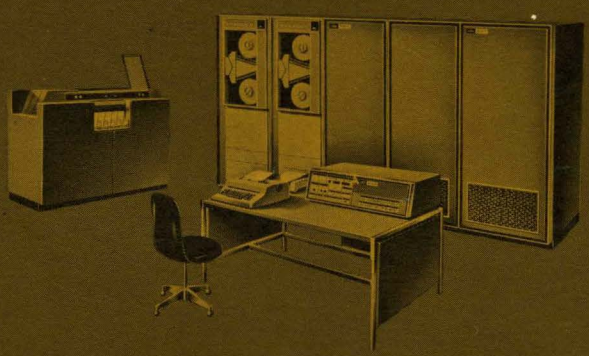
SP-201

An Introduction to the

RAYTHEON 520 PROGRAMMING LIBRARY



RAYTHEON COMPUTER

RAYTHEON

An Introduction to

THE RAYTHEON 520 PROGRAMMING LIBRARY

An Introduction to

THE RAYTHEON 520 PROGRAMMING LIBRARY

COMPLEMENTING THE RAYTHEON 520 . . .

. . . is the Programming Library, an extensive set of programming aids designed to take full advantage of machine characteristics. The basic elements in the Library are integrated so that they form an extremely effective TOTAL OPERATING SYSTEM. Raytheon makes the entire contents of this Programming Library available to every 520 installation, giving to each user means for exploiting his computing and data processing with minimum programming effort.

The design specifications and the development of the Programming Library are the result of extensive analysis by and with specialists in many fields, each of whom was required to impress his bias on the Library and its contents. Every effort was made to reveal and give consideration to the complete needs of any given 520 SYSTEM configuration in many operating environments. Versatility was also a primary goal in developing the 520 SYSTEM Programming Library. To this end, standard structural characteristics were devised to provide similar frameworks for each of the separate programs. They can thus be considered as modules suitable for combining to form many special-purpose programs simply and quickly. In addition to reducing programming time — which was a primary design specification — this approach simplifies library maintenance activities.

These efforts produced, as the basic component of the 520 TOTAL OPERATING SYSTEM, a highly-sophisticated executive/monitor control routine called BOSS. BOSS provides each user means for continuous and effective operation of his computing installation with a low-cost/high-production yield.

The Programming Library also includes FLEXTRAN®, a newly-conceived, completely flexible translator of the compiler-assembler type; 520 FORTRAN, substantially extended version of FORTRAN II; and an IBM 1620 Simulator. The TOTAL OPERATING SYSTEM also includes a comprehensive set of mathematical subroutines, utility and service routines to allow complete assistance in the operation of the central processor and its many peripheral options, and a completely-automatic, multi-level diagnostic program.

FLEXTRAN® is an advanced syntax-directed programming system of the compiler-assembler type. This type of automatic programming system provides users that extensive linguistic capability that is desirable for ease in developing and testing programs. Thus, a program may be written in a language that is essentially the machine's own natural language or a programmer may create a virtually unlimited set of his own macro-instructions, either for temporary or permanent use. More than 140 such macro-instructions have been defined. The following list is a cross-section sample of these instructions: Table Look Up, Push Down Queue, Floating Compare for Less, Store Address Field of A Register, Double Precision Multiply, Compare A Register and Limits, and Transfer and Set Return. Subsets of them are oriented to particular problem-solving environments. For example, there are instruction sets for the scientific and data systems environments. Thus, FLEXTRAN® and the Raytheon 520 System's natural language combine to provide a programmer with ability to create linguistic capability for every use, including the creation of translators of other computers' source language programs.

Raytheon 520 FORTRAN is a substantially extended FORTRAN II. It includes such features as load and list/link routines, interchange of input/output units without re-programming, the facility to accept any standard FORTRAN II statement, the most extensive diagnostics of any FORTRAN II, and the ability to handle mixed-mode arithmetic. It also has a special feature which allows complete modification (augmenting, diminishing or altering the contents) of the FORTRAN Library to suit the particular needs of a given problem without becoming involved in the complex mechanics of the Library. 520 FORTRAN is especially fast in compiling and generates object programs that require less memory and give faster object execution speeds. It is further enhanced by the very desirable characteristics of being easy to learn and to use.

The 1620 Simulator gives to users of that computer the means to expand their computing capabilities to the level of the faster, high-performance 520 System without the high pressure that often accompanies a changeover, a conversion from one computer to another. Use of the 520 System also provides all the added benefits of the 520 Programming Library. The Raytheon 520 simulates the 1620 to give program execution up to three times faster than the 1620 Mod I.

FORTRAN source programs, incidentally, could be recompiled and executed by the Raytheon 520 at a speed as much as fifty times faster than the 1620.

Finally, Raytheon gives extensive programming support and thorough education and training in computer use. Programming support is provided in the field, at the customer's site, by a staff of competent Computer Applications Specialists assigned to offices from coast to coast. Training includes courses on system operation, programming and maintenance. Membership in the Raytheon Users' Group, which was formed by users for their mutual benefit, is open to users of the Raytheon 520 System. Exchange of program materials and sharing of knowledge about how to exploit effectively their computer systems are primary motives of its members.

Taken together, these programs, routines, languages and services are the basic ingredients of a comprehensive Support Schedule that assures a user he can make efficient and effective use of his 520 System.

BOSS

BOSS, the Raytheon 520 Monitor and I/O Control System, provides automatic control of the operation of the 520 System. BOSS achieves its goal of enabling efficient "hands off" operation of a Raytheon 520 SYSTEM by:

1. Supervising the running of separate jobs; a section of BOSS resides in memory at all times to insure continuous operations and to permit linkage control between and among the programs in the Library.
2. Calling system programs needed to translate symbolic programs from assembler or compiler language to machine language.
3. Supervising loading of programs and library subroutines; providing a "Load and Go" mode, which permits a program to be compiled or assembled and then executed.
4. Supervising the printing of diagnostics and memory dumps as requested by the program or as needed when a program is terminated.
5. Interpreting all basic input/output commands to insure legality and allow them to be executed if they reference a non-restricted and free device.
6. Permitting the accumulation of time for any single job from the time the ID card is interpreted until the end of the job.

THE ASSEMBLER SYSTEM

FLEXTRAN® 520 is an automatic programming system of the compiler-assembler type. It provides the programmer with an extensive capability to ease the writing and testing of programs. This is accomplished in a language that is one-for-one with object code or at a macro level combining many machine instructions in frequently used problem-oriented programming functions.

FLEXTRAN® was designed to provide fast assembly, efficient object code (by means of both generator structure and the directive or declarative forms that allow the programmer to exercise significant control over the compiler-assembler itself), and extensive source-language statements to aid program testing. All of these design points are successfully welded into an automatic programming system that effectively copes with the demand for such a system. With appropriate auxiliary storage, FLEXTRAN® can operate in the "load-and-go" mode.

A FLEXTRAN® program is composed of statements in either imperative or declarative form. An imperative statement is one that is equivalent to one or more of the instructions to be executed by the "target" computer (for example, Add Exponent, Test for Zero). A declarative form (or directive) is one that instructs the compiler-assembler to do something (for example, "BLOC" specifies the beginning location of a program block). A declarative form may also convey information to the compiler-assembler (e.g., "QUIT" marks the end of the program to be assembled). Finally, a declarative form may be ignored by the compiler-assembler (e.g., "*" may be used to identify a statement containing comments only).

In addition to recognizing the 64 instructions that are a 520 System's machine language, FLEXTRAN® permits the use of more than 140 macro instructions. Some of these are oriented to particular kinds of environments. For example, one subset of macro instructions is concerned with data systems applications; it includes instructions like:

Convert Channel N
Test Upper and Lower Limits
Output Word to Control Register
Input 24-bit Data Word from Channel N
Data Quality Check
Convert to Engineering Units

Another set of macro-instructions has been designed specifically to implement the coding of programs for an analog-digital hybrid computer simulation. These include such functions as:

Analog Computer Mode Select
Read Analog Channel and Scale
Scale and Write Analog Channel
Skip on Analog Channel Greater
Skip on Analog Channel Less

Set Pot
Read Pot Setting
Read Analog Element

A specially selected group of the macro instructions are further distinguished as the basic set of general-purpose instructions. So distinguished, they facilitate the sharing of programs written by any user of a Raytheon 520 System.

The programmer has complete control over the format of any segment of the assembled object program. With this control the programmer can make every decision concerning space-time tradeoffs. All necessary housekeeping functions are provided by the compiler-assembler.

The compiler-assembler puts out both absolute and relocatable object code. The loader part of the Raytheon 520 Operating System can load separately assembled programs, irrespective of the sequence in which the segments were assembled; the loader provides all necessary linkage.

All input and output are handled through the Input/Output Control System (IOCS). Input and output are "file oriented;" the concept of a file permits free and easy assignment of peripheral devices. For example, if output to the line printer is file oriented, that output may be reassigned to another peripheral device if the printer cannot be used. Thus, useful work may continue to be done even when part of a system is inoperable.

Each program in the Raytheon 520 Programming Library has a structure that may be listed in either of two groups — Systems or Scientific Applications. Every program — including FLEXTRAN® itself — has the basic characteristics of a subroutine. Thus, the elements of the Library can be regarded as two types of modules, and as determined by the type of problem, may be used as standard building blocks out of which an unlimited variety of special-purpose programs can be constructed with ease and simplicity. Not only is programming effort eased by this artifice, Library maintenance is facilitated too.

RAYTHEON 520 FORTRAN

For the Raytheon 520, FORTRAN II has been extended to give 520 users the most efficient, useful, and convenient programming system available for this class of computers. Some of the outstanding advantages are:

Fast compilation combined with efficient object code

Simplified syntax to save programming time and reduce errors

First-pass source deck error checking and diagnostic printout

Relocatable compiled programs

I/O devices named at object load time to allow interchange without reprogramming

Overlapped computing and block transfer I/O operations

Load-link-list facilities to allow merging of separately compiled programs and simplify debugging

Editor routine to permit easy modification and expansion of FORTRAN Library to fit installation needs

Compatibility with IBM FORTRAN II

These are some of the general features that contribute to making Raytheon 520 FORTRAN the procedure-oriented system with the most benefits for the user. Another set of major advantages results from the elimination of most of the redundancies and logically unnecessary restrictions of earlier versions of FORTRAN. In addition, the meanings of some of the other features have been generalized so that they apply to more situations. These changes have made 520 FORTRAN easier to learn, easier to use without making minor errors, and much more powerful than earlier versions. At the same time, these changes have led to development of a processor capable of both fast compilation and the production of highly efficient object code. The result, from the user's viewpoint, is a welcome saving through strict conservation of time and of memory space during both program generation and actual production.

Raytheon 520 FORTRAN is compatible with 7090 FORTRAN II and compatible with all other FORTRAN II versions given very minor changes. For example, statements may involve components not present in the 520 System. Such a problem can be handled automatically, however, because the Raytheon compiler provides the facility of reassigning input/output devices at object program time.

To illustrate how the restrictions of earlier FORTRAN versions have been overcome for the Raytheon 520, consider the greater freedom in program preparation indicated by these characteristics:

Statement labels may be alphanumeric

Variable names may be up to 40 characters, instead of only six

Standard representation of array elements is allowed in equivalence statements

The replacement operator has been completely generalized

Mixed mode expressions are permitted

Levels of exponentiation are unlimited

Alphanumeric fields may be stored as arrays

A subscript may be any arithmetic expression and may take on negative values

Subscripts may be subscripted

The number of allowable subscripts is unlimited

Assigned GO TO statements have been expanded

In a DO statement, values following the equal sign may be any constant, variable, or arithmetic expression

Reverse DO loops are allowed

A DO loop may be entered from outside its range

Arbitrary rules for writing statements have been removed

Mixed mode input/output statements are allowed

The best way of assessing the advantages of these characteristics, of course, is to compare them to FORTRAN II procedures in actual practice. The following example is a brief illustration of the benefits to be realized. The first set of 17 statements was written under the restrictions imposed by FORTRAN II; it produces 46 instructions for the Raytheon 520. By rewriting these statements to take advantage of the flexibility of Raytheon 520 FORTRAN, we have six statements yielding 29 instructions. Note the twofold saving that results: a 65 per cent reduction in source language and a 37 per cent reduction in object code.

FORTRAN II		RAYTHEON 520 FORTRAN
	I=ISTAR	
	J=I + 1	
	K=J + 1	K=1 + (J=1 + (I=ISTAR))
110	A=I	
	B=J	
	AB=A * B	
	X=AB/RATIO	110 X=(AB=I*J)/RATIO
115	C=K	
	RP=X/C	
	RPP=RP + .501	
	L=RPP	115 IF((L=(RP=X/K)+.501)−MAX) 400,400,600
	IF (L-MAX) 400, 400, 600	
400	IF (L-K) 600, 401, 401	400 IF(L-K) 600,401,401
401	D=L	401 IF(RP-L) 402,410,403
	IF (RP-D) 402, 410, 403	
403	DP=D + .1	403 IF(L+.1−RP) 600,410,410
	IF (DP-RP) 600, 410, 410	

THE 1620 SIMULATOR

For the convenience of Raytheon 520 users who are making a transition from an IBM 1620, a simulator has been prepared to allow direct execution of the 1620 programs by the 520.

Some general features of the 1620 Simulator are:

All basic and optional 1620 instructions are simulated

Machine-language 1620 programs can be run without reprogramming

Simulated 1620 programs are executed on the 520 as much as three times faster

IBM 1402 Card Reader/Punch units and 1311 Disk Packs from the 1620 System can be used with the 520

Memory requirements for simulation can be seen from the following table:

IBM 1620	RAYTHEON 520
20,000 digits	8,192 words core, 512 BIAX
40,000 digits	16,384 words core, 512 BIAX
60,000 digits	20,480 words core, 512 BIAX

The 1620 Simulator includes provisions for paper tape input and output, typewriter, card reader/punch, and line printers.

The speed advantage of the Raytheon 520 varies according to the type of 1620 program and the field lengths of operands within the program. As an average, a speed ratio of three-to-one in favor of the 520 can be expected. Input/output simulation takes place at the operating rates of Raytheon peripheral devices.

After the Simulator is loaded, a set-up routine produces a request for input describing the equipment

configuration to be simulated. The same routine arranges for the Simulator to operate according to the 1620 specified by the user.

A console routine is provided to give the user control of the simulation process. It supplies a means for the operator to enter data and examine the status of the simulated 1620 console. It also provides indications of error conditions, halts, and peripheral equipment status.

The user is informed of the condition of the simulated console switches and indicators by typewritten messages. Provision has been made for interruption of the automatic operation and inquiries or control by the operator.

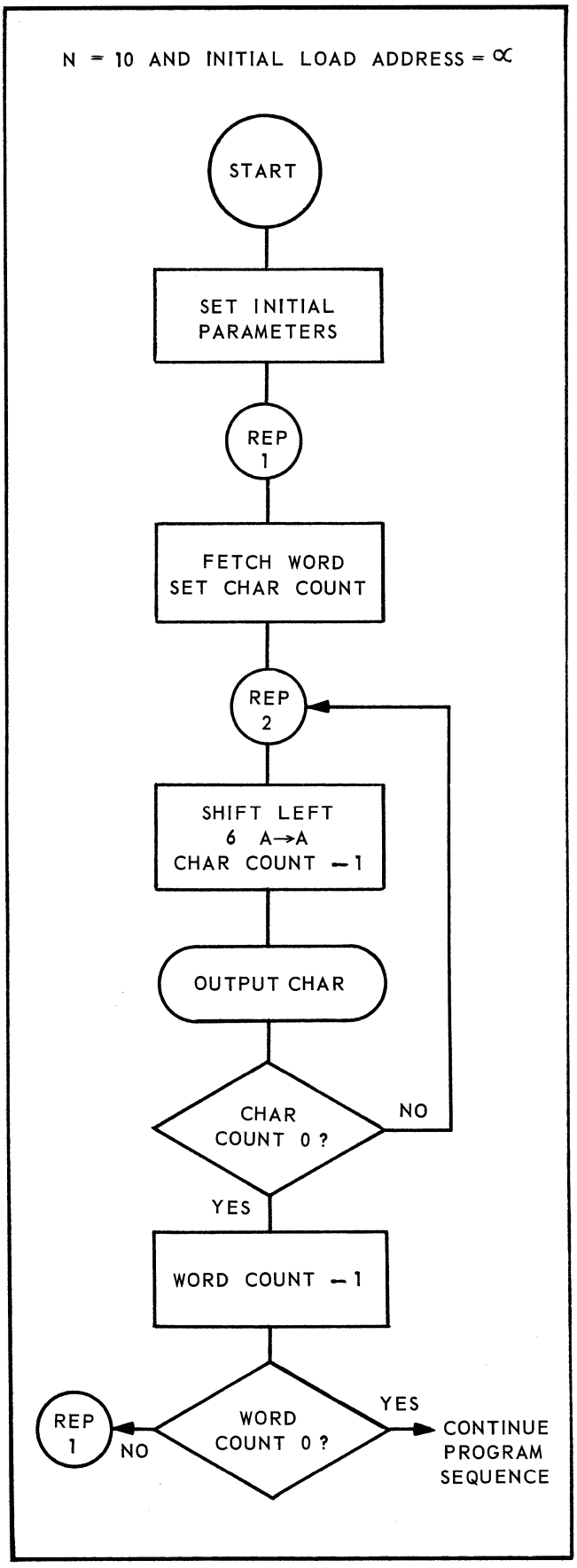
Those programs written in 1620 FORTRAN language can be compiled by the Raytheon 520 FORTRAN processor; the resultant machine code can be executed on the Raytheon 520 without use of the 1620 Simulator at speeds as much as 50 times greater than the speed of the 1620 Mod I.

WRITING IN 520 BASIC LANGUAGE

The Raytheon 520 basic language instruction set provides extensive flexibility for handling of data, especially where individual bits, or fields, of a character, or word, are moved or tested. The instructions are especially well-suited for subroutines where loops are made and for data I/O routines. The routines can easily be made efficient in length and speed. The following is an illustration of the ease with which a data output of N words, each with four 6-bit characters, is made at machine language level (of course, the code is written in basic FLEXTRAN® format):

8-word instruction sequence:

```
        LDI     P,L     set initial load address -> L
                        and loc counter + 1
        LRC     10      set word count -> 10
        ADR     α       initial load address
REP 1   LDI     L,A     load word
        CLD     4       set CHAR count -> 4
REP 2   SLS     A,A     shift left six A -> A
        CDL     D,D     CHAR count - 1
        DTR     2,A     output least significant 6 bits
        NOP
        TNZ     D,F     CHAR count 0?
        BTR     2       no -> REP 2
        CDL     N,N     yes -> word count - 1
        TNZ     N,X     word count 0?
        BTR     5       no -> REP 1
        NOP
        . . .           yes -> continue program
                        sequence
```



Utility Routines . . .

These include routines that provide such assistance as program loads, typewriter I/O, formatted memory dumps, data conversion, card I/O, system editing, magnetic tape I/O, paper tape I/O, paper tape editing, conversion of paper tape to cards and cards to magnetic tape.

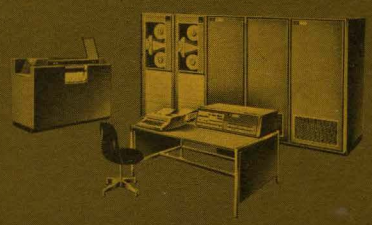
Service Routines . . .

Diagnostic routines are provided for both hardware and software. An automatic, monitor-controlled hardware diagnostic with modular construction allows the user to select from the complete set of modules only those that test his hardware. For example, one set of modules can test a minimal system consisting of CPU, Typewriter, Paper Tape Reader and Punch and Line Printer. Modular construction permits easy addition or deletion of test routines as system configurations change, thus enabling the user to achieve economies in storage and maintenance. It provides automatic fault detection with special examination features for each type of failure. The collection of software diagnostics is a comprehensive one. FLEXTRAN® and FORTRAN, of course, are rich in natural linguistic capability to provide dynamic information about program status. In addition, cross-reference listings, for example, aid in developing and testing programs. Also available are selective and complete trace routines, selective dumps, etc.

Mathematical Subroutines . . .

Raytheon provides a comprehensive set of high-speed mathematical subroutines. Routines exist that will accomplish the following tasks:

Polar to Rectangular Coordinate Conversion
Binary to BCD (four 6-bit words)
BCD to Binary (four 6-bit words)
Square Root
Sin
Cos
Arctan
Exponent
Log
Floating Point to Fixed Point
Fixed Point to Floating Point
Angular Unit Measurement Conversion
Fourth Order Euler Integration
Fourth Order Runge-Kutta Integration
Gray to Binary (12-bit)



SALES OFFICES

Alabama

RAYTHEON COMPUTER
Holiday Office Center, Suite 47
Huntsville, Alabama
Phone (205) 881-3245
TWX 205 533-1112

California

RAYTHEON COMPUTER
2700 South Fairview Street
Santa Ana, California 92704
Phone (714) 546-7460
TWX 714 546-0444

Massachusetts

RAYTHEON COMPANY
Bedford Laboratory
P. O. Box 508
Bedford, Mass. 01730
Phone: (617) 274-7100, Ext. 643 and 644

Texas

RAYTHEON COMPUTER
204 East Main
Arlington, Texas
Phone (817) CR 5-5361
TWX 817 274-3917

Washington, D. C.
Eastern Regional Office

RAYTHEON COMPUTER
4217 Wheeler Avenue
Alexandria, Va. 22334
Phone (703) 836-7616
TWX 703 931-4247

Represented By
PIVAN ENGINEERING COMPANY
3535 Peterson Ave. Chicago 45, Ill.
KEystone 9-4838

RAYTHEON

RAYTHEON COMPUTER

2700 SOUTH FAIRVIEW ST., SANTA ANA, CALIFORNIA 92704

LITHO U.S.A. 4-65