Systems Engineering Laboratories
810B Real Time Executive
Programming System

Introduction

At Systems Engineering Laboratories, Incorporated, Software is computer intelligence. Therefore, the Real Time Executive (RTX) software system has been developed to supply SEL 810B Computer users with a modular monitor that enables maximum utilization of the nanosecond processing speeds of the SEL 810B Computer for foreground application programs, while additionally affording background batch processing capabilities with minimum system overhead.

The RTX software system controls, sequences, schedules and provides for the most efficient allocation of facilities for operation of the real-time system. Included in the RTX software system is a Task Scheduler which enables programs to be executed in the foreground on the basis of a priority interrupt structure which is controlled by both hardware and software. Software scheduling includes time of day, time interval and program sequencing thereby eliminating the need for the programmer to plan for program concurrence.

Both core-memory-only and expanded core-disc systems are supported by RTX. Foreground programs may be both resident and non-resident in the core-disc system. Non-resident programs are stored on the disc in either a relocatable or an absolute format and, as they become active, are assigned core memory on a dynamic basis.

Optional background operation permits assembling, compiling, loading, executing and debugging of user-created FORTRAN IV and/or Macro Assembler programs. Inclusion of this option is specified at System Generation (SYSGEN) time. User-supplied system parameters are specified to the System Generator program which then customizes RTX to the unique demands of the installation.

317-000001-000

Printed in United States of America

Real Time Executive (RTX) Features

- SYSGEN builds a monitor personalized around your hardware configuration and your software requirements. For instance, the I/O handlers can be specified as either resident or non-resident at the time of SYSGEN, thereby allowing the user a trade-off between core memory utilization and system response time.
- OPEN-ENDED DESIGN in such areas as IOCS, Monitor Services, and Operator Communicator are designed so that each user can add his own functions with a minimum of effort. Documentation is provided to assist the user in performing this task.
- MINIMUM RELOCATABILITY OVERHEAD due to design of the load module format.
- REENTRANT MONITOR SERVICES, I/O HANDLERS AND I/O EDITORS provide for overlapping of I/O operations as well as the ability to achieve maximum overlap of I/O and computing.
- CORRECTIVE ACTION PROVISION (CAP) provides for secondary peripheral unit substitution when the primary peripheral fails such that the I/O operation may be completed.
- MODULARITY enables selection of only those monitor features desired by the user.
- TASKS CAN BE INTIATED by hardware interrupts, elapsed time, operator request, and by other tasks.
- TASKS CAN BE SCHEDULED ON THE BASIS of time, immediate (hardware), round robin (time-slicing), and by operator request.
- USER CAN ADD, DELETE AND MODIFY TASKS ON LINE.
- FORMATTED DEVICE PROCESSING provides a FORTRAN-like edit and format capability available through machine language which alleviates the user from formatting and logging data. This feature is optional, re-entrant (one copy, shared by all tasks), and can be specified as resident or non-resident at SYSGEN time.
- NO PRACTICAL LIMIT ON THE NUMBER OF PROGRAMS results in up to 256 uniquely defined tasks recognized by RTX, each task being an integral number of MAP's (512 words)
- DYNAMIC MEMORY ALLOCATION
- CONNECT/DISCONNECT TASKS TO DIFFERENT SOFTWARE INTERRUPT LEVELS
- ON-LINE, CORE-DISC MODIFICATION
- OVERLAY CONSTRUCTION CAPABILITIES
- BACKGROUND/FOREGROUND OPERATION
- FORTRAN IV (USASI Standard) allows in-line assembly language coding as well as other convenience features.
- DEVICE INDEPENDENT PROCESSORS (FORTRAN IV, MACRO ASSEMBLER, ETC.) accept input from any compatible device and perform output to any compatible device. Device assignment may take place through an operator request, via a job control specification, or dynamically at run time.
- TRACING FEATURES WITH BACKGROUND DEBUGGING

Principles of Operation

At Systems Engineering Laboratories, Operation means productive action. This concept is put to work for you when using RTX. Envision the system in operation, scanning the task control blocks and testing the activation state of several tasks. The operator loads the background control command input medium with a job stack and keys in a command on the ASR-33 which activates the background operation. When the RTX control program has been notified that the background task has been activated, it causes the Background Operating System (BOS) Executive to be loaded into the background area. When all higher priority tasks have been cared for, control is then transferred to the BOS Executive and the background job is given control of the CPU and proceeds to process the first job.

At this time an interrupt requests that a non-resident foreground task be activated. At the next timer interrupt, the scheduler determines that this is now the highest priority active task and that space is available in dynamic core for allocation.

This task is then transferred from disc storage by the resident disc handler while the background program is allowed to continue to execute computer instructions, simultaneous to the actual transfer on a memory cycle stealing basis. The necessary relocation of the program is performed by the Link Loader and, when the relocation is completed, the task is placed into execution.

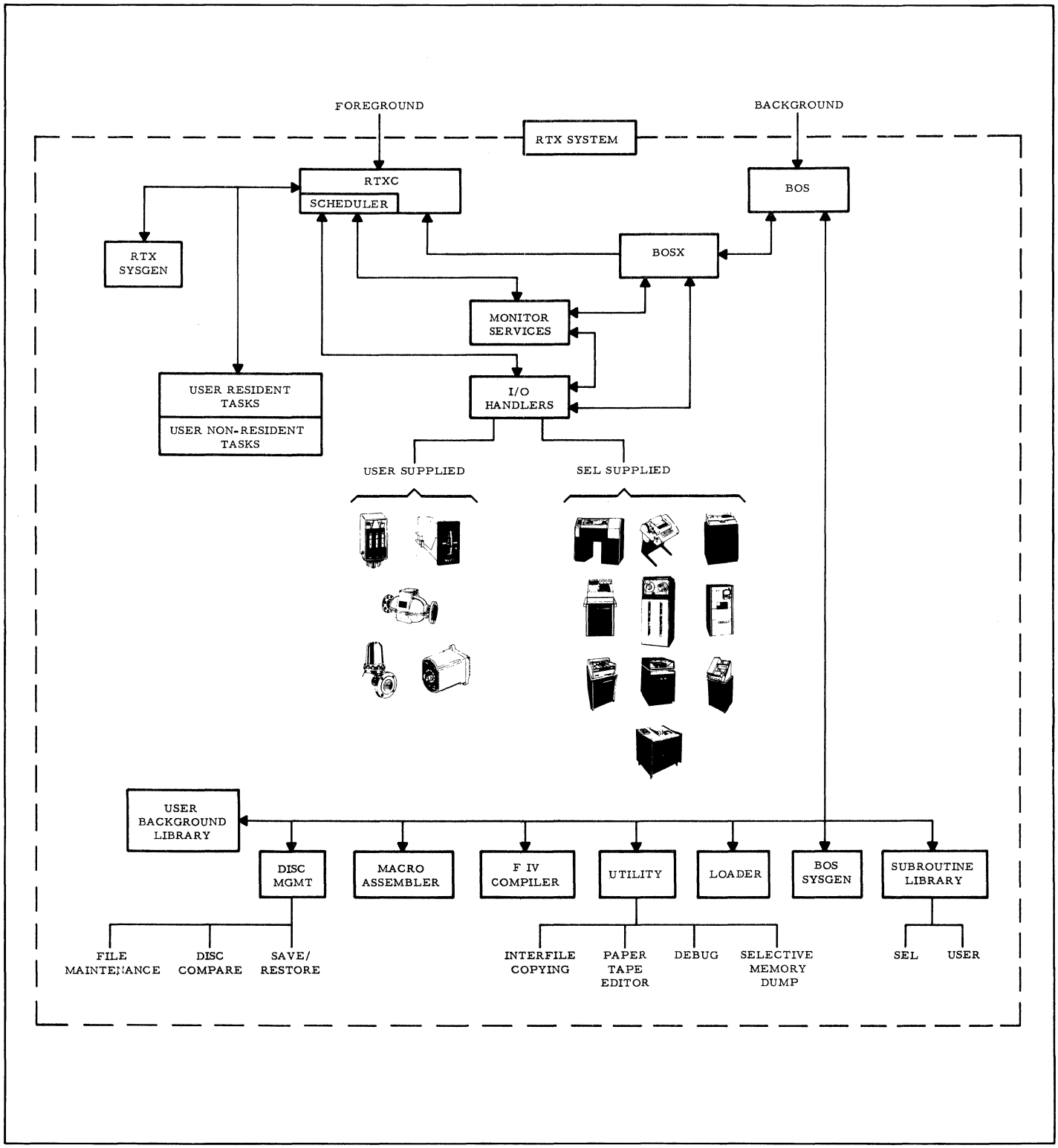
After executing for a period of time, the task requests input from the real-time process. It is placed into a locked state, the I/O is scheduled for operation, and the background operation proceeds. When the real-time process I/O terminates, the foreground task is unlocked and placed into execution. Upon operating for a time increment, the elapsed time shows that a lower priority task has bcome activated. It is not processed beyond activation because it is of lower task priority. Later, another external interrupt causes execution of a directly connected task. Since the task is higher in priority than any of the RTX tasks, it runs immediately. The interrupted non-resident foreground task continues when the directly connected task terminates. When the foreground task terminates, its memory is released if so initiated, and becomes part of the available dynamic core pool. At this time the lower priority task, previously queued, is placed into execution. The foreground task and the background job continue to share Central Processing Unit time. Upon termination of the foreground task, the background job stack is processed until an end-of-job command is encountered. At this point, the system reverts to the task control block scanning loop.

Design Goals Exceeded!

At System Engineering Laboratories, RTX development means customer features. During this development the RTX software has met and exceeded all the following basic design goals:

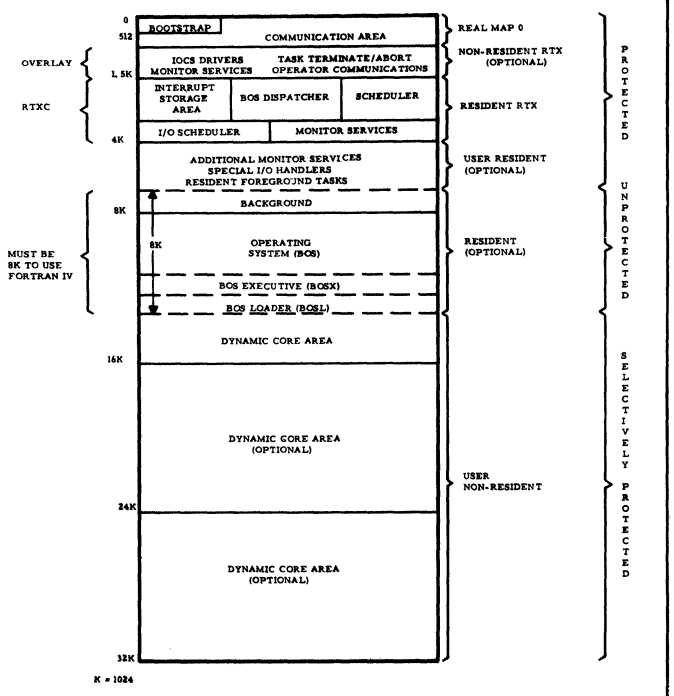
- Quick response to interrupts — 3-10 microseconds for directly connected interrupts.
- High degree of multi-programming — place maximum work into memory.
- Maximum response time to resident monitor connected foreground programs of 150 cycles.
- Modular system components — ease of interfacing.
- Extensive set of operator control commands.
- High background capability — extensive batch processing facilities.
- Rich job control language.
- Customer "Hands On" control of the system — gives users access to the decision-making capability of the RTX where appropriate.
- Surpass overhead limitations of TSX and MPX.

RTX Block Diagram



RTX Multiprogramming Executive Control Routine (RTXC)

Memory Map



The RTXC Routine is the central module controlling and scheduling all system activities other than those tasks which are tied directly to interrupt levels (these tasks operate at a priority level above the Executive Control Routine).

The components of RTXC include:

- Task Scheduler
- Dynamic Core Allocator
- Loading Services
- Operator Communicator
- I/O Handlers (IOCS)
- Monitor Services

Because response to these demands for the above services must be very quick, most of these routines are link-edited into a core image and are contained in the system bootstrap file. Certain RTXC functions run as tasks (e.g., the input and output scheduler) and the user has the choice of whether these tasks are to be core resident or not.

Task Scheduler

Task scheduling is one of the most important features of the RTX programming system. Every 60 Hz an interrupt is generated by the real-time clock at a high priority hardware interrupt level. Here, the time-of-day clock and a time-slice counter (set by the SYSGEN process) are incremented. When the time-slice count reaches zero, an interrupt is generated at a lower level where task scheduling will occur.

These software maintained task priorities (priority levels) may be viewed as an extension of the hardware priority interrupt structure. Monitor services exist to permit great flexibility in the software alteration of the present status of any specific task level including:

- Activate/Deactivate
- Reassign
- Terminate
- Abort
- Lock/Unlock
- Protect/Unprotect

While scheduling occurs normally on a time dependent basis, certain action within the monitor will force rescheduling providing that no other higher levels are active. These include:

- Request for I/O transfer, wait until complete
- Disc transfer complete interrupt

Scheduling overhead varies from system to system primarily on the size and number of non-resident foreground programs and on the core memory size. Design criteria dictate a fast response to program task activation requests so that all scheduling tables are kept in core memory. A system with 50-75 non-resident programs, averaging approximately three MAPs in length, (with a 16K system) will not tax system overhead above 5-10 percent.

Dynamic Core Allocator

The primary function of the RTX System is to perform real-time data processing. To accomplish this, the computer memory is divided into foreground and background.

RTX's sole purpose is to control the execution of varied tasks in a dynamic real-time environment. Tasks can thus be divided into the following categories:

Resident Monitor (Systems Engineering Laboratories supplied)

Non-Resident Monitor (Systems Engineering Laboratories supplied)

Resident Foreground Tasks (User supplied)

Non-Resident Foreground Tasks (User supplied)

Background Task (System Engineering Laboratories supplied, and/or User written)

The core memory allocation map (see chart) demonstrates that core memory is allocated in one of three distinctpriority dependent manners.

Highest — permanently resident tasks

Next — tasks to which memory is allocated from the dynamic core area

Lowest — jobs operating in the background

Once allocated, the Background Operating System (BOS) cannot alter its memory dynamically. Memory is allocated to tasks in the Dynamic Core Area on the basis of an integral number of MAPs (512 words). Maximum core that can be allocated at one time is 16K.

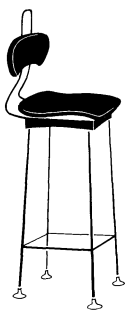
Loading Services

RTX Bootstrap (BTLD) —

This resident module is capable of reloading the resident core image RTX modules as well as the non-resident RTX initialization module. This can be called as a privileged monitor service or by bootstrapping the disc.

Link Loader (LKLD) —

A link-edited module has the capability of being loaded anywhere into a previously allocated area in Dynamic Core via the resident Link Loader.



Certain non-resident foreground tasks cannot afford the overhead of such relocation. These tasks exist in core image form on the disc and are loaded by LKLD as well. A bit in the task identification block informs the Link Loader as to whether the task is relative or absolute. Construction of overlays can easily be accomplished by calls to LKLD.

Background Loader (BOSL) —

This loader accepts the object output from the Macro Assembler or the FORTRAN IV Compiler and will link-edit a module for execution in the background. Procedures for loading the appropriate library programs are also provided for in the BOSL. A non-resident SYSGEN task will edit a link-edited module into the proper format for the selective inclusion in the foreground task library.

Operator Communicator

An extensive set of operator control functions are provided. Since these functions will be in demand only when operator intervention is required or desired, they will not be part of the resident monitor; rather, the control functions will be brought into core dynamically when needed. Essentially, these programs run as non-resident foreground tasks.

The functions provided include:

- Display and alter core
- Update current date and time
- Disc dump, load and compare
- Change device status (remove and insert from/into service)
- Display and alter scheduling tables:
  - Initiate a task
  - Abort a task
  - Hold a task
  - Alter a task's priority

I/O Control System (IOCS)

Re-entrant I/O handlers are provided for standard Systems Engineering Laboratories' peripherals. These are modular subroutines that need not be included in a system unless the particular device is present. An important consideration in a high speed real-time data acquisition system is dedicated I/O interrupts for each device to maximize I/O through put.

Any task operating under the RTXC (and not directly connected to interrupts) has the I/O capability of operating with:

- Symbolic files
- Formatted I/O
- Direct I/O (unformatted)

Monitor Services

A software interrupt triggering capability exists within the SEL 810B hardware by the execution of a special non-privileged augmented instruction. This instruction generates a priority interrupt to occur on a special level for interrupt service "N" where "N" is specified in the next word of the instruction.

A uniform monitor service sequence is thus provided as well as a common vehicle for communication between either protected or unprotected tasks and RTX.

Monitor services include the following functions:

- Secure and release a stack of dynamic storage
- Number system conversion
- Schedule I/O processors to input/output data
- Redefine device characteristics
- Assign/deassign/reassign devices to files
- Allocate and deallocate an integral number of memory MAPs to a task.
- Task abort
- Test availability status of a given service
- Status about a given peripheral device
- Status about a given file
- Current time of day
- Current date

System Generation

A comprehensive SYSGEN program is provided which allows a user to personalize a system to his particular needs.

The user supplies a set of parameters to the system generation program permitting the user to describe his computer hardware configuration and define his software operating environment. These parameters include:

- Peripheral devices available
- Interrupt assignments
- Initial task scheduling priorities
- System input file
- Memory size
- File/device equivalence
- Dynamic core allocation
- Disc file extents
- Background capability
  - a. Start Location
  - b. Priority Level
  - c. Desired Processors { Systems Engineering Laboratories supplied / User supplied }

Background Operating System (BOS) Executive (BOSX)

The Background Operating System (BOS) consists of a resident dispatcher (part of the resident RTXC) and non-resident processors which are activated by either operator requests to process user job and/or by job control within the job stack.

The capabilities of BOS include:

- Searching and loading specified programs from magnetic tape
- Writing and editing of files
- Initiating and controlling job processing
- Batching of assemblies and compilations
- Disc file maintenance

The SEL 810B BOS can run as a stand-alone operating system on the SEL 810B Computer or as the background in the RTX programming system.

BOS is completely device independent allowing run time selection of I/O devices and does support all of Systems Engineering Laboratories' standard peripherals. A SYSGEN program for the BOS is also supplied by Systems Engineering Laboratories.

BOS Processors

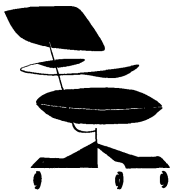
Compiler

The SEL 810B FORTRAN IV Compiler meets or exceeds all USASI[1] standards. Convenience features included in a one pass operation are:

- No reserved identifiers
- Hollerith and logical constants
- Optional in-line assembly coding
- A-spec conversion
- Variable formatting — lists specified at run time
- Item and area tracing
- Chaining

Macro Assembler

The SEL 810B Macro Assembler translates programs written in symbolic code into meaningful machine language code. Designed to operate in the minimum computer hardware configuration, the



1 As suggested by USA Standards Institute Section Committee X3, Computers and Information Processing.

assembler makes use of all available memory and all standard peripheral devices. Operating in a two pass mode, some of the features include:

13 special pseudo-ops to facilitate coding

Subroutine call and naming capability compatible for loading with FORTRAN IV programs.

I/O free-selection of Symbolic Input, Listing Output, and Binary Output is made at load time.

Selection of alternative mass storage device for second pass

Relocatable or absolute output

Addresses may be expressed in symbols, octal, or decimal formats

Address arithmetic may be performed with any combination of the above formats

Literal addresses and constants

Data pseudo-op allows entering octal, decimal, fixed point, floating point, alphanumeric, and symbolic address data

The assembler is further enhanced by its macro capability. In-line coding is generated according to the respective prototype and parameter list assigned to a given macro call name. The elements within the Macro Assembler may be either internal (set symbols), fixed, or user-supplied parameters. Nested macros to any depth are permitted as well as character string concatenation capabilities.
The SEL 810B RTX is written in the Macro Assembler language.

RTX Hardware Requirements

At Systems Engineering Laboratories, hardware requirements are minimized while the RTX features are optimized.

Minimum hardware requirements for a basic system include:

SEL 810B Computer with 8K of memory

Variable Base Register (VBR)

Why? To utilize multiprogramming and the dynamic relocation of programs.

Index Register

Why? To facilitate coding, especially the handling of reentrant monitor services.

RTX Interrupt Module

Why? To allow the monitor to effectively disable all interrupts of a lower priority with one instruction. In addition, a special monitor call instruction is implemented which will permit a uniform calling sequence to monitor service modules. This also facilitates communication between both protected and unprotected tasks and RTX.

Second Group of Priority Interrupt Modules

Why? To provide dedicated I/O interrupts for all standard Systems Engineering Laboratories' devices as well as program protect and instruction trap.

60 Hz Real Time Clock

Why? To provide the softwear scheduler with a time reference for program sequencing.

Program Protect and Instruction Trap

Why? To guarantee system integrity in a dynamic environment.

Binary I/O Device

Why? For SYSGEN, either the card reader or the high speed paper tape reader.

Minimum hardware requirements for an advanced background / foreground disc system which will allow non-resident task scheduling, and dynamic core allocation include:

SEL 810B Computer with 16K to 32K of memory

Block Transfer Control (BTC) for Disc

Why? To cut down on overhead

Random Access Mass Storage Device

Why? To provide for monitor bootstrap, systems residence, non-resident foreground programs, file storage, etc., must be either:

Fixed Head (head per track) Disc

Movable Head Disc

Optional items that also receive full RTX support include:

Card Reader

Card Punch

Line Printer

High Speed Paper Tape Reader/Punch

Magnetic Tape with BTC { 7 Track / 9 Track }

Additional Disc File Units

Data Acquisition Systems including the Systems Engineering Laboratories' Accurelay Analog Input Subsystem

Customer Software Support

At Systems Engineering Laboratories, customer application programming means user-task oriented, over-all software responsibility. Our Programming Department is organized and staffed to serve the best needs of our computer customers and users.

Systems Engineering Laboratories maintains two programming sections whose services are available to its customers.

The *Software Product Control Section* has the responsibility for all standard software products delivered by Systems Engineering Laboratories. In addition to insuring the integrity of the software delivered to customers, it is their responsibility to evaluate software modifications recommended by customers and to implement those which will increase or enhance the existing software product line.

Within this section is the *Software Support Group* which maintains constant contact with customers in order to supply technical assistance in the software area.

The *Applications Software Section* has the responsibility for servicing all other customer needs not covered by the software control section. The staff of the applications section is composed of many professionals, whose backgrounds cover a broad spectrum of mathematical and engineering experience. These programmers are true software analysts, and one or more is assigned to each system project which has special hardware or specific custom software requirements.

Applications programmers work directly with system engineers to insure that a project is delivered in the form of a totally engineered hardware/software system. They follow each project into the field and provide support through final customer acceptance.

The applications programmer is available for direct consultation with a customer and his programmers. Such sessions often prove valuable to all concerned since they usually result in project definition beyond the detail of contract. Typical applications programming jobs are illustrated in the following examples:

- Customizing of standard software packages to meet customer specifications.
- The generation of special system diagnostic programs.
- The design and implementation of high-speed data acquisition and processing systems.
- The design and imulementation of industrial process control systems.
- The programming of custom radar tracking and telemetry systems.

Applications programmers have programmed every peripheral or special interface device delivered by Systems Engineering Laboratories. Many special subroutines, executives, and programming techniques are often available to the applications programmer which are not yet released to the standard software library. A customer can avail himself of this experience by the project consultation method described above or by the direct purchase of an applications programmer's time.

All custom software delivered to a customer is well documented. The company insists that all documentation delivered includes flow charts, listings, detailed functional operating procedures, and theory of operation. Software documentation is published in bound volumes and delivered to the customer as a separate part of the project documentation.

Customer Services

At Systems Engineering Laboratories, Incorporated, Customer Services is Support. This support includes thorough programming training courses, complete software service and maintenance, documentation, benefits of a well organized USER's group, and the overall reassurance of Systems Engineering Laboratories' professional programming assistance.

Programming training courses are offered on a regularly scheduled basis at our Fort Lauderdale, Florida, facility. Professional instructors span a software area from class room computer concepts to actual computer program assembly and debugging practice in Systems Engineering Laboratories' computer center. If desired, a Systems Engineering Laboratories' instructor will be dispatched to conduct programming training at the customer facility. Programming courses are two weeks in length, the last week including "hands on" for solution of class assigned problems. Following the two-week programming class is a three-day RTX SYSGEN class where students get actual experience using RTX.

Complete software documentation is available including Reference manuals (external specifications), Maintenance manuals (including flow diagrams) and Listing and Coding manuals on all Systems Engineering Laboratories' software.

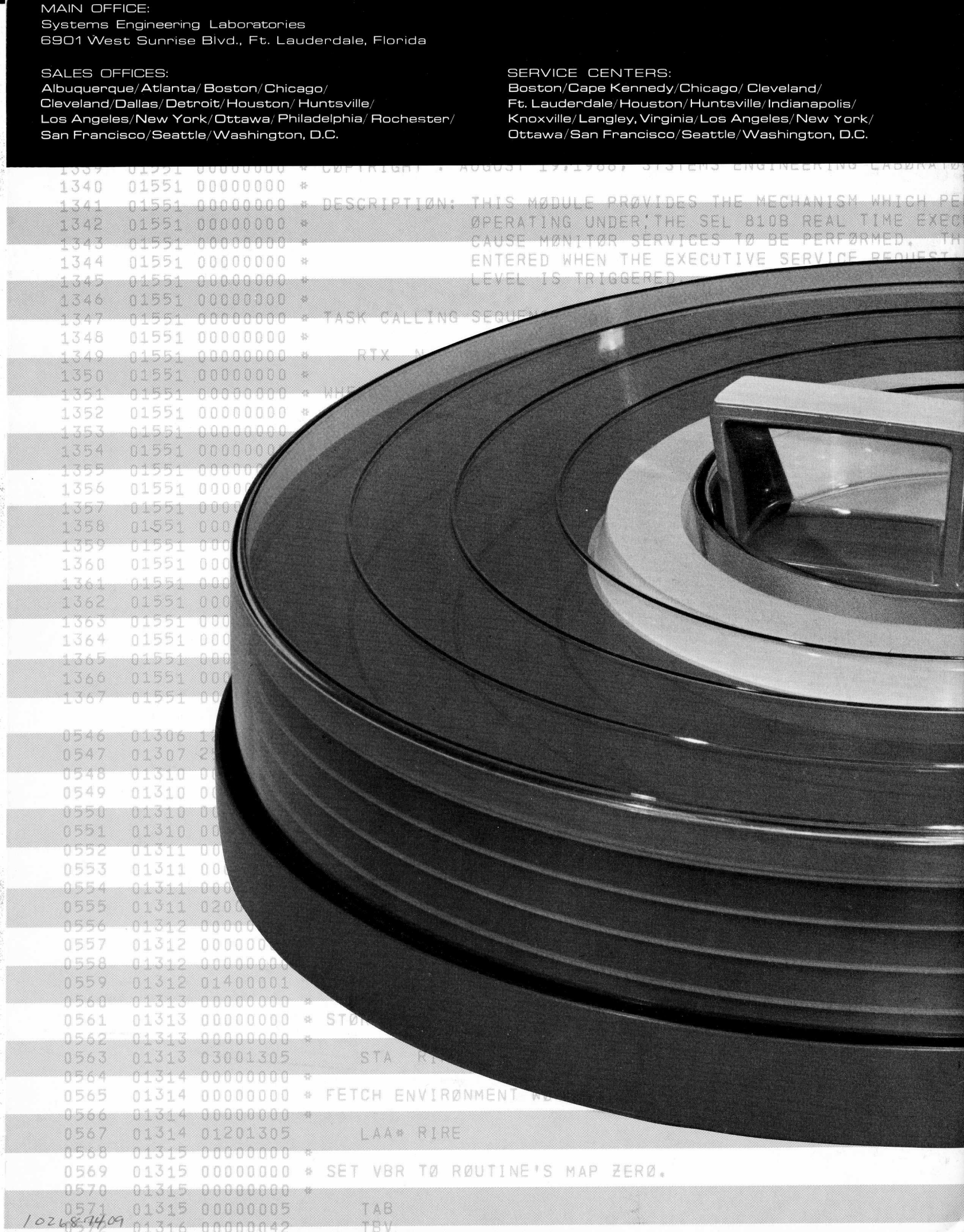
Because of our emphasis on customer satisfaction, Systems Engineering Laboratories has assumed an active roll in assisting and supporting a software USER's group. This organization is made up of interested customers who wish to share the knowledge they have gained in using Systems Engineering Laboratories' equipment and exchanging programs they have developed. As a result, this group is an effective means of sharing software knowledge and determining user needs.


RTX
810B REAL
TIME EXEC

MAIN OFFICE:
Systems Engineering Laboratories
6901 West Sunrise Blvd., Ft. Lauderdale, Florida

SALES OFFICES:
Albuquerque/Atlanta/Boston/Chicago/
Cleveland/Dallas/Detroit/Houston/Huntsville/
Los Angeles/New York/Ottawa/Philadelphia/Rochester/
San Francisco/Seattle/Washington, D.C.

SERVICE CENTERS:
Boston/Cape Kennedy/Chicago/Cleveland/
Ft. Lauderdale/Houston/Huntsville/Indianapolis/
Knoxville/Langley, Virginia/Los Angeles/New York/
Ottawa/San Francisco/Seattle/Washington, D.C.



102687409