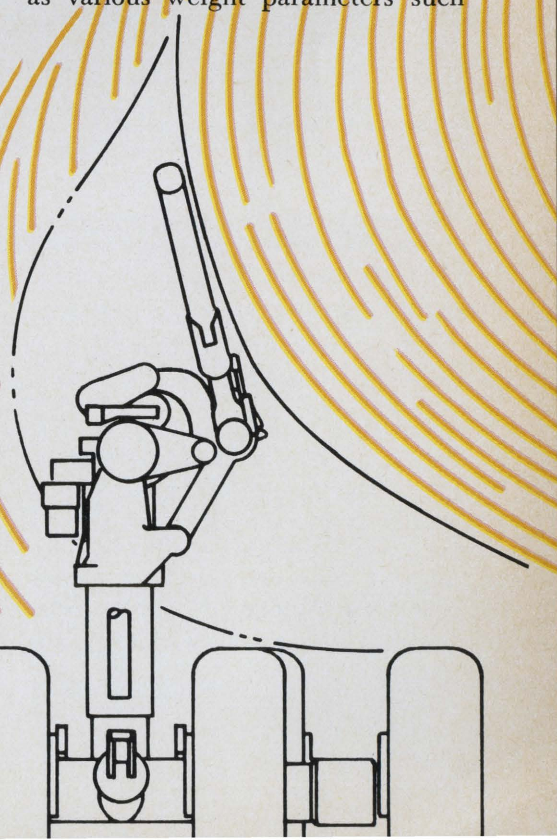
The six bogie types are shown in Figure 2. They all have advantages and disadvantages in specific applications. For instance, Type 3 proved to be the most suitable for the underfloor gear, while Type 5 was the best for a side fuselage mounting. A wing-mounted gear would probably take advantage of the smaller tire sizes afforded by Type 4.

A typical landing gear may weigh some four per cent of aircraft gross weight. For a 750,000 pound aircraft, the general weight category of the C-5, this amounts to 30,000 pounds. An essential goal of the parametric study was to attempt to reduce total landing gear weight to the lowest possible value. The study flow and considerations involved are charted in Figure 3.

Tire data, footprint data, and tire spacing charts were prepared for analysis prior to initiation of landing gear design. A matrix of pod fairing shapes was established and, from this, wetted areas were established for each gear variant being considered. Thus, for any given bogie configuration, the wetted area changed whenever a tire size was changed—unless the tire was so small that it was housed entirely within the natural contour of the fuselage. Weight calculations for each landing gear design, along with pod and flotation data were fed into a digital computer.

Each landing gear variant was appropriately coded for easy identification. An A-4-16 gear for instance would be wing-mounted (type A), having a type 4 bogie, and a total of 16 main gear wheels per airplane—thus, it would have one of these 8-wheel bogies on each wing.

The computer output showed the flotation of each landing gear as well as various weight parameters such



as landing gear weight, fuselage or wing weight change, and effect on take-off gross weight. This output was put through a sorting process which listed all gears in order of flotation, weight, and weight/flotation. The best of 2600 landing gear variants were selected from this assortment for each gross weight design. A second screening further evaluated kneeling capability, practicability, and rough field capability to produce a final listing for selection of the optimal main landing gear for the C-5.

The nose landing gear was omitted from this study. It was obvious at the outset that this would be fairly conventional, and its only effect would be to modify the flotation values. After the main gear had been selected, the nose gear flotation effect was calculated for that particular gear.

Tires

Tires must have adequate static load rating, minimum size for stowage, maximum contact area for flotation, minimum weight, a wheel

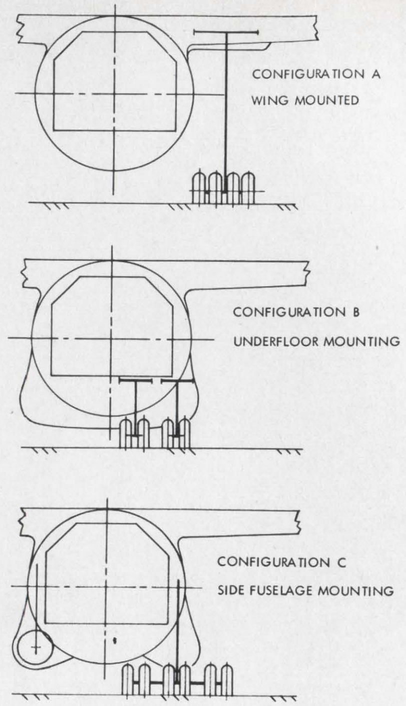


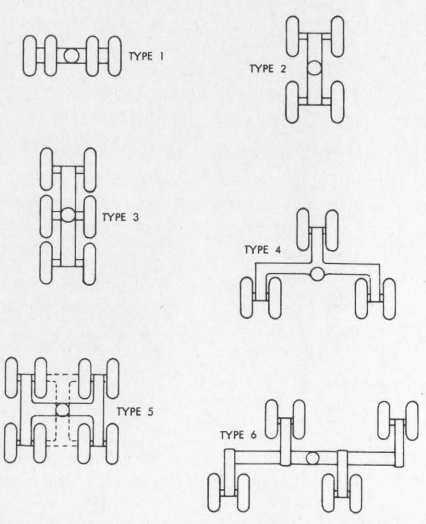
Fig. 1. Installation Types

diameter consistent with brake requirements, and a section height large enough to "swallow" the bumps on the airfield.

It is evident that an incompatibility exists between these factors. A smaller tire is better for minimum stowage space, and minimum weight. A large tire is better for contact area, load capacity and, as it usually has a larger wheel, it is then better for brake installation. For maximum bump capability a large tire on a small wheel is best.

The ultimate tire selection is one of the most important factors in choosing an optimum landing gear. Consequently, 27 tires, all having

Fig. 2. Bogie Types



SOFTWARE PROFESSIONALS

WE'RE LOOKING FOR PEOPLE WITH PROBLEMS

We have plenty of solutions. What we need are problems...and people who are good at thinking them up.

We're lucky to have some of the most outstanding software people in the country out here in Sunnyvale, California. And we need the best. Because our assignment is the total information system segment of orbiting satellite systems. Includes planning, commanding, data management and post flight analysis. We not only design information systems to fill present needs, but anticipate and project information system segments of space systems that will most likely be needed in 1970 or '75.

It should come as no surprise to learn we have more operational software development and more operating on-orbit experience in complex spacecraft than anyone in America.

Now, since most people are trained as problem-solvers, the people we hire will be quite extraordinary. In a way, we're asking for trouble. That is, we're asking for people who habitually create more problems than they're worth...to most other companies.

What we'll do when we find them is make their problems worth more. With us, they'll be creating more worth than problems, even though they're still creating problems.

Get it?

If you do, you may have real problems. The kind we're looking for.

If you're one of the professionals with a degree in EE, Physics or Math and have 2-10 years applicable experience, see if you can create problems for us in any of the following areas:

MATH SPECIALISTS—COMPUTER APPLICATIONS Provide detail design, development, maintenance, modification and improvements for command and control, event and data evaluation computer programs for use on assigned satellite projects.

PROGRAMMERS Coordinate with systems engineers, ops analysis engineers, and users and/or the customer to determine basic requirements for computer program development and modification. Document, design, flow chart, code, check-out and validate computer programs.

ANALYSIS ENGINEERS—COMMAND AND CONTROL Provide technical coordination to assure compatibility of computer software programs and data base for operational vehicles and on-orbit operational support for the hardware/software interface areas. Background in analysis and operation of airborne systems and sub-systems with emphasis on digital command and telemetry systems. Knowledge of computers, associated software. An understanding of software logic and Jovial language required.

SPECIALISTS — VALIDATION AND TEST Establish test procedures, at the system, sub-system, and module level from preliminary and final design specifications of software programs. Conduct extensive validation tests to ensure compatibility with all developed programs.

Please send details of your educational background and experience, including salary requirements, to Mr. I. Cutter, General Electric Co., Space Systems, Room 164-K, 1003 West Maude Ave., Sunnyvale, California 94086.

Command & Information Systems Operations

SPACE SYSTEMS

GENERAL ELECTRIC

An Equal Opportunity Employer (M/F)

adequate load rating, were considered in the parametric study.

As indicated in Figure 4, an approximate logarithmic increase in flotation is achieved with a reduction in tire load, a coverage being defined as a sufficient number of passes to completely cover the runway traffic lane one time with adjacent tire tracks.

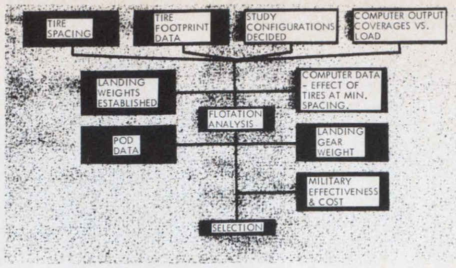


Fig. 3. Method for Flotation Evaluation

Tire load can be decreased by increasing the number of wheels. The greater the number of wheels, however, the more difficult it is to avoid wheels tracking one behind the other, thereby limiting load distribution over the terrain. Besides degradations in reliability, maintainability, cost, weight and stowage capabilities, increasing the number of wheels requires a larger pod fairing, adding to the aircraft drag. Keeping these factors in mind, main landing gears with 8, 12, 16, 18, 24 and 32 wheels were considered as configurations during the course of the study.

Measured contact areas applicable to 35 per cent tire deflection were used as study criteria. No attempt was made to evaluate pressures needed to withstand operational loading conditions for each tire size. This would have involved checking load distributions among tires for all bogie configurations, with and without flat tires, for standard and substandard airfields, and under all loading conditions such as braked turn, steady turn and 2g taxi operation. For the purposes of the study, there was very little to be gained from making such calculations.

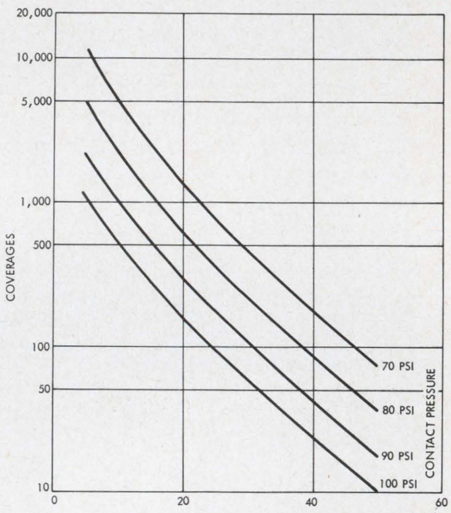


Fig. 4. Flotation vs. Tire Load and Pressure

Figure 5 shows the tire footprint data used in the analysis. The columns show some of the parameters used in calculating flotation. At that time, the flotation criteria indicated that there would be no interaction between tires spaced 4R or more apart, where R is the radius of the circle equivalent to the contact area. A column is provided, therefore, to show the minimum tire spacing for optimum flotation.

Tire Spacing

Among all the various bogie configurations, there were cases where 4R tire spacing could not be used, and there were other cases, such as

Systems Programmers and Systems Analysts:

Give IBM a hand and we'll try to make it a free one.

If predictable thinking were what we wanted, we'd just program it. But what we're looking for are imagination and ingenuity. If you have these qualities, and you qualify for a job with IBM, you can be sure you'll get to use them.

Immediate openings

We have immediate openings in these areas of data control:

SYSTEMS PROGRAMMING—You could work close to the system, very much into its own particular nuances and idiosyncrasies, and adapt programs to maximize the data processing potential of the computer. Or write entirely new programs to link the actual problem to its solution.

Or you could develop symbolic languages for both commercial and scientific use, and concern yourself with computer performance, serviceability, systems analysis, component evaluation, and the development of diagnostic programs.

SYSTEMS ANALYSIS—Because you're close to the problem, its thorns and its kinks, your job is creating an overall data processing strategy that solves it. The problem itself could be just about anything: payroll, inventory control, process control, long-range financial forecasts—to name a few.

A Bachelor's degree and one year's experience are required for the above positions.

Grow with IBM

Today's major growth industry is information handling and control. And IBM is a leader in that field. As a result, the professional challenges at IBM are more demanding, more sophisticated than ever. Our job is to help our customers solve their information-handling problems. And IBM people are at work helping them solve a myriad of problems from diagnosing a strange disease to urban redevelopment.

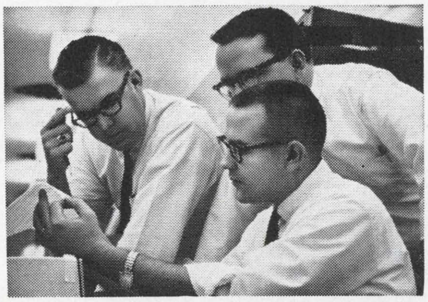
So if you're a problem-solver who wants a personal sense of achievement and recognition for your hard work in an exciting growth company, consider IBM.

Write us now

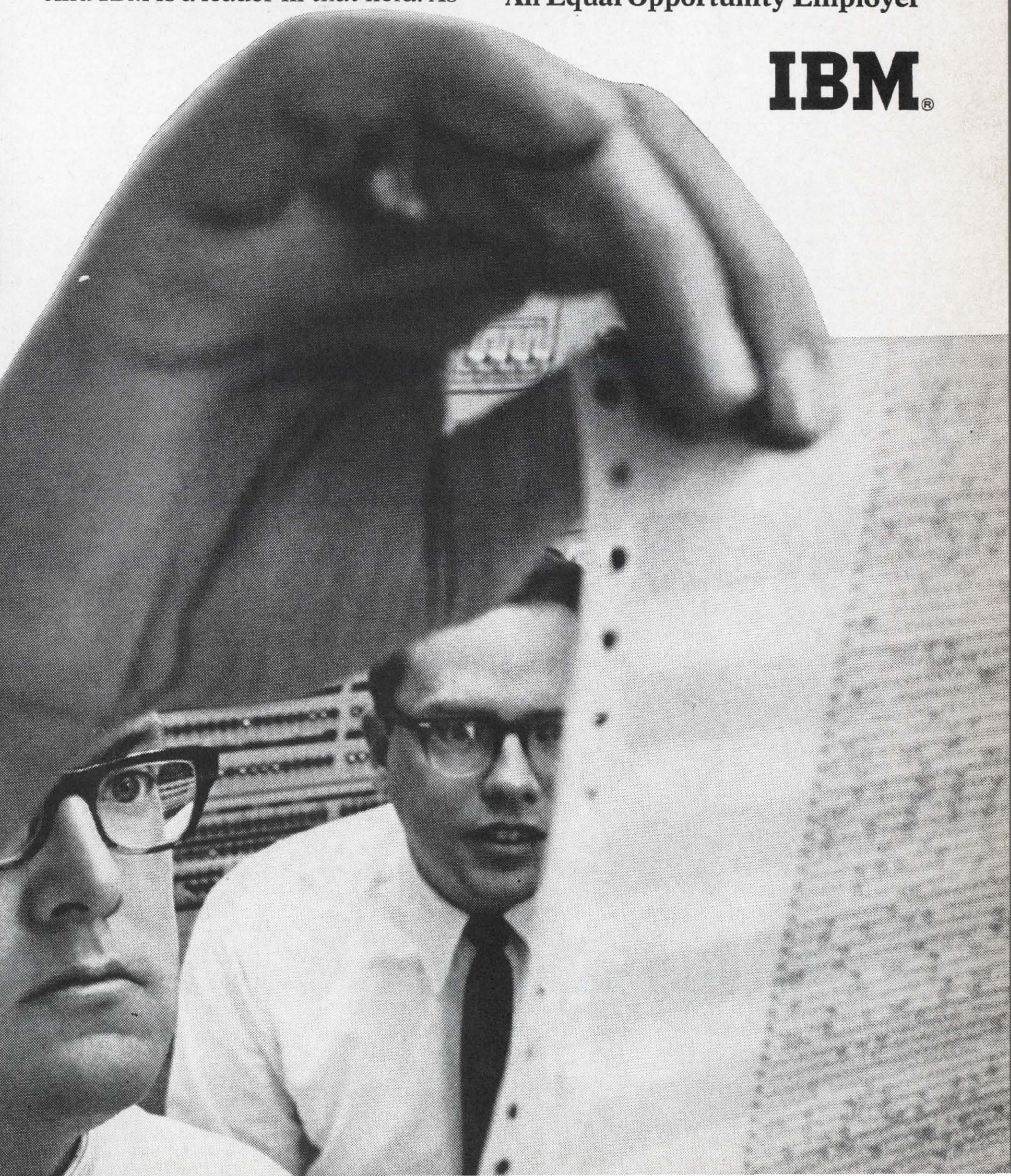
Jerry Hinkley is the man to talk to. Write to him at IBM Corp., Dept. BK1039, 24th Floor, 425 Park Ave., New York, N.Y. 10022.

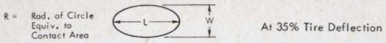
An Equal Opportunity Employer

IBM®



IBM programmers and systems analysts work in small teams on an IBM System/360 computer.





Tire Type	Tire	Contact Area A	$\frac{A}{\pi}$	$\sqrt{A}$	R $.564\sqrt{A}$	W $.874\sqrt{A}$	L $1.457\sqrt{A}$	4R	80+W	.75W
III	13.50 - 16	181.5	57.8	13.47	7.60	11.77	19.63	30.40	91.77	8.83
	15.00 - 16	219	69.7	14.80	8.35	12.94	21.56	33.40	92.94	9.71
	16.00 - 16	244	77.7	15.62	8.81	13.65	22.76	35.24	93.65	10.24
	17.00 - 16	268.5	85.5	16.38	9.24	14.32	23.87	36.96	94.32	10.74
	15.50 - 20	230	73.2	15.16	8.55	13.25	22.09	34.20	93.25	9.94
	15.50 - 18	244	77.7	15.62	8.81	13.65	22.76	35.24	93.65	10.24
	17.00 - 20	264	84.0	16.25	9.17	14.20	23.68	36.68	94.20	10.65
	20.00 - 20	382	121.6	19.54	11.02	17.08	28.47	44.08	97.08	12.81
	19.00 - 23	330	105.0	18.16	10.24	15.87	26.46	40.96	95.87	11.90
	25.00 - 28	526	167.3	22.94	12.94	20.05	33.42	51.76	100.05	15.04
VII	30 x 7.7	68	21.6	8.25	4.65	7.21	12.02	18.60	87.21	5.41
	30 x 8.8	68	28.0	9.38	5.29	8.20	13.67	21.16	88.20	6.15
	46 x 9	115	36.6	10.73	6.05	9.38	15.63	24.20	89.38	7.04
	36 x 11	120	38.2	10.96	6.18	9.58	15.97	24.72	89.58	7.19
	38 x 11	126	40.1	11.22	6.33	9.81	16.35	25.32	89.81	7.36
	40 x 12	163	51.9	12.77	7.20	11.16	18.61	28.80	91.16	8.37
	39 x 13	147	46.8	12.12	6.84	10.59	17.66	27.36	90.59	7.94
	44 x 13	203	64.6	14.25	8.04	12.45	20.76	32.16	92.45	9.34
	40 x 14	183	58.2	13.56	7.65	11.85	19.76	30.60	91.85	8.89
	44 x 16	219	69.7	14.80	8.35	12.94	21.56	33.40	92.94	9.71
	46 x 16	220	70.0	14.83	8.36	12.96	21.61	33.44	92.96	9.72
	56 x 16	278	88.5	16.67	9.40	14.57	24.29	37.60	94.57	10.93
	49 x 17	261	83.1	16.15	9.11	14.12	23.53	36.44	94.12	10.59
VIII	41 x 15.0-18	186	59.2	13.64	7.69	11.92	19.87	30.76	91.92	8.94
	50 x 20.0-20	276	87.5	16.58	9.35	14.49	24.16	37.40	94.49	10.87
	45 x 24 - 16	307	97.7	17.52	9.88	15.31	25.53	39.52	95.31	11.48
	56 x 30 - 15	603	192.0	24.56	13.85	21.47	35.78	55.00	101.47	16.10

Fig. 5. Tire Footprint Data

four tires side-by-side on one axle, where it was of interest to learn the effect of close tire spacing. An empirical formula was developed to ascertain minimum spacing when two tires are mounted on one wheel, as with the B-58 Hustler, or with wheels very close together. This showed:

Where:

$S_{min} = 0.40\,(2H_t + W_t + W_f)$

S_{min} = Minimum tire spacing

H_t = Tire section height

W_t = Tire section width

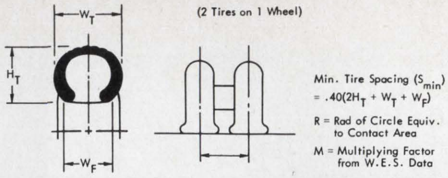
W_f = Width between wheel flanges.

Results of this formula were tabulated for various tire types and sizes with two tires on one wheel, and for a bogie beam located between wheels. Figure 6 shows the tabulation for two tires on one wheel.

Computer Data

Charts were prepared for each of the permutations of the 660 configurations investigated. In order to develop these charts, the flotation was calculated for each case, the weight of the gear was evaluated, and pod and drag effects were determined. Then, for each configuration, tire size and gross weight, the following data was supplied to the computer:

Structure weight (pod + airframe structure effect)
Landing gear weight.
Wetted area increase due to pod (S)
Flotation
Factor K_1—see below
Factor K_2—see below



Tire Type	Tire	S°_{min}	R	S/R	M_{max}	$S_{op\ max}$	M_{min}
III	13.50 - 16	19.52	7.60	2.57	1.36	30.40	1.0
	15.00 - 16	21.18	8.35	2.54	1.365	33.40	1.0
	16.00 - 16	22.18	8.81	2.52	1.37	35.24	1.0
	17.00 - 16	23.88	9.24	2.58	1.355	36.96	1.0
	15.50 - 20	21.80	8.55	2.55	1.363	34.20	1.0
	15.50 - 18 *	22.60	8.81	2.57	1.36	35.24	1.0
	17.00 - 20	23.70	9.17	2.58	1.355	36.68	1.0
	20.00 - 20	28.64	11.02	2.60	1.35	44.08	1.0
	19.00 - 23	26.49	10.24	2.59	1.355	40.96	1.0
	25.00 - 28	35.34	12.94	2.73	1.319	51.76	1.0
VII	30 x 7.7	10.90	4.65	2.34	1.416	18.60	1.0
	30 x 8.8	12.52	5.29	2.37	1.408	21.16	1.0
	46 x 9	12.44	6.05	2.06	1.485	24.20	1.0
	36 x 11	15.84	6.18	2.56	1.362	24.72	1.0
	38 x 11	15.84	6.33	2.50	1.377	25.32	1.0
	40 x 12	17.62	7.20	2.45	1.388	28.80	1.0
	39 x 13	18.10	6.84	2.65	1.339	27.36	1.0
	44 x 13	19.22	8.04	2.39	1.403	32.16	1.0
	40 x 14	19.52	7.65	2.55	1.363	30.60	1.0
	44 x 16	21.80	8.35	2.61	1.348	33.40	1.0
	46 x 16	21.80	8.36	2.61	1.348	33.44	1.0
	56 x 16	22.94	9.40	2.44	1.390	37.60	1.0
	49 x 17	23.70	9.11	2.60	1.352	36.44	1.0
VIII	41 x 15.0 - 18	20.30	7.69	2.64	1.341	30.76	1.0
	50 x 20.0 - 20	26.50	9.35	2.83	1.293	37.40	1.0
	45 x 24 - 16 *	29.40	9.88	2.98	1.256	39.52	1.0
	56 x 30 - 15 *	38.30	13.85	2.77	1.308	55.40	1.0
	° Minimum for two tires on one wheel * Hypothetical tires						

Fig. 6. Tire Spacing

In the aerodynamic evaluation of the pods, it was recognized that each pod would have different drag characteristics, and by holding a constant payload-range capability for each airplane weight, the pod drag can be interpreted as a change in take-off gross weight. In other words, a high drag pod will necessitate additional fuel to meet that payload-range capability, and this additional fuel is reflected in increased take-off weight. Thus K_1 is a factor predicated upon statistical and wind tunnel data, differing for each weight and pod type considered. The value of K_1 for a side fuselage-mounted gear varied from 10.8 for a 550,000 pound airplane to 16.9 for a 750,000 pound airplane, and the change in take-off weight due to pod drag was given by

$$\triangle TOGW = K_1 \times S_w$$

where S_w is the difference between the pod wetted area and the wetted area of the fuselage or wing covered by that pod.

Factor K_2 reflects the effect of incremental weight changes on take-off gross weight at a constant wing loading and thrust loading. Thus, increasing the diameter of a tire increases the weight of the gear, and this results in an airplane weight increase of K_2W, where W is the structure weight plus gear weight.

The value of K_2 did not change with landing gear type, and varied from 3.0 for 550,000 pound airplanes to 3.9 for 750,000 pound airplanes.

Thus, the weight effect was recognized in total by the following equation:

$$\triangle TOGW = (K_1 \times S_w) + (K_2 \times W)$$

For wing-mounted gears this equation was further extended to include the effects of flap span reduction.

Flotation data was calculated for both optimum and minimum tire spacing for each configuration being considered, and the methods used are given in AFSCM 80–1—Handbook of Instructions for Aircraft Designers.

Inflight Tire Deflation

The computer analysis also showed the effects of using inflight tire deflation. This is a system which deflates the tires in flight such that the tires are at their optimum deflection at landing, just as they were at take-off. Take-off tire pressure is predicated upon take-off weight. During flight, weight is reduced due to fuel usage, and when the airplane lands its required tire pressure may well be considerably less than it was at take-off. Flotation is improved whenever tire pressure is reduced, and the inflight tire deflation system provides this reduction. However, any additional system has a cost, weight, maintainability and reliability impact, so the merits of such a system were evaluated as part of this study.

Computer Analysis

One set of charts was prepared showing all the data pertaining to each specific landing gear configuration. For instance, under the heading of a C-4-24 gear—a side fuselage-mounted gear with a type 4 bogie and 24 main gear wheels per aircraft—the chart showed the flotation and weight effect for each of the 17 possible tire sizes, with and without inflight tire deflation, at optimum and minimum tire spacing for each of the three gross weights considered.

Computer data then sorted out all the gears so that one sheet showed them in order of weight, one sheet

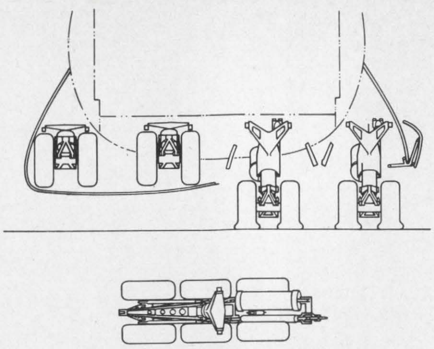


Fig. 7. Underflows Landing Gear

showed them in order of flotation, and a third sheet showed them in order of flotation per pound of weight. Thus, one gear may have indicated that it had extremely high flotation, but unless its weight was compatible, it would not necessarily be a choice gear.

An operational analysis had been conducted concurrently with the parametric study which showed the flotation that the airplane should have for each gross weight. For instance, it was shown that a 550,000 pound airplane would have to have far better flotation than the 750,000 pound airplane, as more of the "light" airplanes would be required to land at a given field in order to land a specific total load. The precise flotation values were determined by mission analysis, and these showed which landing gears were best suited for the particular airplanes.

For each airplane weight a graph was plotted which showed how all the "best" gears related to each other in terms of airplane gross weight and flotation. Knowing what the flotation value should be for that weight, it was then possible to select the best main landing gears for each of the three gross weights. In the graphical analysis there were 184 chosen gears, and from this 28 were chosen for final screening.

Landing Gear Selection

The final selection did not involve any computer operation. It was, rather, a logic analysis. The initial studies had shown that the airplane should have a kneeling capability to lower the cargo floor such that the lower fuselage was very close to the ground. If any of the selected gears were not capable of providing this capability then they were eliminated. For instance, an underfloor gear with large 56 x 16 size tires would not be able to provide this feature because of the large tire diameter situated between the cargo floor and the ground.

Various aspects of practicability were then considered, and if a particular arrangement did not appear too practical it was eliminated. A typical case would be four large tires side-by-side at optimum spacing. Such a design would lead to severe axle design and braking problems.

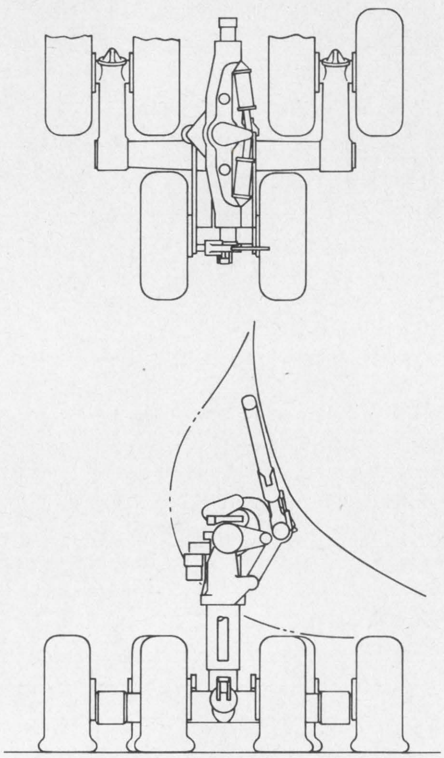


Fig. 8. Selected Main Landing Gear

The tires of the select gears were then evaluated for bump capability, and at that time a six-inch bump height was used as the criteria. To "swallow" such a bump, the tire had to have six inches between its normal loaded radius and its flat tire radius. If such a capability was not available then that gear was rejected.

Any of the remaining gear configurations having a high weight-to-flotation ratio were eliminated, and this left nine gears for final evaluation.

A maintenance, reliability and cost analysis was made on these gears, and from this two configurations emerged as the most suitable main landing gears for the C-5. They were the present twin tricycle arrangement, using a Type 5 bogie mounted on the side of the fuselage, and an underfloor arrangement using a type 3 bogie and having 24 main gear wheels.

The underfloor arrangement is shown in Figure 7. It was believed that main gear steering would not be required with such an arrangement. The retraction method was relatively simple in that the shock strut merely rotated fore and aft. In addition, the bogie was relatively light due to minimization of torsional loads. A comparison of the two competing designs revealed, however, that the underfloor-mounted gear incurred extremely severe door problems and had less flotation than the finally-selected side-fuselage-mounted design.

Finally-Selected Main Landing Gear

A diagram of the finally-selected six-wheel bogie and tire arrangement is presented in Figure 8. Other details of the gear can be seen in Figure 9. The mechanism required for crosswind positioning is located at the top of the gear. It rotates the bogie up to 20 degrees left and right of neutral depending upon the magnitude of the crosswind during landing.

The kneeling system is shown in Figure 10. The main gear is suspended from the fuselage main frames at two trunnion points. These two trunnions are part of a large aluminum alloy yoke forging. The shock strut is free to slide up and down through a hole in the center of the forging. The top of the shock strut terminates in a T-shaped crosshead. Ballscrews are suspended from the crosshead and pass through ballnuts housed inside the yoke. By rotating the ballnuts, the crosshead is separated from the yoke and the yoke slides downward around the shock strut until it bottoms out on the bogie beam.

A chain drive connected to an air motor, powered by engine bleed air drives the ballnuts within the yoke. The system can be braked to a stop at any point of travel desired.

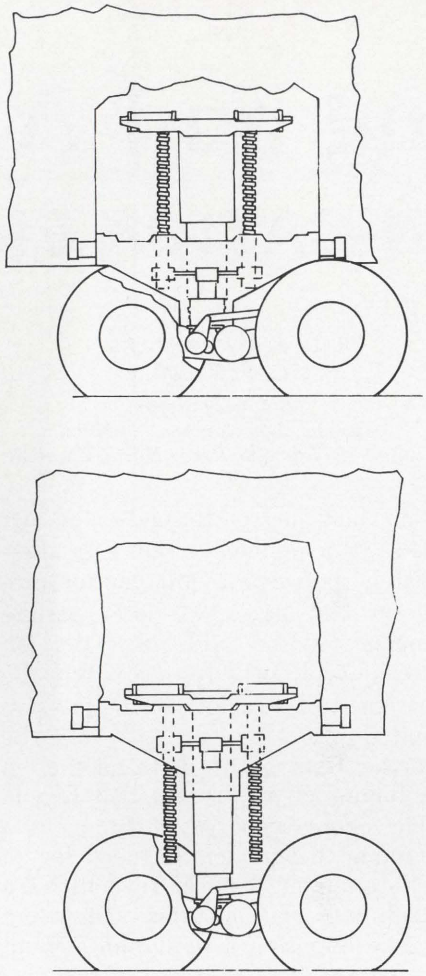


Fig. 10. Kneeling System

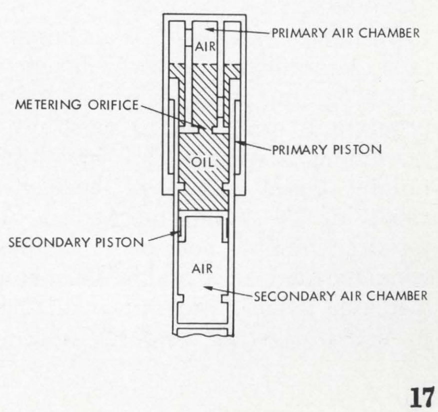
To retract the landing gear, a rotation cylinder on the top of the crosshead rotates the strut and bogie through 90 degrees so that the forward wheels are facing the aircraft centerline. The landing gear is then swung about its trunnion points until it reaches its retracted position beneath the cargo floor.

The shock strut is a unique double-acting design in which there are two chambers instead of one. As shown in Figure 11, it has a primary chamber which acts in the same way as a conventional shock strut. But at the base of that chamber there is a secondary piston which floats at the top of an air-charged secondary chamber. This chamber is preloaded by compressed air in excess of maximum static requirements, and when the landing gear encounters airfield roughness this secondary chamber acts in the same way as a hydraulic surge damper, absorbing unsprung mass momentum and attenuating peak loads. In this way the loads which would be transmitted to the airframe structure are greatly reduced.

At the base of the shock strut—where it attaches to the bogie—there is a universal joint. This permits the bogie to rotate in any plane as it traverses roughness, ensuring that all three wheel pairs of that bogie are always in contact with the ground for optimum load distribution.

In summary, the C-5 landing gear has undergone a long and rigorous development. With its 24 main gear wheels, arranged in an optimum pattern, it can operate at maximum gross weight on the same runways as present-day jets which are less than half the weight of the C-5. At a reduced weight it can make more than a hundred landings and take-offs from a bare soil field. With close attention having been paid to detail design it is predicted that it will meet its stringent maintainability guarantees, and this, together with its kneeling system, will minimize the amount of equipment required to support the airplane at remote bases. One interesting feature of the kneeling system is that it is possible to kneel individual main gears. In this case, instead of the fuselage being lowered, one gear is raised, with the other three gears supporting the airplane. This obviates the necessity for jacking to accomplish routine landing gear maintenance—a feature which paid off during taxi tests when a tire blew. The affected gear was raised clear of the ground and the airplane taxied back to the hangar for a tire change. It is also of interest to report that the test pilots have found the airplane extremely easy to handle on the ground, despite its tremendous size and its 24 main gear wheels. One indication of this is the fact that it has turned 180 degrees in 112 feet width of runway—well within its guarantee. ■

Fig. 11. Dougle-Acting Shock Strut



Scheduling and Allocating

—IN A COMMUNICATION ENVIRONMENT—

L. J. Endicott, Jr.
L. S. Kreger
IBM Corporation
Systems Development Division
Research Triangle Park, North Carolina

■ The primary purpose of systems programming has always been to provide general programs which can be used by many users, to avoid having everyone code the same routines. It is still an important purpose. Today, however, there is an additional, more pressing reason for system programs: to guide the user in fully utilizing all of the system's resources—both software and hardware—at his disposal. Guidance can be provided through the structure of the language and the allocation of resources.

Allocation of resources includes scheduling, which in a priority system occurs at the time a priority is established. Scheduling must not be confused with dispatching, which is a simple queue-management function. Scheduling includes: *what* gets on the dispatcher's queue, *how* it gets on the queue, *when* it gets on the queue, and where it goes into the queue.

Allocation may be static or dynamic. Static allocation is predetermined and fixed by the application program for a specific application. There is no allocation manager to assign resources. Dynamic allocation occurs when resources allocations—including CPU time—are made or changed during program execution.

This "rate of change" is an important determinant of the efficiency of a program and varies widely from program to program. In most programming, resources are not optimally shared because of the inefficiency of the sharing mechanism. A resource might not be available when needed, or it might be made available when a program is unable to use it. Phyical resource allocation may be performed once per IPL, or assignments may pass alternately between a job control program and successive jobs. Assignments may be made once per job step, or physical resources may be assigned to a region during job-step initialization, which then can be passer from one task to another in a hierarchy of tasks in that region. The more resources existing in a system, the greater the need for an allocation mechanism that allows a high rate of change in assignment while imposing a minimum amount of overhead.

A mechanism has been designed, as part of the Queued Telecommunications Access Method (QTAM), which allows a high rate of change of resource allocation, and an external language which can be modified by the user. (This is an IBM Type I Program, available in Operating System/360 and Disk Operating System/360.) The language is structured from several types of delimited macros. The user is shown what types are to be used in which construction of his program—guiding him in the construction of his program. These facilities aid the user to achieve maximum utilization of his total system's resources. Although QTAM was developed for scheduling and allocating of communication systems, its principles are applicable to any control program. However, it was designed with the special requirements of a Tele-processing environment in mind.

The Tele-Processing Environment

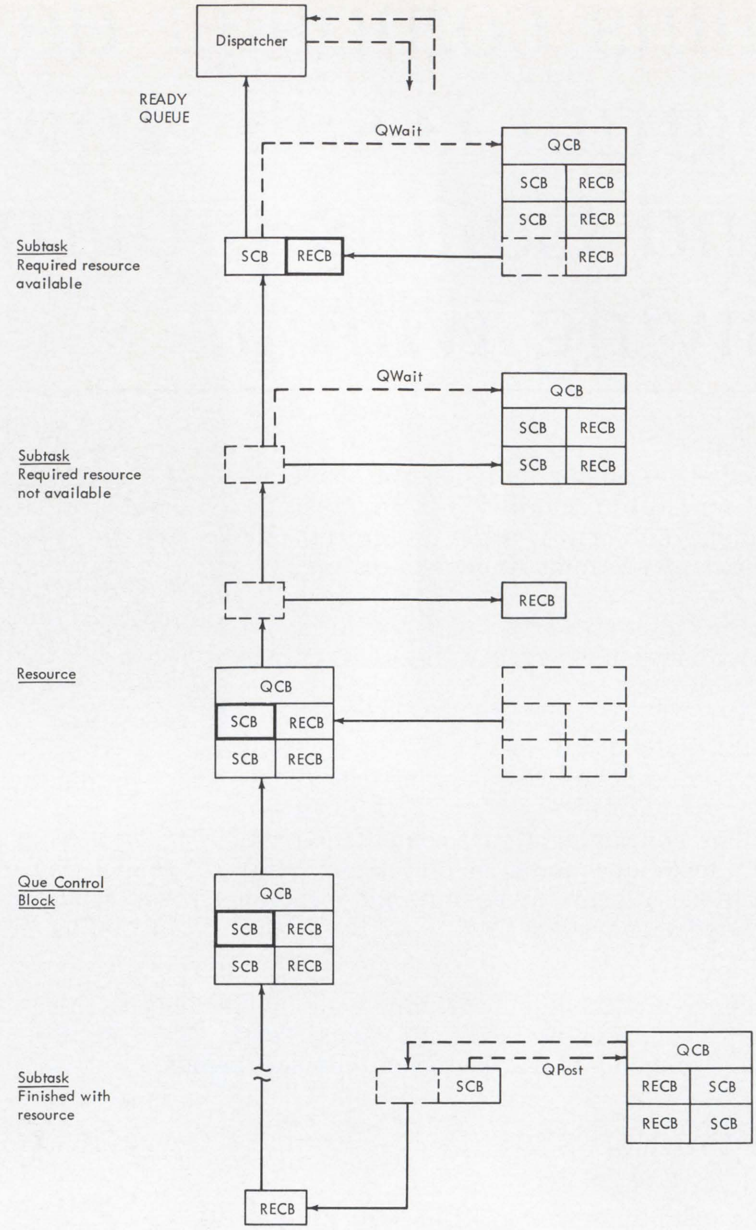
Tele-processing—data communication—I/O is not generically different from tape or disk I/O; its differences are in degree rather than in kind. The need for dynamic allocation with a high rate of change results from the peculiarities of a Tele-processing environment. Among these are:

Slower Data Rate—The transmission rate for the IBM 1050 Data Communications System, for example, is just under 15 characters per second. At this rate, it becomes intolerable to tie up a full record area for the entire time it takes to read a complete message. A 400-character record will require more than 26 seconds to read in after the first character has been received. If only half of the space were made available for only half of the time, over 5,000 200-character records conceivably could be read and processed from our fastest tape drives.

Greater Number of I/O Channels—A 100-line system is not considered impossible in a Tele-processing environment. A 100-line system with an average of five terminals on a line and five I/O components per terminal would amount to an admittedly quite large system of 2,500 I/O devices. Exceptional, perhaps, but not impossible.

Higher Error Rate—Error rates of only one in several million characters are achievable with tapes or disks. However, because of the long transmission lines, etc., one transmission error in 5,000 characters may be considered quite acceptable in some Tele-processing applications. Further, these are usually more critical than in telephone and teletype transmissions where the meaning in context often neutralizes errors.

More Device Control Considerations—A Start I/O for a Tele-



Legend

QCB - Queue Control Block
SCB - Subtask Control Block
RECB - Resource Element Control Block

Figure 1. Disposition of Ready Queue Elements

processing operation addresses only as far as the line adapter on a control unit. Many steps must be taken beyond that to operate the actual device at a desired terminal station. Polling and addressing, reading and writing responses, asking and answering "Who are You's"—all of these functions greatly extend the time and complicate the process of transmitting and receiving data. Many telecommunications devices were never intended to be attached to a computer; it is up to the systems program to make up the difference.

More Man/Machine Interface Considerations—A new set of problems arises when the I/O device is actually being operated by a human. The program must be able to carry on a conversation and appear intelligent while doing it. Meanwhile, the man on the line is keying in data, making mistakes (some of which he recognizes), asking to cancel in the middle of a transaction on which processing has already begun, asking for something to be repeated that he lost because he was out of paper, and becoming irritated when the program hesitates in order to handle a higher-priority situation.

Highly-Variable Data Structure and Volume—Tele-processing I/O is normally not done at the request of a processing program; on the contrary, the program is usually executed at the request of the I/O. A read to a line is simply an invitation to whatever is out there to begin sending. When the start I/O is given, the program may not know whether there is anything to be read, how much is to be read, or the content or format of the data. In normal data processing, 80-character (card-size) blocks are often used. Tele-processing blocks may vary from eight to 32,000 characters. Also, instead of three or four types of data, as many as 50–80 may be received for processing.

It is the cumulative effect of all these factors which complicates the programming in a communications environment. The most important factors are the variable structure of the data and the greater number of I/O channels. The variable length of messages makes core allocation more complex. Sometimes, blocks must be divided between core and tape or disk. The large number of types of data encountered in Tele-processing requires as many different types of programs.

A large number of resources to be serviced, and a relatively small number of other resources with which to service them, makes it necessary for all resources to be shared. The solution is to break up the resources into the smallest practical resource units, and to dynamically allocate them as needed to the smallest practical work unit.

In order to achieve efficient movement of resources through the QTAM system, subtasks were created. A subtask controls the manipulation of a resource. Optimally, a subtask should require only a single resource in order to perform the unit of work for which it was created. A subtask waits for a resource, operates upon it, and passes it to the next subtask. Thus, the multiplication of resources strongly affects the tasking concept.

Programming Aids

The Queued Telecommunications Access Method provides many aids to the construction of a Tele-processing system: addressing, polling, dialing, answering, sending, receiving, buffering, routing, queuing, error detection and correction, header analysis and synthesis, logging, translation, etc. Here, however, we are primarily interested in the way in which these facilities are offered to guide the user in constructing a more efficient system.

QTAM is much more than an access method. The user first must construct a complete message-control program. Once this is done, he can proceed to write any number of message-processing programs, using the simple OPEN, GET, PUT, and CLOSE macro instructions as with any other access method. The message-control program is constructed through the use of environmental and procedural macro instructions. The process of using these macro instructions is more like using a high-level compiler language than using typical macro definitions. It is an extremely flexible language, including the possibility, but seldom the necessity, of direct assembler code. At the same time, it is structured in such a way as to lead the user step-by-step through the entire process.

The message-control program is really a generalized switching program: messages are switched from one communications line to another or from a line to a processing program; they are switched from one processing program to another, or from a program to a communications line. The operation of this message-control program is completely asynchronous to that of the processing programs. It continuously monitors the network, reading messages as they appear and writing them in disk queues. Output mes-

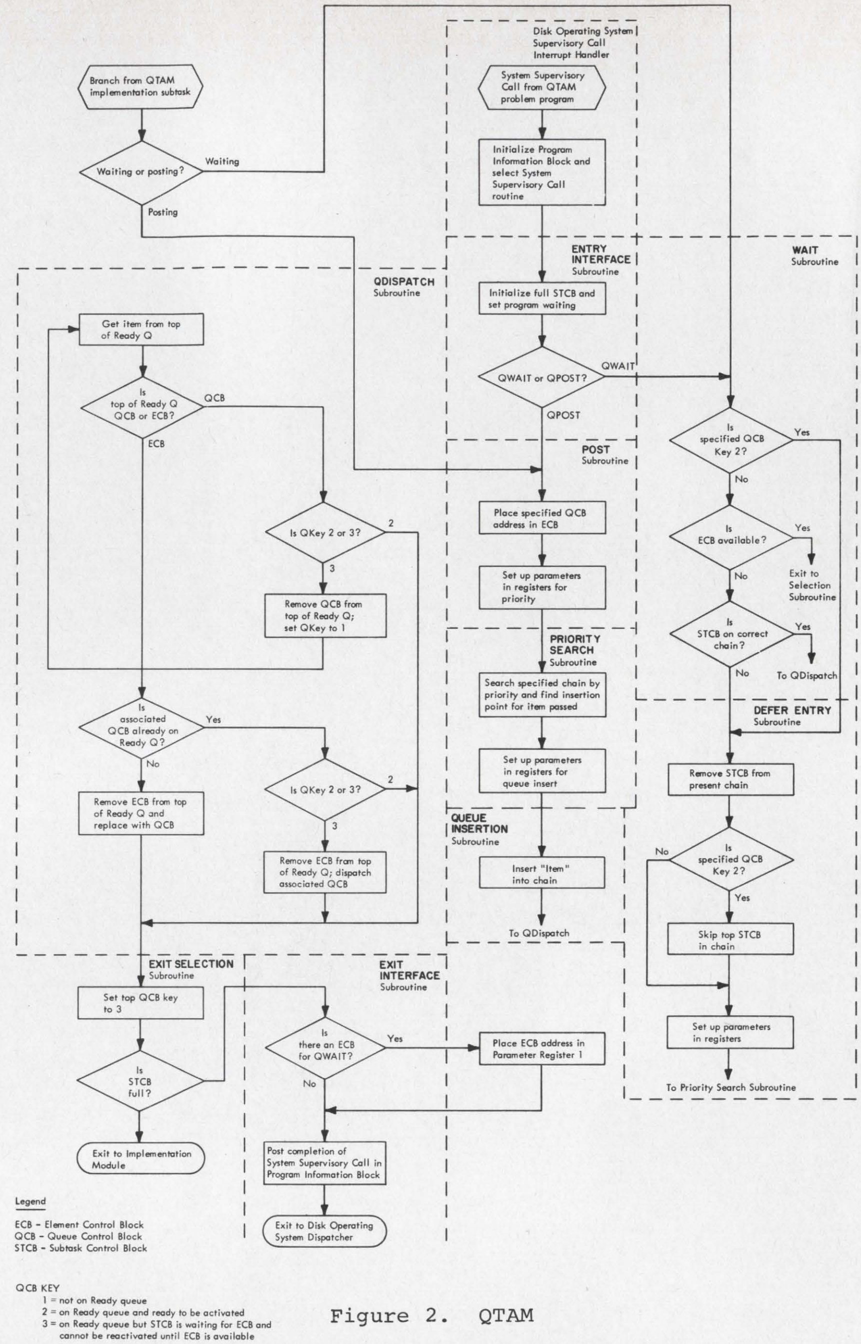


Figure 2. QTAM

sages are taken from disk queues and sent out on the lines as the lines become available. Message-processing programs are loaded as necessary and transfer messages from and to disk (through message control).

Message control is run in the highest-priority partition; the message-processing program (or programs) in lower priority. The user is aware only of this relatively straightforward, 2-program structure. In order to understand the priority structure of QTAM, however, it is necessary to step outside of the operating system framework and look at QTAM as an independent, multi-tasking control program.

Resource Allocation

QTAM resource allocation is controlled by a set of routines executed as part of the operating system supervisors. These allocate resources and dispatch subtasks.

All resources (lines, buffers, and buffer-request blocks) are represented by Resource Element Control Blocks in core. A Resource Element Control Block can represent one type of element at one time, another at a different time. For example, an empty buffer assigned to receive a message is an entirely different type of resource than that same buffer later when it contains a message segment and is waiting to be written on disk.

All programs are organized into subtasks, each of which is represented by a Subtask Control Block. Most of the subtasks are permanently defined. For the most part, these are the functions involved in the actual control of the lines and the disk queues, and they are executed in supervisor state, disabled to interruptions. In addition to these, there are the problem-program subtasks created whenever the message-control program or one of the message-processing programs requests or passes a resource.

The key to the allocation and scheduling mechanism of QTAM is the structure through which a resource is allocated to a subtask—the Ready Queue. The nodes of the allocation and scheduling mechanism are Queue Control Blocks which queue both resources waiting to be operated upon and subtasks waiting for a particular type of resource. There is a Queue Control Block for each type of resource in the system.

Queue Control Blocks can be looked at as "nodes" of the system from two points of view, corresponding to the two different definitions of the word "node." First, each is a node in the sense of a knot which ties together two strings (a string of resource with a string of subtasks). It is also a node in the sense of a point where a moving process becomes still. As a resource is passed from one subtask to another (or, from another point of view, as a subtask passes from one resource to another) it comes to a resting place on the Queue Control Block. If conditions in the system require it, resources and subtasks can be allowed to build up into queues chained to the Queue Control Block.

When a subtask is finished with a resource, it posts or chains the resource directly to the Ready Queue, in proper priority sequence. Thus, at any given moment, the Ready Queue consists of a chain of Queue Control Blocks and Resource Element Control Blocks arranged in a priority sequence. The position of all items on the Ready Queue is determined by the relative priorities of resource elements as they are passed to the queue. Generally speaking, the priority of a resource element is determined by the type of subtask to which it is being passed.

When a subtask requires another resource, it issues a QUEUE WAIT on the proper Queue Control Block. If the resource is available, control is returned to the requesting subtask immediately. If not, the Subtask Control Block for this subtask is chained to the Queue Control Block, control passes to the QTAM dispatcher, and the next item on the Ready Queue is examined. In those cases where more than one subtask may request the same resource, the Subtask Control Blocks may be chained to the Queue Control Blocks in priority sequence, as are the Resource Element Control Blocks.

A subtask requests the resources it requires for its execution from the appropriate Queue Control Block, executes its function, and then passes the resource to the proper Queue Control Block for the next function to be performed. The rate at which a subtask acquires resources is based on the availability of the resources and the priority of the subtask itself. The availability of resources is optimized by allocating only those resources required, and only when they are required.

A more subtle, difficult to describe factor that enters into the QTAM allocation mechanism is the capability to decide the actual priority of a resource dynamically, depending on the current status of the system. For example, a buffer-request block which is "asking" for a buffer for a line operation is normally of lower priority than a buffer-request block "asking" for a buffer for a disk operation.

However, before posting a line buffer-request block to the Ready Queue, QTAM compares the length of the disk I/O request queue to the number of *high*-priority buffer requests. If there are more I/O requests than there are high-priority buffer requests, it is an indication that the priority of the line-buffer request can probably be raised without showing the disk operations.

The effect of all this is to shift the importance of different functions in the system. The tendency is to increase efficiency by allowing the system to decide at the time a resource becomes available how it is to be used. QTAM is self-regulating, allowing optimum resource utilization in a constantly-changing environment.

Conclusions

The approach to allocation and scheduling described here, and demonstrated by the QTAM program, can perhaps be best put into perspective by examining the extent of dynamic allocation achieved in current IBM System/360 systems. In the basic programming systems, all allocation is static within a given job. With Disk Operating System and Operating System "Multi-Programming with Fixed Tasks," multijobbing, and allocation are static throughout a job step. Although these systems share processing time between the various jobs, this must not be confused with dynamic allocation. The assignment of relative priorities to these jobs (or rather

Checkmate!

If you would like to see a regular column on chess with a monthly problem and appropriate prizes for its solution, write:

Feature Editor
software age
1020 Church Street
Evanston, Ill. 60201

the assignment of a job to a partition) determines the allocation of time throughout the job step. The dispatcher simply activates the job to which time was previously allocated.

Operating System "Multi-Programming with Variable Tasks" takes the next step, allowing multitasking at least with a region, with resources now being assigned to a region for the duration of a job step. Within that region, however, the resource can be given from one task to another and then back again. Thus a resource may be considered assigned to one of these "daughter" tasks only for the duration of that task's existence.

In QTAM, however, a resource is never, in a sense, statically assigned to a subtask but is passed to the subtask at the moment that the subtask is dispatched. When a subtask has finished operating on the resource, it can pass the resource on to the next subtask and—most significantly—the subtask can *ask* for another. Furthermore, when a resource becomes available, it can, in effect, ask for a subtask.

This last point is the essential difference between QTAM and other priority-scheduled control programs. By treating subtasks and resources in exactly the same way, by allocating and dispatching them with a single-queue management facility, it becomes possible to base the first level of priority on the resource itself, instead of on a physical partition or even on a task. Since all resources, including physical resources, subtasks, and time, are allocated by a single mechanism, it becomes possible for the first time to base allocation on:

- first, the availability and priority of resources;
- second, the priority of the task.

Most of the facilities provided by QTAM are peculiar to a Tele-processing environment. So too is the absolute requirement for such a high rate of change in resource allocation. However, a single mechanism of this type could be provided for an entire operating system. Such a system should allow a greater utilization of total system resources, with less overhead cost, than when allocation is performed through several different mechanisms. ■

NASA Developed Software Available

■ Computer software developed in activities of the National Aeronautics and Space Administration is now readily available to the U.S. public. Tapes, card decks, run instructions, program logic—complete software packages—can be obtained for only the cost of handling, reproduction, and distribution.

This is the result of a joint effort by NASA and the University of Georgia. In July, 1966, the University of Georgia was awarded a contract by NASA to operate the University's Computer Software Management and Information Center (COSMIC), which disseminates computer programs and computer information emanating from the programs of NASA (and, in some cases, from other sources).

COSMIC is working as a part of the NASA Technology Utilization Program, which seeks to enlarge the return on the public investment in aeronautical and space activities by encouraging secondary uses for the results of such work. Thus, computer programs developed for space use that have broader utility are accessible to business, industry, education, medicine, and other sectors of the economy.

COSMIC is working through the NASA Marshall Space Flight Center Technology Utilization Office, in conjunction with all other elements of NASA. Under the terms of the contract with NASA, COSMIC must operate in an efficient manner designed to make the Center self-supporting on the basis of fees for services rendered.

One of the principal functions of the NASA Technology Utilization Program is to identify technological advances derived from the space effort and to make them available for use by industry, business and education. One especially useful aerospace by-product for many organizations is a wide range of well-documented, operational computer programs. By making these computer programs easily available, NASA expects to contribute directly to the nation's industrial effort. Companies can avoid duplication of programming effort and shorten the task of developing certain computer programs.

The Computer Center at the University of Georgia has gained extensive experience over the last few years in providing computer services and assistance in computer applications to industrial and business firms. The Center employs a professional staff of statisticians, mathematicians, biologists, numerical analysts, engineers, chemists, physicists, and information and computer scientists.

The two major computer systems in the Center are the IBM 360 Model 65 and the IBM 7094 with two IBM 1401 systems serving as input/output peripheral units for the 7094. In addition, an IBM 1620 computer and an EAI TR-20 analog computer are operated on an open-shop basis.

Computer programs to be disseminated by COSMIC are obtained primarily from NASA-supported activities. These programs, many of which have broad applicability, are evaluated to determine their potential utility to American industry, educational institutions, and other prospective users. Availability of programs added to the COSMIC inventory is announced periodically. Any U.S. organization can request to be placed on a mailing list to regularly receive such announcements. Interested parties may request computer program documentation on any program announced by NASA or COSMIC, and may also request source decks which will be provided in tape or card form depending upon the requestor's preference. Program decks over 2,000 cards, however, are not disseminated in card form without special agreement with COSMIC. All requests for tape copy will be provided on tape reels furnished by COSMIC.

Each requestor is charged for duplication, handling and mailing of both computer program documentation packages and source decks.

Effective July 1, 1967, COSMIC instituted a new fee schedule. The fee schedule for computer program dissemination services is structured so as to offset the operational costs of COSMIC. The price of computer program *documentation* is:

Document Size	*Price*
1– 25 Sheets	$ 1.50
26– 50 Sheets	3.00
51–100 Sheets	5.00
101–200 Sheets	10.00
201–300 Sheets	12.50
300-Over	15.00

A complete Computer Program Package, which includes the documentation, source deck, and a listing of the program, is available for $75.00.

There may be a few exceptions to the above fee schedule, especially, extensive programs which require special processing, handling and duplication. These programs which are exceptionally bulky and/or complex are noted in the announcements of the programs. Special fees are also charged for programs of a highly specialized nature that have been developed by NASA but—because of limited use—not stored in the COSMIC inventory.

A complete directory of abstracts for all available computer programs is disseminated periodically.

If you would like to be added to the mailing list for dissemination of this directory, or if you would like additional information, contact: COSMIC Computer Center, University of Georgia, Athens, Georgia 30601. (404) 542–3265. ■

market place

The Classified Ad Department of SOFTWARE AGE

SERVICE AVAILABLE

Part Time Programming Wanted,
360 OS FORTRAN & COBOL
Warwyk, Box 141, Bethpage, N. Y. 11714

SOFTWARE OFFERED

CIRCULATION & MAILING LISTS: complete tested (working) software for card or mag-tape conversion, maintenance and label printing on 360/20. Card version require 20/20, 8K with 2560 and 2203 with 120-c line. Tape version require 20/20, 12K with 2203 or 1403, 2560 and two tape drives. Produces 4-up E-W Cheshire labels @ 15,000/hr with 2203; up to four lines of address, with or w/o additional matchcode line, using unusual 1-cd. format providing 4 cols. (up to 8 items by using zone and digit pchs separately) of classification, demographic and/or expire data. Programs identify "suspect" duplications (i.e. near but not total duplication) for operator decision. Automatic zip markings to speed mailing operations. State selection, every Nth name and other selective label print programs also available. Write or phone H. L. Rothra, PRESS–TECH, Inc., 1020 Church St., Evanston, Ill. 60201. (312) 869–1244.

MOONLIGHT SERVICE

MOONLIGHTERS WANTED: Whatever your experience—COBOL, RPG, NEAT or BAL in business applications, or the wildest machine languages for math/science applications—we have part-time work in your area. Best contract rates, your hours, for analysts, programmers, documentation specialists and top-flight operators qualified for debugging adequately documented programs. Send complete qualifications—areas of experience, machines and languages with which you are familiar to: Dept. 890, SOFTWARE AGE, 1020 Church St., Evanston, Ill. 60201.

market place

CLASSIFIED ADS in software age

where you can reach **110,000** programmers, mathematicians, analysts and EDP managers.

CLASSIFIED ADVERTISING

Non-Display Classified: For firms or individuals offering commercial products or services, 75¢ per word (including name and address). Minimum order $7.50. For Blind Ad Service, an additional $10.00. For "Positions Wanted" Ads, 55¢ per word (including name and address). No minimum. **Payment must accompany copy** except when ads are placed by accredited advertising agencies. Frequency discounts: 5% for 6 months; 10% for 12 months paid in advance.

Display Classified: One inch by one column, $70.00. Column width 2¼". Photographs accepted for an additional $20.00. Advertiser to supply all photo, art, cuts, or camera ready copy.

General Information: One inch display Help Wanted and Employment Services ads **will be** accepted in the classified section. Employment ads ⅛ page or more will appear run of book, will be keyed to the resume form in back of publication, and will qualify to free daily resume service.

Closing Date: 1st of preceding month (for example, May issue closes April 1st).

Send order and remittance to: Classified Dept., Software Age, 1020 Church Street, Evanston, Illinois 60201.

software age
1020 CHURCH STREET • EVANSTON, ILLINOIS • 60201

CLASSIFIED ADVERTISING ORDER FORM

Please refer to the above information for complete data concerning terms, frequency discounts, closing dates, etc. Cash with order.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35

______ Words { @ 55¢ (Reader Rate) / @ 75¢ (Commercial Rate) } = $______

______ Include Photograph @ $20.00 additional (**Display Ads Only**)

☐ Run Ad Blind $10.00 Insert ______ time(s) Total Enclosed $______

NAME ______

ADDRESS ______

CITY ______ STATE ______ ZIP ______

Signature ______

WORD COUNT: Include name and address. Name of city (Des Moines) or of state (New York) counts as one word each. Zone or Zip Code numbers not counted. (Publisher reserves right to omit Zip Code if space does not permit.) Count each abbreviation, initial, single figure or group of figures or letters as a word. Symbols such as 35mm, COD, PO, AC, etc., count as one word. Hyphenated words count as two words.

SA-10

NEW PRODUCTS

(Continued from page 6)

capability, and increasing the number of time sharing computers in the SDS product line to four.

The new Sigma 5 time sharing system is a lower priced version of the time sharing system available with the Sigma 7.

The new Sigma 5 Batch Time-Sharing software permits interactive time sharing for up to eight simultaneous users; at the same time the system runs batch problems written in FORTRAN, COBOL, or assembly language. Batch throughout is maintained at a high level in the time sharing Sigma 5 by dedicating primary system resources and significant percentages of time to batch jobs and assigning on-line users to specified time slices as their demands require.

Sigma 5 time-sharing configurations include eight teletype terminals, 48,000 words of core memory, 6-million bytes of Rapid Access mass storage, and the usual complement of peripherals.

1. The SDS 945 is especially suited for universities and research, or industrial organizations. It provides conversational time sharing for up to 24 simultaneous users plus a large selection of proven programming languages. The SDS 945, which leases for less than $15,000 per month, is one of the lowest priced conversational time sharing systems available in the industry.

2. The SDS 940, which is a more capable time sharing system than the 945, is designed especially for commercial time sharing service centers or large research or business organizations. It serves 32 simultaneous conversational users, provides very fast response, and stores large amounts of data for immediate retrieval.

3. The Sigma 5 time sharing system is specifically intended for applications in which large general purpose scientific problems are solved in a batch mode; at the time a number of programmers and other users perform simultaneous time sharing.

4. Sigma 7, the largest system offered by SDS, can solve scientific oriented problems, perform business data processing jobs and real time tasks, while simultaneously providing conversational time sharing.

With the addition of the Sigma 5 time sharing configuration to the SDS line, SDS now offers a greater number of time sharing systems for a larger variety of applications than any other company.

For more information, circle No. 56 on the Reader Service Card

* * *

Payroll Software System—COBOL system designed for an IBM System 360 with 65K under DOS or OS. It features 50 state accounting, sequential cost allocation, multiple deductions, multi-company (5 levels) accounting, and personnel records. **Federal, state, and city taxes compiled** for quarterly 941-A reporting. Handles hourly, salary, overtime, commissions, bonuses, and advances. Employees may be paid by cash or check with a payroll statement issued for cash payments. Deposit slips can be provided where direct employee deposits are made to a bank. Personnel record is provided which includes job classification and other personnel information.

For more information, circle No. 57 on the Reader Service Card

* * *

A ten-minute color film describing the strides in computer tape technology which have enabled the development of a non-certified, yet guaranteed performance tape, has been announced by 3M Company.

Produced by 3M's Magnetic Products division, the film is shot on location in a modern data processing facility and the company's magnetic tape plant in Hutchinson, Minn. Included is a history of the evolution of computer tape and an extensive look at the quality control behind the new "Guaranteed Performance" product. Narration is by Peter Jennings (ABC News).

3M will provide the projection equipment and personnel for showings in individual data centers.

For more information, circle No. 58 on the Reader Service Card

* * *

An unusually wide line of **perforator tapes in the fanfold format** has been announced by Robins Data Devices, Inc., subsidiary of Robins Industries Corp., College Point (Flushing), N. Y. 11356. Seventy-one types are listed.

The new fanfolds have application in data processing, communications, numerical control, photo-typesetting and computer functions. They are especially useful in short-run applications on data processing equipment and have advantages in storage because fanfolded tape can be filed more readily with original documents.

The perforator tapes come in paper, mylar and laminated materials, oiled, unoiled, and in colors. They are also printed with directional arrows, or unprinted.

For more information, circle No. 59 on the Reader Service Card

* * *

Arvey Corporation, Chicago, has added **three new Mylar reinforced combinations to its present line of perforator tapes.** Identified as R–V–CZ–64, .003″, Mylar/foil/Mylar; R–V–CP–115, .004″, paper/Mylar/paper; and R–V–CT–52D, .0028″, metalized Mylar, the tapes are reported to offer improved performance for N/C applications.

The company states that it produces the most versatile range of perforator tapes for numerical control, data processing, computer programming and equipment testing. Tape thicknesses range from .0043″ down to .0015″. All standard colors, widths and thicknesses are available.

Tapes are also produced to custom specifications. Arvey tapes are opaque to both infrared and visible light and meet the most rigid requirements for photoelectric and electro-mechanical readers; offer exceptional dimensional stability and are sold in guaranteed continuous measured roll lengths without splices. Inspection and testing methods comply with MIL standards.

For more information, circle No. 60 on the Reader Service Card

TROUBLE-TRAN PRESENTS XTRAN'S ADVENTURES IN FORTRAN

Send your ANSWER to the problems posed here in each issue to:

TROUBLE-TRAN EDITOR
software age
1020 Church St., Evanston, Illinois 60201

You can also profit by submitting PROBLEMS for this feature. If your problem in FORTRAN programming is selected for use in this feature, you will receive $25.00

Contest Rules:

1. USA Standard FORTRAN is assumed.
2. CDC-6000 FORTRAN and IBM System/360 FORTRAN IV (level H) are used in verifying answers.

The correct answer bearing the earliest postmark will net the reader submitting it $25.00

The second correct answer with earliest postmark wins $15.00

By GEORGE N. VASSILAKIS
of *TRW*'s Software and Computing Center

PROBLEM OF THE MONTH

Mr. XTRAN was asked to zero-out, in the beginning of his main program, all the labeled CØMMØN storage he was using. He reasoned that, since the main program would always be loaded first and the loader would assign contigious storage to all the blocks of CØMMØN storage, one DØ loop would suffice.

So, here is what he did.

```
   Main Program
   CØMMØN/R/R(5)              10 R(J) = 0.
   CØMMØN/X/X/X/Y/Y/Z            .
   CØMMØN/W/W(10)                .
   DØ 10 J = 1,18                END
```

Will the DØ loop zero-out all CØMMØN storage? Is the statement CØMMØN/X/X/X/Y/Y/Z a legitimate FORTRAN statement? If yes, what does it mean?

ANSWER TO LAST MONTH'S PROBLEM

First, I would like to apologize for the mysterious appearance of the alphabetic character Ø in places where the numeric 0 (zero) should have been. The problem is that the typesetting is done by non-programmers; and, because of publication deadlines, I do not have the opportunity to proofread the final text.

In last month's problem, you were asked to complete the coding that would enable subroutine B to transfer to statement 100 in the middle of subroutine A without subroutine B using the RETURN or CALL statements.

The given coding was:

```
     SUBRØUTINE A                      SUBRØUTINE B
     .                                 .
     IF(1.GT.0) GØ TØ 10               .
100  WRITE(3,110)                      .
110  FØRMAT(14H RIGHT YØU ARE)         .
     RETURN                            .
 10  .                                 IF(1.LT.0) RETURN
     CALL B                            END
     .
     END
```

The key to this problem is to create a label 100 in CØMMØN storage and transfer to it from subroutine B. Here is one solution:

```
     SUBRØUTINE A                      SUBRØUTINE B
     CØMMØN /M/M                       CØMMØN /M/M
     IF(1.GT.0) GØ TØ 10           10  IF(1.LT.0) ASSIGN 10
100  WRITE (3,110)                       TØ M
110  FØRMAT(14H RIGHT YØU ARE)         IF(1.LT.0) GØ TØ 20
     RETURN                            GØ TØ M, (10)
 10  ASSIGN 100 TØ M               20  RETURN
     CALL B                            END
     RETURN
     END
```

This problem worked on the IBM 7094 and the CDC-6500, but did not work on the IBM system/360. I would appreciate hearing from users of other systems.

Time and space do not permit me to go into all the details of why this problem did not work on the IBM 360; however, it has to do with the "base-displacement addressing approach" and the ability to relocate programs and data at load time.

P. S. Once again I would like to ask my readers if there is any interest in rotating problems with FOTRAN algorithms.

June Winners: 1st—Alfred E. Riccomi, Madison, Wisconsin; 2nd—Val Tareski, Urbana, Illinois.

software age

THIS INQUIRY IS IN DIRECT RESPONSE TO YOUR ADVERTISEMENT IN SOFTWARE AGE MAGAZINE

CONFIDENTIAL INQUIRY

Your original copy of this form will be retained at the offices of SOFTWARE AGE and will be used for no other purpose than to notify the specific firms which you have checked (on the reverse side) of your interest.

TYPE OR PRINT CLEARLY FOR PHOTO REPRODUCTION

JOB DESIRED: ______________________

List computer hardware knowledge (names of systems, tape, disk, terminals, etc.): ______________________

Programming specialties and years of experience (commercial, scientific, theoretical, experimental, analog, etc.): ______________________

Systems programming on which you have had development experience (compilers, assemblers, executives, monitors, O.S., etc. Indicate for what computer): ______________________

Programming languages used and extent of experience (COBOL, FORTRAN, etc.): ______________________

Applications programmed (aerospace, banking, insurance, math subroutines, compilers, etc.): ______________________

Systems analysis experience (card design, flow charting, operation analysis, etc.): ______________________

EDP management experience (include years and number of people reporting to you): ______________________

SALARY: __________ (current) __________ (desired) **DATE OF AVAILABILITY:** __________

EDUCATION: Indicate major as well as degree unless self-explanatory.

Degrees			
Years			
Schools			

EMPLOYMENT: Indicate present employment and previous jobs below.

Employer			
City			
Years	to	to	to
Title or Function			

Name __________ Age __________

Home Address __________ Home Phone __________

__________ (city) __________ (state) __________ (ZIP code) U.S. Citizen? __________

Security Clearance __________ Location Preference __________

Marital Status __________

Military Status __________

BE SURE YOU HAVE CHECKED ON REVERSE SIDE THE COMPANIES YOU WANT TO SEE THIS INQUIRY. PUT FORM IN STAMPED ENVELOPE AND MAIL TO:

software age
MAGAZINE
1020 CHURCH ST., EVANSTON, ILL. 60201

AFTER DECEMBER 10, THIS FORM WILL NOT BE PROCESSED
WRITE ADVERTISERS DIRECTLY OR WRITE US FOR CURRENT S/A ISSUE

check your interests here

Fill in the confidential inquiry form on the other side of this sheet. This form provides all the information advertisers require to screen applicants. If further information is desired, you will hear from the advertiser direct. Then, check below the boxes of those companies to which you want copies of your form sent. Mail to Software Age, 1020 Church Street, Evanston, Illinois 60201. **(Please do not send us your own resume. We will only process this form. A new form must be filled out for each issue in which you are answering ads.)**

EMPLOYMENT AND SEARCH AGENCIES

PRODUCTS AND SERVICES
(Use Reader Service Card)

☐ **I do not now receive S/A. Please enter my FREE subscription.**

....
Name   Street Address

....
City   State   Zip Code

................................../...................................
Prime Experience in What Industry   My Specialty

☐ Technical Degree   ☐ Non-Technical Degree   ☐ No Degree

....
Year Born   ☐ I Have Analog/Hybrid Experience

software age

1020 Church Street
Evanston, Illinois 60201