Standard

# IC-7000

TIME-SHARING SYSTEM

## SYSTEM SUMMARY

Standard

*Standard Computer Corporation*

FORM 807010-2

# IC-7000

## TIME-SHARING SYSTEM

# SYSTEM SUMMARY

*Revised Edition*
*October 1969*

FORM 807010-2

*First Printing, Sept. 1969, .8M*
*Revised, Oct. 1969, 2M*

*Copies of this and other STANDARD Computer Corporation publications can be obtained through the nearest STANDARD sales office (see back cover).*

*Comments concerning the contents of this publication should be addressed to STANDARD Computer Corporation, Technical Publications, 633 East Young Street, Santa Ana, California 92705.*

 ***Printed in USA***

## TABLE OF CONTENTS



## LIST OF ILLUSTRATIONS

## CORPORATE PROFILES

*STANDARD Computer Corporation* of Los Angeles, California, and Call-A-Computer, Inc. of Minneapolis, Minnesota, have collaborated hardware/software capabilities to provide time-sharing computer users with a total executive computing system.

*STANDARD* has developed scientific and industrial computer systems which are presently in operation across the nation. The engineering, marketing, and management talent at *STANDARD* has now developed a truly unique, problem adaptable, multi-lingual, time-share computer system—the IC-7000. Behind this development are highly specialized veterans formerly with computer industries such as IBM, UNIVAC, General Electric, Texas Instruments, and others.

*STANDARD Computer* maintains total capabilities for the design, fabrication, assembly, testing, and systems programming. Field service engineer training facilities are located in Santa Ana, California, with sales and demonstration offices in most major cities from Los Angeles to New York.

Call-A-Computer is nationally known as the pioneer of time-sharing software for computer systems. Call-A-Computer's experience is unexcelled, and has opened new horizons for time-sharing users by expanding and perfecting software programming to provide the best time-share security file system available today. Call-A-Computer has major offices and facilities located in major cities throughout the nation.

The joint effort by *STANDARD* and Call-A-Computer is dedicated to providing time-sharing computer system users with the fastest, most powerful, state-of-the-art computer hardware/software system currently available. Advanced technology developed by engineering and corporate individuals in both organizations promise an expanding computing system service. With systems located in major cities in the nation, the time-sharer is promised the availability of unlimited access to the system.

# IC-7000 TIME-SHARING SYSTEM

## MACHINE SYSTEM

### Introduction

The IC-7000 Time-Sharing System, a powerful computing system offered by *STANDARD Computer Corporation* and Call-A-Computer, Inc., designed to meet the highly specialized requirements of time-sharing users. *STANDARD* was able to meet these design criteria with its unique inner-computer micro-programming capabilities.

IC-7000 system's architecture is based on state-of-the-art hardware modularity combined with a powerful program library of sophisticated software. The time-share user thus has at his disposal a total system far advanced to that of any competitive predecessor. A typical IC-7000 Time-Sharing System configuration is shown in Figure 1.

Multiple stored program execution exists on many levels. This system hardware capability permits concurrent instruction execution by the Arithmetic and Language Processor (ALP) and Supervisory Processor Unit (SPU) (Figure 2), and the data channels, and provides greater volume throughput and parallel memory accesses.

The IC-7000 time-sharing executive is divided into two areas. One area directs and controls the ALP, and the other runs in the SPU. The ALP part is small and serves two functions: one is the handling of ALP interrupts, and the other is the communications interface with the SPU. The SPU handles all other functions, which amounts to about 90% of the time-sharing executive. SPU executive functions include I/O scheduling, memory allocation, program swapping, and memory compacting.

The user's remote terminal communicates with the system via low speed lines through programmable remote concentrators. The concentrators provide terminal interface with the system, and concentrate all low speed input lines to a single high speed line. Concentrator programmability permits handling of diverse terminal types, speeds, and codes. Each remote concentrator communicates with the local programmable communications multiplexer via a high speed line. System main memory is accessed by the communications multiplexer via one of the IC-7000 system channels.

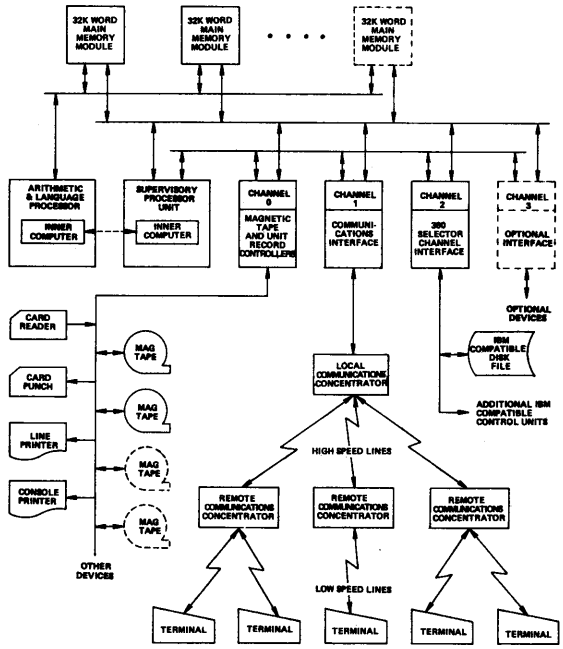


*Figure 1. Typical IC-7000 Time-Sharing System Configuration*

In summary, a user benefits by fast turn-around time due to an extremely high execution speed. This is achieved with the IC-7000 system by virtue of the four-level computer execution overlap, utilizing two specialized processors (ALP and SPU), the local multiplexer, and remote concentrators, all of which provide the user with a total time-share computing system.

### System

The system is designed to do multi-programming for time-sharing. Dual base registers incorporated into the hardware allow a program to be stored anywhere in core. When relocated by the base address register, the program appears to originate at a memory address of zero. One base address register relocates the procedure (or program instructions) and the other address register relocates data. This allows a program to be split in core to facilitate core allocation. One machine instruction saves all system status and registers for a program at the completion of each time slice.

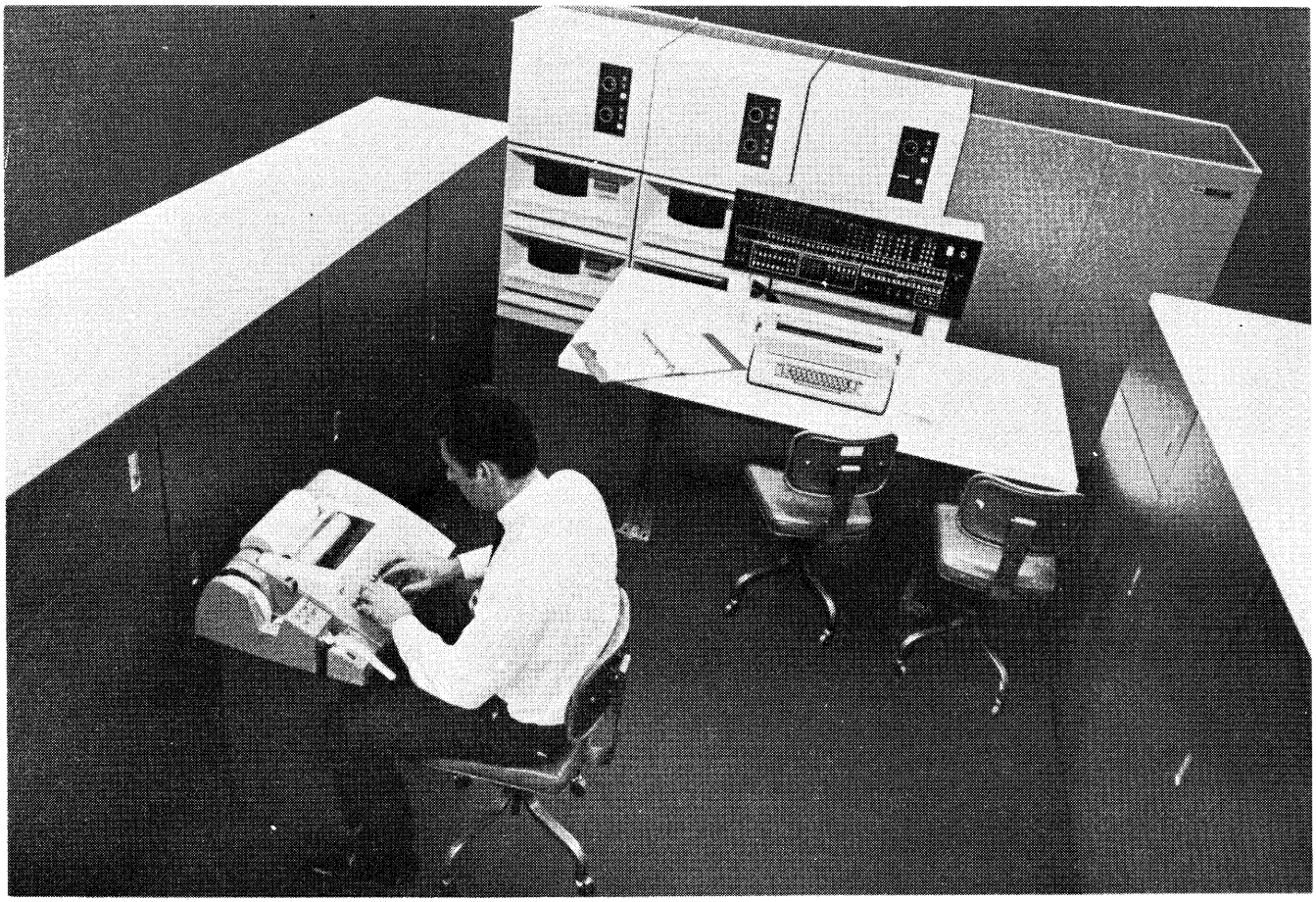

*Figure 2. Arithmetic and Language Processor (left), Desk Console and Disk Storage Drive (center), and Supervisory Processor Unit (right)*

## Arithmetic and Language Processor

The IC-7000 system's computational center is contained in the Arithmetic and Language Processor (ALP). Control functions for data manipulation, and the basic interfaces for main memory, supervisory processor, and the desk console are all housed within the ALP. Monolithic integrated circuitry is utilized throughout the ALP which employes a parallel, look-ahead data transfer method of data access. Fast data transfer is accomplished using the ALP's inner computer and the parallel look-ahead technique.

The ALP shares main memory with the SPU and the data channels. It can test the SPU to find if it is active, and it can also load the contents of its program counter. If the SPU is idle, the ALP can start it. Likewise, if the ALP is halted, the SPU can command it to proceed. In addition, the SPU can trap the ALP making it save necessary information and execute an instruction from a fixed memory location. Otherwise, the two processors run independently. The ALP performs all of its input/output operations by placing entries in memory queues serviced by the SPU. (The time-sharing executive and file system performs all I/O for the general time-sharing user.)

User jobs, run by the ALP, have access to multiple high-speed accumulators, index registers, and working areas in control memory of the inner computer. An expandable number of main memory units, each a 32K word memory (36 bits per word), interface the ALP through memory crossbars. An ALP control console permits dynamic monitoring and snapshot capability of computer facilities. The inner computer has its own control console with similar capability. Control memory (ALP inner computer) is a 2048-word core memory with an access time of 500 nanoseconds.

The arithmetic/logic unit in the ALP performs signed fixed-point, floating-point, and double-precision floating-point arithmetic, logic operations, and shifting, rotating, comparing, and testing operations.

The ALP control unit decodes instructions and controls the data transfers needed to access memory and perform all operations instructed by the program. Memory relocation base and protection registers make it possible to access a 256K - word memory under limitations imposed by the executive. System programs are therefore free from interference by applications programs.

Supervisory Processor Unit

The SPU shares main memory with the Arithmetic and Language Processor. It can trap the ALP, storing all data and execute an instruction from a fixed memory location. It can start the ALP (specifying the starting address) if the ALP is halted. It can wait to be restarted by the ALP. The ALP can obtain the address in the SPU program counter and can test to see if the SPU is in operation. Other than these operations, the two processors run independently. Neither processor can execute programs written for the other. Because of this, instructions in an SPU program are called *directives* to distinguish them from ALP *instructions.*

Four data channels connect the SPU with peripheral-device controllers for tape, disk-file, communications multiplexers, and other peripheral devices. The channels communicate directly with the system main memory. Once initiated by an input/output directive executed by the SPU, the channels operate independently, executing commands stored in main memory. Their operation thus overlaps execution of the SPU program. When operations are completed, the SPU program is interrupted for another command.

The arithmetic/logic unit performs modulo-$2^{36}$ and modulo-$2^{18}$ addition and subtraction (by two's complement addition), AND, OR, Exclusive-OR and negate logic, and shifting, rotating and testing operations. During arithmetic and logic operations, the unit sets indicators if the result is zeros or if no carry propagates from the highest-order bit operated on. These are called previous result indicators (PRI's) and can be tested by conditional-branch directives. The previous-result indicators store their condition until another arithmetic or logic directive is executed.

A control unit decodes directive words. This unit controls the data transfers needed to access memory for operands and to perform the directed operation.

Main Memory

The IC-7000 system main memory is constructed on a modular system of expandable memory capability. The self-contained memory modules are each a 32,768 word capacity with 36-bits per word, plus parity. Main memory units interface system processors and data channels through memory cross-bars to provide parallel system access.

Data Channels

There are three types of channels. One type operates on serial and unit record devices such as tapes, printers, and card readers. A second channel interfaces with the communication multiplexer, allowing for control and data information transfer to or from the SPU and main memory. The third channel interfaces with IBM compatible devices such as disk (2314, 2311) or drum. A fourth channel is optional and can be used as any of the previously described channels.

The channels execute command strings directly from main memory and independently transfer data to or from main memory.

## THE INNER-COMPUTER CONCEPT

The most important quality of a computer language is its ability to control communications between the various functional stations (memory, registers, arithmetic units, and input/output devices). In most present data computer systems, the relationships among the functional stations are frozen by the design and wiring of the system. Such fixed relationships mean that a particular computer can execute only one machine language effectively and cannot utilize the full potential of each functional station.

The IC-7000 system ALP and SPU are each a "computer-within-a-computer". This structure eliminates the fixed link between the various functional stations experienced in conventional computers. In the ALP and SPU all functional stations communicate with each other through the inner com-

puter. This inner computer is programmed to control communication between the various functional stations.

Both the ALP and SPU are divided into two parts: the external functional stations, and the inner computer. The external functional stations consist of the main memory, arithmetic units and registers, and the console. They perform the function of similar devices on conventional computers. The inner computer consists of a scheduler, a control memory, translators, mini-instruction registers and decoders, indicators and display registers, and a mini-engine. The inner computer takes the place of much of the wiring and control logic in a conventional computer.

When a program is run, the inner computer fetches an instruction from the main memory, and performs the necessary indexing and indirect addressing operations by means of a MINIFLOW*-controlled wired-in-sequence. The instruction from main memory is then translated and a MINIFLOW microprogram in the control memory is entered. This directs the inner computer through all the steps necessary to execute a particular instruction. The next instruction is then fetched from main memory and the entire process is repeated until the program is terminated.

## INNER COMPUTER

Each control memory contains 2,048 words of 18 data bits plus parity. The control memory is used to store MINIFLOW microprogram subroutines, data and constants used by the MINIFLOW system or the hardware, and can be used as a buffer area for information transmitted between the ALP and the SPU. Control memory and main memory are independent and fully overlapped.

The scheduler responds to the service requests of the I/O channels, the operator's console, the program of the target machine, and interrupts of other processors in the system. It does so on a fixed priority basis to achieve the least interference between the simultaneous operations of these devices. The scheduler passes control to certain entry points in the wired-in-sequence depending upon the type of request honored.

A wired-in-sequence can be used optionally under MINIFLOW microprogramming control to increase throughput of complex sequences. For example, one of the sequences could fetch the instruction, decode it, perform indexing and indirect address operations, update the instruction counter, fetch the operands required by the instruction, and fetch the first mini-pair instruction. Another sequence might save certain registers by storing them in predetermined control memory locations, when a point of intervention restores them at a later time. The scheduler will pass control to one of several sequences depending upon the information stored in the entry word of the MINIFLOW microprogram subroutine for the particular target machine instruction being executed.

The translators decode the target machine instruction and point to a word in the entry table called the Control and Transfer Vector, which controls the wired-in-sequence and gives the starting address of the routine necessary to complete the MINIFLOW microprogram sequence. The translations also precondition certain control flip-flops in order to pass on specific information about instruction characteristics to the MINIFLOW microprogram.

The mini-operation register receives the 18-bit mini-instructions two at a time as they are read from control memory. The bit configuration is sent to the instruction decoder which sends the appropriate control signals throughout the system to perform the operations.

MINIFLOW microprogram sequence control and shift counting is done by the Mini-Engine registers and adders. Certain instructions make the mini-engine registers available to the programmer. The indicators and display registers are sets of flip-flops which hold hardware and MINIFLOW microprogram status. The registers in this category include the display register, general indicators, and secondary I/O indicators. The bits in the indicators may be individually set and reset by mini-instructions. Many are connected to lamps on the operator's console; some are controlled by hardware; some directly control hardware functions.

### Two Instruction Sets

The ALP and SPU both execute instructions from main memory concurrently. Since the functions they perform for the system are not similar, their respective instruction sets are different. Those codes which the ALP perform are called *instructions*, while those which the SPU perform are called *directives*. Each unit has its own MINIFLOW microprogramming system that directs the execution of the instructions and directives.

**MINIFLOW is a trademark of STANDARD COMPUTER Corporation.*

### Super Instructions

A major innovation of the inner computer concept is the "super-instruction". Instructions and functions can be invented as the need arises. In a conventional computer, such inventions can only be implemented if the computer is rewired. In the IC-7000, super-instructions can be assembled into the MINIFLOW microprogram and tested in a very short time period. Sub-routines that are often used can be taken out of main memory and also assembled into the MINIFLOW system. The operation code in main memory that initiates the execution of the sub-routine in control memory is called a super-instruction. Since control memory execution is much faster than main memory, the creation of super-instructions is effectively the act of problem adapting the hardware to the application. In this case, the IC-7000 is a proven problem-adapted time-sharing machine.

Super-instructions such as Store Slave Status (STSS), and Load Slave Status (LDSS) were implemented to facilitate the program switching demands of a highly responsive time-sharing system. These operations save and restore program registers and indicators. The STSS operation stores the registers and indicators, starting at the addressed memory word and storing the number of words specified in the count field of the instruction. The LDSS reloads the registers and indicators from memory, starting at the addressed memory location and continuing through the number of words specified in the count field.

## I/O DEVICES

I/O devices communicate with the IC-7000 system through the various channels. System I/O devices include the following:

### Desk Console

The operator's console, which is physically located on the Desk Console (Figure 3), has been designed for ease of operation, debugging, and maintenance of the IC-7000 system. The console contains the keys, switches, and lamps necessary for manual and semi-automatic control over the system and the visual checking of information in the computer. Power to the system may be controlled from the console. All memory and register locations can also be displayed. An execute entry function permits execution of console-keyed instructions without disturbing main memory. Address stop control provides several optional stop modes. Console functions are readily mastered by a trained operator. A console printer keyboard is physically mounted in the console which allows the system to communicate with the operator at a print-out speed of 15 characters per second.

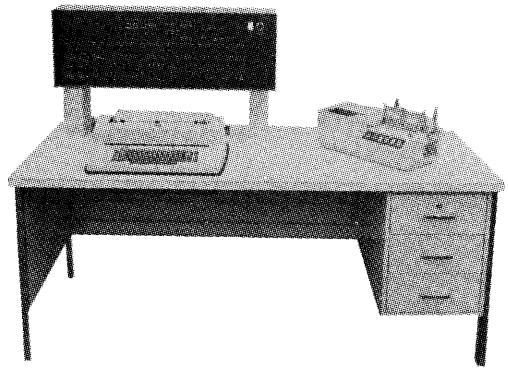

*Figure 3. Desk Console*

### Disk Storage Drive

The Disk Storage Drive (Figure 4), is a large capacity, high speed, direct access storage device that is ideally suited for real-time data processing and teleprocessing applications.

Direct access storage enables record retrieval without sequential searching through a file. Thus, when only part of the records of a given run are active, a direct access storage device provides significantly better throughput than a device that handles each record sequentially. Direct access storage also enables random inquiries during record processing. If desired, a direct access device can also process records sequentially. Therefore, with a direct access storage device, records can be stored sequentially or randomly.

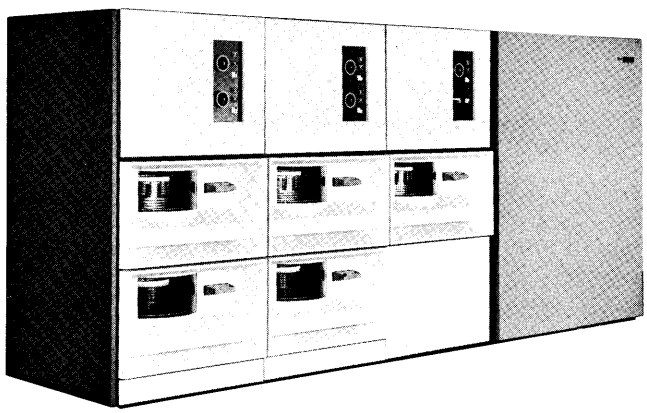

*Figure 4. Disk Storage Drive*

Each disk storage drive utilizes an 11 disk, 20 surface removable Disk Pack (IBM 2316, or equivalent) with a rotation cycle of 25 milliseconds and a data rate of 290 KB. Two-hundred cylinders with an average cylinder access time of 50 milliseconds are available with a total data capacity per drive of either 28.8 million 8-bit bytes, or 6.4 million 36-bit words.

A disk pack can be quickly removed from or installed in any module location. This interchangeability of disk packs makes possible a high degree of program flexibility and a virtually unlimited off-line storage capacity. Additional disk storage drives are available that are IBM 2311 compatible.

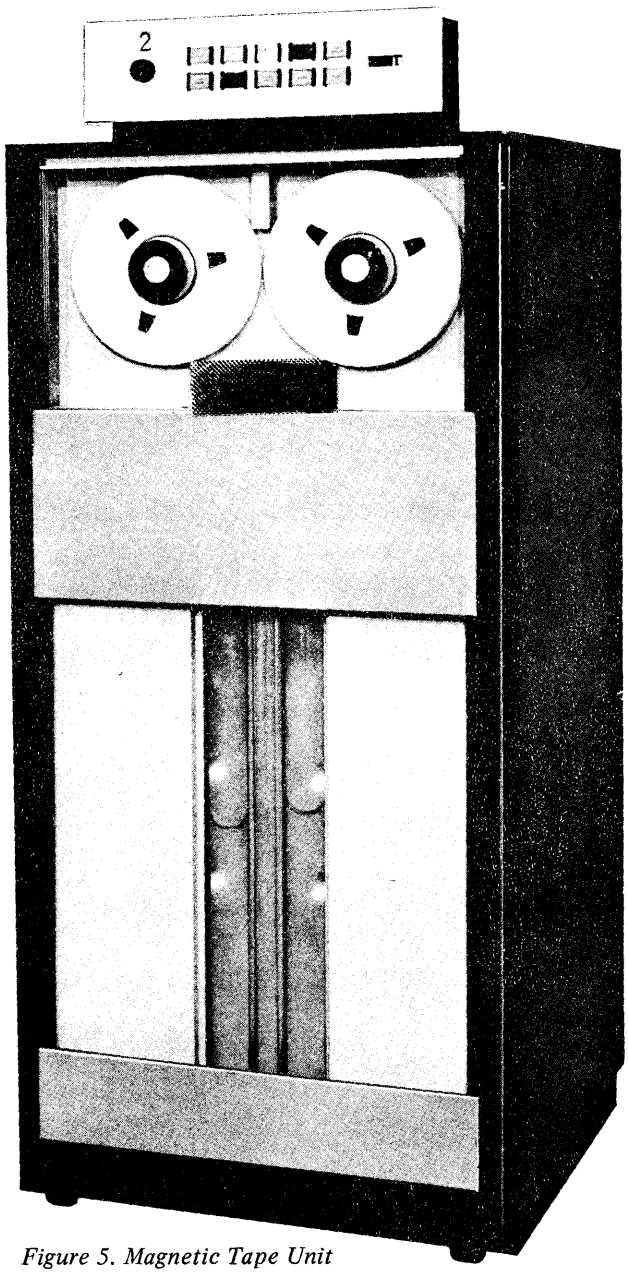

*Figure 5. Magnetic Tape Unit*

## Magnetic Tape Unit

The Magnetic Tape Unit (Figure 5) consists of a single-capstan tape transport with associated control panel and a read/write amplifier designed for use with high performance computer systems. The units available feature a 60 or 90kHz transfer rate, 7-track tape speed of 75 or 112.5 inches per second (ips), and a packing density of 556 or 800 bits per inch (bpi). Motion commands to the transport are derived from the computer system and a convenient control panel provides tape switches for loading and unloading of tape on the transport. Local and remote commands to the drive electronics are via the read/write amplifier chassis. The writing and reading of data is a function of the read/write magnetic head. In the write mode input data is received from the computer by the write amplifier and transferred to the write portion of the read/write magnetic head. Upon command from the computer, data written on tape is read in the read section of the read/write head and transferred to the computer.

## Line Printer

The Line Printer (Figure 6) is a high-speed, rotating drum, buffered line printer, capable of printing up to 1250 lines per minute. The printer uses the latest

*Figure 6. Line Printer*

asynchronous electronic timing circuitry to ensure an even, non-wavy print line at rated speed. Up to 6-part forms can be printed with clarity.

The revolving print drum incorporates a 64-character set. The standard width is 132 print positions.

The full font of characters is presented to electromagnetically actuate hammers once during each revolution. One hammer is provided for each column. The resulting print-out is clean and crisp.

Character set decoding is accomplished within the loadable magnetic core control memory. Vertical spacing, either 6 or 8 lines per inch, is controlled with a prepunched paper tape loop providing 8 channels of skipping, including overflow detection (12 channels available as an option).

Operator switch controlled code translation is included with every line printer. This feature allows the five H-chain characters, which differ from their A-chain counterparts, to be translated to A-chain characters; thus eliminating the need to physically convert from one chain to another.

Horizontal and vertical alignment is accomplished through use of a calibrated dial, and a transparent scale, which indicates the exact position of all print hammers

The line printer also has self-check features such as: low paper, out of paper, parity errors, fuse failure, and skip error.

A similar line printer model is available with a 96-character set (full ASCII).

## Card Reader

Two card reader models are available for the IC-7000 system. The 800 card per minute (cpm) Model 1011 Card Reader (Figure 7) accepts both Hollerith and binary cards, separately or intermixed. The card input/output hopper capacity is 2000 cards with special, last-batch switch and error hopper features. The Model 1012 Card Reader contains all the features of the 800 cpm machine, but at a rated 1500 cpm. Both card reader models contain extensive error checking procedures during the card read cycle to prevent misread data from entering the system.

Both card readers offer reliability, rugged service, and high speed input to the computer system.

The card reader features easy loading and unloading during operation. All controls and indicators are conveniently located and are easily accessible to the operator.

Use of solid-state electronics, plug-in modules, pull-out chassis, and convenient test points contribute to greatly simplify routing maintenance.

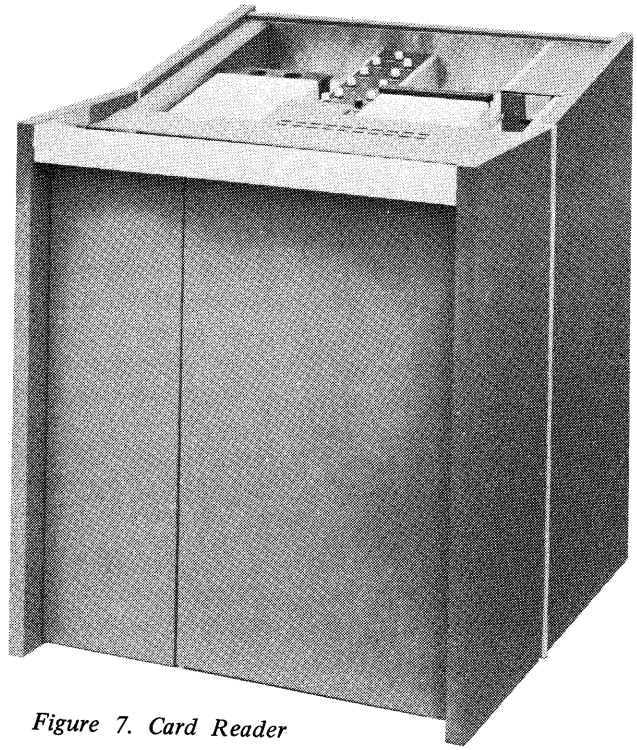

*Figure 7. Card Reader*

## Card Punch

The Card Punch (Figure 8) is a medium-speed unit capable of punching 80-column cards asynchronously at 100 to 275 cpm, depending upon the number of columns to be punched. Cards are punched column-by-column, each card being incrementally stepped through as many column positions as the processor indicates.

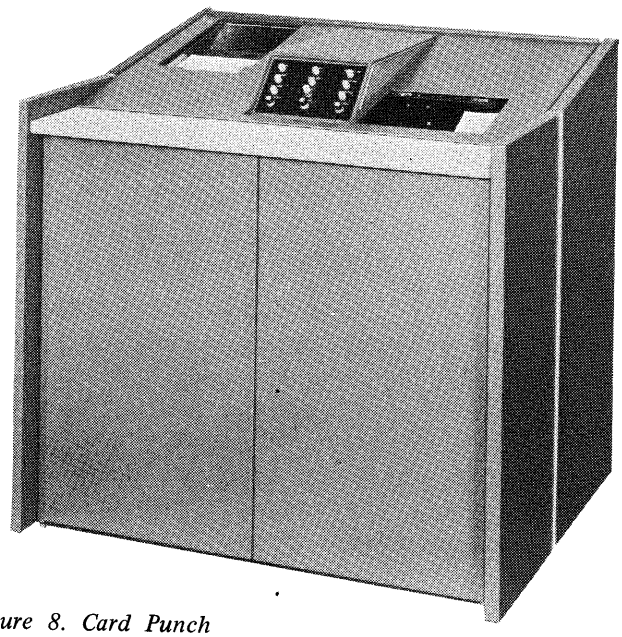

*Figure 8. Card Punch*

To insure utmost reliability in data transfer, electronic checking of reading and timing circuitry is performed during every card cycle as the card travels through the transport.

In addition to checking the punch circuitry and card movement through each card cycle, the card punch also verifies the data punched into the card, thus ensuring punch reliability. Punched data is verified by developing pulses from the energized punch cycle and transmitting this information to the processor for comparison. The card punch is equipped to photoelectrically count the punches in each card and compare the total count with the number of punches ordered by the processor.

When a mispunched card is detected, the condition is indicated on the operator control panel and the unit is automatically shut down. When desired, the rejected mispunched card can be offset without interrupting machine operation. The card offset mechanism offsets a mispunched card approximately ½ inch in the output hopper.

The rejected card can also be routed to the reject pocket, under programmed control of the computer. This does not alter the speed of the device.

Both input and output hoppers hold 1000 cards each and are located for ease in loading and unloading. Both operations may be performed with the machine in operation. Punched cards are stacked in the same order in which they are punched. Simplicity of design and operation assures maximum operator efficiency with only brief training.

Solid-state electronic circuitry, combined with the use of plug-in modules, pull-out chassis, conveniently located test points, and easy access to the entire transport assembly and punching head, contribute to reduced maintenance.

## MEMORY RELOCATION, PROTECTION AND SEGMENTATION

The memory-protection features, contained in the Arithmetic and Language Processor, process each user program as if it were the only program in memory. Simultaneously, the memory protection feature prevents executive and other system program interference. Processor hardware and MINIFLOW systems provide memory protection features automatically, under control of three time-sharing mode control bits. These are called relocate, protect, and slave. A fourth control bit is called data segment mode. Only data segment mode is controlled by the time-sharing user. Relocate, protect, and slave are controlled by a privileged instruction, and are saved in the trap status word when a trap occurs.

### Relocate

When the relocate bit is set all memory accesses are relocated; the processor modifies each address before it uses the address to access an instruction, operand, or indirect address word. The processor modifies each address by adding it to the contents of a base register. The base register is 10 bits long and corresponds to the 10 most significant bits of an 18-bit address; thus, addresses are relocated in increments of 256 words. There are two base registers, both located by a privileged instruction. Two of the registers are described under Data Segment Mode.

### Protect

If the protect bit is set, the processor tests every address before it uses the address to access memory. The processor compares the address to the contents of an upper-limit register. If the address is more than 256 words greater than the limit, the processor traps itself in an exception trap.

The limit register is 10 bits long, corresponding to the 10 most significant bits of an 18-bit address. There are two upper-limit registers, both loaded by a privileged instruction (LDMS). The two registers are described under Data Segment Mode. If the relocate bit is set along with the protect bit, the processor adds the relocation base register to the address and then compares the result with the upper-limit register.

### Slave

When the slave bit is set, the processor cannot execute a privileged instruction. Instead, it traps itself in an exception trap. Thus, only a master program using privileged instructions controls the processor memory-protection and relocation features.

### Data Segment Mode

When the data segment mode bit is set, the processor treats instruction addresses differently from operand and indirect-address-word addresses. Memory is considered to be divided into two parts: A

data phase. All instructions are in the procedure segment, while all operands or indirect address words are in the data segment. For this reason, there are two sets of base and upper-limit registers for relocation and protection. The processor uses the procedure base register and the procedure limit instruction address, and uses the data base register and the data limit register to relocate and limit the addresses for operands and indirect-address words.

## PROGRAMMING SYSTEM

### Introduction

The unique and advanced *STANDARD Computer* IC-7000 hardware was designed in conjunction with Call-A-Computer time-sharing software specications. This combination of knowledge and experience has resulted in a time-sharing system embodying many conceptual innovations.

A time-sharing system must contend with two separate and quite different tasks. First, to run the user's problem programs, and second, to regulate and control the time-sharing environment so as to maximize the throughput of the system. In the conventional approach of one CPU, the second function becomes overhead to the first function.

The *STANDARD*/Call-A-Computer approach was to design instruction sets to meet each of the two functions. An arithmetic and language processor (ALP) was micro-programmed to handle compilations and program execution. A second instruction set, composed of "directives", was designed for the Supervisory Processor Unit (SPU). Directives are oriented toward table look-up functions, input/output supervision and block data manipulation.

### Time-Sharing Executive

The time-sharing executive is operated on by both the ALP and the SPU and maximizes the concurrent parallel main storage access of the two processors. The ALP part of the executive optimizes the throughput of the many jobs contending for resources. Scheduling of functions, job scheduling, swap scheduling, internal timer function, and various interrupts, are handled by the SPU. The SPU also updates tables and adjusts boundaries, handles billing, core allocation, job termination, and overlays.

Requests for I/O are passed to the SPU from the ALP. The SPU is responsible for driving the file access routines and scheduling channel activities. The SPU also communicates with the communications multiplexer by sending and receiving data and sense information across a channel. The SPU manages the buffer pool and posts I/O completions and pertinent status to control blocks which can be interrogated by the ALP. Both ALP and SPU operate independently and asynchronously on data.

### Swapping

Increased system capacity is realized by moving programs between high speed disk and core memory. This process is known as swapping. Simultaneous execution of a large number of programs is accomplished using the swapping method.

During reallocation procedures, the SPU determines which programs are to be swapped out of core and which are to be brought into core from disk. Programs that are not affected by reallocation requests are processed simultaneously with swapping procedures.

Programs are swapped as follows: For each job in the job mix, a descriptor resides in one of three major queues within the system. An internal queue contains the descriptor of those jobs ready to execute and awaiting a time-slice. An I/O queue contains those jobs waiting for the completion of a previously given I/O request. Jobs on disk that are waiting for core allocation are kept on the external queue. When the SPU swaps a job out of core, it moves the associated job descriptor from the internal queue to the external queue. Conversely, when a job is swapped into core, the SPU places the descriptor in the internal queue and removes it from the external queue. No interrupts are necessary between the SPU and ALP in the performance of swapping procedures. The ALP works its way down the internal queue, giving time-slices to those jobs whose descriptors it finds there.

### Time-Sharing Statistics

Extensive statistics are kept by the time-sharing executive. The statistics are used by accounting routines to aid in user billing. They are also used by the executive to analyze the nature of the job mix, and its effectiveness in optimizing the throughput of the mix.

System statistics processed by the executive include:

Number of jobs in core awaiting execution.

Number of jobs awaiting completion of I/O events.

Number of jobs on disk awaiting core allocation.

Number of time slices for each job.

Core size of each job.

Data words the job has transferred across the channels.

Number of requests the job has made for various supervisor services.

Job start time.

Number of times the job has been swapped.

### Dynamic Time Slicing

Periodically, the SPU takes time to determine the load balance of the job mix, and to dynamically adjust the time-slice duration and initiates core reallocation requests. This optimizes the total throughput of the job mix. Statistics that have been gathered are interrogated at this time. Factors are given to the dynamic time-slice algorithm and, if necessary, a new time slice duration is calculated. Dynamic time-slice usage permits the executive to determine if the job mix is I/O bound or computer bound, and adjusts the time-slice allocations accordingly. Time-share users are thus guaranteed optimum system time for whatever job type processed. The system optimizes itself to give the best user response time possible. At peak usage periods, degradation is held to a minimum. The following are some of the statistical factors considered:

Terminal output buffer service.

Swap density—how many jobs have been moved in or out of core since the last analysis.

User counts.

Terminal input bound users.

Users who are computer bound.

Contractual guaranteed time-slices in the mix.

Size of jobs in core.

### Data Segment Mode

Important ramifications in the performance of the time-sharing system, as related to the data segment mode, provide a good example of the advantage of using the integrated hardware/software design as implemented in the IC-7000 system.

Compilers in this system are re-entrant and operate in the data segment mode. Therefore, there is only one copy of a compiler in core and it stays there as long as it is in use. Each user has a segment which the compiler uses as a work area. This segment is 675 words long. If a job is swapped out, the 675 words are written to disk. Restoration of jobs is therefore fast.

Object code, generated by the compilers, is executed in the data segment mode. One set of relocation and limit registers is assigned to the procedure segment and a second set is assigned to the data segment. Instruction operand addresses are modified by the relocation register of the data segment. Instructions in the procedure segment are thus protected and can never be altered. If the program is scheduled to be swapped out, only the data segment is saved while the procedure segment can be overlayed, since its original copy is stored on disk. The assembly language programmer has the necessary instructions to turn the data segment mode switch on or off. It is possible to code the assembly language programs in either data segment mode or single segment mode. Since the supervisor does not know how the assembly language programmer has used the data segment mode switch in his program, the supervisor tags all assembly language modules and, when swapping, saves the complete program.

## ASSOCIATIVE FILE SYSTEM

The first of its kind ever offered to time-sharing users, this file system retrieves files by an associative process. This process uniquely identifies the address of each file on disk by keys that it associates with the disk file address. The keys include user number, program name, and other keys which the user may specify. The system builds an association between each key and disk-file address at storage time. At retrieval time the system searches a balanced-tree directory to find the storage area associated with each key given, then retrieves the single file that all the given keys point to in common. The associative file system makes it possible for the user to devise powerful information retrieval systems of his own.

As an example of how the associative file system works, consider a user who has, in the course of doing business, filed travel expense report data by department each month. He desires to know the travel expense reports filed by sales and marketing, in the month of July, that were not tagged closed. He retrieves the information as follows:

Since the user started his file, he has created the following keys:

File Names: Reports, Travel
Month: Jan., Feb., March, etc.
Department: Sales, Engineering, Marketing, etc.
Status tags: Flag, Hold, Closed, etc.

After interrogating the system, the user types:

**OLD**

and the system responds:

**NAME–**

From all the keys, the user selects and types:

**REPORTS, TRAVEL, JULY, SALES, MARKETING, CLOSE**

and the system responds:

**KEYS–3, 9, 5, 2, 7, 4**
**?–**

For each key name, the system assigned a number. The system then asked what is needed (?–). Using the key number, the user types:

**(2 or 7) (3, 9, 5) (NOT 4)** which is a representation of his problem restated as follows:

sales or marketing, and reports, travel, July, and not
(2) (7) (3) (9) (5)
tagged close.
(4)

The system then responds:

**12 FILES FOUND**

and the user types in:

**EDIT MERGE 1-12**

The system responds:

**READY**

The user types in:

**RENAME**

and the system responds:

**NAME–**

The user types in:

**SUMDATA**

and the system responds:

**READY**

The user has created a file, Sumdata, that contains the input for his summary program: Sales or Marketing July travel reports not tagged closed.

## SECURITY

The IC-7000 time-sharing system provides the most rigorous security system available, to protect user files from unauthorized access. The system also ensures that a user is billed only for computer time that he actually uses. The basic security system lets the user designate files as public, for fast access, or highly proprietary–with a broad range in the degree of security provided. In addition, the system allows the user to program his own security routines with any degree of security he desired.

Typical security routines for user access to files are shown below. In these examples, user "type-ins" to the system are shown underlined, and system responses are shown not underlined.

Fast access for low security files–name of file grants anyone access:

**OLD** **NAME–TALLY**
**READY**

Medium security, fast access–name of file, password(s) and user number required for access:

**OLD** **NAME–TALLY**
**PASSWORD(S)–BAR**XER,FOXTA**
**READY**

Top secret security–user program queries the user who requests access. Program may be as involved as is practicable, can deny access or even identify terminal requesting access, then disconnect terminal:

**OLD** **NAME–TALLY**

**PASSWORD(S)–BARKER, FOXTAIL**

**CREDENZA**

**WHO ARE YOU? *X*9**

**HOW OLD ARE YOU? NINETY****

**MOTHER'S MAIDEN NAME? SMITH**

**SMITH? JONES**

**READY**

Hard-copy Security: Passwords and query responses can include non-printing characters.

Specified Limits of Access: Both the password and the query-response method of security specify the degree of access granted.

The user who successfully gains access to a file can read file identification, read, write, append, execute, or adjust security, or any combination of these as specified by security data for that file. For example, one set of passwords may allow the user to only append the file; another, to read and write; a third, unlimited access, etc.

## USER ARRANGEMENTS

Once contact with the system is made, the user is verified by the supervisor and user contract arrangements are retrieved from the disk file.

User requests are then checked against his contract for validity. For example, a user may not ask to use a tape drive unless it was specified so at contract time. The user may contract time-share service for any or all of the following:

Guaranteed time slices.

Use of central computer peripheral devices (i.e., printer, tape drives, card reader).

Rate adjustments at certain hours.

User limited to certain hours only.

## ROAD BLOCKS

User may enter jobs that have roadblocks associated with them. For example: A large job is to be run and the results are not needed until the next day. The rates after a certain hour decrease. So, the job can be entered through the terminal with the stipulation that it be started at a later specified hour, and then the terminal can be disconnected until the following day. The executive starts the job at the specified time. Jobs can be held for days and then run if the user so desires.

## OPERATING MODES

In time-sharing mode, the user has the computer on-line at his fingertips. He can compile programs, edit programs and text, enter input data, and receive output data. If the user has previously prepared his program and data on tape or cards, he can attach his terminal and then let the system read and execute his program in the remote batch processing mode. If the program and data stored on secondary storage is in the system, he can connect his terminal and start the job, disconnect, and allow the system to run it in the background mode. His terminal is now free to start another time-sharing job, or he may shut down his terminal until later when he dials again and connects to interrogate the results of his previous background execution. The software system is designed so that all three modes of operation look the same; the user has all modes available at all times. There is no limit to the number of jobs a user may have in execution concurrently.

## SOURCE PROGRAM MANIPULATION

Source programs may be entered from keyboard, paper tape, or cards from the particular terminal. Once entered, they can be stored, compiled, or updated. Individual lines can be added, deleted, or changed. Line numbers can be assigned by the system or the user. Source programs can be compiled, and the resulting object code saved in a compressed format that conserves disk space. This object code can be updated on the file with the same file manipulation techniques available to source program manipulation. Modified source programs can be renamed and saved in the new version. Debugged source program segments can be merged with other debugged segments to give a complete program.

Source or object programs are stored using the Associative File System. They are under security protection and can be easily retrieved by the legitmate user.

## DISK FILE SIZE WARNING

There is no limit to the amount of disk space a user may use, as long as he is willing to pay for it. However, to prevent unnecessary large files due to program errors or a tactical error, the system warns the user after a certain number of blocks have been written. The user has the option to cancel the job or continue.

## LANGUAGES

All languages used in the IC-7000 time-share system can be used intermixed. The system permits programs written in any language to call upon subroutines written in any other language. Complete language versatility is offered time-share users of the IC-7000 computing system.

### FORTRAN

IC-7000 FORTRAN is an expanded time-sharing version of A.S.A. FORTRAN IV. It allows completely mixed-mode arithmetic expressions, unrestricted nesting of function calls, and multi-dimensional arrays. Array elements can be addressed by any valid mixed-mode expressions used as subscripts. The compiler accepts optional picture formats as more powerful alternatives to the FORTRAN format statement.

### BASIC (Expanded)

BASIC is a versatile language, practical for professional users in many fields. BASIC is simple to use, but a powerful language providing arithmetic capabilities, logic comparisons, subscripting, common trigonometric functions, lists, and arrays. It has extensive capabilities for matrix manipulation. Call-A-Computer BASIC has been further expanded to include extended precision arithmetic operations, string manipulation, picture formatting, and subroutine compatibility with FORTRAN. This includes the passing of arguments by parameter list or by common storage assignment. An interactive diagnostic aid system operates in parallel with the compiler to help the user correct and de-bug his programs.

### Assembly Language

The assembly program accepts IC-7000 mnemonics and symbolic expressions, and translates them into machine language. It accepts pseudo-operations that aid the user in coding and debugging programs and is completely compatible with other IC-7000 compilers and assemblers in its linkage format.

The IC-7000 instruction set includes over 200 mnemonics. Because of the IC-7000's unique fourth generation MINIFLOW system concepts, many of these mnemonics fall into a set of instructions known as super-instructions. For example, the mnemonic "FFAD" converts a 27-bit integer in the addressed memory word (bits 0, 9-35) to a normalized, single-precision, floating-point number and adds it to the floating-point number in the accumulator. The result resides in that accumulator plus an adjacent accumulator in double-precision format. There is a complete family of these mixed mode arithmetic instructions. Another super-instruction set deals with bit-string manipulation.

### IMPLAN

IMPLAN is an implementation language oriented toward the creation of compilers, assemblers, and other system software modules. For this reason, special emphasis has been placed on bit, byte, and string manipulations. For example: BIT MOVE obtains a variable-length bit-field from the source, and inserts it into a variable position in the destination variable.

### COBOL

The COBOL language is source level compatible with Level F COBOL. It is primarily designed to assist in compilation and debugging but will execute Level F COBOL statements.

### EDIT

The text edit system makes it easy to revise text, data files, or programs. With the text edit system, the computer can be commanded to file, revise, rearrange, delete, insert, extract, and combine lines or fragments of text. The user does this by typing commands into the IC-7000 terminal. Commands like "find the name Smith wherever it appears in this text and change it to Smythe." But, not exactly like that. Because it is a computer that has to obey these commands, the commands must be in computer language, typed according to rules. To correct Smythe's misspelled name, for instance, you type:

```
EDIT $FIND 'SMITH' $REPLACE 'SMYTHE' 100
```

The number 100 after the command is the number of tries it will take to get every misspelled 'Smith', with some tries left over for good measure. This example typically illustrates the case and simplicity of text editing at the terminal.

Briefly, the text edit works two ways: The first is by line number (each line numbered sequentially as it is typed). An edit command is then entered by the typist to move lines, duplicate, delete, interweave, or type out lines called by number. The second way text edit works is by looking up strings of characters assigned, and handling these strings according to your commands. The Smith/Smythe example shows how handy the text edit system can be with strings.

## APPLICATIONS

Time-share computer system usage by industrial and manufacturing engineers to develop piece work pricing, and price modifying, has become widespread. The time-shared terminal also offers a computer to methods study engineers and planners, as a personal tool for use as long as needed, and at any location.

Industrial engineering usage includes job costing, machine utilization analysis, employee efficiency rating, machine maintenance scheduling and analysis, expansion or rearrangement, budgeting and scheduling, investment analysis, and optimum job routing and material utilization studies. The time-shared associative file system stores all tables, formulas, and formats for various analyses, scheduling, and pricing usages mentioned. The output can be formatted to meet individual user requirements. With respect to price information, the terminal could actually generate a job voucher, thus eliminating procurement paperwork associated with a job. In summary, the time-share terminal user in industry and manufacturing performs in his function faster, and more efficiently, using the IC-7000 Time-Sharing System as his personal tool.


go with the
innovators

# Standard


Standard Computer Corporation
1411 W. Olympic Boulevard
Los Angeles, California 90015
(213) 387-5267

777 N. First Street, Suite 600
San Jose, California 95112
(408) 294-7150

55 S. 36th Street
Boulder, Colorado 80302
(303) 428-0529

7718 Tanglecrest Drive
Dallas, Texas 75240
(214) 239-9627

8 South Michigan Avenue
Suite 720
Chicago, Illinois 60605
[redacted]
(312) 641–2227

17500 Northland Park Court
Southfield, Michigan 48076
(313) 352-1710

Suite 1325 – Land Title Building
Broad & Chestnut Streets
Philadelphia, Pennsylvania 19102
(215) LO3-6350

7100 France Avenue South
Minneapolis, Minnesota 55435
(612) 926-0706

235 Bear Hill Road
Waltham, Massachusetts 02154
(617) 891-5083

2 West 45th Street
New York, New York 10036
(212) 661-1834